

Oracle® Application Server

Adapters for Tuxedo User's Guide

11g Release 1 (11.1.1)

E16089-01

March 2010

Oracle Application Server Adapters for Tuxedo User's Guide, 11g Release 1 (11.1.1)

E16089-01

Copyright © 2009, 2010, Oracle and/or its affiliates. All rights reserved.

Primary Author: David Goldman

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

Preface	ix
Audience	ix
Documentation Accessibility	ix
Related Documents	x
Conventions	x
1 Introduction	
OracleAS Adapters for Tuxedo Overview	1-1
Adapter for Tuxedo	1-1
Adapter for Tuxedo/Q.....	1-3
Adapter for Tuxedo Gateway.....	1-3
Buffer Types.....	1-3
Return Codes	1-4
OracleAS Adapters for Tuxedo Architecture	1-4
Tuxedo Adapter for Inbound Architecture	1-4
The Flow from Oracle Application Server to the Legacy Application and Back	1-5
Tuxedo Gateway Adapter for Synchronous Inbound Architecture	1-5
Tuxedo Gateway Adapter Components.....	1-6
2 Configuring OracleAS Adapters for Tuxedo in Oracle Studio	
Configuring Oracle Connect Overview	2-1
Configuring OracleAS Adapters for Tuxedo in Oracle Studio	2-1
Securing Access to Oracle Connect	2-2
Setting Password Access to Oracle Studio	2-2
Specifying Users with Administrative Rights.....	2-3
Configuring Run-Time User Access to OracleAS Adapter for Tuxedo	2-4
Setting up the Tuxedo OracleAS Adapter	2-5
Before You Begin.....	2-5
Modeling Interactions for the Tuxedo OracleAS Adapter.....	2-6
Configuring a Tuxedo OracleAS Adapter.....	2-6
Starting Outbound Interactions Metadata Import	2-7
Getting a JOLT Input File for Tuxedo Adapter	2-9
Applying Filters for Tuxedo Adapter (JOLT).....	2-10
Generating Interactions (JOLT).....	2-10
Configuring Tuxedo Records for Tuxedo Adapter.....	2-10

Importing the Metadata Definitions for Tuxedo Adapter	2-11
Getting an FML or VIEW Import File for Tuxedo Adapter	2-11
Adding FML Records for Tuxedo Adapter (FML and View).....	2-12
Getting Tuxedo Records for Tuxedo Adapter (FML and View Files).....	2-13
Generating Interactions for Tuxedo Adapter (FML and View Files).....	2-13
Starting Interactions Metadata Import	2-15
Setting up the Tuxedo Queue OracleAS Adapter	2-19
Modeling Interactions for the Tuxedo Queue OracleAS Adapter	2-19
Configuring the Tuxedo Queue Adapter	2-19
Generating Inbound Interactions	2-20
Defining the Tuxedo Queue Unstructured Records	2-23
Setting up the Tuxedo Gateway OracleAS Adapter	2-25
Configure the Tuxedo Gateway as a Regular Adapter	2-25
Configuring the Tuxedo Gateway Adapter in Oracle Studio	2-26
Setting Properties for the Tuxedo Gateway Adapter	2-27
Starting a Metadata Import for Tuxedo Gateway Adapter	2-27
Getting a JOLT Input File for Tuxedo Gateway Adapter	2-29
Applying Filters for Tuxedo Gateway Adapter (JOLT)	2-30
Generating Inbound Synchronous Interactions for Tuxedo Gateway Adapter (JOLT)	2-30
Configuring Tuxedo Records for Tuxedo Gateway Adapter.....	2-31
Importing the Metadata Definitions for Tuxedo Gateway Adapter	2-31
Getting an FML or VIEW Import File for Tuxedo Gateway Adapter	2-31
Adding FML Records for Tuxedo Gateway Adapter (FML and View).....	2-32
Getting Tuxedo Records for Tuxedo Gateway Adapter (FML and View Files).....	2-33
Generating Inbound Synchronous Interactions for Tuxedo Gateway Adapter (FML and View Files)	2-34
Configure a Non-Transactional Tuxedo Gateway as a Tuxedo Server.....	2-35
Configure a Transactional Tuxedo Gateway as a Tuxedo Group.....	2-36
Starting the Tuxedo Gateway Transaction Group	2-38
Adapter Schema for the Tuxedo Gateway Adapter	2-38

3 Integrating OracleAS Adapters for Tuxedo with WebLogic

Overview of Integrating OracleAS Adapters for Tuxedo.....	3-1
Configuring the Tuxedo Adapter for Outbound.....	3-2
Configuring the Tuxedo Queue Inbound Adapter.....	3-3
Configuring the Tuxedo Gateway Synchronous Inbound Adapter.....	3-4
High Availability.....	3-5

4 Integrating OracleAS Adapters for Tuxedo

Configuring OracleAS Adapter for Tuxedo	4-1
Creating JCA Configuration Files	4-1
Create JCA Configuration Files for the Tuxedo and Tuxedo Queue Adapters.....	4-1
Create JCA Configuration Files for the Tuxedo Gateway Adapter.....	4-2
Configuring the Binding for Outbound Interactions.....	4-3
Configuring the Binding for Tuxedo Queue Inbound Interactions.....	4-4

Configuring the Transactional Gateway Process	4-5
Configure the BPEL Transactional Gateway Process using JDeveloper.....	4-6
Configure the Resource Adapter Connection Factory.....	4-6

5 Configuring an OracleAS Adapter for Tuxedo

Creating Outbound Interactions with the OracleAS Adapter for Tuxedo	5-1
Create the toUpper Interaction	5-1
Prepare the System	5-1
Set up Machine Access to Oracle Connect	5-2
Add a Tuxedo Adapter	5-2
Import the Metadata	5-3
Create the JCA Configuration Files	5-4
Creating Inbound Interactions with the Tuxedo Queue Adapter.....	5-4
Create the DEQUEUE Event Queue.....	5-4
Prepare the System	5-4
Set up Machine Access to Oracle Connect	5-4
Add a Tuxedo Queue Adapter	5-5
Import the Metadata.....	5-6
Create the JCA Configuration Files	5-6

6 Creating the Project in JDeveloper Examples

Getting Started.....	6-1
Deploy a BPEL Inbound Process.....	6-2
Beginning Tasks	6-2
Design-Time Configuration.....	6-2
Create a BPEL Inbound Process.....	6-3
Use a Third-Party Adapter for the Inbound Process	6-4
Design the BPEL Inbound Process	6-6
Deploy the Project.....	6-7
Create the Connection	6-8
Run-time Configuration.....	6-8
Triggering the Event using the XML Utility	6-8
Mediator Inbound Process	6-9
Beginning Tasks	6-10
Create a Mediator Inbound Process	6-10
Use a Third-Party Adapter for the Inbound Process	6-12
Create a File Adapter	6-13
Design an Inbound Mediator	6-14
Create Mapping Rules and Map the Files	6-15
Deploy the Project	6-15
Deploy a BPEL Outbound Process.....	6-15
Beginning Tasks	6-16
Design-Time Configuration.....	6-16
Before You Begin.....	6-16
Create a BPEL Project for a BPEL Outbound Process.....	6-17
Create a File Adapter to Read the String.....	6-17

Design a BPEL Outbound Process to Read the String	6-18
Create a Partner Link to the Tuxedo Adapter	6-19
Edit the Invoke_1 Activity	6-20
Edit the Assign_1 Activity	6-21
Create a Process to Write the Upper-Case Letter String	6-22
Runtime Configuration	6-23

7 Troubleshooting OracleAS Adapter for Tuxedo

Troubleshooting the Daemon	7-1
Starting the Daemon	7-1
Task: Starting the Oracle Connect Daemon	7-2
Shutting Down the Oracle Connect Daemon.....	7-2
Monitoring the Daemon During Run Time.....	7-2
Daemon (Computer) Options	7-2
Workspace Options	7-3
Server Options.....	7-4
Daemon Logs	7-4
The Daemon Log Monitor	7-5
The Workspace Log Monitor.....	7-5
The Server Log Monitor	7-6
Resolving Communication Errors.....	7-6
Resolving Specific Errors.....	7-7

8 Advanced Features of OracleAS Adapter for Tuxedo

Configuring the Daemon for High Availability	8-1
Adding a New Daemon Workspace Configuration.....	8-1
Editing the Workspace	8-2
Configuring the Server Mode.....	8-2
Configuring a Binding Environment	8-5
Debug.....	8-6
General.....	8-7
Language	8-8
Modeling	8-8
ODBC	8-8
OLEDB	8-8
Optimizer	8-8
Query Processor	8-8
Parallel Processing	8-9
Transactions	8-9
Tuning.....	8-9
XML.....	8-9
Migration Considerations.....	8-9
Security Considerations	8-10
Setting Design Time Security	8-10
Setting Run-Time Security	8-11
Encrypting Network Communications	8-11
Transaction Support.....	8-15

A Advanced Tuning of the Metadata

Editing Adapter Interactions	A-1
Adapter Metadata General Properties	A-1
Adapter Metadata Interactions	A-3
Adapter Metadata Interactions for the Tuxedo and Tuxedo Queue Adapters	A-3
Adapter Metadata Interactions for the Tuxedo Gateway Adapter	A-5
Adapter Metadata Schema Records	A-7
Adapter Metadata XML	A-8

B OracleAS Adapters for Tuxedo Message Buffer Support and Data Type Support

OracleAS Adapters for Tuxedo Message Buffer Support	B-1
User-Defined Message Buffers	B-2
Data Type Support	B-2
Data Type Mapping	B-2
Data Type Handling	B-2
Header Record Structure.....	B-3

C Advanced Tuning of the Daemon

Control	C-1
Logging	C-3
Security	C-5
Workspaces	C-7
General.....	C-7
Server Mode	C-11
Security	C-16

D Globalization Settings

Defining the Language and Code Page	D-1
--	-----

E Editing XML Files in Oracle Studio

Preparing to Edit XML Files in Oracle Studio	E-1
Making Changes to the XML File	E-2
Remove Objects	E-2
Add DTD Information.....	E-2
Edit Namespaces	E-3
Add Elements and Attributes.....	E-5
Replace an Element.....	E-5

Index

Preface

This guide is the primary source of user and reference information on OracleAS Adapter for Tuxedo, which enables client applications to access transactions running under Tuxedo through the Sun J2EE Connector Architecture (J2CA) API.

This document describes the features of OracleAS Adapter for Tuxedo.

This preface covers the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This manual is intended for Oracle integration administrators who perform the following tasks:

- Installing and configuring OracleAS Adapter for Tuxedo
- Diagnosing errors
- Using OracleAS to access Tuxedo transactions

Note: You should understand the fundamentals of OracleAS, WebLogic, the UNIX and Microsoft Windows operating system before using this guide to install or administer OracleAS Adapter for Tuxedo.

Note: For the purposes of this version of the Oracle Weblogic Server, Oracle Application Server refers to the Oracle WebLogic Server and OracleAS refers to Oracle WLS.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to

facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit <http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Other Product One Release 7.0 documentation set or in the Oracle Other Product Two Release 6.1 documentation set:

- *Oracle Application Server Adapter Concepts Guide*
- *Oracle Application Server Adapter Installation Guide*
- *Oracle Application Server Adapter Concepts Guide*
- *Oracle Application Server Containers for J2EE User's Guide*
- *Oracle Application Server Containers for J2EE Services Guide*
- *Oracle Application Server Containers for J2EE Security Guide*

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.
<i>italic</i>	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.

Introduction

This chapter provides an overview of the features and architecture of OracleAS Adapters for Tuxedo.

This chapter contains the following topics:

- [OracleAS Adapters for Tuxedo Overview](#)
- [OracleAS Adapters for Tuxedo Architecture](#)
- [Tuxedo Gateway Adapter for Synchronous Inbound Architecture](#)

OracleAS Adapters for Tuxedo Overview

OracleAS Adapters for Tuxedo includes the following adapters:

- [Adapter for Tuxedo](#)
- [Adapter for Tuxedo/Q](#)
- [Adapter for Tuxedo Gateway](#)

These adapters integrate Oracle Application Server with the legacy applications. OracleAS Adapters for Tuxedo model services running on the BEA Tuxedo application server. The BEA Tuxedo application server provides two programming interfaces, Common Object Request Broker Architecture (CORBA) and Application to Transaction Monitor Interface (ATMI). ATMI supports a programming interface that offers procedural library-based programming using a set of C or COBOL procedures. ATMI also provides an interface for communication, transaction, and data buffer management. In addition, the ATMI interface and BEA Tuxedo system implement the X/Open distributed transaction processing (DTP) model for transaction processing.

Adapter for Tuxedo

OracleAS Adapter for Tuxedo includes the following features:

- Improved performance by providing direct access to the platform where OracleAS Adapter for Tuxedo runs.
- Models Tuxedo services as a collection of interactions, where each interaction is mapped to a specific Tuxedo service, along with the input and output records.
- Captures and maintains a metadata schema for the Tuxedo system by importing Tuxedo metadata either from FML and VIEW files or from a JOLT file, and transforming this metadata into mapping definitions for Oracle Connect.
- Supports Tuxedo/T synchronous calls.
- Maps Tuxedo message structures to XML data and back.

- Supports local transactions by acting as a Resource Manager (RM). This is achieved by the adapter responding to XML transactional verbs and issuing the corresponding ATMI transactional calls. The back-end service is responsible for the transaction semantics and implementation.
- Supports full client authentication with the Tuxedo server. An optional use of OracleAS Adapter and password is also supported.
- Supports the buffer types as shown in the following table.

Table 1–1 Buffer Types and Corresponding Field Types

Buffer Type	Field Type
FML (both 16-bit and 32-bit are supported).	CARRAY
	Short
	Long
	Char
	Float
	Double
	String
VIEW (both 16-bit and 32-bit are supported).	Short
	Long
	Char
	Float
	Double
	String
X_C_TYPE	Short
x_common	Long
	Char
	Float
	Double
	String
x_octet	
CARRAY	
String	
XML	

- Supports Post events (EventBroker). This enables Java applications to subscribe to Tuxedo events, and receive them as J2CA 1.5 message inflow. Events are posted to one or several queues, according to ruling policies.
- Supports writing messages to a queue (enqueue). This is performed by executing the enqueue interaction for which the event is defined as the interaction input record.

OracleAS Adapter for Tuxedo does not support global transactions and cannot participate in a distributed transaction.

See Also: *Oracle Application Server Adapter Concepts Guide*. BEA Tuxedo Documentation.

Adapter for Tuxedo/Q

OracleAS Adapter for Tuxedo/Q is used for pulling messages from a queue. A queue is referred to as an outbound channel, and it is logically equivalent to an outbound interaction.

To pull messages from the queue, the `getEvents` interaction is executed. This interaction is automatically added to the adapter schema. When executed, all messages are pulled from the queue. In addition, a record that unions all the output records is created and is being used as the interaction output record.

A selected message can be removed from the queue, using the `dequeue` interaction. When executed, a message with specific attributes is removed from the queue.

Adapter for Tuxedo Gateway

OracleAS Tuxedo Gateway adapter is used for synchronous inbound communication. In this case, the Tuxedo Gateway adapter calls an application and also waits for a response. This enables seamless integration and/or migration of Tuxedo services from a Tuxedo-only environment into a heterogeneous environment where Tuxedo services can call Oracle products such as BPEL and ESB in exactly the same way they call other Tuxedo services. Interactions created under this adapter indicate services that are exposed by the Tuxedo Gateway as a Tuxedo services.

The Tuxedo Gateway adapter provides support for synchronous access from Tuxedo services to Oracle services. Using the Tuxedo Gateway adapter, you can call Oracle products such as the BPEL Process Manager and ESB in the same way that other Tuxedo services are called.

The Tuxedo Gateway adapter supports two-phase commit transactions. For information, see [Configuring the Transactional Gateway Process](#).

Buffer Types

The Tuxedo Gateway adapter supports the buffer types as shown in the following table.

Table 1–2 Buffer Types and Corresponding Field Types

Buffer Type	Field Type
FML (both 16-bit and 32-bit are supported).	CARRAY
	Short
	Long
	Char
	Float
	Double
	String
VIEW (both 16-bit and 32-bit are supported).	Short
	Long
	Char
	Float
	Double
	String

Table 1–2 (Cont.) Buffer Types and Corresponding Field Types

Buffer Type	Field Type
X_C_TYPE	Short
x_common	Long
	Char
	Float
	Double
	String
x_octet	
CARRAY	
String	
XML	

Return Codes

The table below describes the codes returned when calling the Tuxedo Gateway adapter services.

Table 1–3 Return Values and Codes

Event	Return
Success	Return value (<i>rval</i>): TPSUCCESS Return code (<i>rcode</i>) 0 Note: The data and the data length are returned.
Fail	Return value (<i>rval</i>): TPFFAIL The return code depends on the reason the call failed. The following are the codes (<i>code</i>) returned when the call fails: <ul style="list-style-type: none"> ▪ <code>unavailable = 1</code> ▪ <code>badinput = 2</code> ▪ <code>applicationError = 3</code> ▪ <code>generalError = 4</code> Note: No data is returned.

OracleAS Adapters for Tuxedo Architecture

This section describes the following:

- [Tuxedo Adapter for Inbound Architecture](#)
- [Tuxedo Gateway Adapter for Synchronous Inbound Architecture](#)

Tuxedo Adapter for Inbound Architecture

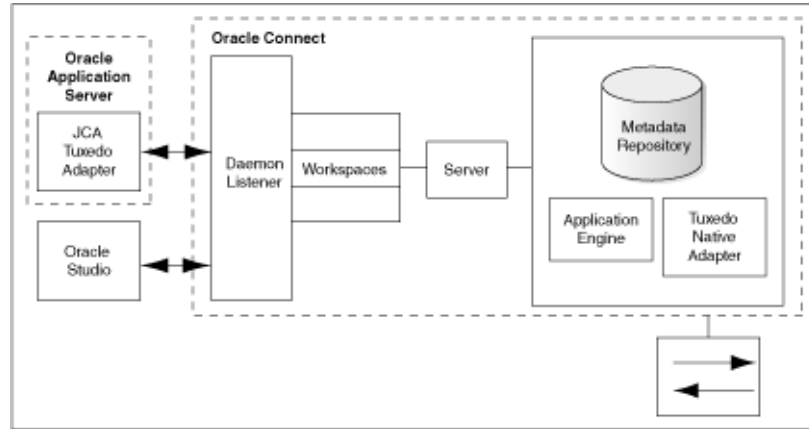
The following components comprise the architecture for Oracle AS Adapters for Tuxedo:

- J2CA 1.5 Tuxedo adapter: The J2CA Tuxedo adapter is a standard resource adapter that is compliant with J2EE Connector Architecture, providing J2EE components connectivity.

- Oracle Connect: Oracle Connect runs on the legacy system and handles requests from the J2CA 1.5 Tuxedo adapter that runs within Oracle Application Server Containers for J2EE (WebLogic).
- Oracle Studio: Oracle Studio is the configuration tool for Oracle Connect. Configuration tasks using Oracle Studio are performed on a Windows platform. Oracle Studio uses perspectives that enable you to generate specific information necessary to model OracleAS Adapters for Tuxedo.

Figure 1–1 illustrates the components of OracleAS Adapter for Tuxedo.

Figure 1–1 Architecture of OracleAS Adapter for Tuxedo



OracleAS Adapter

See Also: *Oracle Application Server Adapter Concepts Guide*

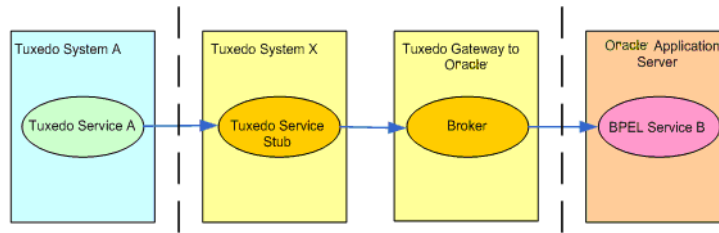
The Flow from Oracle Application Server to the Legacy Application and Back

The J2CA 1.5 Tuxedo adapter converts the J2CA interaction invocation received from an application client to the XML format and passes the XML format to Oracle Connect on the legacy server. The daemon listens for the request coming from the J2CA 1.5 Tuxedo adapter client and assigns a server process to handle the request. The properties of the server process (such as connection pooling requirements) are defined by a workspace definition within the daemon. The server process includes an instance of the application engine, which converts the XML format into native structures understandable by Tuxedo and passes the converted XML to the back-end adapter. The back-end adapter builds an interaction based on the metadata for the back-end adapter stored in the repository and the incoming converted XML and passes it to the legacy application. The results of this execution are passed back to the application engine, using the back-end adapter, where these results are converted to XML and passed back to the client.

Tuxedo Gateway Adapter for Synchronous Inbound Architecture

The Tuxedo Gateway adapter is set up to allow a Tuxedo service call an Oracle service. This is done without the Tuxedo Service “knowing” that it is calling a different type of service. The following diagram shows how a Tuxedo service calls an Oracle service using the Tuxedo Gateway adapter.

Figure 1–2 Tuxedo Gateway Adapter Flow



In this example, a regular Tuxedo service wants to call an OracleAS service, in this case BPEL. The Tuxedo service does not recognize this service and cannot call it. The two intermediate steps are added by the Tuxedo Gateway adapter to allow the Tuxedo service to call the OracleAS service. The following table describes the four steps in this flow.

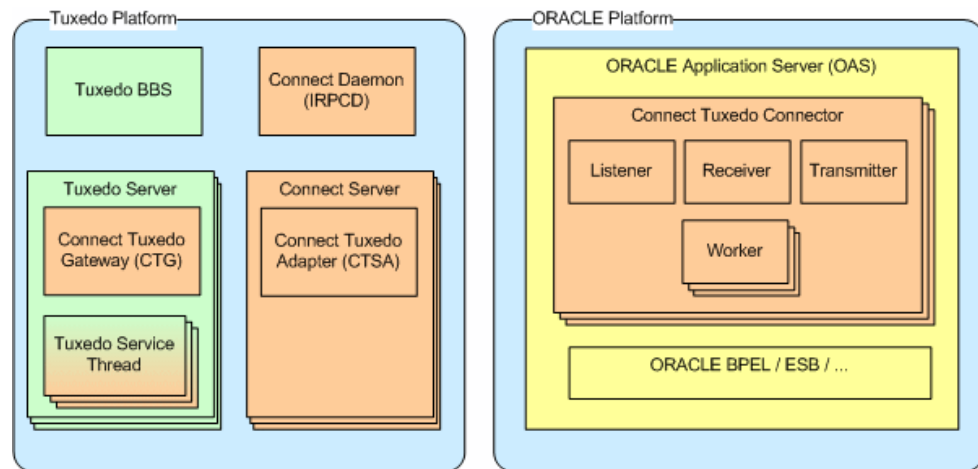
Table 1–4 Tuxedo Gateway Adapter Flow Description

Step	Description
Tuxedo Service A'	This is any regular Tuxedo service.
Tuxedo Service Stub	This is an intermediate step added when you use the Tuxedo adapter, which acts as a Tuxedo Server. This component accepts the requests from the Tuxedo service and forwards it to the Tuxedo Gateway, then receives the response and returns it to the original service.
Tuxedo Gateway to Oracle	This is an intermediate step added when you use the Tuxedo Gateway adapter. This component accepts the work requests from the Tuxedo Service Stub and forwards them to the OracleAS. When the service request is executed, any response is returned to the Tuxedo Gateway and then to the Tuxedo Service Stub, which forwards the response to the originating Tuxedo service.
Oracle Application Server	This is the service that is being called by the Tuxedo service. It can be a BPEL or ESB service implementation.

See Also: Oracle Application Server Adapter Concepts Guide, Oracle Application Server Containers for J2EE User’s Guide, Oracle Application Server Containers for J2EE Services Guide, and Oracle Application Server Containers for J2EE Security Guide.

Tuxedo Gateway Adapter Components

The following diagram shows the system components for the Tuxedo Gateway adapter:

Figure 1–3 Tuxedo Gateway Adapter Components

The following table describes each of the components:

Table 1–5 Tuxedo Gateway Adapter Components Description

Component	Description
Tuxedo BBS	A Tuxedo component that registers Tuxedo services and manages Tuxedo servers.
Tuxedo Server	A process that hosts Tuxedo services. This is managed by the Tuxedo BBS. You can configure multiple instances of the Tuxedo Server.
Connect Tuxedo Gateway (CTG)	The main adapter. This adapter handles service requests from Tuxedo.
Tuxedo Service Thread	A special Tuxedo service function that relays Tuxedo service requests to the gateway and returns the response from the gateway (all in the same process). There can be more than one Tuxedo Service Thread on a Tuxedo server. This is dependant on the Tuxedo configuration.
Connect Daemon	The Connect daemon that starts the Connect server processes for clients. The daemon also provides notification about the Tuxedo Gateway servers to the Tuxedo JCA connector.
Connect Server	A process managed by the Connect daemon that hosts native application adapters.
Connect Tuxedo Adapter (CTSA)	The native Connect adapter to Tuxedo services.
Oracle Application Server (OAS)	The J2EE container hosting the JCA Tuxedo resource adapter

Table 1–5 (Cont.) Tuxedo Gateway Adapter Components Description

Component	Description
Connect Tuxedo Connector	<p data-bbox="683 262 1365 342">A JCA 1.5 based connector that supports both inbound and outbound synchronous interaction with Tuxedo. The following are the components of the Connect Tuxedo Connector.</p> <ul data-bbox="683 359 1365 764" style="list-style-type: none"><li data-bbox="683 359 1365 489">■ Listener: A connector thread that creates/terminates the Receiver/Transmitter thread pair to interact with each new Connect Tuxedo Gateway process that is started by Tuxedo. A listener thread is created for each deployed connector and this thread works with one Connect daemon.<li data-bbox="683 506 1365 585">■ Receiver: A connector thread that gets call request batches from a Connect Tuxedo Gateway instance. A receiver thread is created for each Tuxedo server process started by Tuxedo.<li data-bbox="683 602 1365 699">■ Transmitter: A connector thread that sends call response batches to a Connect Tuxedo Gateway instance. A transmitter thread is created for each Tuxedo server process started by Tuxedo.<li data-bbox="683 716 1365 764">■ Worker: A pool of connector threads that synchronously invoke endpoints based on call requests.

Configuring OracleAS Adapters for Tuxedo in Oracle Studio

This chapter includes the following topics:

- [Configuring Oracle Connect Overview](#)
- [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#)
- [Securing Access to Oracle Connect](#)
- [Setting up the Tuxedo OracleAS Adapter](#)
- [Setting up the Tuxedo Queue OracleAS Adapter](#)
- [Setting up the Tuxedo Gateway OracleAS Adapter](#)
- [Adapter Schema for the Tuxedo Gateway Adapter](#)

Configuring Oracle Connect Overview

You can configure Oracle Connect using Oracle Studio. To use Oracle Studio, first configure it to enable access to the platform where OracleAS Adapters for Tuxedo run.

Before configuring Oracle Connect, ensure the following requirements are fulfilled:

- You have permission to access the platform where OracleAS Adapters for Tuxedo runs.
- Oracle Connect daemon is running on this computer.
- The following requirements are for the Tuxedo Q Adapter only:
 - Tuxedo queue space and a queue are defined.
 - The TMQUEUE server is set to handle Q-space.
 - The EventBroker server process is set. Either TMUSREVT for application events, or TMSYSEVT for system events.

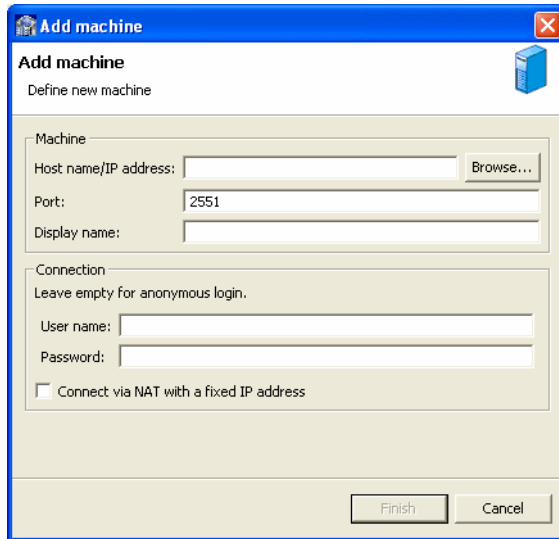
Note: For further information on BEA Tuxedo, see BEA Tuxedo documentation.

Configuring OracleAS Adapters for Tuxedo in Oracle Studio

Perform the following steps to configure OracleAS Adapters for Tuxedo in Oracle Studio:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**. Oracle Studio opens, displaying the Design perspective and the Welcome pane.
2. Right-click **Machines** in the Configuration Explorer and select **Add Machine**.
The Add Machine screen is displayed, as shown in the following figure:

Figure 2–1 The Add Machine screen



3. Enter the name of the computer you want to connect to, or click **Browse** to select the required computer that uses the default port (2551).
4. Specify the OracleAS Adapter and password of the user who was entered as the administrator during Oracle Connect installation.

Note: Selecting **Anonymous connection** enables all users having access to the computer to be an administrator.

5. Click **Finish**. The selected computer is displayed in the Configuration Explorer.

Securing Access to Oracle Connect

Oracle Studio includes mechanisms to secure access to Oracle Connect both during design time and run time.

During design time, the following security mechanisms can be applied:

- [Setting Password Access to Oracle Studio](#)
- [Specifying Users with Administrative Rights](#)

During run time, client access to Oracle Connect is provided by the user profile:

- [Configuring Run-Time User Access to OracleAS Adapter for Tuxedo](#)

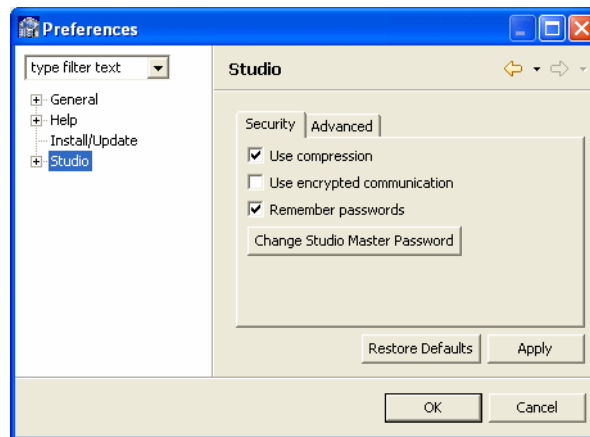
Setting Password Access to Oracle Studio

Initially, any operation performed using Oracle Studio does not require a password. You can set a password so that the first operation that involves accessing the server from Oracle Studio requires a password to be entered.

Perform the following steps to set password access to Oracle Studio:

1. From the **Start** menu, select **Programs, Oracle** and then select **Studio**. Oracle Studio opens.
2. Select **Window** from the menu bar and then select **Preferences**. The Preferences window is displayed.
3. Click **Studio**, as shown in the following figure:

Figure 2–2 The Preferences screen



4. Click **Change master password** to open the Change Master Password screen.
5. In the Change Master Password screen, leave the **Enter current master password** field empty and type a new master password.
6. Confirm the new password.
7. Click **OK**.

Specifying Users with Administrative Rights

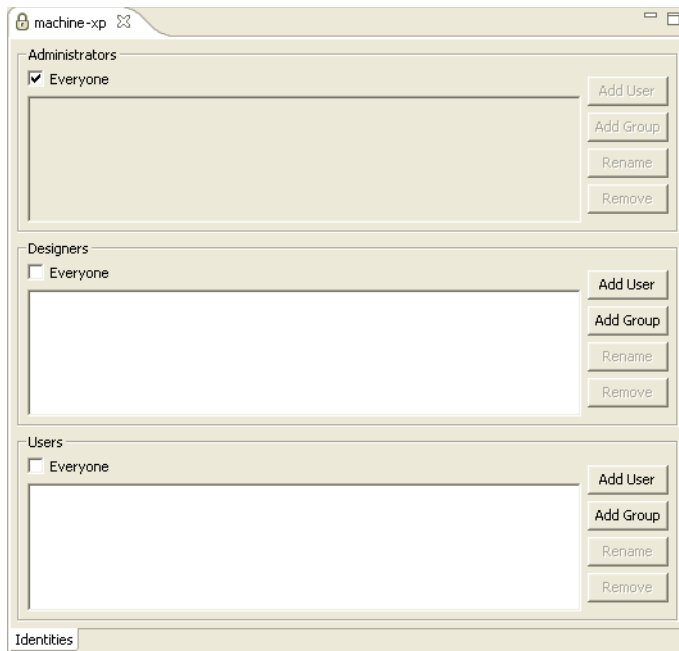
By default, only the user who was specified as an administrator during the installation is authorized to modify settings on that computer from Oracle Studio. The administrator can authorize other users to make changes or view the definitions for a selected computer.

Note: By default, all users should be given administrative rights during installation.

Perform the following steps to specify users with administrative rights:

1. From the **Start** menu, select **Programs, Oracle** and then select **Studio**. Oracle Studio opens, showing the Design perspective.
2. Right-click the required computer in the Configuration Explorer and select **Administration Authorization**.

The Administration Authorization screen is displayed, as shown in the following figure:

Figure 2–3 The Administration Authorization screen

The screen has three sections:

- Administrators: Administrators can view and modify all the definitions for the selected computer in Oracle Studio. On initial entry to Oracle Studio, every user is defined as a system administrator.
 - Designers: Designers can view all the definitions for the computer in Oracle Studio and can modify any of the definitions under Bindings and Users for the selected computer.
For example, Oracle Studio database administrator can add new data sources and adapters and can change metadata definitions for a table in a data source.
 - Users: Users can view all the definitions for the selected computer in Oracle Studio. Regular users cannot modify any of the definitions.
3. Add users or groups by clicking **Add User** or **Add Group** for the relevant sections. The user or group that is added must be recognized as a valid user or group for the computer.

Note: Once a name has been added to a section, only the user or group that logs on with that user name has the relevant authorization.

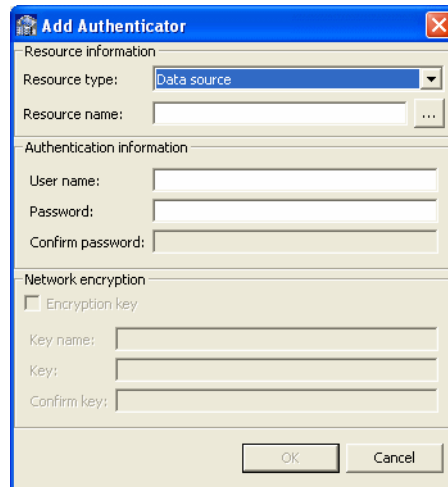
Configuring Run-Time User Access to OracleAS Adapter for Tuxedo

During run time, client access to Oracle Connect is provided by the user profile. A user profile contains name and password pairs that are used to access a computer, data source or application during run time, when anonymous access is not allowed.

Perform the following steps to configure run-time user access to OracleAS Adapters for Tuxedo:

1. In the Configuration Explorer, expand the machine to set the user name and password.
2. Expand the **Users**.
3. Right-click the user profile and select **Open**. The user profile editor is displayed.
4. In the User editor, click **Add** to open the Add Authenticator screen, as shown in the following figure.

Figure 2–4 The Add Authenticator screen



5. Select the **Remote Machine** resource type.
6. Enter the name of the OracleAS Adapter for Tuxedo computer defined in Oracle Studio.
7. Enter the user name and password used to access the computer and confirm the password.
8. Click **OK**.

Setting up the Tuxedo OracleAS Adapter

This section contains the following topics:

- [Before You Begin...](#)
- [Modeling Interactions for the Tuxedo OracleAS Adapter](#)

Before You Begin...

This section describes some parts of the configuration process that you should be aware of before you start.

Workspace Server: You should ensure that the workspace server mode is set to `reusable` or `single client`. It is recommended to use the `reusable` mode. For more information on setting the workspace server mode, see [Configuring the Server Mode](#).

Maximum number of servers: Ensure that this property is set to a number that is large enough to handle your system requirements. The default value of 3 may not provide enough servers. For more information on how to set the maximum number of servers, see [Configuring the Server Mode](#).

Environment properties: Ensure that you set the environment properties required by the Tuxedo adapter to work with tuxedo, such as TUXDIR.

Setting the environment properties in UNIX may be done by editing the `nav_login` script to set up all of the required environment variables.

In Windows, enter the information in the Windows Environment Variables dialog box. To open the Environment Variables dialog box, right-click **My Computer** on the Windows desktop and select **Properties**. Click the **Advanced** tab, then click **Environment Variables**.

Modeling Interactions for the Tuxedo OracleAS Adapter

Modeling interactions for OracleAS Adapter for Tuxedo involves defining an Oracle Connect adapter using Oracle Studio.

All the definitions specified in Oracle Studio are written to the platform where OracleAS Adapters for Tuxedo run.

This section contains the following:

- [Configuring a Tuxedo OracleAS Adapter](#)
- [Starting Outbound Interactions Metadata Import](#)
- [Getting a JOLT Input File for Tuxedo Adapter](#)
- [Applying Filters for Tuxedo Adapter \(JOLT\)](#)
- [Generating Interactions \(JOLT\)](#)
- [Configuring Tuxedo Records for Tuxedo Adapter](#)
- [Importing the Metadata Definitions for Tuxedo Adapter](#)
- [Getting an FML or VIEW Import File for Tuxedo Adapter](#)
- [Adding FML Records for Tuxedo Adapter \(FML and View\)](#)
- [Getting Tuxedo Records for Tuxedo Adapter \(FML and View Files\)](#)
- [Generating Interactions for Tuxedo Adapter \(FML and View Files\)](#)
- [Starting Interactions Metadata Import](#)

Configuring a Tuxedo OracleAS Adapter

To work with Oracle Connect, you must configure the adapter definition on the platform where OracleAS Adapters for Tuxedo run to handle the interactions to and from a Tuxedo service.

Perform the following steps using Oracle Studio to specify these definitions:

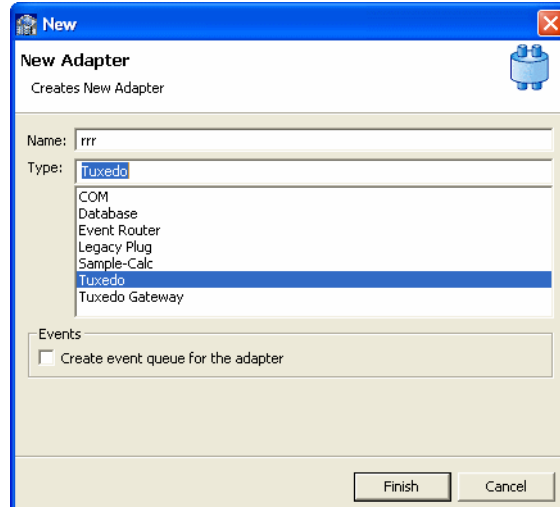
1. From the **Start** menu, select **Programs, Oracle** and then select **Studio**.
2. In the Configuration Explorer, expand the machine defined in [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#).
3. Expand the **Bindings**. The binding configurations available on this computer are listed.
4. Expand the binding where you want to define the adapter. The binding configuration includes branches for data sources and adapters that are located on the machine.
5. Right-click **Adapters** and select **New Adapter** to open the New Adapter wizard.

6. Enter a name for the adapter.

Note: The word *event* is a reserved word and cannot be used to name an adapter.

7. Select **Tuxedo** as the adapter type from the **Type** list, as shown in the following figure:

Figure 2–5 The New Adapter screen



Select **Create event queue for the adapter** to create an event queue for the adapter.

8. Click **Finish**. The adapter is added to the adapters list and its definition opens for editing.

Note: Other adapters that are displayed in the **Type** list are not supported with the version of Oracle Connect installed at the site.

Starting Outbound Interactions Metadata Import

Oracle Connect requires metadata describing the adapter interactions, including the structures used to pass information to and from the adapter.

If either BEA JOLT bulk loader file or Tuxedo configuration and FML or VIEW source files describing the adapter are available, you can import them using the Metadata Import wizard in Oracle Studio Design perspective to generate interaction metadata. If either BEA JOLT bulk loader file or Tuxedo configuration and FML or VIEW source files describing the input and output structures are not available, you must manually define the metadata. For more information of the metadata definition, see [Advanced Tuning of the Metadata](#).

The following information is required during the import procedure:

BEA JOLT bulk loader file or Tuxedo configuration and FML or VIEW source files. These are copied to the computer running Oracle Studio as part of the import procedure.

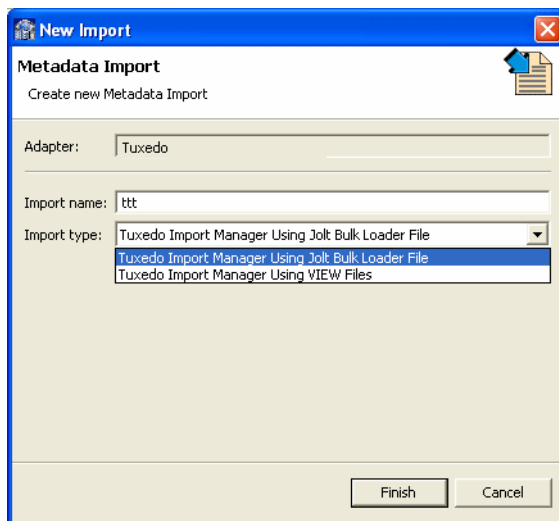
Using Oracle Studio, complete the Metadata Import steps to generate interaction metadata, as follows:

1. From the **Start** menu, select **Programs, Oracle** and then select **Studio**.
2. In the Configuration Explorer, expand the machine where you are using the Tuxedo adapter defined in [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#).
3. Expand **Bindings**. The binding configurations available on this computer are listed.
4. Expand the binding with the adapter you are using.
5. Expand the **Adapters**.
6. Right-click the Tuxedo adapter defined in [Configuring a Tuxedo OracleAS Adapter](#).
7. Select **Show Metadata View** to open the Metadata tab, with the Tuxedo adapter displayed under the adapters list.
8. Right-click the Tuxedo adapter and select **New Import**. The New Metadata Import screen is displayed.
9. Enter a name for the import. The name can contain letters and numbers and the underscore character only.
10. Select the metadata import type. You can import the metadata either from a BEA JOLT file or from Tuxedo record definition files (FML or VIEW files).

Note: A JOLT file consists of definitions of Tuxedo services along with the metadata, using the BEA JOLT Bulk Loader. FML and VIEW files are metadata files used by Tuxedo with a configuration file that includes the Tuxedo services.

For a JOLT file, select **Tuxedo Import Manager Using Jolt Bulk Loader File**. For FML or VIEW files, select **Import Manager Using VIEW Files** as shown in the following figure:

Figure 2–6 The Metadata Import screen



11. Click **Finish**. The Metadata Import wizard is displayed.

If JOLT files are used, see [Getting a JOLT Input File for Tuxedo Adapter](#) and follow the instructions.

If FML or VIEW files are used, see [Getting an FML or VIEW Import File for Tuxedo Adapter](#).

The metadata is imported based on the options specified and it is stored on the computer where OracleAS Adapter for Tuxedo runs. An XML representation of the metadata is generated.

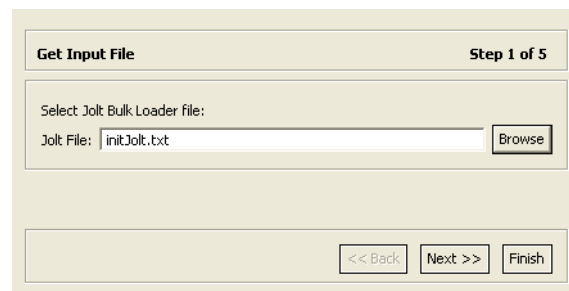
After the import is complete, you can view the metadata in Oracle Studio Design perspective **Metadata** tab. You can also make any fine adjustments to the metadata and maintain it, as necessary.

See Also: [Advanced Tuning of the Metadata](#) for details about fine tuning the adapter metadata

Getting a JOLT Input File for Tuxedo Adapter

If you selected the **Tuxedo Import Manager using JOLT Bulk Loader File**, the first step is **Get Input Files**.

Figure 2–7 *Get Input Files (JOLT)*



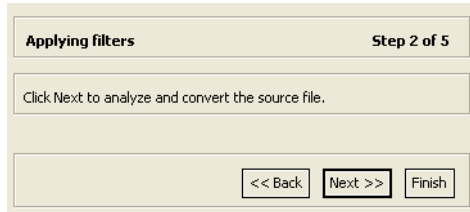
In this step you must select the JOLT file where you are importing the metadata. Do the following to select a JOLT file.

1. In the Get Input File step that is displayed in the Editor, click **Browse**. The Select From Resources screen is displayed, which provides the option to select files from the local computer or to copy the files from another computer.
2. If the files are on another computer, right-click My FTP Sites and select **Add** or double-click **Add FTP site**. The Add FTP Site screen is displayed.
3. Enter the name or IP address of the server where the JOLT bulk loader file or the FML or VIEW files reside and enter a valid OracleAS Adapter and password to access the computer (if anonymous connection is used, select Anonymous connection), then click **OK**.
4. Right-click the required computer and select **Set Transfer Type**. Enter the transfer type (ASCII or BINARY) and click **OK**.
5. Click the added site and locate the necessary file. To change the root directory, right-click the computer and select Change Root Directory. Enter the root directory and click **OK**.
6. Select the required file or files and click **Finish**. The selected file or files are displayed in the **Get Input File** step.
7. Click **Next** to for [Applying Filters for Tuxedo Adapter \(JOLT\)](#).

Applying Filters for Tuxedo Adapter (JOLT)

In the Applying Filters step the filters are applied automatically as shown in the figure below.

Figure 2–8 The Apply Filters screen

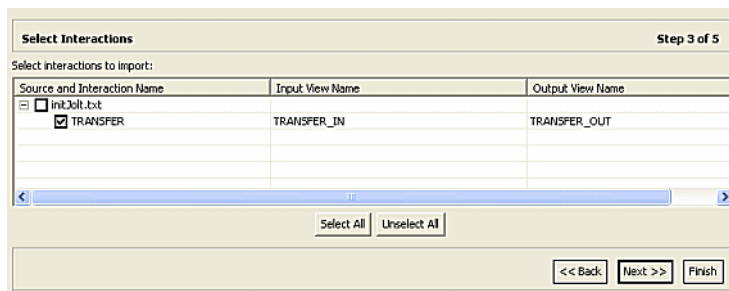


Click **Next** to analyze and convert the source file. The next step is for [Generating Interactions \(JOLT\)](#).

Generating Interactions (JOLT)

The **Select Interactions** step is used to select the interactions to be used. This step is displayed below. The screen lists the services from the input JOLT file as interactions, along with the input and output structures to use with each interaction.

Figure 2–9 The Select Interactions screen

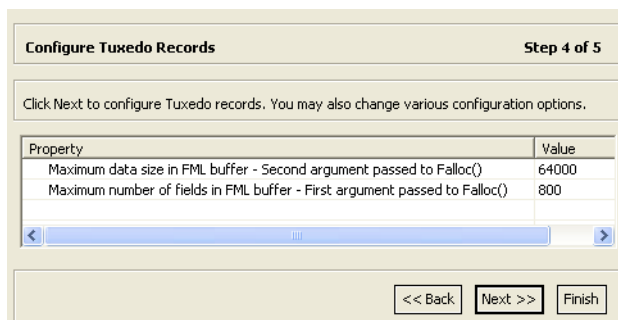


Select the interactions to implement from the list, and click **Next**. The next step for [Configuring Tuxedo Records for Tuxedo Adapter](#).

Configuring Tuxedo Records for Tuxedo Adapter

The Configure Tuxedo Records screen is displayed. It is used to specify how the Tuxedo records should be configured, as shown in the following figure:

Figure 2–10 The Configure Tuxedo Records screen



Click **Next** for [Importing the Metadata Definitions for Tuxedo Adapter](#).

Importing the Metadata Definitions for Tuxedo Adapter

In the Import Metadata step, select **Yes** to transfer the metadata from the Windows computer to the computer where the Tuxedo Gateway adapter runs, and then click **Finish**.

The metadata is imported based on the options specified and it is stored on the computer where OracleAS Adapter for Tuxedo runs. An XML representation of the metadata is generated.

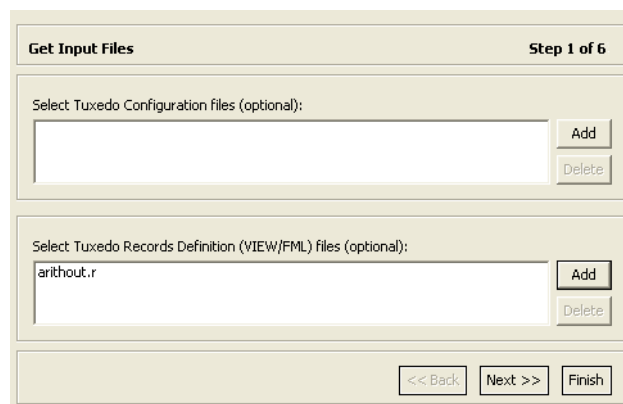
After performing the import, you can view the metadata in Oracle Studio Design perspective **Metadata** tab. You can also make any fine adjustments to the metadata and maintain it, as necessary.

See Also: [Advanced Tuning of the Metadata](#) for details about fine tuning the adapter metadata

Getting an FML or VIEW Import File for Tuxedo Adapter

In this step you can add Tuxedo configuration files and Tuxedo record definition files (FML or VIEW). The step is displayed in the Editor as shown below.

Figure 2–11 Get Input Files Step



This step is optional for both types of files. To get the import files, do the following.

1. In the Get Input Files step that is displayed in the Editor, click **Add** for either type of file. The **Add** button is at the top and bottom of the screen. The Select Resources screen is displayed, which provides the option to select files from the local computer or to copy the files from another computer.
2. If the files are on another computer, right-click My FTP Sites and select **Add** or double-click **Add FTP site**. The Add FTP Site screen is displayed.
3. Enter the name or IP address of the server where the JOLT bulk loader file or the FML or VIEW files reside and enter a valid OracleAS Adapter and password to access the computer (if anonymous connection is used, select Anonymous connection), then click **OK**.
4. Right-click the required computer and select Set Transfer Type. Enter the transfer type (ASCII or BINARY) and click **OK**.

5. Click the added site and locate the necessary file. To change the root directory, right-click the computer and select Change Root Directory. Enter the root directory and click **OK**.
6. Select the required file or files and click **Finish**. The selected file or files are displayed in the **Get Input File** step.

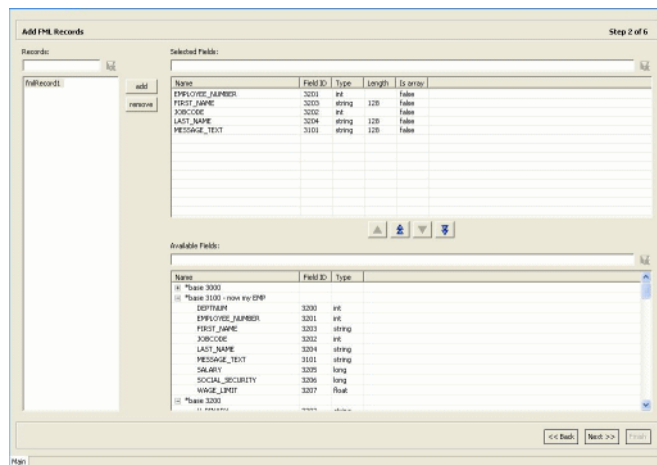
Note: Each file should be of equal size.

7. Click Next for [Adding FML Records for Tuxedo Adapter \(FML and View\)](#).

Adding FML Records for Tuxedo Adapter (FML and View)

The Add FML records step is displayed in the Editor as shown below.

Figure 2–12 Get Tuxedo Records Step



In this step, add FML records by selecting fields from the files imported in the first step. If you did not import any files, you can skip this step and add records in the [Getting Tuxedo Records for Tuxedo Adapter \(FML and View Files\)](#) step.

To add FML records

1. In the Records column, enter a name for the record.

Notes:

- Use an expression to filter the records displayed in the list.
 - User the + and - buttons to change the record order in the list.
-

2. From the **Available Fields** section, select a field or multiple fields to add to the record.
3. Click the Use Selected Button to add the selected fields to the record. The Use Selected button is a blue triangle that points up.

Note: To use all of the fields, click the Use All button (the button with the double triangle pointing up).

- Click **Next** for [Getting Tuxedo Records for Tuxedo Adapter \(FML and View Files\)](#).

To remove fields from an FML record

- Select a record from the Records column.
- From the **Selected Fields** section, select a field or multiple fields to remove from the record.
- Click the Remove Selected button to remove the selected fields. The Remove Selected button is a blue triangle that points down.

Note: To remove all of the fields, click the Remove All button (the button with the double triangle pointing down).

Getting Tuxedo Records for Tuxedo Adapter (FML and View Files)

The Get Tuxedo Records step is displayed in the Editor as shown below.

Figure 2–13 Get Tuxedo Records Step

Record Name	Field Type	Size
Record1	STRING	8

The User can add records after the following actions:
 Press on Add Record button.
 Select Field Type by clicking on the Field Type cell.
 Set Size when the type is not XML.

In this step, add additional simple record definitions that are not included in the FML or VIEW files used for the import.

Enter the name of the record and select the buffer type from the list in the **Field Type** field.

Select one of the following message buffer types:

- A message buffer of type `STRING` is wrapped within a record containing a single field of type string.
- A message buffer of type `CARRAY` is wrapped within a record containing a single field of type binary with a fixed size.
- A message buffer of type `XML` is wrapped within a record containing a single field of type XML.

Click **Next** for [Generating Interactions for Tuxedo Adapter \(FML and View Files\)](#).

Generating Interactions for Tuxedo Adapter (FML and View Files)

In the Add Interaction step, you can add interactions to the Tuxedo Gateway adapter. The Add Interaction screen is displayed in the following figure.

Click **Add** to add as many interactions as required. Provide the following information for each interaction:

Figure 2–14 The Add Interactions screen

Add Interactions Step 3 of 6

Name	Mode	Input	Output	Description	Add
interaction1	sync-send-r...				Remove

You need to supply records to continue the import process

Interaction-Specific Parameters

Input Buffer Type:

Output Buffer Type:

No Transaction No Reply Expected

No Blocking Request No Timeouts

Signal Restart

Interaction Type:

Queue Space Name:

Queue Name:

Event Name:

Main

- **Name:** The name of the interaction. You can change the default name.
- **Mode:** The interaction mode. You can select one of the following:
 - `sync-send-receive`: The interaction sends a request and expects to receive a response. This is the default mode.
 - `sync-receive`: The interaction expects to receive a response.
 - `sync-send`: The interaction sends a request and does not expect to receive a response.
- **Input:** Identifies an input record. The input record is the data structure for the interaction. The records generated from the input files specified at the beginning of the procedure are listed. Select the relevant record for the interaction.

Note: You must specify an input record for each interaction before you can click **Next**.

If the interaction does not require an input record, the record specified here is ignored.

- **Output:** Identifies an output record. The output record is the data structure for the results of the interaction. The records generated from the input files specified at the beginning of the procedure are listed. Select the relevant record for the interaction.

The records include those generated from the input files, and those defined in the previous step

Note: You must specify an output record for the interaction if the mode is set to `sync-send-receive` or `sync-receive`, before you can click **Next**.

- **Description:** Free text describing the interaction.
- **Interaction-Specific Parameters:** Tuxedo specific parameters, as listed in the following table:

Table 2–1 Interaction-specific Parameters

Parameter	Description
Input Buffer Type	The type of buffer used for the input.
Output Buffer Type	The type of the buffer to use for the results of an outbound interaction.
No Transaction	Enables to run a service, regardless of transaction context. This parameter should always be checked.
No Reply Expected	For future use.
No Blocking Request	Avoids a FROM request submission if a blocking condition exists.
No Timeouts	Ignores blocking timeouts.

- **Signal Restart:** When selected, if a signal interrupts any underlying system calls, the interrupted system call is reissued.
- **Interaction Type:** Your options are:
 - **SERVICE:** Sends a request and synchronously awaits its reply. This is the default setting.
 - **ENQUEUE:** Places a message in the queue.
If you select this option, you must also fill in the fields Queue Space Name and Queue Name with the relevant values to assign the queue where the message is to be placed.
 - **POST:** Places a message into one or more queues, according to rules and filters predefined according to events that have been subscribed to.
If you select this option, you must also fill in the field Event Name.

Click **Next** for [Configuring Tuxedo Records for Tuxedo Adapter](#).

Starting Interactions Metadata Import

Inbound interactions are defined as events in Oracle Studio. An event adapter is defined automatically when you selected the Tuxedo adapter with **Create event queue for the adapter** selected, as described in [Configuring a Tuxedo OracleAS Adapter](#). The event adapter is defined with the same name as the adapter with the word event appended to it.

The adapter and the event adapter are linked by Oracle Studio. You can skip from the adapter definition to the event definition by right-clicking the adapter, or event, in the Configuration explorer, and selecting **Linked Event** or **Linked Adapter**, respectively.

The event adapter requires metadata describing the inbound interactions, including the structure used to pass the information.

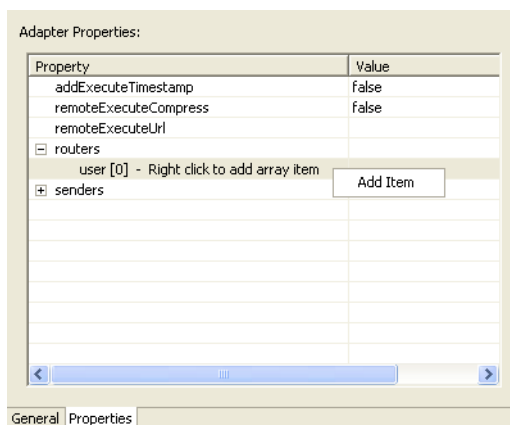
Note: The generation of inbound interactions involves similar steps to the steps described to generate outbound interactions. For more information, see [Starting Outbound Interactions Metadata Import](#).

During the import procedure, the BEA JOLT bulk loader file or Tuxedo configuration and FML or VIEW source files are used. These are copied to the computer running Oracle Studio as part of the import procedure.

Perform the Metadata Import steps to generate inbound interaction metadata, as follows:

1. In the Configuration Explorer, right-click the Tuxedo adapter defined in [Configuring a Tuxedo OracleAS Adapter](#).
2. Select **Linked Event** to skip to the event adapter.
3. Right-click the event adapter and select **Open**.
4. Click the **Properties** tab to add the names of Oracle Application Server users who can retrieve inbound interactions and users who can send inbound interactions.
5. To add Oracle Application Server users, expand the **Routers**, right-click **user**, and then select **Add Item**, to open the following screen:

Figure 2–15 The Adapter Properties screen



6. Enter the name of the Oracle Application Server user in the **Value** column for the item added.
7. To add users, expand the **Senders**, right-click **Users**, and then select **Add item**. A new entry is added under **User**.
8. Enter the name of the user in the **Value** column for the item added.
9. Click **Save** to save the changes.
10. Right-click the event adapter in the Configuration Explorer and select **Show Metadata View** to display the **Metadata** tab with the event adapter displayed under the Events list.

11. Right-click **Imports** and select **New Import**. The New Import screen is displayed.
12. Enter a name for the import. The name can contain letters, numbers and the underscore character only.
13. Select **Event Queue Import Manager Using Tuxedo View/FML Files** as the import type, and then click **Finish**.

After defining an import type, the Metadata Import wizard opens in Oracle Studio.

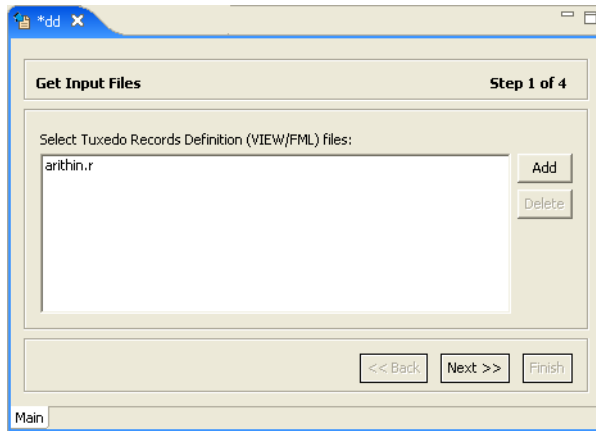
14. In the Get Input Files step that is displayed in the Editor, click **Add** for either type of file. The **Add** button is at the top and bottom of the screen. The Select Resources screen is displayed, which provides the option to select files from the local computer or to copy the files from another computer.
15. If the files are on another computer, right-click **My FTP Sites** and select **Add** or double-click **Add FTP Site**. The Add FTP Site screen is displayed.
16. Enter the server name or IP address where the FML or VIEW files reside and enter a valid OracleAS Adapter and password to access the computer (if anonymous access is used, select Anonymous connection), then click **OK**. The FTP site is added to the list of available sites.

Note: The selected server is accessed using the OracleAS Adapter as the high-level qualifier, enabling you to browse and transfer files.

17. Right-click the computer and select **Set Transfer Type**. Enter the transfer type (ASCII or BINARY) and click **OK**.
18. Expand the added site and locate the necessary FML or VIEW files. To change the high-level qualifier, right-click the computer and select **Change Root Directory**. Enter the new high-level qualifier enclosed in quotes, and click **OK**.
19. Select the required FML or VIEW file or files and click **Finish**.

The selected file or files are displayed in the Metadata Import wizard, as shown in the following figure:

Figure 2–16 The Get Input Files screen



20. Click **Next**.
21. Click **Next** to analyze and convert the selected FML or VIEW files.
The Add Events screen is displayed.
22. Click **Add** to add an event.
The Add Events screen is displayed.
23. Add as many interactions as required. For each interaction, provide the required information, as listed in the following table:

Table 2–2 Interaction Parameters

Property	Description
Name	The name of the interaction. You can change the default name.
Mode	The Interaction mode.
async-send	The interaction sends a request and does not expects to receive a response.
Input	Identifies an input record. The input record is the data structure for the interaction. The records generated from the FML or VIEW files specified are listed. Select the relevant record for the interaction. Note: You must specify an input record for each interaction before you can click Next
Description	Free text describing the interaction.

24. Click **Next** to generate the metadata definitions for the adapter.

25. For **Do you want to transfer the metadata to the server**, select **Yes** to transfer the data from the Windows computer to the target platform, then click **Finish**.
26. The metadata is imported based on the options specified. An XML representation of the metadata is also generated. After performing the import, you can view the metadata in Oracle Studio Design perspective **Metadata** tab. You can also make any fine adjustments to the metadata and maintain it, as necessary.

See Also: [Advanced Tuning of the Metadata](#) for details about fine tuning the adapter metadata

Setting up the Tuxedo Queue OracleAS Adapter

The Tuxedo adapter is configured using Oracle Studio.

This section contains the following topic:

- [Modeling Interactions for the Tuxedo Queue OracleAS Adapter](#)

Modeling Interactions for the Tuxedo Queue OracleAS Adapter

Modeling interactions for OracleAS Adapter for Tuxedo Queue involves defining an Oracle Connect queue adapter using Oracle Studio.

All the definitions specified in Oracle Studio are written to the platform where OracleAS Adapter for Tuxedo Queue runs.

This section contains the following:

- [Configuring the Tuxedo Queue Adapter](#)
- [Generating Inbound Interactions](#)
- [Defining the Tuxedo Queue Unstructured Records](#)

Configuring the Tuxedo Queue Adapter

To work with Oracle Connect, you must configure the queue adapter definitions on the platform where OracleAS Adapter for Tuxedo Queue runs to handle the events from a Tuxedo Queue.

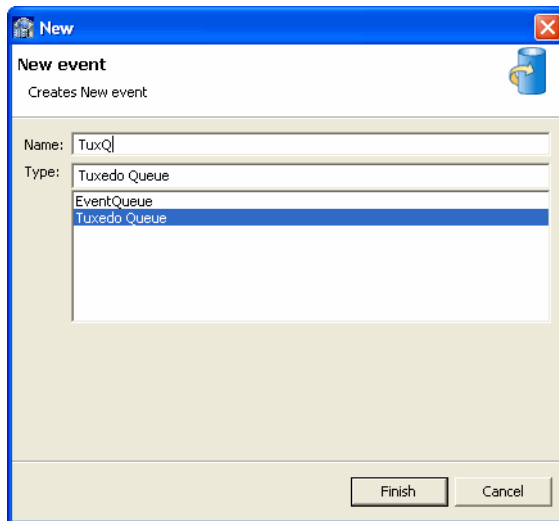
Perform the following steps to specify these definitions:

1. From the **Start** menu, select **Programs, Oracle** and then select **Studio**.
2. In the Configuration Explorer, expand the computer defined in [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#).
3. Expand the **Bindings**. The binding configurations available on this computer are listed.
4. Expand the binding where you want to define the adapter. The binding configuration includes branches for data sources and adapters that are located on the computer.
5. Right-click **Events** and select **New Event**.
The New Event wizard is displayed.
6. Enter a name for the event queue.

Note: The word *event* is a reserved word and cannot be used to name an event.

7. Select **Tuxedo Queue** as the event type from the **Type** list, as shown in the following figure:

Figure 2–17 The New Event screen



8. Click **Finish**.

Generating Inbound Interactions

The queue adapter requires metadata describing the inbound interaction, including its structure.

During the import procedure, Tuxedo FML or VIEW configuration source files are copied to the computer running Oracle Studio as part of the import procedure. Alternatively, this procedure enables you to manually define the queue adapter metadata.

Before generating the interactions, note the following:

- All the events described in a single Tuxedo Queue adapter, should have the same Tuxedo buffer type.
- In case that FML/FML32 buffer type is used, a common field with the same FBName must be included in all events. This field should contain the record/event name.
- The interaction is of `async-send` type (it does not expect to receive a response).

Perform the following steps to generate inbound interaction metadata:

1. In the Configuration Explorer, select the Tuxedo Queue adapter defined in [Configuring the Tuxedo Queue Adapter](#).
2. Right-click the queue adapter in the Configuration Explorer and select **Show Metadata View**.

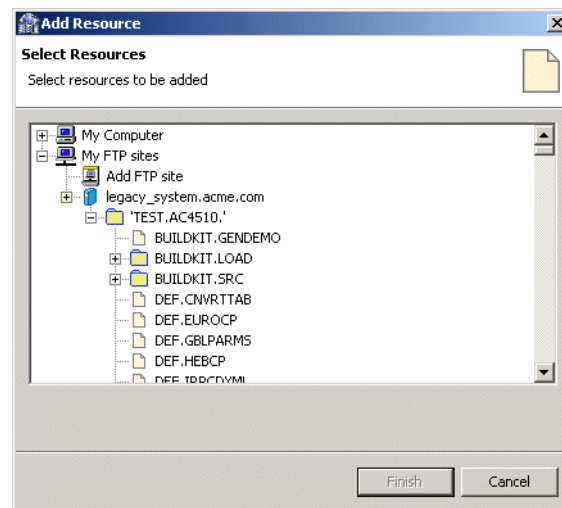
The **Metadata** tab is displayed, with the queue adapter displayed under the Adapters list.

3. Right-click **Imports** and select **New Import**.

The New Import screen is displayed.

4. Enter a name for the import. The name can contain letters, numbers and the underscore character only.
5. Select the import type from the **Import Type** list. You have the following options:
 - Tuxedo Queue Import Manager for XML/STRING/CARRAY Buffers: This option does not require importing any files. It enables you to manually define the required Tuxedo record.
 - Tuxedo Queue Import Manager for VIEW/VIEW32 Buffers:
 - Tuxedo Queue Import Manager for FML/FML32 Buffers:
6. Click **Finish**.
The next step depends on the selected import type. If either the VIEW/VIEW32 Buffers or the FML/FML32 Buffers options is selected, then the Metadata Import wizard opens, in which case, proceed to the following step. If the XML/STRING/CARRAY Buffers option is selected, proceed now to [Defining the Tuxedo Queue Unstructured Records](#).
7. Click **Add**.
8. The Select Resources screen is displayed, which provides the option to select files from the local computer or copy the files from another computer.
9. If the files are on another computer, right-click **My FTP Sites** and select **Add**. Optionally, double-click **Add FTP Site**. The Add FTP Site screen is displayed.
10. Enter the server name or IP address where the required reside and enter a valid OracleAS Adapter and password to access the computer (if anonymous access is used, select Anonymous connection), then click **OK**. The FTP site is added to the list of available sites.

Figure 2–18 The Select Resources screen



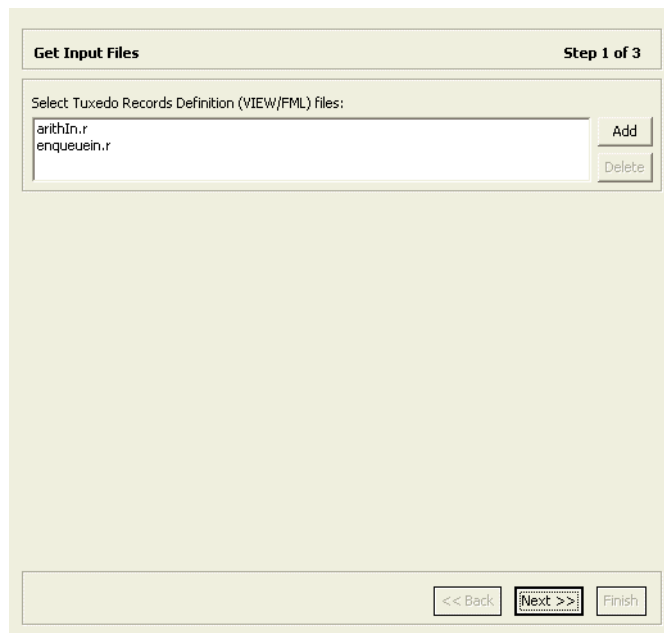
11. Right-click the computer and select **Set Transfer Type**. Enter the appropriate transfer type, and click **OK**.

12. Expand the added site and locate the files. To change the directory, right-click the computer and select **Change Root Directory**. Enter the new directory name, and click **OK**.

13. Select the required file or files and click **Finish**.

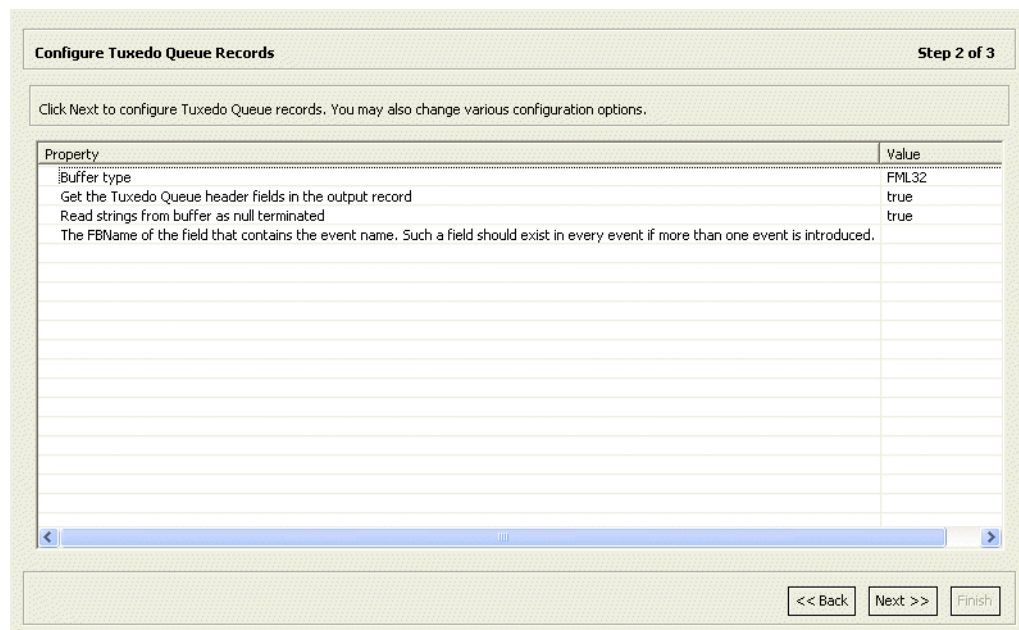
The selected file or files are displayed in the Metadata Import wizard, as shown in the following figure:

Figure 2–19 The *Get Input Files* screen



14. Click **Next**.

The Configure Tuxedo Records screen is displayed, as shown in the following figure:

Figure 2–20 The Configure Tuxedo Queue Records screen

15. Ensure that the settings are correct for the following properties:
 - Buffer type: Indicates the buffer type, as read from the FML/VIEW file. This property should not be modified.
 - Get Tuxedo Queue header field in the output record: Indicates that the header field of each record is read. The default setting is true.
 - Read strings from buffer as null terminated: Indicates that strings are handled as null terminated. The default setting is true.
16. Specify the FBName of the field that contains the event name. This field is common to all incoming events and it should include the record name. This property is required only for FML/FML32 files.
17. Click **Next** to generate the metadata definitions for the queue adapter.
18. Specify if you want to transfer the data to the server, then click **Finish**.

The Import wizard generates record structures, which are used for the record structures for the inbound interactions, and the metadata is imported based on the options specified and it is stored on the target platform.

An XML representation of the metadata is also generated. After performing the import, you can view the metadata in Oracle Studio Design perspective **Metadata** tab, under the **Imports** for the Queue adapter. You can also make any fine adjustments to the metadata.

See Also: [Advanced Tuning of the Metadata](#) for details about fine tuning the adapter metadata.

Defining the Tuxedo Queue Unstructured Records

You can manually define the required Tuxedo records when all the events in the Tuxedo queue are of the same type and are unstructured. Only one record is defined.

Perform the following steps to manually define the required Tuxedo records, as follows:

1. In the Configuration Explorer, select the Tuxedo Queue adapter defined in [Configuring the Tuxedo Queue Adapter](#).
2. Right-click the Tuxedo Queue adapter and select **Show Metadata View**.
The **Metadata** tab is displayed, with the queue adapter displayed under the Adapters list.
3. Right-click **Imports** and select **New Import**.
The New Import screen is displayed.
4. Enter a name for the import. The name can contain letters, numbers and the underscore character only.
5. Select **Tuxedo Queue Import Manager for XML/STRING/CARRAY Buffers** from the **Import Type** list.
6. Click **Finish**.
The Get Tuxedo Records screen is displayed, as shown in the following figure:

Figure 2–21 The Get Tuxedo Records screen

Get Tuxedo Records Step 1 of 3

Add Records

Record Name	Field Type	Size

The User can add records after the following actions:
 Press on Add Record button.
 Select Field Type by clicking on the Field Type cell.
 Set Size when the type is not XML.

Pay Attention:
 Only one record can be added in order to be able to continue.

7. Click **Add Record**. A new record entry is added to the records list, with a default type.
8. Select the field type from the **Field Type** list. Your options are:
 - STRING (the default)
 - CARRAY
 - XML
 - X_OCTET
9. Specify the maximum buffer size in the Size column. This is not required if XML was selected as the field type.
10. Click **Next**.

11. Specify if you want to transfer the data to the target platform, then click **Finish**.

The record structure is generated, and the metadata is imported to and stored on the target platform.

An XML representation of the metadata is also generated. After performing the import, you can view the metadata in Oracle Studio Design perspective **Metadata** tab, under the **Imports** for the Queue adapter. You can also make any fine adjustments to the metadata.

See Also: [Advanced Tuning of the Metadata](#) for details about fine tuning the adapter metadata.

Setting up the Tuxedo Gateway OracleAS Adapter

The Tuxedo Gateway adapter is configured using Oracle Studio. When you set up the adapter, a new Binding and Workspace with the same name as the new Tuxedo Gateway adapter are automatically created. The adapter is added to the new binding. You also import metadata to the adapter with the Oracle Studio import manager.

Note: In the Workspace property sections, ensure not to change the values for the following:

- **Server type:** This must be left to its default setting of `external`. For more information, see [General](#).
 - **Workspace server mode:** This must be left to its default setting of `multiClient`. For more information, see [Server Mode](#).
-
-

When you create interactions in Oracle Studio, each interaction defined on the gateway, represents a service exposed by the Gateway as tuxedo server service.

Before you start setting up the Tuxedo Gateway adapter, ensure that the daemon is running.

To set up the Tuxedo Gateway adapter, you should:

- [Configure the Tuxedo Gateway as a Regular Adapter](#)
- [Configure a Non-Transactional Tuxedo Gateway as a Tuxedo Server](#)

Configure the Tuxedo Gateway as a Regular Adapter

Use Oracle Studio to configure the OracleAS Tuxedo Gateway.

The following topics describe how to configure the interactions (services):

- [Configuring the Tuxedo Gateway Adapter in Oracle Studio](#)
- [Setting Properties for the Tuxedo Gateway Adapter](#)
- [Starting a Metadata Import for Tuxedo Gateway Adapter](#)
- [Getting a JOLT Input File for Tuxedo Gateway Adapter](#)
- [Applying Filters for Tuxedo Gateway Adapter \(JOLT\)](#)
- [Generating Inbound Synchronous Interactions for Tuxedo Gateway Adapter \(JOLT\)](#)
- [Configuring Tuxedo Records for Tuxedo Gateway Adapter](#)

- [Importing the Metadata Definitions for Tuxedo Gateway Adapter](#)
- [Getting an FML or VIEW Import File for Tuxedo Gateway Adapter](#)
- [Adding FML Records for Tuxedo Gateway Adapter \(FML and View\)](#)
- [Getting Tuxedo Records for Tuxedo Gateway Adapter \(FML and View Files\)](#)
- [Generating Inbound Synchronous Interactions for Tuxedo Gateway Adapter \(FML and View Files\)](#)

Configuring the Tuxedo Gateway Adapter in Oracle Studio

This section provides details on how to use the Oracle Studio to create an Oracle Tuxedo Gateway adapter.

To work with Oracle Connect, you must configure the adapter definition on the platform where OracleAS Adapters for Tuxedo run to handle the interactions to and from a Tuxedo service.

Perform the following steps using Oracle Studio:

1. From the **Start** menu, select **Programs, Oracle** and then select **Studio**.
2. In the Configuration Explorer, right-click the machine and select **New** and then **Tuxedo Gateway**. The following is displayed.

Figure 2–22 Tuxedo Gateway Adapter Configuration



3. Enter the following information:
 - **Name:** Enter a name for the new adapter. When you finish this process a new adapter, binding, and workspace are created. The binding and workspace have the same name as the adapter. Do *not* use a name that is in use as a binding or workspace on your machine.
 - **Machine:** Select a machine where you want to configure the new adapter. By default, the machine you selected to open the dialog box is selected.
 - **Daemon:** Select the daemon where you want to create the adapter's workspace.
4. Click **Finish** to create the new adapter, workspace, and binding.

Setting Properties for the Tuxedo Gateway Adapter

To edit the Tuxedo Gateway adapter properties, open the Oracle Studio Design perspective and find the binding for the Tuxedo Gateway adapter. The binding has the same name you gave to the Tuxedo Gateway adapter. For information on how to create a Tuxedo Gateway adapter, see [Configuring the Tuxedo Gateway Adapter in Oracle Studio](#).

Do the following to edit the Tuxedo Gateway adapter properties.

1. From the **Start** menu, select, **Programs, Oracle**, and then select **Studio**.
2. Expand the **Machines** folder.
3. Expand the server machine that you are working with.
4. Expand the **Bindings** folder, and then expand the binding the name of Tuxedo Gateway adapter that you are working with.
5. Expand the **Adapter** folder.
6. Right-click the adapter and select **Open**.

The adapter configuration editor opens in the editor, which displays the properties for the adapter.

7. Edit the adapter parameters as required.

`logLevel`: To set the logging level for the Tuxedo Gateway adapter select from the following log levels. The log levels are listed in order of completeness (None means that no logs are produced, and Debug produces logs with all possible information).

- None
- API
- Info
- Debug

Note: The Tuxedo Gateway adapter logging goes to the Tuxedo log.

`timeout`: Sets the service request timeout in the Oracle Tuxedo Gateway Server. Enter a time in seconds.

`tmsConnectTimeout`: Sets the connection timeout in the Oracle Tuxedo Gateway TMS. If the transaction level is 0PC, this property is not relevant. The default value is 30 seconds. For information on setting this property, see [Starting the Tuxedo Gateway Transaction Group](#).

`transactionLevelSupport`: Enter the transaction type supported. The options are:

- 0PC: Does not support transactions
- 2PC: Supports XA transactions.

Starting a Metadata Import for Tuxedo Gateway Adapter

Oracle Connect requires metadata describing the adapter interactions, including the structures used to pass information to and from the adapter.

If either BEA JOLT bulk loader file or Tuxedo configuration and FML or VIEW source files describing the adapter are available, you can import them using the Metadata Import wizard in Oracle Studio to generate interaction metadata. If either BEA JOLT

bulk loader file or Tuxedo configuration and FML or VIEW source files describing the input and output structures are not available, you must manually define the metadata.

The following information is required during the import procedure:

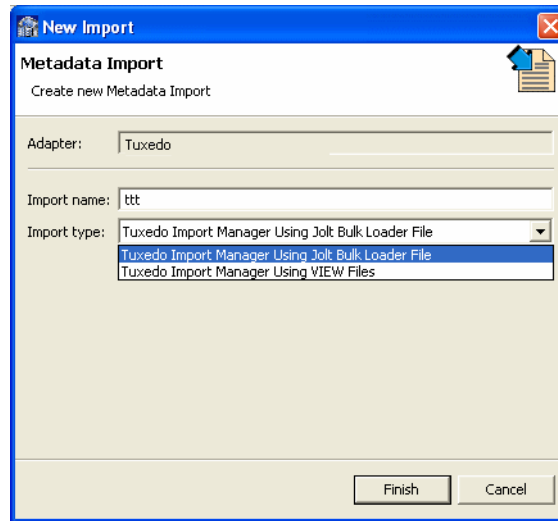
BEA JOLT bulk loader file or Tuxedo configuration and FML or VIEW source files. These are copied to the computer running Oracle Studio as part of the import procedure.

Using Oracle Studio, complete the Metadata Import steps to generate interaction metadata, as follows:

1. From the **Start** menu, select **Programs, Oracle** and then select **Studio**.
2. In the Configuration Explorer, expand the machine where you are using the Tuxedo Gateway adapter. For more information, see [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#).
3. Expand **Bindings**. The available binding configurations are listed.
4. Expand the binding that was created when you created the adapter. This binding has the same name as the Tuxedo Gateway adapter.
5. Expand the **Adapters**.
6. Right-click the Tuxedo Gateway adapter defined in [Configuring the Tuxedo Gateway Adapter in Oracle Studio](#).
7. Select **Show Metadata View** to open the Metadata tab, with the Tuxedo Gateway adapter displayed under the adapters list.
8. Right-click the Tuxedo Gateway adapter and select **New Import**. The New Metadata Import screen is displayed.
9. Enter a name for the import. The name can contain letters and numbers and the underscore character only.
10. Select the metadata import type. You can import the metadata either from a BEA JOLT file or from Tuxedo record definition files (FML or VIEW files).

Note: A JOLT file consists of definitions of Tuxedo services along with the metadata, using the BEA JOLT Bulk Loader. FML and VIEW files are metadata files used by Tuxedo with a configuration file that includes the Tuxedo services.

For a JOLT file, select **Tuxedo Import Manager Using Jolt Bulk Loader File**. For FML or VIEW files, select **Import Manager Using VIEW/FML Files** as shown in the following figure:

Figure 2–23 The Metadata Import screen

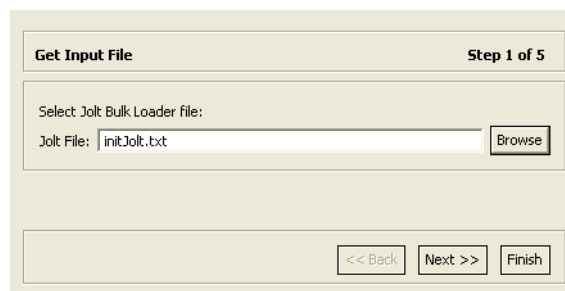
11. Click **Finish**. The Metadata Import wizard is displayed.

If JOLT files are used, see [Getting a JOLT Input File for Tuxedo Adapter](#) and follow the instructions.

If FML or VIEW files are used, see [Getting an FML or VIEW Import File for Tuxedo Adapter](#).

Getting a JOLT Input File for Tuxedo Gateway Adapter

If you selected the **Tuxedo Import Manager using JOLT Bulk Loader File**, the first step is **Get Input Files**.

Figure 2–24 Get Input Files (JOLT)

In this step you must select the JOLT file where you are importing the metadata. Do the following to select a JOLT file.

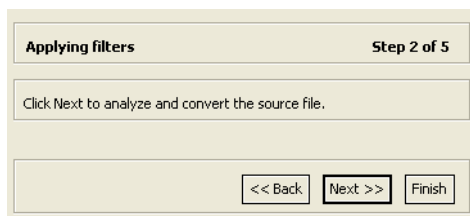
1. In the Get Input File step that is displayed in the Editor, click **Browse**. The Select Resources screen is displayed, which provides the option to select files from the local computer or to copy the files from another computer.
2. If the files are on another computer, right-click My FTP Sites and select **Add** or double-click **Add FTP site**. The Add FTP Site screen is displayed.
3. Enter the name or IP address of the server where the JOLT bulk loader file or the FML or VIEW files reside and enter a valid OracleAS Adapter and password to access the computer (if anonymous connection is used, select Anonymous connection), then click **OK**.

4. Right-click the required computer and select Set Transfer Type. Enter the transfer type (ASCII or BINARY) and click **OK**.
5. Click the added site and locate the necessary file. To change the root directory, right-click the computer and select Change Root Directory. Enter the root directory and click **OK**.
6. Select the required file or files and click **Finish**. The selected file or files are displayed in the **Get Input File** step.
7. Click **Next** to for [Applying Filters for Tuxedo Adapter \(JOLT\)](#).

Applying Filters for Tuxedo Gateway Adapter (JOLT)

In the Applying Filters step the filters are done automatically as shown in the figure below.

Figure 2–25 The Apply Filters screen



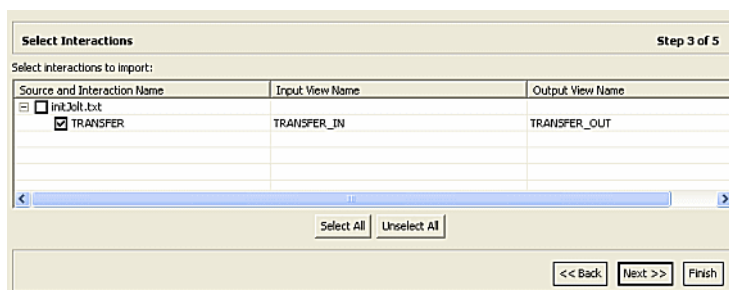
Click **Next** to analyze and convert the source file. The next step is for [Generating Interactions \(JOLT\)](#).

Generating Inbound Synchronous Interactions for Tuxedo Gateway Adapter (JOLT)

The **Select Interactions** step is used to select the interactions to be used. This step is displayed below. The screen lists the services from the input JOLT file as interactions, along with the input and output structures to use with each interaction.

Note: Interactions in Oracle Connect are the Tuxedo Services that you are using.

Figure 2–26 The Select Interactions screen



Select the interactions to implement from the list, and click **Next**. The next step for [Configuring Tuxedo Records for Tuxedo Adapter](#).

Configuring Tuxedo Records for Tuxedo Gateway Adapter

The Configure Tuxedo Records screen is displayed. It is used to specify how the Tuxedo records should be configured, as shown in the following figure:

Figure 2–27 The Configure Tuxedo Records screen

Property	Value
Maximum data size in FML buffer - Second argument passed to Falloc()	64000
Maximum number of fields in FML buffer - First argument passed to Falloc()	800

Click **Next** for [Importing the Metadata Definitions for Tuxedo Adapter](#).

Importing the Metadata Definitions for Tuxedo Gateway Adapter

In the Import Metadata step, select **Yes** to transfer the metadata from the computer with Oracle Studio to the computer where the Tuxedo Gateway adapter runs, and then click **Finish**.

The metadata is imported based on the options specified and it is stored on the computer where OracleAS Adapter for Tuxedo runs. An XML representation of the metadata is generated.

After performing the import, you can view the metadata in Oracle Studio Design perspective **Metadata** tab. You can also make any fine adjustments to the metadata and maintain it, as necessary.

See Also: [Advanced Tuning of the Metadata](#) for details about fine tuning the adapter metadata

Getting an FML or VIEW Import File for Tuxedo Gateway Adapter

In this step you can add Tuxedo configuration files and Tuxedo record definition files (FML or VIEW). The step is displayed in the Editor as shown below.

Figure 2–28 Get Input Files Step

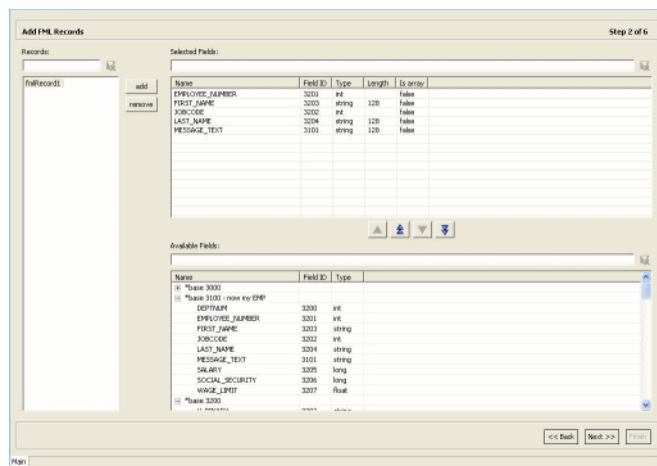
This step is optional for both types of files. To get the import files, do the following.

1. In the Get Input Files step that is displayed in the Editor, click **Add** for either type of file. The **Add** button is at the top and bottom of the screen. The Select Resources screen is displayed, which provides the option to select files from the local computer or to copy the files from another computer.
2. If the files are on another computer, right-click **My FTP Sites** and select **Add** or double-click **Add FTP site**. The Add FTP Site screen is displayed.
3. Enter the name or IP address of the server where the JOLT bulk loader file or the FML or VIEW files reside and enter a valid OracleAS Adapter and password to access the computer (if anonymous connection is used, select Anonymous connection), then click **OK**.
4. Right-click the required computer and select Set Transfer Type. Enter the transfer type (ASCII or BINARY) and click **OK**.
5. Click the added site and locate the necessary file. To change the root directory, right-click the computer and select Change Root Directory. Enter the root directory and click **OK**.
6. Select the required file or files and click **Finish**. The selected file or files are displayed in the **Get Input File** step.
7. Click **Next** for [Adding FML Records for Tuxedo Gateway Adapter \(FML and View\)](#).

Adding FML Records for Tuxedo Gateway Adapter (FML and View)

The Add FML records step is displayed in the Editor as shown below.

Figure 2–29 Get Tuxedo Records Step



In this step, add FML records by selecting fields from the files imported in the first step. If you did not import any files, you can skip this step and add records in the [Getting Tuxedo Records for Tuxedo Gateway Adapter \(FML and View Files\)](#) step.

To add FML records

1. In the Records column, enter a name for the record.

Notes:

- Use an expression to filter the records displayed in the list.
- User the + and - buttons to change the record order in the list.

2. From the **Available Fields** section, select a field or multiple fields to add to the record.
3. Click the Use Selected Button to add the selected fields to the record. The Use Selected button is a blue triangle that points up.

Note: If you want to use all of the fields, click the Use All button (the button with the double triangle pointing up).

4. Click **Next** for [Getting Tuxedo Records for Tuxedo Gateway Adapter \(FML and View Files\)](#).

To remove fields from an FML record

1. Select a record from the Records column.
2. From the **Selected Fields** section, select a field or multiple fields to remove from the record.
3. Click the Remove Selected button to remove the selected fields. The Remove Selected button is a blue triangle that points down.

Note: If you want to remove all of the fields, click the Remove All button (the button with the double triangle pointing down).

Getting Tuxedo Records for Tuxedo Gateway Adapter (FML and View Files)

The Get Tuxedo Records step is displayed in the Editor as shown below.

Figure 2-30

Get Tuxedo Records Step 2 of 6

Add Records

Record Name	Field Type	Size
Record1	STRING	8

The User can add records after the following actions:
 Press on Add Record button.
 Select Field Type by clicking on the Field Type cell.
 Set Size when the type is not XML.

In this step, add additional simple record definitions that are not included in the FML or VIEW files used for the import.

Enter the name of the record and select the buffer type from the list in the **Field Type** field.

Select one of the following message buffer types:

- A message buffer of type `STRING` is wrapped within a record containing a single field of type string.
- A message buffer of type `CARRAY` is wrapped within a record containing a single field of type binary with a fixed size.
- A message buffer of type `XML` is wrapped within a record containing a single field of type XML.

Click **Next** for [Generating Interactions for Tuxedo Adapter \(FML and View Files\)](#).

Generating Inbound Synchronous Interactions for Tuxedo Gateway Adapter (FML and View Files)

In the Add Interaction step, you can add interactions to the Tuxedo Gateway adapter.

Note: Interactions in Oracle Connect are the Tuxedo Services that you are using.

Click **Add** to add as many interactions as required. Provide the following information for each interaction. The Add Interaction screen is displayed in the following figure:

Figure 2–31 The Add Interactions screen

Name	Mode	Input	Output	Description
interaction1	sync-send-receive	Record1	Record1	cdd

Interaction-Specific Parameters

Output Buffer Type: XML

- **Name:** The name of the interaction. You can change the default name.
- **Mode:** The interaction mode. You can select one of the following:
 - `sync-send-receive`: The interaction sends a request and expects to receive a response. This is the default mode.
 - `sync-receive`: The interaction (Tuxedo service) that expects to receive a response.
 - `sync-send`: The interaction sends a request and does not expect to receive a response.
- **Input:** Identifies an input record. The input record is the data structure for the interaction. The records generated from the input files specified at the beginning of the procedure are listed. Select the relevant record for the interaction.

Note: You must specify an input record for each interaction before you can click **Next**.

If the interaction does not require an input record, the record specified here is ignored.

- **Output:** Identifies an output record. The output record is the data structure for the results of the interaction. The records generated from the input files specified at the beginning of the procedure are listed. Select the relevant record for the interaction.

The records include those generated from the input files, and those defined in the previous step.

Note: You must specify an output record for the interaction if the mode is set to `sync-send-receive` or `sync-receive`, before you can click **Next**.

- Description: Free text describing the interaction.
- Interaction-Specific Parameters: Tuxedo specific parameters, described below:
 - **Output Buffer Type:** The type of the buffer to use for the results of an outbound interaction

Click **Next** for [Configuring Tuxedo Records for Tuxedo Adapter](#).

Configure a Non-Transactional Tuxedo Gateway as a Tuxedo Server

The non-transactional Tuxedo Gateway can be configured as a regular Tuxedo server that advertises the services (interactions) configured when you [Configure the Tuxedo Gateway as a Regular Adapter](#). This is done using the UBB configuration file.

In the UBB configuration file, add a new entry to the SERVERS section called `<acgw_tuxedo>`, where `acgw_tuxedo` is the name of the executable TNS provided in the Oracle Connect `bin` folder. You must use lowercase letters. The following is an example of the `acgw_tuxedo` entry:

```
"acgw_tuxedo"      SRVGRP="GROUP3" SRVID=3
  CLOPT=" -- -wMSGATEWAY"
  RQPERM=0666      REPLYQ=N          RPPERM=0666      MIN=1    MAX=10    CONV=N
  SYSTEM_ACCESS=FASTPATH
  MAXGEN=1         GRACE=86400       RESTART=N
  MINDISPATCHTHREADS=10  MAXDISPATCHTHREADS=10  THREADSTACKSIZE=0
  SICACHEENTRIESMAX="500"
```

The configuration in this entry is the standard Tuxedo Server configuration, which may also be used to configure the number of server occurrences and concurrently dispatched threads that each server process that is supported by the Tuxedo Gateway. In addition to the standard configuration, the following parameters should be passed to the Tuxedo Server using the CLOPT property:

- CLOPT=" -- -w<workspace_name> [-u<username> [-p<password>]] [-l[<ip>]:<port>]

Where:

- `workspace` is the name of the workspace where the Tuxedo Gateway adapter is defined. It has the same name as the adapter.

- Enter information for these parameters if you are using a secure daemon:
 - username is the name of a user with permission to access the daemon
 - password is the password for this user
- ip is the ip address of the computer where the daemon is running (optional)
- port is the port in which the daemon is running. This is optional unless the daemon is running on a port that is not the default port.

Note: The Tuxedo configuration must be set to thread capable.

After you finish configuring the Tuxedo Adapter in Oracle Studio and [Configure a Non-Transactional Tuxedo Gateway as a Tuxedo Server](#), you can start Tuxedo.

Note:

Ensure that you set the following in the Tuxedo environment before running Tuxedo:

For Windows:

* Add \$NAVROOT/bin to the PATH

For UNIX:

* Add \$NAVROOY/bin to the PATH

* Add \$NAVROOT/lib to the LD_LIBRARY_PATH

Configure a Transactional Tuxedo Gateway as a Tuxedo Group

The transactional Tuxedo Gateway should be defined as a separate Tuxedo group. This group should configure the Oracle Tuxedo Gateway TMS and the Oracle Tuxedo Gateway Server.

The TMS executable named `acgw_tuxedo_xa` is supplied in the Oracle Connect `bin` folder or directory. The supplied TMS is not always sufficient. If you cannot use this TMS, you can build a new one using the following parameters:

- XA_SWITCH: `acgw_xa_switch`
- XA library name: `tuxgateway.lib`

Note: Run the `nav_login` before running the `buildtms` command to build `acgw_tuxedo-xa`.

The Tuxedo Gateway Server advertises the services (interactions) configured when you [Configure the Tuxedo Gateway as a Regular Adapter](#).

The configuration is done by adding two new entries to the UBB configuration file. You must use lowercase letters for the value definitions.

You must add a new entry to the GROUPS section. This group should define the Oracle Tuxedo Gateway TMS as follows:

```
TMSNAME=acgw_tuxedo_xa OPENINFO="Oracle_BPEL:<connect information>"
```

In the TMS that you define, the connect information must match the connect information that is defined in your Tuxedo Server. The following is an example of a group entry:

```
"GROUP3"          LMID="TUX1"   GRPNO=3
TMSCOUNT=2 TMSNAME=acgw_tuxedo_xa OPENINFO="Oracle_BPEL:-wINTTEST5040 -1:2551"
```

Add a new entry to the **SERVERS** section called `<acgw_tuxedo>`, where `acgw_tuxedo` is the name of the executable TNS provided in the Oracle Connect `bin` folder. The following is an example of the `acgw_tuxedo` entry:

```
"acgw_tuxedo"      SRVGRP="GROUP3" SRVID=3
CLOPT=" -- -wMSGATEWAY"
RQPERM=0666      REPLYQ=N          RPPERM=0666      MIN=1    MAX=10    CONV=N
SYSTEM_ACCESS=FASTPATH
MAXGEN=1         GRACE=86400        RESTART=N
MINDISPATCHTHREADS=10  MAXDISPATCHTHREADS=10  THREADSTACKSIZE=0
SICACHEENTRIESMAX="500"
```

The configuration in this entry is the standard Tuxedo Server configuration, which may also be used to configure the number of server occurrences and concurrently dispatched threads that each server process that is supported by the Tuxedo Gateway. In addition to the standard configuration, the following parameters should be passed to the Tuxedo Server using the `CLOPT` property:

- `CLOPT=" -- <connect information>"`

The connect information is defined as follows:

```
-w<workspace_name> [-u<username>
[-p<password>]] [-l[<ip>]:<port>]
```

Where:

- `workspace` is the name of the workspace where the Tuxedo Gateway adapter is defined. It has the same name as the adapter.
- Enter information for these parameters if you are using a secure daemon:
`username` is the name of a user with permission to access the daemon
`password` is the password for this user
- `ip` is the ip address of the computer where the daemon is running (optional)
- `port` is the port in which the daemon is running. This is optional unless the daemon is running on a port that is not the default port.

Notes: The Tuxedo configuration must be set to thread capable.

Note:

Ensure that you set the following in the Tuxedo environment before running Tuxedo:

For Windows:

* Add `$NAVROOY/bin` to the `PATH`

For UNIX:

* Add `$NAVROOY/bin` to the `PATH`

* Add `$NAVROOT/lib` to the `LD_LIBRARY_PATH`

Starting the Tuxedo Gateway Transaction Group

In most cases, you should not start the transaction group until at least one message driven bean application is activated in the Oracle Application Server for the Tuxedo Gateway adapter.

If you start the Tuxedo group when no message driven bean application is activated, TMS cannot start, therefore the group can be used only without transactions.

When activated, TMS waits for a connection from the resource adapter. The wait time is defined in the `tmsConnectTimeout` property of the Tuxedo Gateway adapter.

When determining the value for the `tmsConnectTimeout` property you should decide whether to use a large or small value. If you start the transaction group separately, it is recommended to use a large value, such as 9999999. In this case, the TMS waits for the connection for a long period. This gives time to activate the message driven bean when necessary.

If you start the entire Tuxedo configuration, it is recommended to use a small value for the `tmsConnectTimeout` property. In this case a large value may prevent other groups from starting.

Adapter Schema for the Tuxedo Gateway Adapter

The XML schema describing the adapter interactions can be viewed by selecting the **Source** tab when you view the metadata as XML. For more information, see [Appendix E, "Editing XML Files in Oracle Studio"](#).

Integrating OracleAS Adapters for Tuxedo with WebLogic

This section includes the following topics:

- [Overview of Integrating OracleAS Adapters for Tuxedo](#)
- [Configuring the Tuxedo Adapter for Outbound](#)
- [Configuring the Tuxedo Queue Inbound Adapter](#)
- [Configuring the Tuxedo Gateway Synchronous Inbound Adapter](#)

Overview of Integrating OracleAS Adapters for Tuxedo

Oracle Application Server provides a complete Java 2 Enterprise Edition (J2EE) environment that executes on the Java virtual machine (JVM) of the standard Java Development Kit (JDK). WebLogic is J2EE certified and provides all J2EE specific containers, APIs, and services. WebLogic supports the J2CA 1.5 standards.

J2CA defines standard Java interfaces for simplifying the integration of applications with the EIS. OracleAS Adapters are deployed as resource adapters within the WebLogic container.

The contract between the WebLogic client application and the resource adapter is defined by the common client interface (CCI). The contract between the WebLogic container and the resource adapter is defined by the service provider interface (SPI). The SPI API addresses the connection management, transaction management and the security management.

Connection management enables application components to connect to an EIS and leverage any connection pooling provided by the application server.

Transaction management enables an application server to use a transaction manager to manage transactions across multiple resource managers.

Lifecycle management contracts enable an application server to initialize a resource adapter instance during the deployment of the adapter or application server startup. In addition, it enables the application server to notify the resource adapter instance during server shutdown or undeployment of the adapter.

The lifecycle contract provides the mechanism for the application server to manage the lifecycle of the resource adapter instance.

Work management contracts enable the resource adapter to perform its logic by using threads dispatched by an application server, rather than creating threads on its own. The handshake is done through a `Work` instance submission. This makes the

application server threads management more efficient, providing better control over their execution contexts (such as security and transaction).

Message inflow, which refers to inbound communications from an EIS to the application server. For more information, see [Configuring the Tuxedo Queue Inbound Adapter](#).

See Also: Oracle Application Server Adapter Concepts Guide, Oracle Application Server Containers for J2EE User's Guide, Oracle Application Server Containers for J2EE Services Guide, and Oracle Application Server Containers for J2EE Security Guide.

Configuring the Tuxedo Adapter for Outbound

Deploy the Oracle Connect RAR into the WebLogic server.

You may create the appropriate connection factories for each Tuxedo Adapter.

This table describes the connection factory properties that are relevant for the Tuxedo Adapter. Do not change the values of any of the other properties.

Table 3–1 Connection Factory Properties for Outbound Interactions

Property	Description
<code>eisName</code>	Required. Sets the name of the adapter to use. The adapter is defined in the Oracle Connect server using Oracle Studio, as described in Setting up the Tuxedo OracleAS Adapter .
<code>serverName</code>	Required. Enter the name of the server with the TCP/IP address or host name where the Oracle Connect daemon is running. For more information, see Advanced Tuning of the Daemon for details about the daemon.
<code>workspace</code>	Optional. Specifies the name of an Oracle Connect server workspace to use. The default workspace is Navigator. For more information, see Workspaces for details about workspaces.
<code>portNumber</code>	Optional. Specifies the TCP/IP port where the Oracle Connect daemon is running on the server. The default port is 2551.
<code>userName</code>	Optional. Specifies a user who can access the Oracle Connect server. The user is defined in the Oracle Connect daemon configuration. For more information, see Daemon Security and Workspace Security for details about users allowed to access an Oracle Connect server.
<code>password</code>	Optional. Specifies a valid password for the user.
<code>firewallProtocol</code>	Optional. Specifies the firewall protocol used: either <code>none</code> or <code>fixedNat</code> (the Nat protocol using a fixed address for the daemon). The default is set to <code>none</code> .
<code>connectTimeout</code>	Optional. Specifies the connection timeout in seconds. The default is 0, indicating that there is no connection timeout.
<code>encryptionProtocol</code>	Optional. Specifies the name of encryption protocol to use. The default is set to RC4. If the value of the property is not defined, the RC4 protocol is used.

Table 3–1 (Cont.) Connection Factory Properties for Outbound Interactions

Property	Description
encryptionKeyName	Optional. Specifies the name of the symmetric encryption key to use.
encryptionKeyValue	Optional. Specifies the value of the symmetric encryption key to use.
fakeXa	For outbound interactions only. When set to <code>true</code> , the XA APIs are internally converted to local transaction APIs. The default is set to <code>false</code> .
useNamespace	Optional. Set to <code>true</code> or <code>false</code> . When set to <code>true</code> , the metadata record schema uses namespaces.
exposeEventStreamMetadata	Optional. When set to <code>true</code> , standard XSD schema is used for the Tuxedo/Q adapter inbound events metadata. When <code>false</code> , the native schema is used.

Configuring the Tuxedo Queue Inbound Adapter

Deploy the Oracle Connect RAR into the WebLogic server.

You may create the appropriate connection factories for each Tuxedo Queue Adapter.

This table describes the connection factory properties that are relevant for the Tuxedo Queue Adapter. Do not change the values of any of the other properties.

Table 3–2 Connection Factory Properties for Queue Inbound Processes

Property	Description
eisName	Required. Sets the name of the adapter to use. The adapter is defined in the Oracle Connect server using Oracle Studio, as described in Setting up the Tuxedo Queue OracleAS Adapter .
messagesInBatch	Specifies the maximum number of messages that can be moved to an endpoint in batch. The default is set to 50.
support2PC	Enables global transaction support, where applicable. The default is set to <code>False</code> . Do not set to <code>true</code> .
waitTime	Defines the maximum response time (in seconds) for Oracle Connect to send a response to the Tuxedo queue adapter. The default is set to 30 seconds.
retryInterval	Defines the sleep time after any detected problem before restarting inbound activity. The default is set to 15 seconds.
serverName	Required. Sets the TCP/IP address or host name where the Oracle Connect daemon is running. For more information, see Advanced Tuning of the Daemon for details about the daemon.
workspace	Optional. Specifies the name of an Oracle Connect server workspace to use. The default workspace is <code>Navigator</code> . For more information, see Workspaces for details about workspaces.

Table 3–2 (Cont.) Connection Factory Properties for Queue Inbound Processes

Property	Description
portNumber	Optional. Specifies the TCP/IP port where the Oracle Connect daemon is running on the server. The default port is 2551.
userName	Optional. Specifies a user who can access the Oracle Connect server. The user is defined in the Oracle Connect daemon configuration. For more information, see Daemon Security and Workspace Security for details about users allowed to access an Oracle Connect server
password	Optional. Specifies a valid password for the user.
firewallProtocol	Optional. Specifies the firewall protocol used: either <code>none</code> or <code>fixedNat</code> (the Nat protocol using a fixed address for the daemon). The default is set to <code>none</code> .
connectTimeout	Optional. Specifies the connection timeout in seconds. The default is 0, indicating that there is no connection timeout.
encryptionProtocol	Optional. Specifies the name of encryption protocol to use. The default is set to <code>null</code> . The RC4 protocol is supported.
encryptionKeyName	Optional. Specifies the name of the symmetric encryption key to use.
encryptionKeyValue	Optional. Specifies the value of the symmetric encryption key to use.
connectionFactory	This property is used only for the BPEL or ESB environment. It specifies the <code>connectionFactory</code> class instance that contains the adapter connection properties. If any connection property is defined on both the <code>activationSpec</code> and the <code>connectionFactory</code> , the <code>activationSpec</code> value is used.

Configuring the Tuxedo Gateway Synchronous Inbound Adapter

Deploy the Oracle Connect RAR into the WebLogic server.

You may create the appropriate connection factories for each CICS Queue Adapter.

This table describes the connection factory properties that are relevant for the Tuxedo Gateway Adapter. Do not change the values of any of the other properties.

Table 3–3 Connection Factory Properties for Synchronous Inbound Processes

Property	Description
eisName	Required. Sets the name of the adapter to use. The adapter is defined in the Oracle Connect server using Oracle Studio, as described in Setting up the Tuxedo Gateway OracleAS Adapter .
serviceName	Required. Sets the name of the Tuxedo service. The service is defined as an interaction in the Oracle Connect Server using Oracle Studio.

Table 3–3 (Cont.) Connection Factory Properties for Synchronous Inbound Processes

Property	Description
<code>waitTime</code>	Defines the maximum response time (in seconds) for Oracle Connect to send a response to the Tuxedo gateway adapter. The default is set to 30 seconds.
<code>retryInterval</code>	Defines the sleep time after any detected problem before restarting inbound activity. The default is set to 15 seconds.
<code>endpointTimeout</code>	Defines the timeout for endpoint thread activation (in seconds). The default is 600 seconds.
<code>serverName</code>	Required. Sets the TCP/IP address or host name where the Oracle Connect daemon is running. For more information, see Advanced Tuning of the Daemon for details about the daemon.
<code>portNumber</code>	Optional. Specifies the TCP/IP port where the Oracle Connect daemon is running on the server. The default port is 2551.
<code>userName</code>	Optional. Specifies a user who can access the Oracle Connect server. The user is defined in the Oracle Connect daemon configuration. For more information, see Daemon Security and Workspace Security for details about users allowed to access an Oracle Connect server.
<code>password</code>	Optional. Specifies a valid password for the user.
<code>firewallProtocol</code>	Optional. Specifies the firewall protocol used: either <code>none</code> or <code>fixedNat</code> (the Nat protocol using a fixed address for the daemon). The default is set to <code>none</code> .
<code>connectTimeout</code>	Optional. Specifies the connection timeout in seconds. The default is 0, indicating that there is no connection timeout.
<code>encryptionProtocol</code>	Optional. Specifies the name of encryption protocol to use. The default is set to <code>null</code> . The RC4 protocol is supported.
<code>encryptionKeyName</code>	Optional. Specifies the name of the symmetric encryption key to use.
<code>encryptionKeyValue</code>	Optional. Specifies the value of the symmetric encryption key to use.
<code>connectionFactory</code>	This property is used only for the BPEL or ESB environment. It specifies the <code>connectionFactory</code> class instance that contains the adapter connection properties. If any connection property is defined on both the <code>activationSpec</code> and the <code>connectionFactory</code> , the <code>activationSpec</code> value is used.

High Availability

The Oracle legacy adapters for OracleAS support High Availability using Oracle AS clusters with active-active and active-passive topologies.

In an active-active topology Oracle Application Server provides an active-active redundant model for all its components with OracleAS Clusters. In an OracleAS

Cluster, two or more Oracle Application Server instances are configured to serve the same application workload. These instances can reside on the same computer or on different computers. The active instances may be front-ended by an external load balancer, which can redirect requests to any of the active instances, or by some other application-level configuration, such as address lists, to distribute the requests.

In an active-passive topology Oracle Application Server supports Active-passive model using OracleAS Cold Failover Clusters. In this case two or more application server instances are configured to serve the same application workload, but only one is active at any time.

Note: The Tuxedo Gateway adapter cannot work with OracleAS cluster and therefore does not support High Availability topology.

Integrating OracleAS Adapters for Tuxedo

To configure the interactions OracleAS Adapters for Tuxedo, you must perform the following tasks in the specified order:

1. [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#)
2. [Integrating OracleAS Adapters for Tuxedo with WebLogic](#)
3. [Configuring OracleAS Adapter for Tuxedo](#)

See Also:

[Configure the BPEL Transactional Gateway Process using JDeveloper](#)
[Configure the Resource Adapter Connection Factory](#)

Configuring OracleAS Adapter for Tuxedo

This chapter includes the following topics:

- [Creating JCA Configuration Files](#)
- [Configuring the Binding for Outbound Interactions](#)
- [Configuring the Binding for Tuxedo Queue Inbound Interactions](#)

Creating JCA Configuration Files

The following sections describe how to use Oracle Studio to:

- [Create JCA Configuration Files for the Tuxedo and Tuxedo Queue Adapters](#)
- [Create JCA Configuration Files for the Tuxedo Gateway Adapter](#)

Perform the following steps to create a binding file and a WSDL file for the Tuxedo or Tuxedo Queue adapters.

Create JCA Configuration Files for the Tuxedo and Tuxedo Queue Adapters

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. In the Design perspective, Configuration view, expand the **Machine** folder.
3. Expand the machine defined in [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#) on page 2-1.

4. Expand the **Bindings**. The binding configurations available on this computer are listed.
5. Expand the **NAV** binding. The NAV binding configuration includes branches for data sources and adapters that are located on the computer.
6. Do *one* of the following:
 - For the Tuxedo adapter, expand the **Adapters** folder and right-click the Tuxedo adapter you defined in [Setting up the Tuxedo OracleAS Adapter](#) on page 2-5 and select **Generate JCA Configuration Files**.
 - For the Tuxedo Queue adapter, expand the **Events** folder and right-click the Tuxedo Queue you defined in [Setting up the Tuxedo Queue OracleAS Adapter](#) on page 2-19 and select **Generate JCA Configuration Files**.

The JCA Configuration Files dialog box opens.

7. In the **Save to** field, type the path to the location where you want to save the files, or select **Browse** to browse to a location.
8. In the **JNDI Location** field, change the default value for the WebLogic Connection Factory JNDI location, if necessary.

You should ensure that a connection factory exists on the WebLogic server. The Connection Factory is mandatory for Outbound interactions and optional for Inbound interactions.

For information on the binding file, see [Configuring the Binding for Outbound Interactions](#) or [Configuring the Binding for Tuxedo Queue Inbound Interactions](#).

9. Click **OK** to create the files in the correct location.

The View Generated Content dialog box opens. It provides links to the binding file (*.jca) and the WSDL file that are created. If you want to view or edit the files, click the appropriate link in the dialog box.

Use these files when working with the BPEL Process Manager or Mediator.

Create JCA Configuration Files for the Tuxedo Gateway Adapter

Perform the following steps to create a binding file and a WSDL file for the Tuxedo Gateway adapter.

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. In the Design perspective, Configuration view, expand the **Machine** folder.
3. Expand the machine defined in [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#) on page 2-1.
4. Expand the **Bindings**. The binding configurations available on this computer are listed.
5. Expand the **TGW** binding that you created in [Setting up the Tuxedo Gateway OracleAS Adapter](#) on page 2-25. The TGW binding configuration includes branches for data sources and adapters that are located on the computer.
6. Expand the **Adapters** folder and right-click the Tuxedo Gateway adapter you defined in [Setting up the Tuxedo Gateway OracleAS Adapter](#) on page 2-25 and select **Generate JCA Configuration Files**.

The JCA Configuration Files dialog box opens.

7. In the **Save to** field, type the path to the location where you want to save the files, or select **Browse** to browse to a location.

8. In the **JNDI Location** field, change the default value for the WebLogic Connection Factory JNDI location, if necessary.

You should ensure that a connection factory exists on the WebLogic server. The Connection Factory is optional for Inbound interactions.

For more information on the binding file, see [Configuring the Binding for Tuxedo Queue Inbound Interactions](#).

9. Click **OK** to create the files in the correct location.

The View Generated Content dialog box opens. It provides links to the binding file (*.jca) and the WSDL file that are created. If you want to view or edit the files, click the link the dialog box.

Use these files when working with the BPEL Process Manager or Mediator.

Configuring the Binding for Outbound Interactions

You should not edit the WSDL file generated by Oracle Studio. You may edit the binding file to configure the `ExecutionTimeout` property in the `AttuInteractionSpec`.

Note: The `FunctionName` property must have the same value as the corresponding operation.

The following is an example of an outbound binding file:

```
<?xml version="1.0" encoding="UTF-8"?>
<adapter-config adapter="Legacy Adapter" name="calc"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
  <connection-factory UIConnectionName="pcbpel" csfKey="pcbpel"
location="eis/legacy/calc"/>

  <endpoint-interaction operation="add" portType="calcPortType">
    <interaction-spec
className="com.attunity.adapter.oracle.AttuInteractionSpec">
      <property name="FunctionName" value="add"/>

      <property name="ExecutionTimeout" value="120"/>
    </interaction-spec>
  </endpoint-interaction>
  <endpoint-interaction operation="display" portType="calcPortType">
    <interaction-spec
className="com.attunity.adapter.oracle.AttuInteractionSpec">
      <property name="FunctionName" value="display"/>

      <property name="ExecutionTimeout" value="120"/>
    </interaction-spec>
  </endpoint-interaction>
  <endpoint-interaction operation="div" portType="calcPortType">
    <interaction-spec
className="com.attunity.adapter.oracle.AttuInteractionSpec">
      <property name="FunctionName" value="div"/>

      <property name="ExecutionTimeout" value="120"/>
    </interaction-spec>
  </endpoint-interaction>
  <endpoint-interaction operation="mul" portType="calcPortType">
    <interaction-spec
```

```

className="com.attunity.adapter.oracle.AtтуInteractionSpec">
  <property name="FunctionName" value="mul"/>

  <property name="ExecutionTimeout" value="120"/>
</interaction-spec>
</endpoint-interaction>
<endpoint-interaction operation="sub" portType="calcPortType">
  <interaction-spec
className="com.attunity.adapter.oracle.AtтуInteractionSpec">
  <property name="FunctionName" value="sub"/>

  <property name="ExecutionTimeout" value="120"/>
</interaction-spec>
</endpoint-interaction>
</adapter-config>

```

Configuring the Binding for Tuxedo Queue Inbound Interactions

Note: When you work with the Tuxedo Gateway adapter metadata (inbound synchronous), the server must be running.

When you work with the Tuxedo Gateway adapter, only one instance can service a Tuxedo Gateway Adapter.

You should not edit the WSDL file generated by Oracle Studio. You may edit the binding file to configure the `AttuActivationSpec` properties. Most of these properties can also be provided using the Connection Factory.

For a list of these properties, see:

- [Configuring the Tuxedo Queue Inbound Adapter](#)
- [Configuring the Tuxedo Gateway Synchronous Inbound Adapter](#)

For information on the properties you that must be defined in the binding, see [ActivationSpec Properties Defined in the Binding](#).

The binding file specifies the name of the adapter's connection factory as the value of the `adapterInstanceJndi` attribute of the `<jca:address>` element in the `<service>` section. If a connection factory exists on the application server, its properties are taken. Otherwise, the properties specified by the `AttuActivationSpec` are used. If a value is specified by both the connection factory and the `AttuActivationSpec`, the `AttuActivationSpec` property overrides the value in the connection factory. If you want to use the value specified in the connection factory, you must delete the property from the binding file.

The following is an example of an inbound binding file for a queue adapter:

```

<?xml version="1.0" encoding="UTF-8"?>
<adapter-config adapter="Legacy Adapter" name="eventQ1"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
  <resource-adapter
className="com.attunity.adapter.oracle.AtтуResourceAdapter"/>

  <endpoint-activation operation="eventStream" portType="eventQ1PortType">
<connection-factory UIConnectionName="pcbpel" csfKey="pcbpel"
location="eis/legacy/OracleDisam_CDCQueue"/>
  <activation-spec
className="com.attunity.adapter.oracle.AtтуActivationSpec">

```

```

<property name="MessagesInBatch" value="50"/>
<property name="WaitTime" value="30"/>
<property name="EisName" value="eventQ1"/>
<property name="ServerName" value="localhost"/>
<property name="PortNumber" value="2551"/>
<property name="UserName" value="" />
<property name="Password" value="" />
<property name="Workspace" value="eventQ1"/>
<property name="RetryInterval" value="15"/>
<property name="ConnectTimeout" value="0"/>
<property name="Compression" value="true"/>
<property name="FirewallProtocol" value="" />
<property name="EncryptionProtocol" value="" />
<property name="EncryptionKeyName" value="" />
<property name="EncryptionKeyValue" value="" />
</activation-spec>
</endpoint-activation>
</adapter-config>

```

The following table describes the properties you must configure in the binding file.

Table 4–1 ActivationSpec Properties Defined in the Binding

Property	Description
messagesInBatch	Optional. The maximum number of events that can be sent by the adapter to the SOA process. The default value is 50.
support2PC	Optional. Set to <code>true</code> to enable 2PC support. You should set this value to <code>true</code> only if the Oracle adapter is configured for two-phase commit. The default value is <code>false</code> .
waitTime	Optional. Defines the maximum response time (in seconds) for Oracle Connect to send a response to the CICS queue adapter. The default is set to 30 seconds.
retryInterval	Optional. The amount of time the resource adapter waits if no events are found in the server before issuing the next request. The default is set to 15 seconds.

Configuring the Transactional Gateway Process

The Oracle Adapter Framework does not support its own resource manager, therefore the inbound transactional process cannot be supported directly under the Adapter

Framework. To configure the transaction gateway process, you must perform the following steps:

- [Configure the BPEL Transactional Gateway Process using JDeveloper](#)
- [Configure the Resource Adapter Connection Factory](#)

Configure the BPEL Transactional Gateway Process using JDeveloper

Configure the Oracle Gateway inbound process for a specific Tuxedo service as explained in [Configuring OracleAS Adapter for Tuxedo](#). You must select a the Tuxedo Transactional Gateway adapter. For more information, see [Setting up the Tuxedo Gateway OracleAS Adapter](#) for more information. The automatically created WSDL contains the `com.oracle.adapter.oracle.AttuGWTxnActivationSpec`. The properties for this activation spec are not in use and should not be changed.

Next, deploy the created BPEL process. When you deploy a process that includes this activation spec, the Tuxedo Gateway resource adapter that works under the Adapter Framework ignores the activation of the corresponding work thread. As a result, only the BPEL application is activated in the BPEL domain.

Configure the Resource Adapter Connection Factory

You should configure a connection factory that provides connectivity to the transactional gateway adapter. For a list of the properties to configure, see [Connection Factory Properties for Synchronous Inbound Processes](#).

In addition, the resource adapter connection factory contains two properties that enable connectivity from the application server environment to the BPEL domain. These properties are:

- `bpelDomain`: The name of the BPEL domain. The default value is `default`.
- `bpelDomainPassword`: Password for the BPEL domain. The default value is `bpel`.

You should not configure the connection factory if the corresponding properties have default values. You can use your existing connection factory or create a new one to edit these properties.

Configuring an OracleAS Adapter for Tuxedo

This chapter provides an example for creating an OracleAS adapter for Tuxedo. To work with an OracleAS adapter for Tuxedo, you must also configure either a BPEL process or a Mediator process. For information on using BPEL or Mediator, see [Creating the Project in JDeveloper Examples](#).

Requirements

The following is required to perform the tasks described in this chapter:

- Computer with Windows operating system
- Computer with Windows or UNIX operating system
- Oracle Connect v 11.1.1.2 installed on the Windows or UNIX computer
- Oracle Studio v 11.1.1.2 installed on the Windows computer
- Tuxedo installed on the Windows or UNIX computer with Oracle Connect

Creating Outbound Interactions with the OracleAS Adapter for Tuxedo

To work with the OracleAS Adapter for Tuxedo and create outbound interactions you create Tuxedo interactions using the Tuxedo adapter and import the metadata.

Create the toUpper Interaction

You use a Tuxedo adapter to create the interactions used to get the data that you need. The interactions are used as services when you set up the endpoint in JDeveloper. For more information on using JDeveloper to create an endpoint, see [Deploy a BPEL Outbound Process](#).

Do the following to create the interactions.

- [Prepare the System](#)
- [Set up Machine Access to Oracle Connect](#)
- [Add a Tuxedo Adapter](#)
- [Add a Tuxedo Adapter](#)

Prepare the System

Make sure to create the toUpper and toLower View files in Tuxedo and then set the correct environment properties on the Windows computer. For information on how to create the view files, see the Tuxedo documentation.

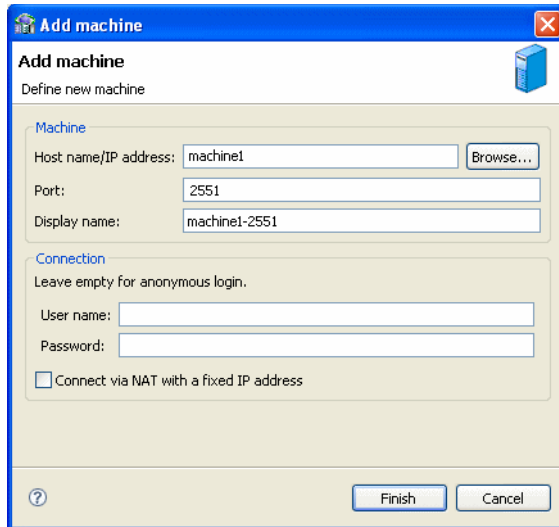
Set up Machine Access to Oracle Connect

You begin by configuring access to the Windows computer with Tuxedo and Oracle Connect. You configure this access with Oracle Studio. Open Oracle Studio and follow the directions [Securing Access to Oracle Connect](#). Enter the following information in the Add machine dialog box:

- **Host name/IP address:** Enter the name or IP address for the computer with Tuxedo.
- **Port:** Enter the port number where the daemon is running. The default port is 2551.
- **Display name:** You do not need to enter any information in this field (By default, the display name is the host name and the port number).
- **User name:** If the computer you are accessing needs an administrator password. If so enter the name of the computer's administrator.
- **Password:** If necessary, enter the computer administrator's password.
- **Connect via NAT with fixed IP address:** Select this if the machine uses the NAT (Network Address Translation) firewall protocol, with a fixed configuration, mapping each external IP to one internal IP, regardless of the port specified.

The following figure shows the Add Machine dialog box with the correct information:

Figure 5–1 Add Machine Dialog Box



Add a Tuxedo Adapter

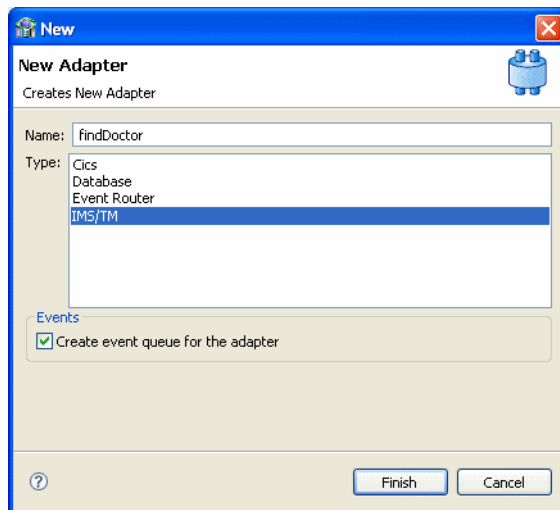
You now create an adapter, which lets you create interactions to get the information you are seeking. To create an adapter, do the following

1. Expand the computer you just added and then expand the **NAV** binding.
2. Right-click the **Adapters** folder and then select **New adapter**.
3. Enter the following information for your adapter, then click **Finish**.
 - **Name:** ToUpper
 - **Type:** Tuxedo

For a detailed explanation on how to set up a Tuxedo adapter, see [Setting up the Tuxedo OracleAS Adapter](#).

The following figure shows the Add adapter dialog box with the correct information:

Figure 5–2 Add Adapter Dialog Box



Import the Metadata

You now import the metadata from Tuxedo into the adapter. For information on how to import data, see [Modeling Interactions for the Tuxedo OracleAS Adapter](#).

In this example you create the TOUPPER interaction.

1. In the Configuration view of the Design perspective, right-click the `toupper` adapter that you created in [Add a Tuxedo Adapter](#) and select **Show metadata view**. The Metadata view opens with the `toupper` adapter expanded and selected.
2. Right-click Imports and select **New Import**.
3. Enter a name for the import and then select **Tuxedo Import Manager Using FML/VIEW Files** as the **Import type** then click **Finish** to view the Metadata Import Wizard.
4. You need the following information for the metadata import:
 - In the **Get Input Files** step, just click **Next**.
 - In the **Add FML Records** step, just click **Next**.
 - In the **Get Tuxedo Records** step, create two records. Name the records **ToUpper_In** and **ToUpper_Out**. Both records should have a **Field Type** of **STRING** and a **Size** of **20**.
 - In the **Add interactions** step, under the **Name** column, enter **TOUPPER**, in the **Input** column, select **ToUpper_In** and in the **Output** column, select **ToUpper_Out**.
 In the **Interaction-Specific Parameters** section, select **STRING** for both the **Input Buffer Type** and **Output Buffer Type**.
 - In the **Import Metadata** step, select **Yes** and then click **Finish**.

Create the JCA Configuration Files

You must create both a WSDL and JCA file to set up your connection with JDeveloper. You create the files using Oracle Studio before setting up the BPEL process or Mediator connections.

For information on how to create JCA configuration files, see [fix link](#). To create the files for the `ToUpper` adapter, do the following:

1. From the Configuration pane in Oracle Studio, expand the NAV binding and the Adapter folder.
2. Right-click the `ToUpper` adapter and select Generate JCA Configuration Files.
3. In the dialog box, save the files to a place you can find when you create the BPEL process or Mediator connection.

Creating Inbound Interactions with the Tuxedo Queue Adapter

To work with the OracleAS Adapter for Tuxedo and create inbound interactions you create Tuxedo interactions using the Tuxedo Queue adapter and import the metadata.

Create the DEQUEUE Event Queue

You use a Tuxedo Queue adapter to create the interactions used to send the data that you need. The interactions are used as services when you set up the endpoint in JDeveloper. For more information on using JDeveloper to create an endpoint, see [Deploy a BPEL Inbound Process](#).

Do the following to create the interactions.

- [Prepare the System](#)
- [Set up Machine Access to Oracle Connect](#)
- [Add a Tuxedo Queue Adapter](#)
- [Import the Metadata](#)

Prepare the System

In order to create an the DEQUEUE interaction, make sure to create a Tuxedo queue space and an individual queue for it. For information, see the Tuxedo documentation.

Set up Machine Access to Oracle Connect

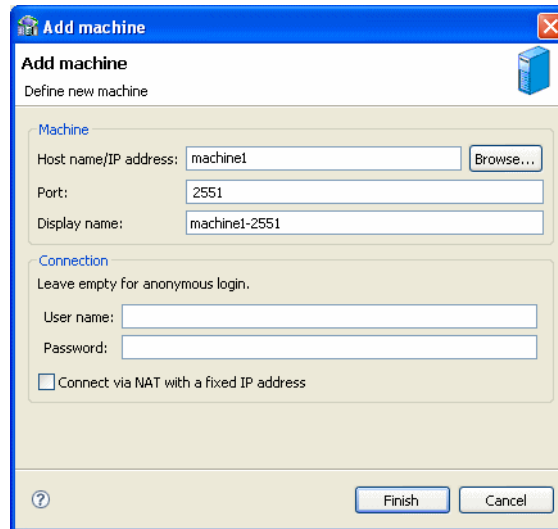
You begin by configuring access to the Windows computer with Oracle Connect and Tuxedo. You configure this access with Oracle Studio. Open Oracle Studio and follow the directions [Securing Access to Oracle Connect](#). Enter the following information in the Add machine dialog box:

- **Host name/IP address:** Enter the name or IP address for the computer with Tuxedo. This should be installed in the same directory as Oracle Connect.
- **Port:** Enter the port number where the daemon is running. The default port is 2551.
- **Display name:** You do not need to enter any information in this field (By default, the display name is the host name and the port number).
- **User name:** If the computer you are accessing needs an administrator password. If so enter the name of the computer's administrator.

- **Password:** If necessary, enter the computer administrator's password.
- **Connect via NAT with fixed IP address:** Select this if the machine uses the NAT (Network Address Translation) firewall protocol, with a fixed configuration, mapping each external IP to one internal IP, regardless of the port specified.

The following figure shows the Add Machine dialog box with the correct information:

Figure 5–3 Add Machine Dialog Box



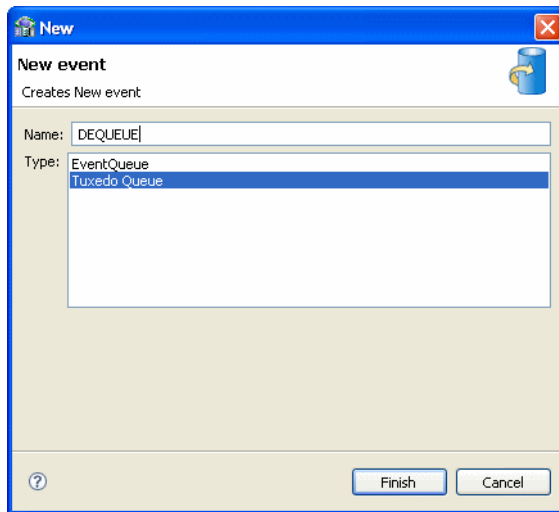
Add a Tuxedo Queue Adapter

You now create an adapter, which lets you create interactions to get the information you are seeking. To create an adapter, do the following

1. Expand the computer you just added and then expand the **NAV** binding.
2. Right-click the **Events** folder and then select **New event**.
3. Enter the following information for your adapter, then click **Finish**.
 - **Name:** DEQUEUE
 - **Type:** Tuxedo Queue

For a detailed explanation on how to set up a Tuxedo Queue adapter, see [Setting up the Tuxedo Queue OracleAS Adapter](#).

The following figure shows the New Event dialog box with the correct information:

Figure 5–4 New Event Dialog Box

Import the Metadata

You now import the metadata from Tuxedo into the adapter. For information on how to import data, see [Modeling Interactions for the Tuxedo Queue OracleAS Adapter](#).

In this example you should import the create the event queue for the DEQUEUE adapter. You must also create the View files in Tuxedo. Do the following to import the metadata:

1. In the Configuration view of the Design perspective, right-click the DEQUEUE adapter that you created in [Add a Tuxedo Queue Adapter](#) and select **Show metadata view**. The Metadata view opens with the toupper adapter expanded and selected.
2. Right-click Imports and select **New Import**.
3. Enter a name for the import and then select **Tuxedo Queue Import Manager for XML/STRING/CARRAY Buffer** as the **Import type** then click **Finish** to view the Metadata Import Wizard.
4. You need the following information for the metadata import:
 - In the **Get Records** step, click **Add Record** and change the name to **str_record** with a size of **20**. Leave **STRING** as the **Field Type**.
 - In the **Configure Tuxedo Queue Records** step, just click **Next**.
 - In the **Import Metadata** step, select **Yes** and then click **Finish**.

Create the JCA Configuration Files

You must create both a WSDL and JCA file to set up your connection with JDeveloper. You create the files using Oracle Studio before setting up the BPEL process or Mediator connections.

For information on how to create JCA configuration files, see [fix link](#). To create the files for the DEQUEUE adapter, do the following:

1. From the Configuration pane in Oracle Studio, expand the NAV binding and the Adapter folder.
2. Right-click the DEQUEUE adapter and select **Generate JCA Configuration Files**.

-
-
3. In the dialog box, save the files to a place you can find when you create the BPEL process or Mediator connection.

Creating the Project in JDeveloper Examples

This chapter contains the following topics:

- [Getting Started](#)
- [Deploy a BPEL Inbound Process](#)
- [Mediator Inbound Process](#)
- [Deploy a BPEL Outbound Process](#)

Getting Started

This topic describes the components necessary to work with the samples and how to prepare your system.

Prerequisites

The following are installation and configuration requirements for using the examples.

- Personal computer running Microsoft Windows XP, Windows Vista, Windows Server 2003, or Windows Server 2008 with one GB of RAM.
- The Tuxedo adapter that you are using (Tuxedo or Tuxedo Queue) must be deployed to the Oracle Application Server.
- OracleAS Tuxedo Adapter must be configured for outbound or inbound processing. See [fix link](#).
- Oracle JDeveloper must be installed.
- Legacy adapter must be deployed.
- Oracle Studio with the necessary machines loaded. See [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#).

This chapter provides examples of the configuration steps that are used when using BPEL to work with the Oracle Tuxedo adapters. You should be familiar with the following before working with these examples:

- How to create J2CA connections. See [Integrating OracleAS Adapters for Tuxedo with WebLogic](#) for more information.
- How to configure the OracleAS adapters for Tuxedo using Oracle Studio. See [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#) for more information.

Integration between the OracleAS adapters for Tuxedo and BPEL has the following processes:

- **Design Time:** The OracleAS adapters for Tuxedo are configured with Oracle Studio as described in [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#).
- **Run time:** After you deploy the BPEL process with JDeveloper, you can test your configurations. See

Deploy a BPEL Inbound Process

This section describes how to create a BPEL inbound process that connects to a Tuxedo adapter. For information on how to configure a Tuxedo adapter for inbound, see [fix link](#).

To configure a process with the BPEL process manager, you use JDeveloper. This section describes how to create a BPEL inbound process for Tuxedo adapters using JDeveloper. There are two configuration types necessary to perform this inbound sample.

- [Beginning Tasks](#)
- [Design-Time Configuration](#)
- [Run-time Configuration](#)

Before you design the inbound process, you integrate and connect your OracleAS adapter for Tuxedo with the BPEL process manager. See [Integrating OracleAS Adapters for Tuxedo](#) for more information.

Beginning Tasks

Before you design the inbound process, you must perform the following:

- Integrate the OracleAS Adapter for VSAM. See [Integrating OracleAS Adapters for Tuxedo with WebLogic](#) for more information.
- Configure the connection for the J2CA VSAM resource adapter. You do this by creating a connection factory. See [Configuring the Tuxedo Queue Inbound Adapter](#) for information on how to do this.
- Write a schema file. First, create an interaction called `findDoctor` using Oracle Studio. See [Configuring the Tuxedo Queue Inbound Adapter](#) for more information.
- Create a BPEL PM Server Connection in JDeveloper.
- Create the WSDL and binding file for the interaction. You do this in Oracle Studio before you begin to build your endpoint in JDeveloper. Oracle Studio creates the WSDL and binding (`jca`) files automatically based on the interaction you create. See [fix link](#) for information on creating these files.

Design-Time Configuration

This section describes the design-time steps necessary to deploy a BPEL inbound process:

- [Create a BPEL Inbound Process](#)
- [Use a Third-Party Adapter for the Inbound Process](#)
- [Design the BPEL Inbound Process](#)
- [Deploy the Project](#)

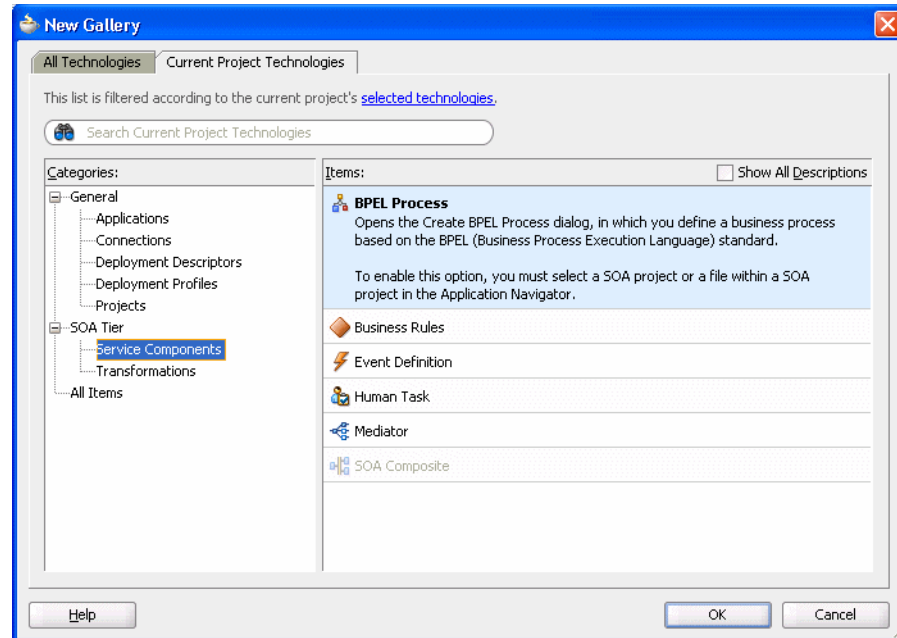
- [Create the Connection](#)

Create a BPEL Inbound Process

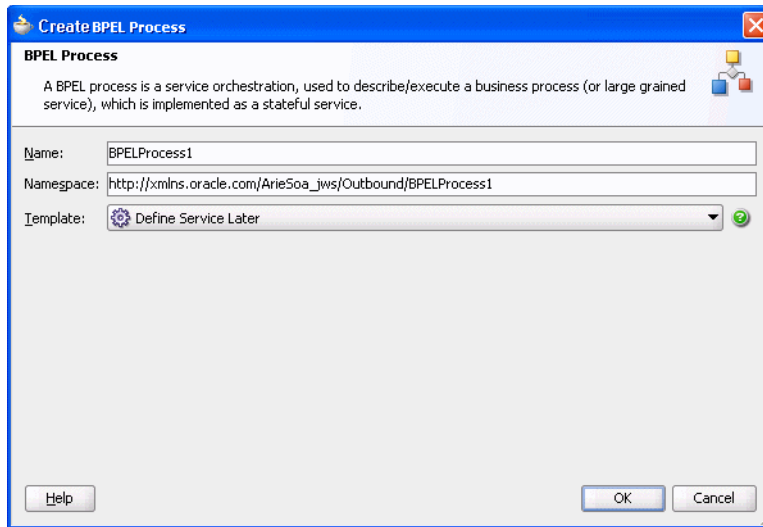
Perform the following to create a BPEL project for inbound.

1. In JDeveloper, create a new SOA project
2. Right-click the SOA project and select **New**. The New Gallery is displayed as shown in [Figure 6-1](#).

Figure 6-1 *New Gallery*



3. In the **Categories** list, expand **SOA Tier** and select **Service Components**.
4. From the **Items** list, select **BPEL Process** then click **OK**. The Create BPEL Process screen is displayed as shown in [Figure 6-2](#).

Figure 6–2 BPEL Process

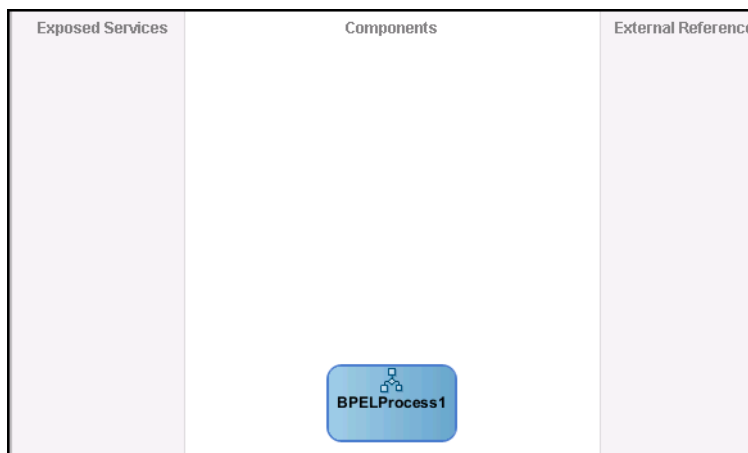
5. Enter the following in the BPEL Process screen:
 - Enter a name for the BPEL process, for example **BPELInbound1**.
 - From the Template list, select **Define Service Later**.
6. Click **OK**. The BPEL process is displayed in the visual editor.

Continue with [Use a Third-Party Adapter for the Inbound Process](#).

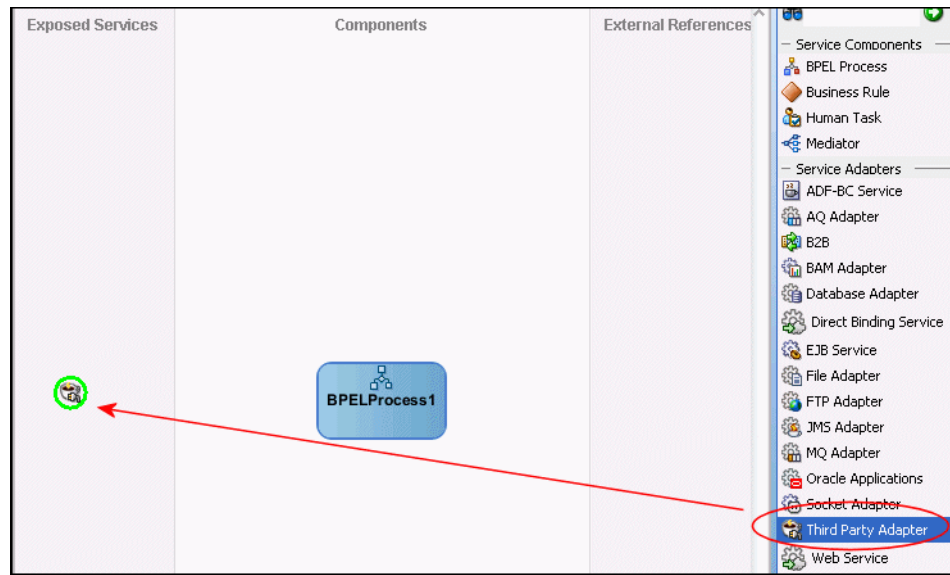
Use a Third-Party Adapter for the Inbound Process

You use a third-party adapter as a partner link to create the inbound process. SOA executes the SQL statement defined in the interaction and passes it to a resource adapter. Perform the following to create the third-party adapter.

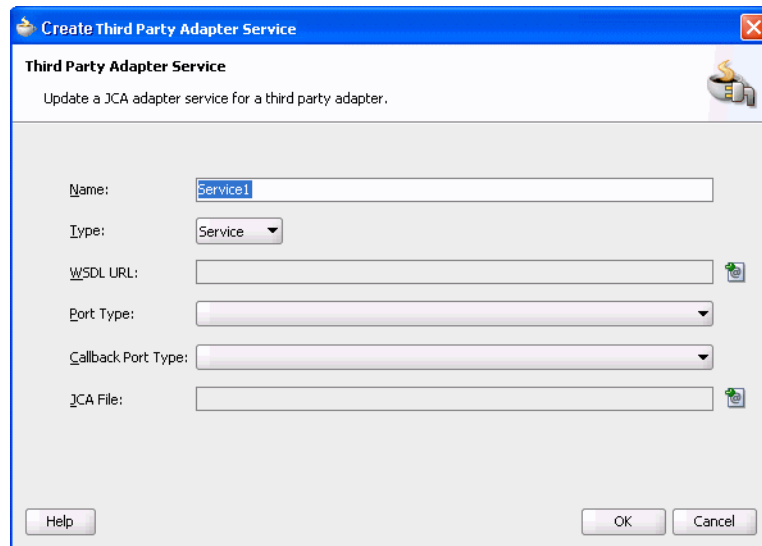
1. From the **Projects** area, on the left side, double-click the `composite.xml`. At the bottom of the visual editor, click the **Design** tab, if necessary. [Figure 6–3](#) shows the Composite view in the visual editor.

Figure 6–3 Composite View

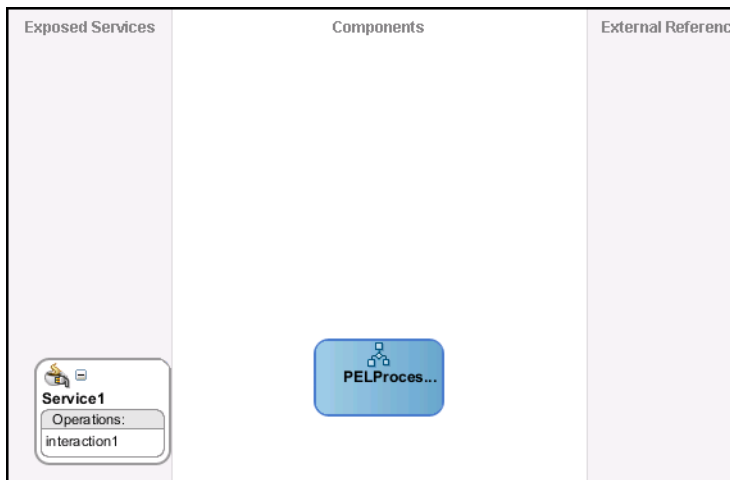
2. Drag a third-party adapter into the **Exposed Services** lane of the visual editor, as shown in [Figure 6–4](#).

Figure 6–4 Third-Party Adapter

When you drag the third-party adapter into JDeveloper, the Create Third Party Adapter Service screen is displayed as shown in [Figure 6–5](#).

Figure 6–5 Create Third-Party Adapter Service

3. Click the button next to the **WSDL URL** field and browse to the WSDL file you are using.
4. In the **Type** list, select **Service**.
5. Click the button next to the **JCA File** field and browse to the binding file you are using.
6. Click **OK** to close. You should now have a composite design as shown in [Figure 6–6](#).

Figure 6–6 Ready to Create BPEL Process (Composite View)

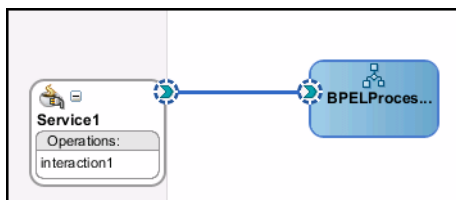
Continue with [Design the BPEL Inbound Process](#).

Design the BPEL Inbound Process

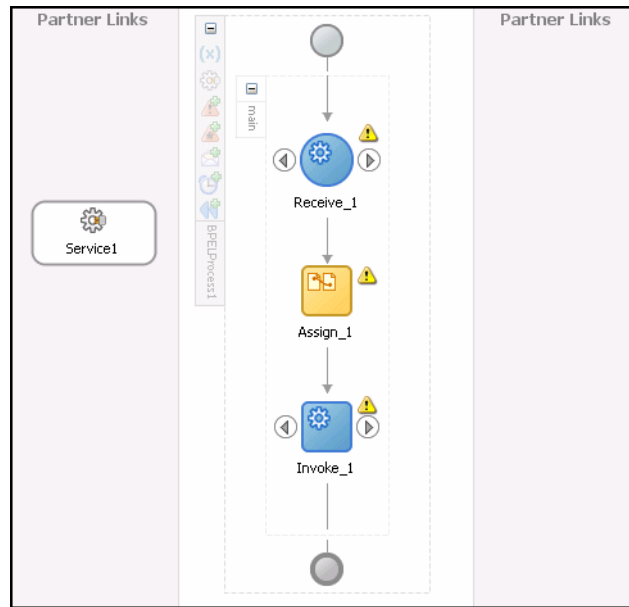
Before you begin to design the inbound process procedure, you may need to configure some activation spec properties in the binding (jca) file. For more information on working with the inbound binding file, see [Configuring the Binding for Tuxedo Queue Inbound Interactions](#).

Do the following to define the BPEL inbound process.

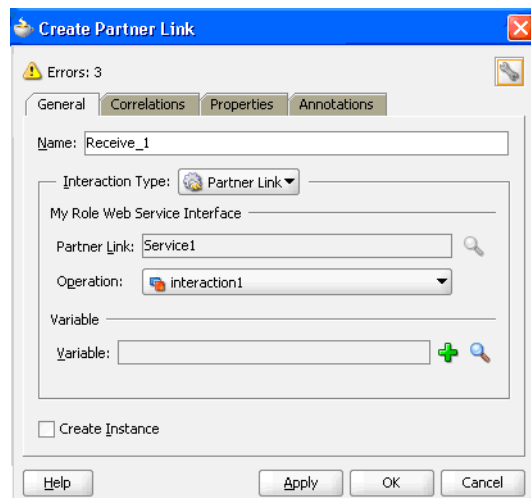
1. In the Composite, connect the BPEL process to the third-party adapter as shown in [Figure 6–7](#).

Figure 6–7 Connect to Third-Party Adapter

2. Double-click the BPEL Process.
3. From the Component Palette, BPEL Process Activities section, drag a Receive, Assign, and Invoke process into the editor as shown in [Figure 6–8](#).

Figure 6–8 Inbound Process

4. In the Visual Editor, connect the Receive activity to the third-party adapter service (Service1). The Edit Receive dialog box is displayed as shown in [Figure 6–9](#).

Figure 6–9 Receive Properties

5. Do the following in the Edit Receive dialog box:
 - Enter a name for the Receive activity.
 - Create the variable. Click the plus sign to the right of the **Variable** field, then click **OK** in the Create Variable dialog box that is displayed.
 - Ensure that the **Create Instance** check box is selected.
6. Click **OK** to close the dialog box and accept the information then [Deploy the Project](#).

Deploy the Project

Perform the following to deploy the project to the Oracle Application Server:

1. From the Application Navigator, right-click the project and find **Deploy** and then select the project you are deploying.
2. From the Deploy wizard, select **Deploy to Application Server**.
3. Continue to click **Next** till the end of the wizard and then click **Finish** to deploy.

Create the Connection

This section describes how to create a connection to the adapter files you are using. Do the following to create the connection.

1. Create a file adapter and give it a name, for example `WriteRes`.
2. Click **Next** and select **Write File**. Give the file a name, for example `Write Message`.
3. Click **Next** and enter the path to the message file you want to write to.
4. In the Message Schema chooser, browse to the file `message.xsd`. You should have made sure that this file was created before you started this process. See the beginning of the section, [Design the BPEL Inbound Process](#) for more information.
5. Select the Message type, `bigOrderEvent` and then click **OK**.
6. Create an Input Variable with the default name for the Invoke activity.
7. Create a Copy Operation for the Assigned activity.

Run-time Configuration

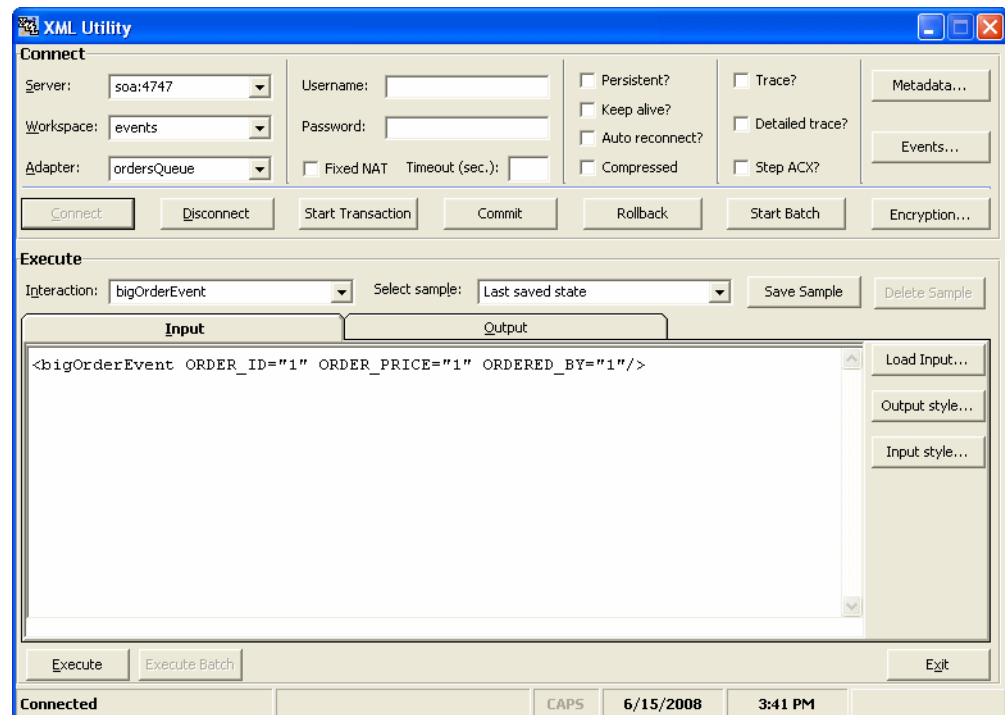
This section describes how to test the project and run it to see that the inbound process is executed successfully. By using the XML utility, you call an event in Oracle Connect that is defined in the file adapter you created. See [Create the Connection](#). Because this event is called from Oracle Connect it is an inbound event. Events that originate with SOA services in the OracleAS, it is an outbound event. For more information, see [Deploy a BPEL Outbound Process](#). The Runtime configuration for inbound has one step called [Triggering the Event using the XML Utility](#).

Triggering the Event using the XML Utility

Use the XML utility to create an SQL statement to trigger the event you are requesting. Do the following to trigger the event.

1. From the Start menu, select **All Programs**, then **Oracle**, then **Server Utilities**, and then **XML Utility**. The XML Utility is displayed.

Figure 6–10 Trigger for Inbound Process



- Enter the following information in the Connect section of the XML utility:
 - Server:** The name of the machine where the OracleAS adapter for Tuxedo is located.
 - Workspace:** The name of the workspace where the OracleAS adapter for Tuxedo is located. A list of available workspaces is shown in the list for this field. The workspaces are defined in Oracle Studio.
 - Adapter:** The name of the OracleAS adapter for Tuxedo. A list of available adapters is shown in the list for this field. The adapters are defined in Oracle Studio.
- Click **Connect** to connect to the selected machine.
- In the **Interaction** field, enter the name of the interaction to call. This is defined in Oracle Studio.
- In the **Input** window, enter a statement to call the interaction. You can see an example in the above figure.
- Click **Execute**. If successful, you get a return similar to the following example:

```
<?xml version="1.0"?>
<bigOrderEvent ORDER_ID="1"ORDERED_BY="1" xmlns+"noNamespace://ordersQueue"/>
```

Mediator Inbound Process

This section describes how to create a Mediator inbound process that connects to a OracleAS adapter for Tuxedo.

You use JDeveloper to configure a process with Mediator. This section describes how to create a Mediator outbound process for Tuxedo resource adapters using JDeveloper. The following topics describe how to deploy a BPEL outbound process.

- [Beginning Tasks](#)
- [Create a Mediator Inbound Process](#)
- [Use a Third-Party Adapter for the Inbound Process](#)
- [Create a File Adapter](#)
- [Design an Inbound Mediator](#)
- [Create Mapping Rules and Map the Files](#)
- [Deploy the Project](#)

Beginning Tasks

Before you design the inbound process, you must perform the following:

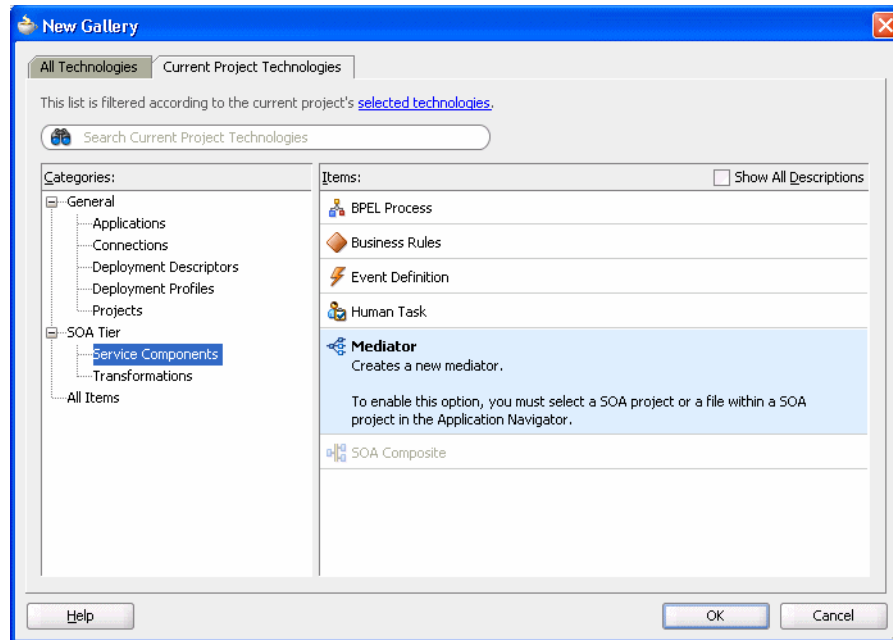
- Integrate the OracleAS Adapter for VSAM. See [Integrating OracleAS Adapters for Tuxedo with WebLogic](#) for more information.
- Configure the connection for the J2CA VSAM resource adapter. You do this by creating a connection factory. See [Configuring the Tuxedo Queue Inbound Adapter](#) for information on how to do this.
- Create the WSDL and binding file for the interaction. You do this in Oracle Studio before you begin to build your endpoint in JDeveloper. Oracle Studio creates the WSDL and binding (jca) files automatically based on the interaction you create. See [fix link](#) for information on creating these files.

Create a Mediator Inbound Process

Perform the following to create an inbound Mediator project.

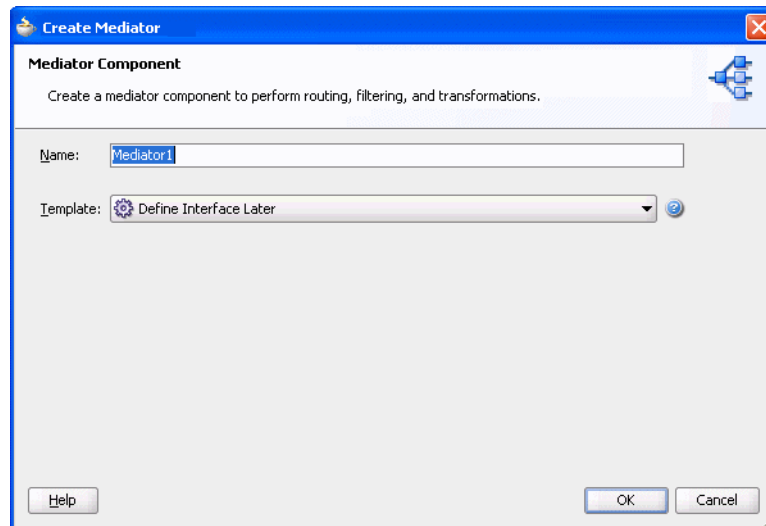
1. In JDeveloper, create a new SOA project
2. Right-click the SOA project and select **New**. The New Gallery is displayed as shown in [Figure 6-11](#).

Figure 6–11 New Gallery



3. In the **Categories** list, expand **SOA Tier** and select **Service Components**.
4. From the **Items** list, select **Mediator** then click **OK**. The Create Mediator screen is displayed as shown in Figure 6–12.

Figure 6–12 Mediator



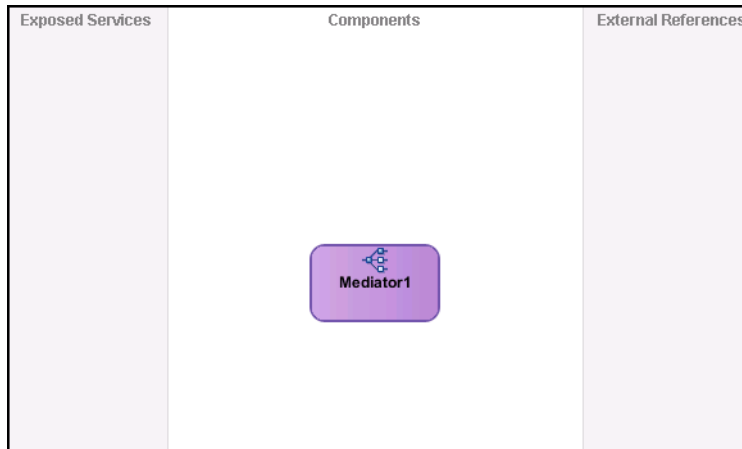
5. Enter the following in the BPEL Process screen:
 - Enter a name for the Mediator process, for example **MedInbound1**.
 - From the Template list, select **Define Service Later**.
6. Click **OK**. The Mediator process is displayed in the visual editor. Continue with [Use a Third-Party Adapter for the Inbound Process](#).

Use a Third-Party Adapter for the Inbound Process

You use a third-party adapter as a partner link to create the inbound Mediator. SOA executes the SQL statement defined in the interaction and passes it to a resource adapter. Perform the following to create the third-party adapter.

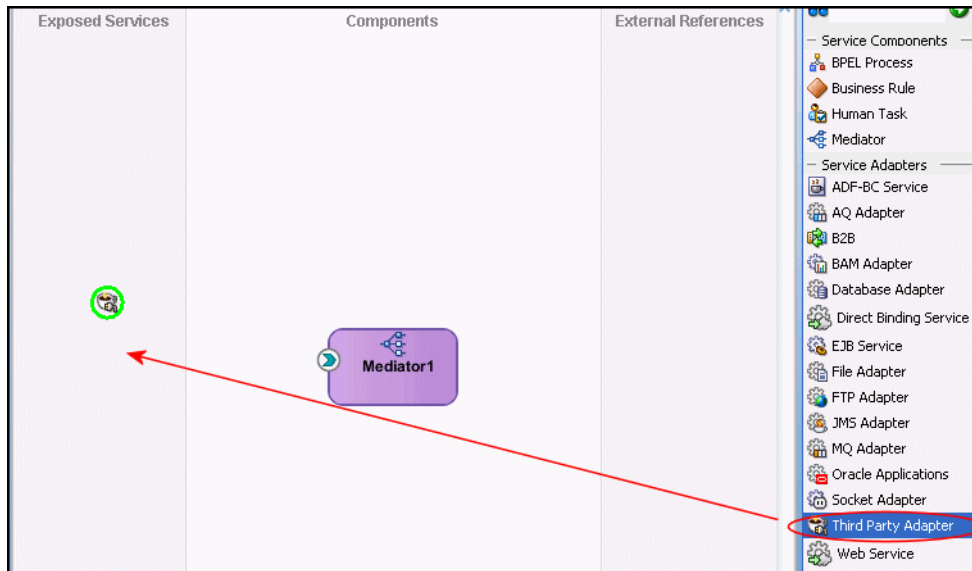
1. From the **Projects** area, on the left side, double-click the `composite.xml`. At the bottom of the visual editor, click the **Design** tab, if necessary. [Figure 6–13](#) shows the Composite view in the visual editor.

Figure 6–13 Composite View

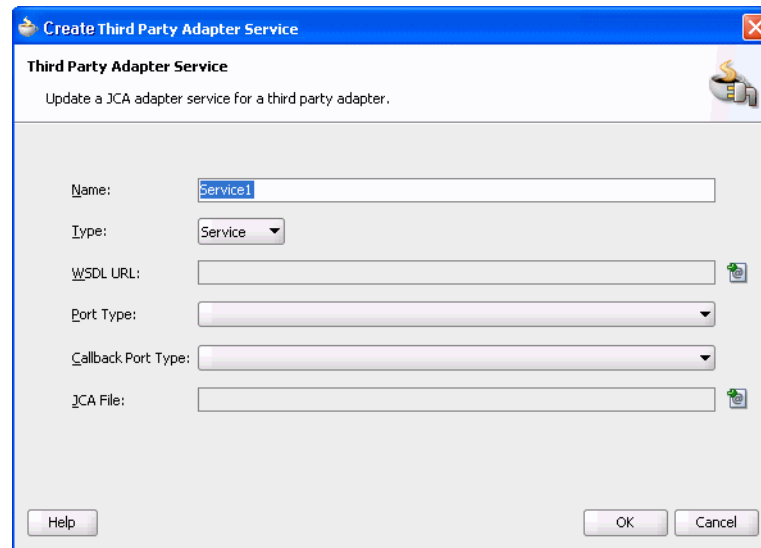


2. Drag a third-party adapter into the **Exposed Services** lane of the visual editor, as shown in [Figure 6–14](#).

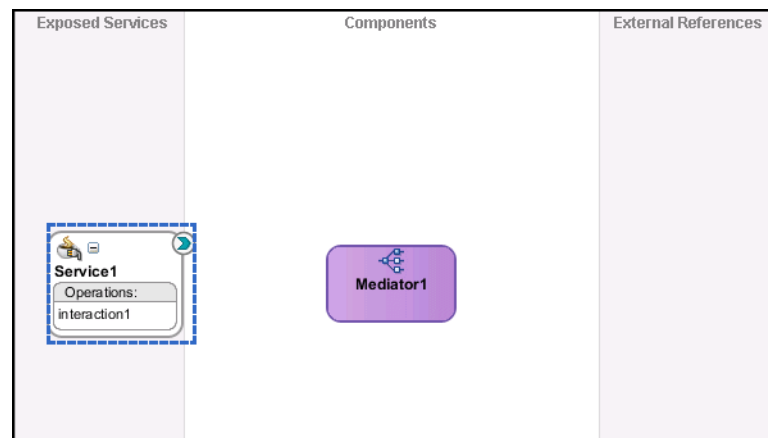
Figure 6–14 Third-Party Adapter



When you drag the third-party adapter into JDeveloper, the Create Third Party Adapter Service screen is displayed as shown in [Figure 6–15](#).

Figure 6–15 Create Third-Party Adapter Service

3. Click the button next to the **WSDL URL** field and browse to the WSDL file you are using.
4. In the **Type** list, select **Service**.
5. Click the button next to the **JCA File** field and browse to the binding file you are using.
6. Click **OK** to close. You should now have a composite design as shown in [Figure 6–16](#).

Figure 6–16 Ready to Create Mediator (Composite View)

Continue with [Create a File Adapter](#).

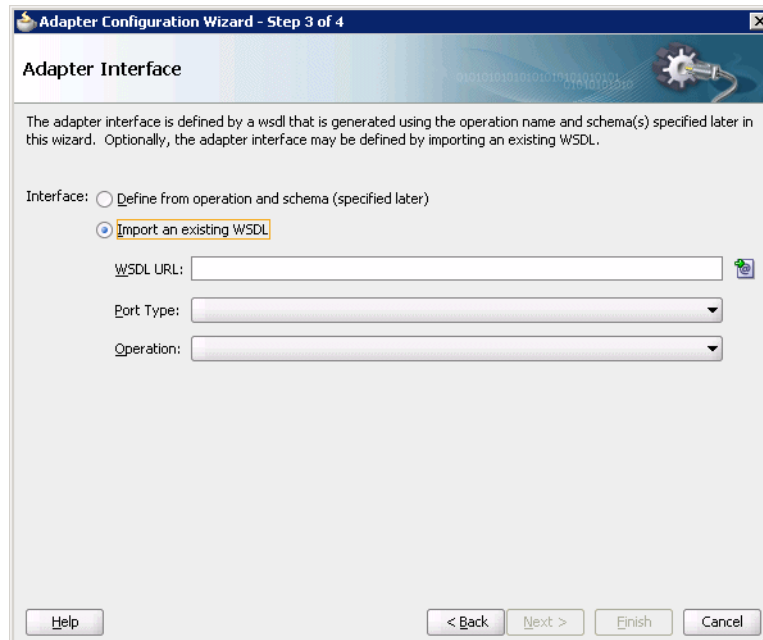
Create a File Adapter

Perform the following to create a file adapter.

1. In the Composite, connect the Mediator to the third-party adapter.

2. From the component pallet, SOA section, on the right, drag a file adapter into the **External References** lane of the visual editor. The Adapter Configuration wizard is displayed.
3. In the first screen of the Adapter Configuration wizard, enter a name for the adapter service, and click next.
4. In the Adapter Interface screen, select the same WSDL file as you selected in [Use a Third-Party Adapter for the Inbound Process](#). [Figure 6–17](#) shows the Adapter Interface screen.

Figure 6–17 File Adapter

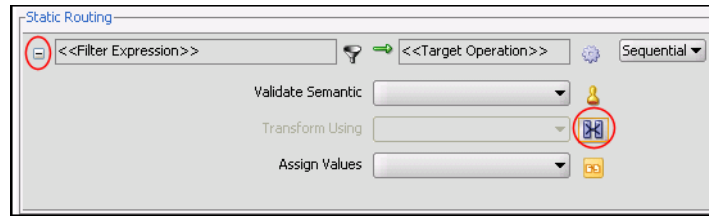


5. Click **Next** to use the default settings for each of the remaining screens in the wizard. When you get to the end click **Finish**.
 6. Connect Mediator to the file adapter and select the operations.
- Continue with [Design an Inbound Mediator](#).

Design an Inbound Mediator

In this step, you design the BPEL process. Perform the following to design the inbound Mediator.

1. In the Composite, double-click the Mediator.
2. Create a static routing rule.
3. Expand the Filter Expression as shown in [Figure 6–18](#).

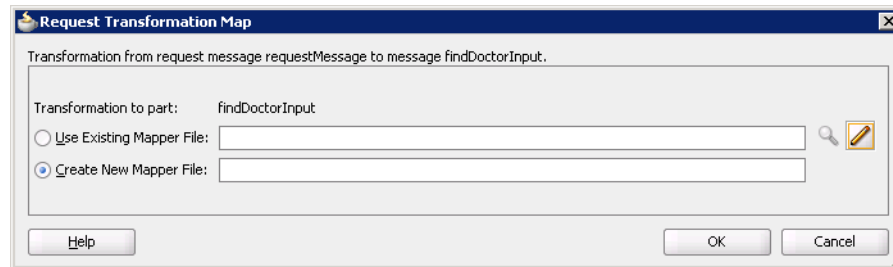
Figure 6–18 Filter Expression

4. Continue with [Create Mapping Rules and Map the Files](#).

Create Mapping Rules and Map the Files

You must perform the following steps to create mapping files and map the files.

1. In the Filter Expression screen, click the icon next to the **Transform using** field as shown in [Figure 6–18](#). The Request Transformation Map screen is displayed as shown in [Figure 6–19](#).

Figure 6–19 Request Transformation Map

2. In the Request Transformation Map, select **Create New Mapper** file. Use the default value.
3. Click **OK**.
4. In the visual editor, click to return to the **mediator.mplan** tab.
5. Click the tab for the Mapper File that you created. Connect the elements.
6. Save and then [Deploy the Project](#).

Deploy the Project

Perform the following to deploy the project to the Oracle Application Server:

1. From the Application Navigator, right-click the project and find **Deploy** and then select the project you are deploying.
2. From the Deploy wizard, select **Deploy to Application Server**.
3. Continue to click **Next** till the end of the wizard and then click **Finish** to deploy.

Deploy a BPEL Outbound Process

This section describes how to create an BPEL inbound process that connects to a Tuxedo adapter. For information on how to configure a Tuxedo adapter for outbound, see [fix link](#).

To configure a process the BPEL Process Manager, you use JDeveloper. This section describes how to create a BPEL outbound process for Tuxedo adapters using JDeveloper. This section contains the following topics.

- [Beginning Tasks](#)
- [Design-Time Configuration](#)
- [Runtime Configuration](#)

Before you design the outbound process, you integrate and connect your OracleAS adapter for Tuxedo with the BPEL process manager. See [Chapter 4, "Integrating OracleAS Adapters for Tuxedo"](#) for more information.

Beginning Tasks

Before you design the outbound process, you must perform the following:

- Integrate the OracleAS Adapter for VSAM. See [Integrating OracleAS Adapters for Tuxedo with WebLogic](#) for more information.
- Configure the connection for the J2CA VSAM resource adapter. You do this by creating a connection factory. See [Configuring the Tuxedo Gateway Synchronous Inbound Adapter](#) for information on how to do this.
- Create a BPEL PM Server Connection in JDeveloper.
- Create the WSDL for the interaction. You do this in JDeveloper Connections explorer before you begin to build your endpoint in JDeveloper. JDeveloper creates the WSDL automatically based on the interaction you create. In the Connections explorer, browse for the Oracle Connect server you configured in Oracle Studio (see *Oracle Application Server Legacy Adapters Installation Guide*), then double-click the findDoctor interaction to view and create the WSDL.

Design-Time Configuration

This section describes the design-time steps necessary to deploy two BPEL outbound processes. The processes use the Web service called TOUPPER. This service reads a string of lower-case letters and writes them in upper-case letters. The following are the steps required to create the outbound process.

- [Before You Begin](#)
- [Create a BPEL Project for a BPEL Outbound Process](#)
- [Create a File Adapter to Read the String](#)
- [Design a BPEL Outbound Process to Read the String](#)
- [Create a Partner Link to the Tuxedo Adapter](#)
- [Edit the Invoke_1 Activity](#)
- [Edit the Assign_1 Activity](#)
- [Create a Process to Write the Upper-Case Letter String](#)

Before You Begin

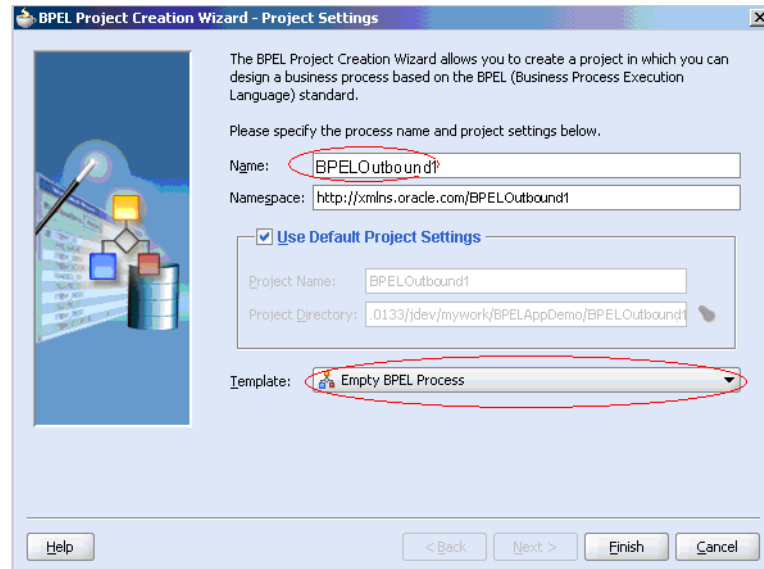
Before you design the outbound process, you define the `Toupper` interaction using Oracle Studio as described in [Creating Outbound Interactions with the OracleAS Adapter for Tuxedo](#).

Create a BPEL Project for a BPEL Outbound Process

Perform the following tasks to create a BPEL project for outbound.

1. In JDeveloper, from the **View** menu, select **Application Navigator**.
2. In the Application Navigator, right-click the application you are working with and select **New Project**. The New Gallery dialog box is displayed.
3. From the **Items** list, select **BPEL Process Project** then click **OK**. The BPEL Project Creation Wizard-Project Settings dialog box is displayed.

Figure 6–20 Outbound Project Settings

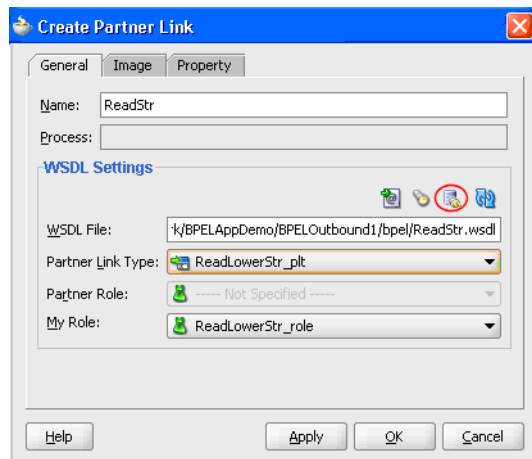


4. Do the following in BPEL Project Creation Wizard-Project Settings dialog box:
 - Enter a name for the BPEL process, for example **BPELOutbound1**.
 - From the Template list, select **Empty BPEL Process**.
5. Click **Next** to review the input/output schema elements or click **Finish**.

Create a File Adapter to Read the String

You create a file adapter to read a file as string input. SOA reads this string and passes it to a resource adapter. The resource adapter then processes the information in this string and returns it in Upper case. Do the following to create the file adapter.

1. Drag a **PartnerLink** into a service lane of the visual editor. The Create Partner Link dialog box is displayed.

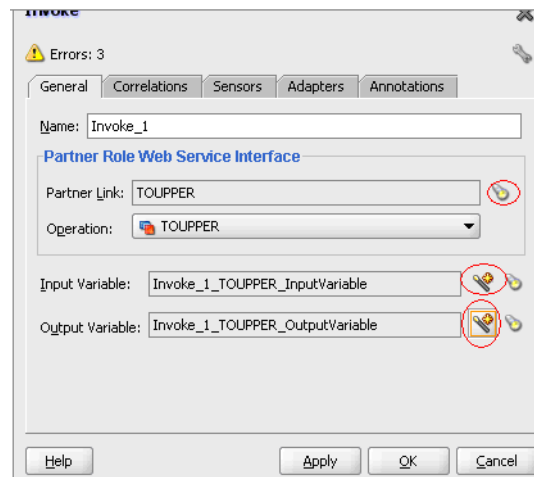
Figure 6–21 Partner Link for File Adapter

2. Click the **Adapter Wizard** button. This is the third button in the **WSDL Settings** section of the Create Partner Link dialog box, as shown in the figure above.
3. Enter the following information in the Adapter Configuration wizard:
 - a. Step 1, Adapter Type: Select **File Adapter**.
 - b. Step 2, Service Name: type `ReadStr`
 - c. Step 3, Operation:
 - Operation Type:** select **Read File**
 - Operation Name:** Type `ReadLowerStr`
 - d. Step 4, File Directories
 - Directory Names are Specified as:** Select **Physical Path**
 - Directory for Incoming Files (physical path):** Enter the full path to the folder where you want the incoming files, for example `C : \ temp`.
 - e. Step 5, File Filtering:
 - Includes Files with Name Pattern:** Enter `*.xml`
 - f. Step 6, File Polling:
 - Polling Frequency:** enter 1 then select **seconds** from the list.
 - g. Step 7, Messages:
 - Schema File URL:** Enter `TOUPPERSchema.xsd`
 - Schema Element:** Enter `ToUpperData`
4. Click **Finish**.

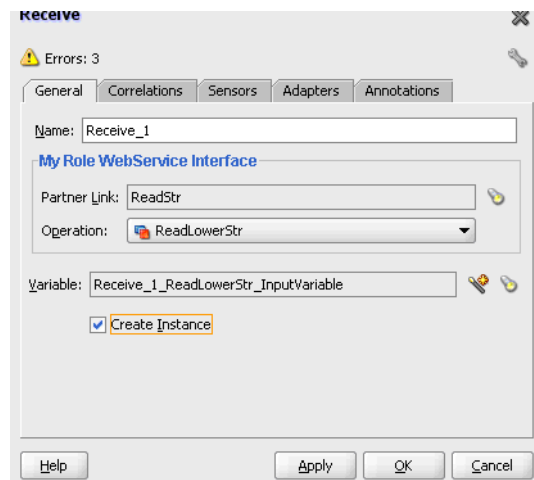
Design a BPEL Outbound Process to Read the String

In this step, you design the BPEL process. Do the following to design the BPEL outbound process:

1. From the Component Palette-Process Activities pane, drag a Receive, Assign, and Invoke activity into the editor for the process you created in the [Create a BPEL Project for a BPEL Outbound Process](#) step.

Figure 6–22 Outbound Process

2. In the Visual Editor, connect the Recieve_1 activity to the ReadStr file adapter. The Edit Receive dialog box is displayed.

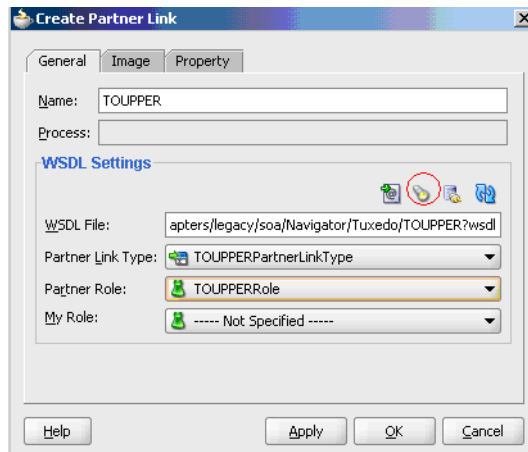
Figure 6–23 Outbound Receive Properties

3. Do the following in the Edit Receive dialog box:
 - Enter a name for the Receive activity, in this example it is `Receive_1`.
 - Create the variable. Click the first button to the right of the **Variable** field, then click **OK** in the Create Variable dialog box that is displayed.
 - Ensure that the **Create Instance** check box is selected.
 - Click **OK** to close the dialog box and accept the information.

Create a Partner Link to the Tuxedo Adapter

In this step, create a partner link to the Tuxedo adapter you are working with. Do the following to create a partner link.

1. Drag a **PartnerLink** into a service lane of the visual editor. The Create Partner Link dialog box is displayed.

Figure 6–24 Partner Link to Tuxedo Outbound

2. From the Create Partner Link dialog box, click the **Service Explorer** button. This button is the second from the left and is shown in the figure in the previous step. The Service Explorer dialog box is displayed.
3. In the Service Explorer, expand the connection you are using, then expand adapters and then applications, and continue to browse until you find the WSDL file you are using. Select the file and click **OK**.

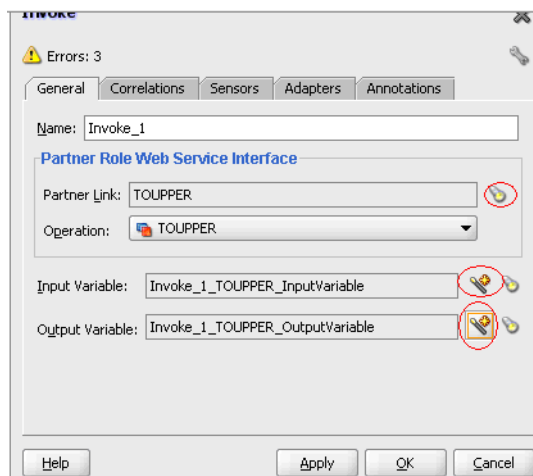
The **WSDL File** field in the Create Partner Link dialog box displays the path to the WSDL file you selected. The **Partner Link Type** field displays the Partner Link defined in the WSDL file.

4. Enter the following information in the Create Partner Link dialog box:
 - In the **Partner Role** list, select **TOUPPERRole**.
 - In the **My Role** list, select **Not Specified**.

Edit the Invoke_1 Activity

Edit the `Invoke_1` activity that you added to the Visual Editor so that it invokes the TOUPPER service. Do the following to edit the `Invoke_1` activity.

1. Double-click the `Invoke_1` activity to edit it. The Invoke dialog box is displayed.

Figure 6–25 Invoke Dialog Box

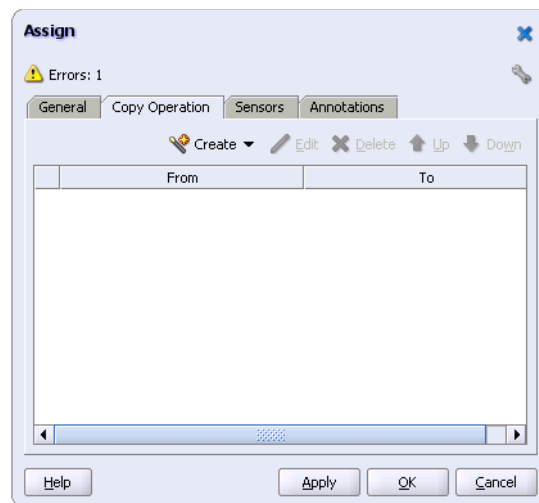
2. In the Invoke dialog box, ensure that the Partner Role Web Service Interface section has the following values:
 - **Partner Link:** TOUPPER
 - **Operation:** TOUPPER
3. Click the **Automatically Create Input Variable** button. It is the first icon to the right of the **Input Variable** field as shown in the figure in the previous step. The Create Variable dialog appears with the input variable.
4. The Input Variable should be Invoke_1_TOUPPER_InputVariable. Click **OK**.
5. Click the **Automatically Create Input Variable** button for the **Output Variable** field.
6. The Output Variable should be Invoke_1_TOUPPER_OutputVariable. Click **OK**.

Edit the Assign_1 Activity

You use the assign activity to assign a value to the resource adapter. Do the following to edit the Assign_1 activity.

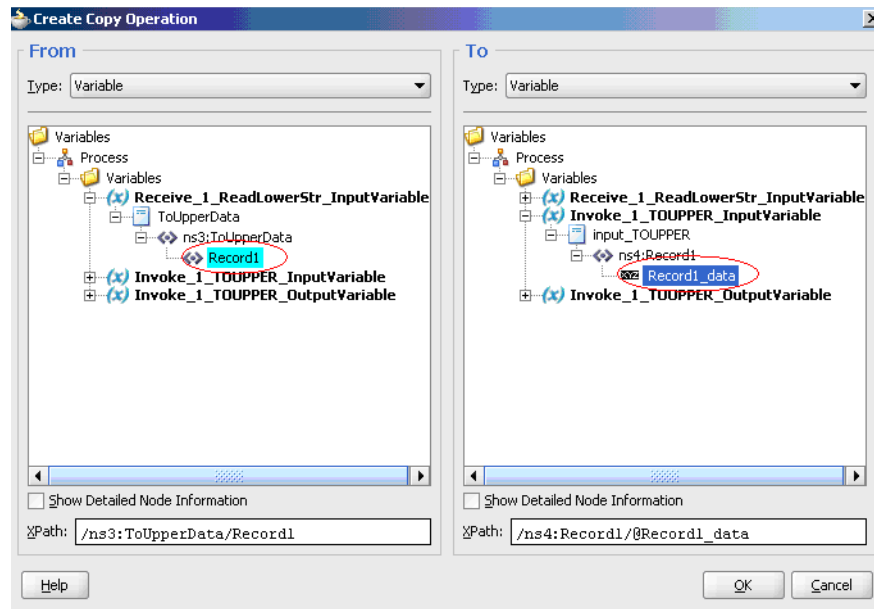
1. Double-click the Assign_1 activity to edit it. The Assign dialog box is displayed.

Figure 6–26 Assign Dialog Box



2. Click the **Create** list and select **Copy Operation**. The Create Copy Operation dialog box is displayed.

Figure 6–27 Create Copy



3. On the **From** side, expand **Variables** then expand **Read_1_ReadLowerStr_InputVariable**, then expand **nsx:ToUpperData** and select **Record1**.
4. On the **To** side, expand **Variables** then expand **Read_1_ReadLowerStr_InputVariable**, then expand **input:ToUPPER**, then expand **nsx:Record1** and select **Record1_data**.
5. Click **OK**. The Copy Operation tab in the Assign window updates to show the rule.

Create a Process to Write the Upper-Case Letter String

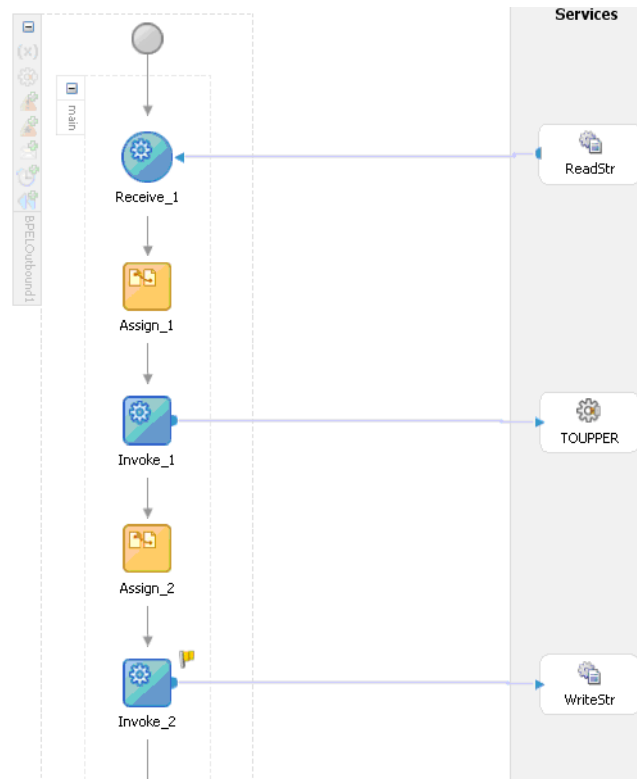
After you create a process that reads a string of lower-case letters, create another process that writes the string in upper case letters. Do the following to create a process that writes the string in upper-case letters.

1. Create a file adapter as described in [Create a File Adapter to Read the String](#). To write the string, use the following values:
 - **Service Name:** WriteStr
 - **Operation Type:** Write File
 - **Operation Name:** WriteUpperStr
 - **Directory for Incoming Files:** C:\tempOut
 - **File Naming Pattern:** *.xml
2. Add an Assign activity and an Invoke activity to the visual editor.
3. Double-click the Invoke_2 activity to edit it. See [Edit the Invoke_1 Activity](#) for an explanation on how to edit an invoke activity. Use the following values for the Invoke_2 activity:
 - **Partner Link:** WriteStr
 - **Operation:** WriteStr
 - **Input Variable:** Invoke_2_WriteStr_InputVariable

- **Output Variable:** Invoke_1_TOUPPER_OutputVariable
4. Double-click the Assign_2 activity to edit it. See [Edit the Assign_1 Activity](#) for an explanation on how to edit an Assign activity. Use the following values for the Assign_2 activity:
 - **From:** Expand **Variables** then expand **Invoke_1_TOUPPER_InputVariable**, then expand **output:ToUPPER**, then expand **nsx:TOUPPER_Response** and select **Record1_data**.
 - **To:** Expand **Variables** then expand **Invoke_2_1_WriteStr_InputVariable**, then expand **ToUpperData**, then expand **nsx:ToUpperData** and select **Record1**.

The following figure shows how the final process looks:

Figure 6–28 Final Outbound Process



Runtime Configuration

This section describes how to test the project and run it to see that the outbound process is executed successfully. Do the following to test the outbound process.

1. Create a file and call it `ToUpperIn.xml` as shown in the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<CU:ToUpperData
  xmlns:CU="http://xmlns.oracle.com/Esb/ToUpperData"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Record1>tuxedo</Record1>
</CU:ToUpperData>
```

2. Save the file to the `C:\temp` folder that was defined for the first file adapter. The file is deleted after the SOA server reads the file and complete the process.

3. Open the C:\tempOut folder that was defined in the second file adapter. A file with the following contents is copied to this folder.

```
<?xml version="1.0"?>  
<ToUpperData xmlns="http://xmlns.oracle.com/Esb/ToUpperData">  
  <Record1 xmlns="">TUXEDO</Record1>
```

Troubleshooting OracleAS Adapter for Tuxedo

Troubleshooting OracleAS Adapters for Tuxedo involves checking various definitions and properties in Oracle Connect, including daemon status, workspace options, server parameters, and various system logs.

This chapter contains the following topics:

- [Troubleshooting the Daemon](#)
- [Resolving Communication Errors](#)
- [Resolving Specific Errors](#)

Troubleshooting the Daemon

Troubleshooting the daemon and the communication between Oracle Application Server and OracleAS Adapters for Tuxedo is performed using Oracle Studio. It is used to monitor the daemon and server activity and control what happens to the daemon and server processes.

Note: The Tuxedo Gateway adapter logging goes to the Tuxedo log.

See Also: [Advanced Tuning of the Daemon](#) for details about the configuration settings

This section contains the following:

- [Starting the Daemon](#)
- [Shutting Down the Oracle Connect Daemon](#)
- [Monitoring the Daemon During Run Time](#)
- [Daemon Logs](#)

Starting the Daemon

The Oracle Connect daemon is started when OracleAS Adapters for Tuxedo are installed. In case you shut down the daemon, as described in [Shutting Down the Oracle Connect Daemon](#), you can restart the daemon as described in the following task.

Note: The Oracle Connect daemon is started only on the platform where OracleAS Adapters for Tuxedo run. It cannot be started remotely using Oracle Studio.

Task: Starting the Oracle Connect Daemon

- Enter the following command:

```
irpcd [-u username [-p password]] -l ip:2551 start
```

Where *username* is the name of a user with a permission to start the daemon, and *password* is the password for this user and *ip* is the ip address of the computer.

Shutting Down the Oracle Connect Daemon

To shut down the Oracle Connect daemon, use Oracle Studio, as follows.

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. Select the computer described in [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#).
3. Right-click the computer and select **Open Runtime Perspective**.
4. In the Runtime Explorer, right-click the computer and select **Shutdown Daemon**.

Monitoring the Daemon During Run Time

Use the Runtime Manager perspective of Oracle Studio to monitor the daemon during Attunity SQLServer-CDC for SSIS.chm. Perform the following steps:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. Right-click the computer described in [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#) and select **Open Runtime Perspective**.

You can manage the daemon by expanding the relevant daemon, workspace or server process, and choosing the relevant option, as described in the following sections.

Daemon (Computer) Options

Right-click the daemon to display the options available for it, including the ability to display the daemon log.

The following table lists the available daemon options:

Table 7–1 Daemon Options

Option	Description
Edit Daemon Configuration	Opens the daemon editor to enable you to reconfigure the daemon. For more information, see Advanced Tuning of the Daemon for details about the configuration settings.
Status	Checks the status of the daemon. The information about the daemon includes the name of the daemon configuration used, the active client sessions, and logging information.

Table 7–1 (Cont.) Daemon Options

Option	Description
Reload Configuration	Reloads the configuration after changes. Any servers currently started are not affected by the changed configuration. For more information, see Advanced Tuning of the Daemon for details about the configuration settings
View Log	Displays the daemon log. For more information, see Daemon Logs .
View Events	Displays the daemon events log.
Daemon Properties	Displays information about the computer where the daemon is running, such as the physical address and any OracleAS Adapter and password needed to access the computer
Shutdown Daemon	Shuts down the daemon on the computer.
Recycle servers	Closes all unused servers and prepares all active servers to close when the client disconnects. New connection requests are allocated with new servers.
Kill servers	Immediately closes all active and unused servers. Note: Use this option with caution, as it may lead to data loss.
Rename	Enables to change the name of the daemon displayed in the Runtime Explorer.
Remove	Removes the computer from the Runtime Explorer.
Refresh	Refreshes the display.

Workspace Options

Right-click a workspace to display the options available for the workspace, including the ability to display the workspace log.

The following options are available at the workspace level:

Table 7–2 Workspace Options

Option	Description
Edit Workspace Configuration	Open the daemon editor to enable you to reconfigure the workspace. For more information, see Advanced Tuning of the Daemon for details about the configuration settings.
Status	Checks the status of the workspace, whether it is available or not.
View Log	View the log for all servers for the workspace. For more information, see Daemon Logs .
View Events	Displays the workspace events log.
Recycle Servers	Closes all unused servers and prepares all active servers to close when the client disconnects. New connection requests are allocated with new servers.
Kill Servers	Immediately closes all active and unused servers. Note: Use this option with caution, as it may lead to data loss.

Table 7–2 (Cont.) Workspace Options

Option	Description
Remove	Removes the selected workspace from the Runtime Explorer.
Disable	Disables the selected workspace.
Refresh	Refreshes the display.

Server Options

Right-click a server to display the options available for the server, including the ability to display the server log.

The options available at the server level are listed in the following table:

Table 7–3 Server Options

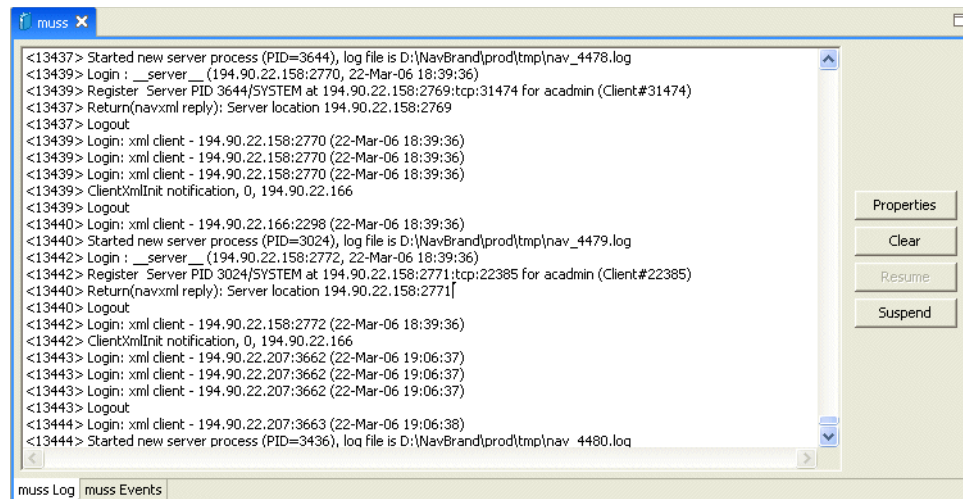
Option	Description
Status	Checks the status of the server. The information about the server includes the server mode and the number of active client sessions for the server.
View Log	View the server log. For more information, see Daemon Logs .
View Events	Displays the server events log.
Kill server	Ends the server process, regardless of its activity status. Note: Use this option with caution, as it may lead to data loss.
Refresh	Refreshes the display.

Daemon Logs

Oracle Connect produces several logs that you can use to troubleshoot problems. The daemon manages the following logs:

- The Daemon log
- The Workspace log
- The Server process log

The Runtime Manager perspective of Oracle Studio provides a monitor for these logs, as shown in the following figure:

Figure 7-1 The Legacy System Log tab

Display the required log by right-clicking the level item want (daemon, workspace or server) and selecting **View Log**. Each log is displayed in a separate tab. You can flick between logs by clicking the appropriate tab.

The Daemon Log Monitor

The daemon log displays activity between clients and the daemon, including clients logging in and logging out from the daemon.

You can change the level of logging by clicking **Properties**. The following levels of logging are available:

- none: The log displays who has logged in and out from the daemon.
- error: The log displays who has logged in and out from the daemon and any errors that have been generated.
- debug: The log displays who has logged in and out from the daemon, any errors that have been generated, and any tracing that has been specified in the daemon configuration.

See Also: [Daemon Logging](#).

The Workspace Log Monitor

The workspace log displays information about the workspace being used by the client.

You can change the level of logging by clicking **Properties**. The following levels of logging are available:

- none: The log displays who has connected and disconnected from the server process.
- error: The log displays who has connected and disconnected from the server process and any errors that have been generated.
- debug: The log displays who has connected and disconnected from the server process, any errors that have been generated, and any tracing that has been specified in the workspace configuration.

See Also: [Workspace General](#).

The Server Log Monitor

The server log displays activity between clients and the server process used by that client to handle the client request.

You can change the level of logging by clicking **Properties**. The following levels of logging are available:

- none: The log displays who has connected and disconnected from the server process.
- error: The log displays who has connected and disconnected from the server process and any errors that have been generated.
- debug: The log displays who has connected and disconnected from the server process, any errors that have been generated, and any tracing that has been specified in the daemon configuration.

See Also: Daemon [Logging](#).

Resolving Communication Errors

When Oracle Studio disconnects from the Tuxedo computer, the computer is displayed in Oracle Studio with an X in a red circle. If this situation occurs, try to access the computer later.

The following table describes the various scenarios that may exist when Oracle Application Server disconnects from the platform where OracleAS Adapters for Tuxedo run.

Table 7–4 Scenarios When a Client Is Disconnected

Scenario	Idle (Not Processing a Client Request)	Processing a Client Request
Explicit Disconnect (client explicitly closes connection or client program terminates)	The server is immediately notified of the disconnect and either becomes available for use by another client or terminates (if it is not reusable).	The server does not know that the client has disconnected and continues processing. When processing completes, the server tries to reply to the client and immediately gets an error that the connection was lost. The server either becomes available for use by another client or terminates (if it is not reusable).
Abrupt Disconnect (client closed without proper shutdown or client system hanged and communication disconnected)	The server does not know that the client has disconnected and remains in the idle state. After timing out based on whichever comes first of the value for the client idle timeout daemon workspace parameter or the TCP/IP KEEPALIVE parameter, the server is notified of the disconnect and either becomes available for use by another client or terminates (if it is not reusable).	The server does not know that the client has disconnected and continues processing. When processing completes, the server tries to reply to the client. After an interval (typically several minutes, depending on the TCP/IP configuration), during which the TCP/IP subsystem retries sending the message to the client, the server assumes that the client has terminated and notifies the server that the connection has been closed. The server either becomes available for use by another client or terminates (if it is not reusable).

To troubleshoot client/server communication problems, you must be familiar with the following:

- Daemon configuration settings.
- Oracle Connect security.
- TCP/IP subsystem. OracleAS Adapters for Tuxedo uses TPC/IP for internal inter-computer communications.

- System details, such as the account name and password of the administrator account, the IP address of the computers involved and whether a portmapper is being used.

Resolving Specific Errors

The following error messages relate to errors received from Oracle Connect.

C005: Could not open the IRPCD log file for write.

Cause: The daemon was not able to create or write to its log file. The log file is viewed using the Oracle Studio Runtime perspective, as described in [Daemon Logs](#). The log file location is set in the daemon configuration as described in [Daemon Logging](#).

Action: Check that the account where the daemon runs has permission to generate/write to the log file.

Action: Check the path specified for the log file in the daemon configuration.

Action: Check that there is no existing log file owned by another user at the specified location.

Action: Ensure that the disk device is not full.

C007: Server initialization failed.

Cause: The daemon failed to start its network service.

Action: Check the processes being run on the system to see whether another daemon or program is using the port specified in the `oc4j-ra.xml` file for the adapter.

Action: Check the TCP/IP subsystem on the current computer by trying to ping it or run FTP or Telnet to or from it.

Action: Check whether the daemon has privileges to use the TCP/IP services on the current computer with the port specified in the `oc4j-ra.xml` file for the adapter.

C008: Setting server event handler failed.

Cause: Internal error.

Action: Contact Oracle Support Services.

C009: IRPCD process has been terminated by user request.

Cause: This message is informational only. The daemon successfully shut down.

Action: No action required.

C00A: Application %s not found.

Cause: The requested workspace does not exist.

Action: Check that the workspace defined in the `oc4j-ra.xml` file is also defined in the daemon configuration on the platform where OracleAS Adapter for Tuxedo runs. Use the Status option in the Runtime Manager perspective.

C00B: Invalid IRPCD client context.

Cause: A non-Oracle Connect program is trying to connect to the daemon.

Action: Check the processes and terminate the relevant process with a system command.

C00C: Daemon request requires a server login.

Cause: A non-Oracle Connect server or program was trying to use a daemon service reserved for Oracle Connect servers.

Action: Check the processes and terminate the relevant process with a system command.

C00D: Daemon request requires a client login.

Cause: The requested daemon requires a valid client login, which was not supplied.

Action: Reissue the command and specify a OracleAS Adapter and password.

Action: Edit the user profile in Oracle Studio to specify a valid OracleAS Adapter and password for the platform where OracleAS Adapters for Tuxedo run.

See Also: [Configuring Run-Time User Access to OracleAS Adapter for Tuxedo.](#)

C00E: Daemon request requires an administrator login.

Cause: The requested daemon service requires an administrative login.

Action: Edit the daemon security in Oracle Studio to specify a valid administrator OracleAS Adapter and password.

See Also: [Daemon Security.](#)

C00F: Anonymous client logins are not allowed.

Cause: The daemon is configured to require a valid username and password, which were not supplied.

Action: Enable anonymous client access in daemon security in Oracle Studio.

See Also: [Daemon Security.](#)

Action: Edit the user profile in Oracle Studio to specify a valid OracleAS Adapter and password for the platform where OracleAS Adapters for Tuxedo runs.

See Also: [Configuring Run-Time User Access to OracleAS Adapter for Tuxedo.](#)

C010: Anonymous server logins are not allowed.

Cause: Internal error.

Action: Contact Oracle Support Services.

C011: Client has already timed out.

Cause: A server process was started on behalf of a client and the client has timed out before the server completed its startup.

Action: Increase the Connect timeout value for the server workspace in the Workspace **General** tab.

See Also: [Workspace General](#)

C012: Invalid username/password.

Cause: Invalid OracleAS Adapter/password supplied when logging on to the daemon.

Action: See the daemon log file because the OracleAS Adapter/password were not accepted.

Action: Edit the user profile in Oracle Studio to specify a valid OracleAS Adapter and password for the platform where OracleAS Adapters for Tuxedo run.

See Also: [Configuring Run-Time User Access to OracleAS Adapter for Tuxedo](#).

C014: Client connection limit reached - try later.

Cause: The maximum number of server processes for the workspace has been reached, and none of the active servers could accept the client connection.

Action: Increase the value of the Set maximum number of servers and Maximum parameter for the Clients per server limit field in the **Server Mode** tab.

See Also: Workspace [Server Mode](#).

Action: Try running the command later.

C015: Failed to start server process.

Cause: The Oracle Connect daemon failed to start a server process or the started server failed upon starting up.

Action: See the daemon and server log files for the reason the server did not start. For example, you might receive an message with a reason specified in the log file similar to the following: [C015] Failed to start NAVIGATOR server process: No server account name defined for anonymous client; code: -1601: SQL code: 0

Action: If you use impersonation, check the user profile on the client. Also see C069.

C016: Unexpected server state.

Cause: Internal error.

Action: Contact Oracle Support Services.

C017: Active daemon clients exist. Shutdown canceled.

Cause: One or more clients are still connected to the daemon.

Action: Wait until all the clients log off the daemon and then retry the shutdown operation.

C019: Request is not granted because someone else is locking it.

Cause: A request to lock a resource managed by the daemon was denied because another user has locked the resource.

Action: Wait for the other user to release the resource.

C01A: Lock %s not found.

Cause: A request to free a resource was denied because the caller did not lock that resource (for example, another user shut down the daemon you are working with).

Action: Contact Oracle Support Services.

C01B: Unexpected error in %s.

Cause: Internal error.

Action: Contact Oracle Support Services.

C01C: Cannot update configuration without _APPLICATIONS lock.

Cause: Internal error.

Action: Contact Oracle Support Services.

C01D: Need to lock the application first.

Cause: Internal error.

Action: Contact Oracle Support Services.

C01F: Cannot set configuration of a deleted application.

Cause: Internal error.

Action: Contact Oracle Support Services.

C020: Failed in looking up host name (gethostname())

Cause: Cannot connect to the remote computer.

Action: Check that the name specified for the computer in the oc4j-ra-xml file is correct.

Action: Check that a domain name server (DNS) is available to look up the host name.

Action: Check the TCP/IP subsystem on the computer by trying to ping it or run ftp or telnet to or from it.

C021: Required variable %s not found

Cause: An environment variable required by the Oracle Connect server was not defined when the server started up.

Action: Check whether the startup script makes any changes to the environment variables used by Oracle Connect.

Action: Check whether the system-defined environment size is sufficiently large for Oracle Connect.

C022: Server failed to connect and register with the daemon.

Cause: An Oracle Connect server started by the daemon was not able to connect or register back with the daemon.

Action: Try to connect again.

Action: Increase the Connect timeout value for the server workspace in the Workspace **General** tab.

See Also: Workspace [General](#)

Action: Check that the startup script for the workspace launches the correct version of Oracle Connect.

Action: Increase the value of the Set maximum number of servers and Maximum parameter for the Clients per server limit in the **Server Mode** tab.

See Also: Workspace [Server Mode](#).

C023: Call made to unregistered module %d.

Cause: Internal error.

Action: Contact Oracle Support Services.

C024: Failed to create a socket.

Cause: An error occurred within the TCP/IP subsystem.

Action: Check whether you have sufficient system privileges.

Action: Check the TCP/IP subsystem on the computer by trying to ping it or run ftp or telnet to or from it.

C025: Failed to set socket option %s

Cause: An error occurred within the TCP/IP subsystem.

Action: Check whether you have sufficient system privileges.

Action: Check the TCP/IP subsystem on the computer by trying to ping it or run ftp or telnet to or from it.

C026: Failed to bind server to port %s

Cause: An Oracle Connect server or daemon was not able to bind to the specified port.

Action: Check whether another program is holding the port that was specified in the oc4j-ra-xml file for the adapter.

Action: Check whether you have sufficient system privileges.

C027: Cannot create TCP service for %s

Cause: An error occurred within the TCP/IP subsystem

Action: Check the TCP/IP subsystem on the computer by trying to ping it or run ftp or telnet to or from it.

C028: Unable to register (%s, %d, tcp)

Cause: This error may happen when a portmapper is used (*host:a*) but the portmapper is not available.

Action: Enable the portmapper.

Action: Avoid using the portmapper (by not using “:a” when starting the daemon).

C029: Failed to create a server thread

Cause: Internal error.

Action: Contact Oracle Support Services.

C02A: Server thread failed to start

Cause: Internal error.

Action: Contact Oracle Support Services.

C02B: Stopping the %s server - no client

Cause: A server that was started by the Oracle Connect daemon to service a client did not get a client connection request within one minute. The server terminates.

Action: In most cases, the client was terminated by a user request, so no specific action is required.

Action: If no client can connect to the server, it may be that the server has multiple network cards and the Oracle Connect daemon is not aware of this. In this case, start the daemon with an IP address.

C02C: Unexpected event - a termination signal intercepted

Cause: Internal error.

Action: Contact Oracle Support Services.

C02D: Modified transport, context unknown/lost

Cause: Internal error.

Action: Contact Oracle Support Services.

C02F: Corrupted arguments passed to procedure

Cause: Internal error.

Action: Contact Oracle Support Services.

C030: Unable to free arguments for %s() of %s

Cause: Internal error.

Action: Contact Oracle Support Services.

C031: Cannot register a non-module RPC %s

Cause: Internal error.

Action: Contact Oracle Support Services.

C032: An IRPCD program is required

Cause: Internal error.

Action: Contact Oracle Support Services.

C033: An IRPCD super-server is required for module events

Cause: Internal error.

Action: Contact Oracle Support Services.

C034: An invalid super-server module ID was specified, %d

Cause: Internal error.

Action: Contact Oracle Support Services.

C035: Out of memory

Cause: Not enough memory to service a client request.

Action: Increase process memory quota or add memory to the system.

C036: Failed to register RPC procedure module %s

Cause: Internal error.

Action: Contact Oracle Support Services.

C037: Failed to register an invalid RPC procedure number %x

Cause: Internal error.

Action: Contact Oracle Support Services.

C038: Cannot re-register RPC procedure number %x

Cause: Internal error.

Action: Contact Oracle Support Services.

C042: Remote call to %s failed; %s

Cause: Remote call to API failed.

Action: Check the daemon log file.

Action: If necessary, change the level of detail written to the log file to help resolve the problem.

See Also: Daemon [Logging](#).

C043: Failed to connect to host %s;%s

Cause: The remote host is not correctly defined to Oracle Connect or is not working.

Action: Check the remote computer definition in the `oc4j-ra.xml` file for the adapter.

Action: Check that the daemon is up on the platform where OracleAS Adapters for Tuxedo run. Use the Status option in the Runtime Manager perspective.

Action: Check the network connection by trying to ping the host computer or run ftp or telnet to or from it.

C045: Failed to create a service thread

Cause: The server failed to create a thread to service a client request.

Action: A system or process quota limit has been exceeded. Either increase the quota or lower the `Clients per server limit` field value in the Workspace **General** tab.

See Also: Workspace [General](#)

C047: %s out of memory

Cause: Not enough memory was available to Oracle Connect to complete a requested operation.

Action: Terminate unnecessary processes running on the server.

Action: Add more memory to the system.

Action: Allow the process to use more memory.

Action: Limit the number of processes the daemon may start. If the demand for servers exceeds the number of available servers, clients get a message telling them the maximum number of servers has been reached and asking them to try again later.

C066: Communication error with the server%s

Cause: Connection to the Oracle Connect daemon or server failed, or an established session with a server has failed.

Action: Check the remote computer definition in the `weblogic-ra.xml` file.

Action: Check that the daemon is up on the platform where OracleAS Adapters for Tuxedo run. Use the Status option in the Runtime Manager perspective.

Action: In case of a network problem, check the network connection by trying to ping the host computer or run ftp or telnet to or from it.

C067: Unexpected error occurred in server function %s

Cause: One of the server functions has exited with an exception, such as an `abend`, or an Invalid Instruction.

Action: Contact Oracle Support Services.

C068: Fail to login daemon

Cause: The daemon is not running on the server computer.

Action: Use the Status in Oracle Studio Runtime Manager perspective to check whether a daemon is running on the server

Action: Have the system administrator reinstall Oracle Connect on the server.

C069: Fail to get server

Cause: The Oracle Connect daemon on the server computer could not start a server process to serve the client. A separate message provides more detail on why the server process could not start.

Action: There are many possible causes of this error. If the cause is not clear from the related message, see the Oracle Connect daemon log file on the server

Action: The resolution to this error is highly dependent on the particular cause. The following are some typical causes and resolutions.

Action: Some process creation quota was exceeded. Either try again later or increase the quota or the other relevant system resources.

Action: The server startup script failed.

Action: The OracleAS Adapter given is not allowed to use the requested server. Use an authorized OracleAS Adapter.

Action: A limit on concurrent clients for a server has been reached. Try again later.

Action: If you use impersonation, check the user profile on the client. Also see C015.

C06A: Failed to connect to server

Cause: The server assigned to the client did not accept the client connection. A separate message provides more detail about why the server process did not accept the connection.

Action: See the daemon and server log files because the server was not available to accept its assigned client.

Action: If a shared server processes are used and many clients are trying to connect to it at the same time, some may get a Connection Refused error if the TCP/IP request queue fills up.

C06B: Disconnecting from server

Cause: A network failure, or a server computer failure or a server program failure caused the connection to stop. The currently active transaction is stopped as well.

Action: Oracle Connect automatically tries to reestablish a connection with a server upon the next SQL command issued against the server. Once the network or computer failure is corrected, the connection to the daemon is reestablished automatically.

C070: Server failed to send reply to the client

Cause: Server terminated unexpectedly.

Action: Unless the client was intentionally stopped (for example, using Control-C), contact Oracle Support Services.

C071: Connection to server %s was disconnected. Cursors state was lost.

Cause: Either a network failure, a server computer failure, or server program failure caused the connection to stop. The currently active transaction is stopped as well.

Action: Normally, Oracle Connect automatically tries to create a session with the server upon the next attempt to access the server. If the network and server are accessible, the next operation should succeed. Otherwise, the network or server should be fixed before connection can be resumed.

Action: In case of a server malfunction that is not related to callable user code, contact Oracle Support Services.

C072: Reconnect to server %s

Cause: This is an informational message only. The client has reestablished its connection with the server.

Action: No action is required.

C073: The parameters passed to the admin server are invalid: %s

Cause: Internal error.

Action: Contact Oracle Support Services.

C074: No authorization to perform the requested operation (%s)

Cause: User/account has insufficient privileges.

Action: Grant administrative privileges to the user/account using the Administrator parameter of the Daemon **Security** or Workspace **Security** tabs.

See Also: Daemon [Security](#) or Workspace [Security](#).

C075: Failed to register daemon in the TCP/IP service table

Cause: Registration of the daemon in the TCP/IP services file has failed.

Action: Check that the account running the daemon has the permissions to update the TCP/IP services file.

E001: Failed in lock/release operation

Cause: A lock or release operation of a global resource has failed. A separate message provides more details. The separate message specifies the cause of this error.

Action: There are various causes for this error, including lack of sufficient privileges or a system resource shortage.

J0006: Operation on already closed connection was requested

Cause: A request using a connection that was closed was attempted.

Action: Reopen the connection and try again.

J0028: Internal Error: Unknown XML tag %s

Cause: Internal error.

Action: Contact Oracle Support Services.

J0030: Internal Error: Method %s needs to be overwritten

Cause: Internal error.

Action: Contact Oracle Support Services.

J0031: Internal Error: Required attribute %s not found in %s verb

Cause: Internal error.

Action: Contact Oracle Support Services.

J0032: Internal Error: %s ACP object was returned instead of %s as expected

Cause: Internal error.

Action: Contact Oracle Support Services.

J0033: Internal Error: Attempt to work with closed socket

Cause: Internal error.

Action: Contact Oracle Support Services.

J0034: Internal Error: corrupted message; %s bytes read instead of %s as expected

Cause: XML sent from the client to the server has become corrupted.

Action: Check compression settings for XML transferred from the client to the server. If the settings are correct, retry sending the request from the client to the server.

J0035: Internal Error: Invalid redirection address %s returned by daemon

Cause: Internal error.

Action: Contact Oracle Support Services.

J0036: %s: %s

Cause: One of the following errors was received from the server: 0 - server.internalError, 1 - client.xmlError, 2 - client.requestError, 3 - client.noActiveConnection, 4 - server.resourceLimit, 5 - server.redirect, 6 - client.noSuchResource, 7 - client.authenticationError, 8 - client.noSuchInteraction, 9 - client.noSuchConnection, 10 - server.notImplemented, 11 - server.xaProtocolError, 12 - server.xaUnknownXID, 13 - server.xaDuplicateXID, 14 - server.xaInvalidArgument, 15 - client.autogenRejected, 16 - server.xaTransactionTooFresh, 17 - server.resourceNotAvailable, 18 - client.authorizationError, 19 - server.configurationError

Action: Review the server log file to determine the problem.

J0037: Internal Error: No ACP response when %s was expected

Cause: Internal error.

Action: Contact Oracle Support Services.

J0039: Internal Error: ACP root is not found in the XML

Cause: Internal error.

Action: Contact Oracle Support Services.

J0040: Internal Error: Input record is required for interaction %s execution

Cause: Internal error.

Action: Contact Oracle Support Services.

J0048: Invalid metadata type %s is passed to %s function

Cause: A request for metadata was not fulfilled.

Action: Check the validity of the request.

J0050: Key of the put method must be of type string

Cause: In either a GET or PUT operation, the key must be a string.

Action: Change the key used in the operation to a valid key.

J0059: Value %s is invalid for attribute %s

Cause: A request for metadata was not fulfilled.

Action: Check the validity of the request.

J0068: Value must be of type string

Cause: In a PUT operation, the value must be a string.

Action: Change the value used in the operation to a valid value.

J0069: Value must be of type MappedRecord

Cause: In a PUT operation, the value must be a mapped record.

Action: Change the value used in the operation to a valid value.

J0070: Value must be of type "MappedRecord[]"

Cause: In a PUT operation, the value must be mapped record array.

Action: Change the value used in the operation to a valid value.

J0071: Bad key for mapped record, #element or #element[] is required

Cause: In a PUT operation, the value must be mapped record array.

Action: Change the key used in the record to a valid key.

J0072: Value must be of type Object[]

Cause: In a PUT operation, the value must be mapped record array.

Action: Change the value used in the operation to a valid value.

J0078: In nonpersistent connection and non keep alive encryption is not supported - ignored

Cause: Encryption is not supported for nonpersistent connections.

Action: There is no action to take. This warning can be ignored.

J0079: Invalid argument passed to %s - Argument: %s, Value: %s

Cause: The value pass.

Action: Change the argument used to a number.

Advanced Features of OracleAS Adapter for Tuxedo

Oracle Connect includes several tuning parameters that can improve performance. Specifically, the daemon can be configured to optimize communication between the OracleAS Adapter for Tuxedo and a client. In addition, the binding environment can be tuned to optimize the request handling.

This chapter contains the following topics:

- [Configuring the Daemon for High Availability](#)
- [Configuring a Binding Environment](#)
- [Migration Considerations](#)
- [Security Considerations](#)
- [Transaction Support](#)

Configuring the Daemon for High Availability

The daemon workspace is responsible for allocating server processes to clients. You can configure a workspace to use a pool of server processes so that a server process is always available for a client request. Use Oracle Studio to maintain daemon and daemon workspace parameters to control the allocation of server processes and their management in a pool.

The daemon manages a pool of server processes on the legacy platform. Oracle Application Server manages connection pooling of the J2CA 1.5 Tuxedo adapter.

You can also have several daemon workspace configurations, enabling you to create individual workspaces for use with different adapters.

Adding a New Daemon Workspace Configuration

Use Oracle Studio to add a new daemon configuration. You can set up different daemon configurations for different situations. On the computer where Oracle Connect is installed, perform the following steps:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. In the Design perspective Configuration view, expand the **Machine** folder.
3. Expand the machine you are working with.
4. Expand the **Daemons**. The available daemon configurations are listed.
5. Right-click **IRPCD** and select **New Workspace**.

6. In the New daemon workspace screen, specify a name for the new workspace and optionally, provide a description.
7. Specify whether you want default settings or copy the properties of an existing workspace.
To copy the properties of an existing workspace, click **Ellipsis** and select the workspace from which you want to copy the properties.
8. Click **Next**. The Select Scenario screen is displayed.
9. Select **Application Server** using connection pooling and click **Next**.
10. Continue through the wizard, specifying the required values for the workspace.
11. To complete the workspace definition, click **Finish**.

The new workspace is now displayed under the IRPCD daemon.

Editing the Workspace

You edit a workspace by using the tabs described in the following table:

Table 8–1 Workspace Properties tabs

Tab	Description
General	Specifies general information including the server type, the command procedure used to start the workspace, the binding configuration associated with this workspace (which dictates the data sources and applications that can be accessed) the timeout parameters, and logging information (which dictates the data sources and applications that can be accessed), the timeout parameters, and logging information.
Server Mode	Contains the workspace server information including features that control the operation of the servers started up by the workspace and allocated to clients.
Security	Specifies administration privileges, user access, ports available to access the workspace and workspace account specifications.

Use Oracle Studio to access these tabs, as follows:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. In the Design perspective Configuration view, expand the **Machines** folder and then expand the machine where you want to edit the workspace.
3. Expand the **Daemons** folder. The daemon available on this computer are listed.
4. Expand the **IRPCD** daemon. The daemon workspaces are listed.
5. Right-click the workspace you are editing and select **Open**.
6. Click the tab that contains the information you want to edit. For full details of the tabs and the fields in these tabs, see [Workspaces](#).
7. After editing the workspace, click **Save**.

Configuring the Server Mode

The server mode determines how the daemon starts new processes. The daemon supports the following server modes:

- **singleClient:** Each client receives a dedicated server process. The account in which a server process runs is determined either by the client login information or by the specific server workspace.
- This mode enables servers to run under a particular user account and isolates clients from each other (because each receives its own process). However, this server mode incurs a high overhead due to process startup times and may use a lot of server resources (because it requires as many server processes as concurrent clients).
- **multiClient:** Clients share a server process and are processed serially.
- This mode has low overhead because the server processes are initialized. However, because clients share the same process, they may impact one another, especially if they issue lengthy queries.

The number of clients that share a process is determined by the Clients per server limit (the maximum number of concurrent clients a server process for the current workspace accepts).

- **multiThreaded:** When Tuxedo runs on a Windows platform, the server process can be multi-threaded.
- **reusable:** This is an extension of the single client mode. When the client processing finishes, the server process does not die and can be used by another client, reducing startup times and application startup overhead.

This mode does not have the high overhead of single client mode because the servers are initialized. However, this server mode may use a lot of server resources (because it requires as many server processes as concurrent clients).

The other modes can be set so that the server processes are reusable by setting the number of times a process can be reused with the Reuse limit value (the maximum number of times a particular server process can be reused or how many clients it can serve before it is retired). Reuse of servers enhances performance because it eliminates the need to repeat initializations. However, reuse runs a risk of higher memory leakage over time. The default for the Reuse limit field value is None, indicating that no reuse limit is enforced.

Set the server mode in the **Server Mode** tab of the daemon workspace editor, as shown in the following figure:

Figure 8–1 The Server Mode Tab

When using any of the server modes you can specify a pool of server processes. These server processes are started when the daemon starts and are maintained in a pool. The server processes are available for use by new client requests from the pool, saving initialization time. Instead of starting a new server process each time one is requested by a client, the client receives a process immediately from the pool of available processes. When the client finishes processing, this server process either dies, or if reusable servers have been specified, it is returned to the pool.

You set up a pool of server processes by specifying the following parameters in the WS Server tab.

- Initial number of servers:** The number of server processes that are prestarted for this workspace when the daemon starts. These are available for use by new client processes with minimal initialization time. Instead of starting a new server process each time one is requested by a client, the daemon immediately allocates (to the client) a server from a pool of available servers. When the number of available server processes drops lower than the value specified in the Minimum number of available servers field, the daemon again starts server processes until the specified number of available servers is reached. The default for this parameter is 0, meaning that no servers are prestarted for this workspace.
- Minimum number of available servers:** The minimum number of server processes in the prestarted server's pool before the Oracle Connect daemon resumes creating new server processes (up to the number specified in the Initial number of servers field value, described earlier). If this parameter is set to a value greater than that of the Initial number of servers field value, the daemon considers the value to be the same as the value specified in the Initial number of servers field. In this case, a new server process is started and added to the pool each time a server process is removed from the pool and allocated to a client). The default for this parameter is 0, which means that new servers are created only when there are no other available servers.

- **Set maximum number of servers:** The maximum number of available server processes pooled for this workspace. If the server is reusable, once a client disconnects from the server, the daemon returns the server to the pool of available servers. If the limit is reached, excess server processes are discarded.

Configuring a Binding Environment

Each binding configuration includes the following information:

- Environment settings, which are used to configure the environment used by any of the adapters defined in the binding.
- Application adapters on the current computer.

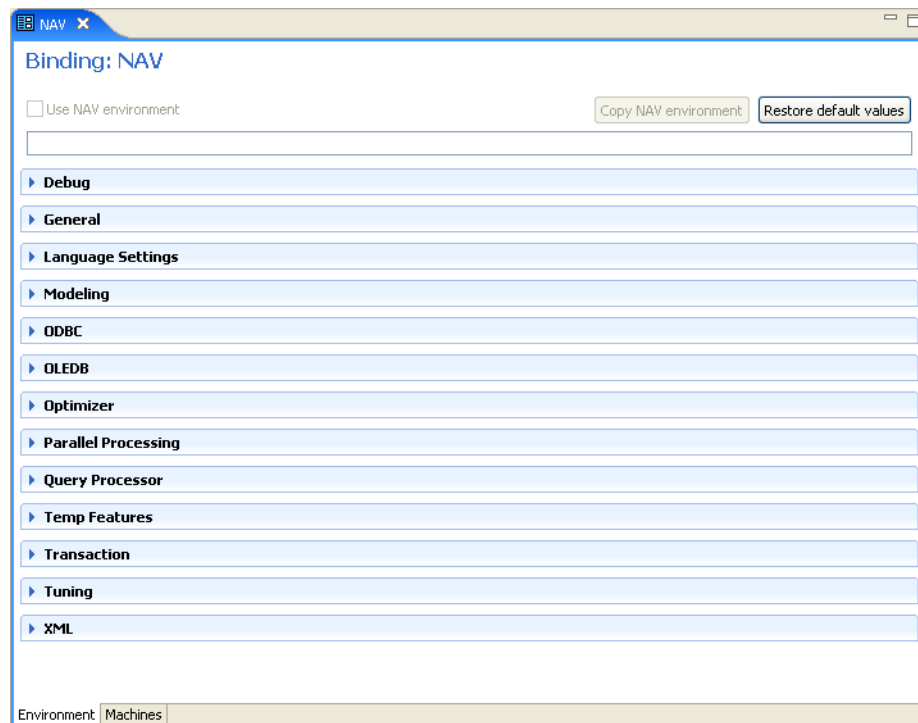
Configuring adapters is described in [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#).

To configure environmental settings in Oracle Studio, perform the following steps:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. In the Design perspective Configuration view, expand the **Machine** folder, then expand the machine where you want to configure the binding.
3. Expand the **Bindings** folder. The binding available on this computer are listed.
4. Right-click **NAV** and select **Open**.
5. In the **Environment** tab edit the required environment settings as needed. To edit an environment setting, click the required property category and then click the required value to edit.

The binding **Environment** tab is displayed in the following figure:

Figure 8–2 The Binding Properties tab



The binding environment is divided into the following categories:

- [Debug](#)
- [General](#)
- [Language](#)
- [Modeling](#)
- [Modeling](#)
- [Modeling](#)
- [ODBC](#)
- [OLEDB](#)
- [Optimizer](#)
- [Query Processor](#)
- [Parallel Processing](#)
- [Transactions](#)
- [Tuning](#)
- [XML](#)

Debug

The following table lists the parameters that define debugging and logging operations:

Table 8–2 Debug Parameters

Parameter	Description
ACX trace	Select this for the input xml sent to the back-end adapter and the output xml returned by the back-end adapter to be written to the log.
GDB Trace	This parameter is not applicable for use with OracleAS Adapter for IMS/DB.
General trace	Select this to log general trace information. The default writes only error messages to the log. Note: Changing the default setting can degrade performance.
Query warnings	This parameter is not applicable for use with OracleAS Adapter for IMS/DB.
Add timestamp to traced events	Select this to add a OracleAS Adapter on each event row in the log.
Query Processor trace	This parameter is not applicable for use with OracleAS Adapter for IMS/DB.
Binary XML Log Level	Select the binary XML log level from the list. The following logging levels are available: <ul style="list-style-type: none"> ■ None ■ API ■ Info ■ Debug

Table 8–2 (Cont.) Debug Parameters

Parameter	Description
Log file	The high-level qualifier of the log file for messages. The following type of message are written to the log: <ul style="list-style-type: none"> ■ Error messages ■ Trace information and information about the query optimization strategy, if <code>generalTrace</code> is set to <code>true</code>.
Trace Directory	This parameter is not applicable for use with OracleAS Adapter for IMS/DB.
Optimizer trace	This parameter is not applicable for use with OracleAS Adapter for IMS/DB.
Transaction extended logging	Select this for the transaction manager to write additional information about transactions to the log.

General

The following table lists the parameters that define general operations where temporary files are written:

Table 8–3 General Parameters

Parameter	Description
NAV_UTIL editor	This parameter is not applicable for use with OracleAS Adapter for IMS/DB.
Temporary Dir	The directory where temporary files are written, including the temporary files created for use by hash joins and for sorting files. The default is the current high-level qualifier.
Year 2000 policy	<p>Determines the way two-digit years are converted into four-digit years. When the <code>year2000Policy</code> parameter is not set, or when it is set to a value outside the range of values defined for the policy, as described in the following paragraphs, a default value of 5 and the Sliding Base Year policy is used. Two policies are provided:</p> <p>Fixed Base Year: <code>year2000Policy</code> is set to a value greater than, or equal to 1900. In this case, the value of <code>year2000Policy</code> is the first four-digit year after 1900 that can be represented by a two-digit year. For example, if <code>year2000Policy</code> is set to 1905, the years 2000->2004 are represented by 00->04. All other two digits are mapped to 19xx.</p> <p>This solution is most required if there is live data at the low end (close to the year 1900), which the user wants to keep with the current two-digit format.</p> <p>The user probably changes the base date only after ensuring that these old dates have been deleted from the data source.</p> <p>Sliding Base Year: <code>year2000Policy</code> is set to a positive value less than 100. In this case, the value of <code>year2000Policy</code> represents the number of years ahead of the current year that can be represented by a two-digit number. With each passing year the earliest year that can be represented by a two-digit number changes to a year later.</p>

Table 8–3 (Cont.) General Parameters

Parameter	Description
Cache buffer size	Enter the number of bytes to be used for a memory buffer on a client computer, which is used by the Oracle Connect client/server to store read-ahead data. The default is 200000.

Language

The following table lists the parameters that define globalization support:

Table 8–4 Language Settings Parameters

Parameter	Description
Language	Identifies the application language. A default Attunity SQLServer-CDC for SSIS.chm is selected based on the value specified for this parameter. For more information, see Globalization Settings .
Code Page	For use with globalization support to identify the Attunity SQLServer-CDC for SSIS.chm for the workspace. For more information, see Globalization Settings .
NLS String	Specifies the Attunity SQLServer-CDC for SSIS.chm used by a field whose data type is defined as <code>nlsstring</code> . Use this for a field whose Attunity SQLServer-CDC for SSIS.chm is other than that of the computer Attunity SQLServer-CDC for SSIS.chm. This parameter includes the following values: <ul style="list-style-type: none"> ■ The name of the Attunity SQLServer-CDC for SSIS.chm. ■ Whether the character set reads from right to left (as in middle eastern character sets). <p>The default is <code>false</code>.</p>

Modeling

The **Modeling** parameters are not applicable with OracleAS Adapter for Tuxedo.

ODBC

The **ODBC** parameters are not applicable for use with OracleAS Adapter for Tuxedo.

OLEDB

The **OLEDB** parameters are not applicable for use with OracleAS Adapter for Tuxedo.

Optimizer

The **Optimizer** parameters are not applicable for use with OracleAS Adapter for Tuxedo.

Query Processor

The **Query Processor** parameters are not applicable for use with OracleAS Adapter for Tuxedo.

Parallel Processing

The **Parallel Processing** parameters are not applicable for use with OracleAS Adapter for Tuxedo.

Transactions

The **Transactions** parameters are not applicable for use with OracleAS Adapter for Tuxedo.

Tuning

The **Tuning** parameters are not applicable for use with OracleAS Adapter for Tuxedo.

XML

The following table lists the parameters that define XML support:

Table 8–5 XML Category Parameters

Parameter	Description
COM maximum XML in memory	Specifies the maximum size of an XML document held in memory. The default is 65535 bytes.
COM maximum XML size	Specifies the maximum size of an XML document passed to another computer. The default is 65535 bytes. Note: When you increase this value for this property, you may need to increase the value for the Maximum XML in memory property in the daemon. For more information on daemons, see Control .
COM XML transport buffer size	Enter the maximum size of the internal communications buffer. The default value (-1) indicates there is no size limit.
XML date format	Enter the date format to use for XML. The options are: <ul style="list-style-type: none"> ■ ISO (the default): The date format is: YY-MM-DDThh:mm:ss [.ss. .] ■ ODBC: The date format is: YYYY-MM-DD HH:MM:SS [.NNN. . .]
Replace invalid XML characters	Select this to replace invalid XML characters with a '?'. This property is used for diagnostic and troubleshooting purposes.
XML trim char columns	Select this to enable padded spaces to be trimmed from XML string columns when the record format is fixed. By default this is selected, and padded spaces are trimmed for fixed size character columns. If you do not want this behavior, clear this check box.

Migration Considerations

You can migrate an adapter configuration from one platform to another. The configuration information is stored in the Oracle Connect repository on the source platform and is exported to an XML file which can then be imported to the target platform. Note that when migrating a configuration, file names and paths that are specific to the source platform must be changed to valid files on the target platform.

Perform the following steps to migrate an adapter configuration using Oracle Studio:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.

2. In the Configuration Explorer, right-click the required computer and select Export XML definitions.
3. Specify the path and name of the XML file, which stores the XML representation and complete configuration.
4. Edit any paths in the XML definition to the paths required on the target platform. For example, the setting for the `serverLogFile` property might need changing, depending on the platform.
5. Set up the target platform in Oracle Studio in the same way you set up the source platform. For more information, see [Configuring OracleAS Adapters for Tuxedo in Oracle Studio](#).
6. In the Configuration Explorer, right-click the target computer and select Import XML definitions.
7. Import the XML file to the target platform.

Security Considerations

Oracle Connect works within the confines of the legacy platform security system. In addition, Oracle Connect provides the following security components:

- A binary XML encryption mechanism, which is activated as follows:
 1. The client's first message to the server includes a pre-defined shared key, including the key name and value in the connection string. The server gets the key value for the key name passed from the client from the native object store (NOS).
 2. The server generates a random 128-bit RC4 session key which is returned encrypted to the client, using the shared key. If no predefined shared key is provided, then a predefined, hard coded key is used (this key is hard coded on the client and on the server).
 3. Passwords are always encrypted when passed over the wire, using an RC4, 128-bit session key, regardless of whether the entire session is encrypted or not.
 4. If a predefined shared key was provided, then the entire session is encrypted. Otherwise, only the password exchange is encrypted (using the hard coded key).
- Credentials: Passwords and OracleAS Adapters exchanged over the network are encrypted using a pre-defined, hard coded, 128-bit RC4 session key.
- Design Time: Security within Oracle Studio to grant access to Oracle Studio itself and to grant access to computers, user profiles and workspaces.
- Runtime: Security used to access Tuxedo, including controlling the daemon for the access.

Setting Design Time Security

Set the design time security as described in the following sections:

- Securing access to Oracle Studio is described in [Setting Password Access to Oracle Studio](#).
- Securing rights to configure a computer in Oracle Studio is described in [Specifying Users with Administrative Rights](#).

- Securing access to user profiles is accomplished by right-clicking the relevant user profile in Oracle Studio and selecting Change Master Password. In the dialog box that is displayed, specify a password that must be provided in the future to edit the specific user profile.
- Securing access to workspaces is accomplished by right-clicking the relevant workspace in Oracle Studio and selecting Set Authorization. In the dialog box that is displayed, specify a valid user and password that must be provided in the future to edit the specific workspace.

Setting Run-Time Security

During Attunity SQLServer-CDC for SSIS.chm, security considerations are implemented as follows:

- When the client request accesses the legacy platform through the daemon, either anonymous access is allowed or a valid user name and password must be provided for the computer in the user profile. The OracleAS Adapter and password properties in the J2CA 1.5 Tuxedo adapter are used at this stage to access the daemon.

Note: The user name used to access the daemon must also be the name of a user profile used.

- Access by the client must be through a valid port, according to the list of ports specified in the Workspace Access section of the Workspace **Security** tab in Oracle Studio. For more information, see Workspace [Security](#).

Note: Access to the legacy platform through a firewall using the NAT protocol is specified when the computer is added to Oracle Studio.

- To be allocated a server process, the client must be granted anonymous access to the workspace or be listed in the Workspace Users section of the Workspace **Security** tab in Oracle Studio. For more information, see Workspace [Security](#).
- The ability to run commands on the daemon, such as starting or stopping a daemon or ending server processes is available only to administrators who have been registered in Oracle Connect as a daemon administrator. A client is registered as a valid daemon administrator in the **Daemon Security** tab in Oracle Studio, as described in [Daemon Security](#).

Note: You can also specify administrators who can run commands only at the level of the workspace. Specify these administrators in the Workspace **Security** tab, as described in [Security](#).

Encrypting Network Communications

The encryption protocol between the client and the server is defined on the client computer.

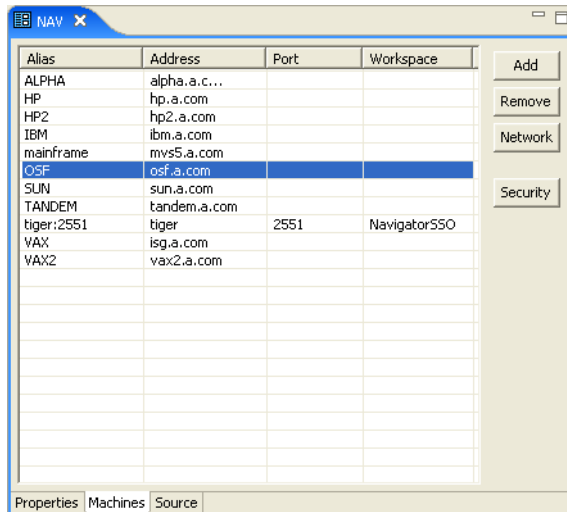
Perform the following steps to encrypt network communications:

1. Right-click the binding in Oracle Studio Design perspective Configuration view, and select **Open**.

2. Select the **Machines** tab.

The **Machines** tab of the Binding editor is displayed, as shown in the following figure:

Figure 8–3 The Binding Editor Machines tab



3. Select the client machine, and click **Network**.

The Network Settings screen is displayed.

4. Select the required encryption protocol from the Encryption Protocol list, and then click OK.

After specifying the encryption protocol, you must specify which servers the client is going to communicate with using the specified encryption protocol.

Perform the following steps to set the server computer for encrypted communications:

1. Right-click the user profile in the Configuration explorer and select **Edit User**.
2. In the User editor, select the client machine and click **Add**.

Figure 8–4 The Add Authenticator screen



3. Configure the authenticator parameters as follows:

- **Resource type:** Specify the resource type as Remote machine.
- **Resource name:** Specify the communication to the machine is encrypted in the following format:

`enckey: machine_name`

where *machine_name* is the machine to which you are connecting.

- **User name:** Specify the name associated with the encryption password and which the daemon on the remote machine looks up.

Multiple clients may specify this name in their user profile. In this case, the user profile on the remote machine must list only this one OracleAS Adapter/password entry for network encryption (rather than listing and looking up multiple OracleAS Adapter/password pairs).

If this user name entry is not specified, the daemon on the remote machine uses the name of the currently active user profile.

- **Password:** Specify the password that is required to pass or access encrypted information over the network.

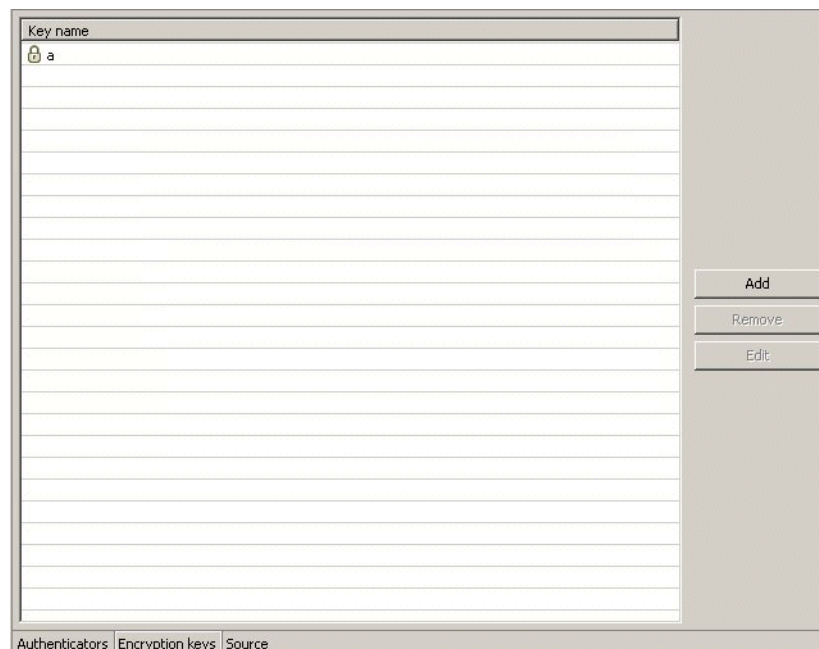
4. Click **OK**.

The server computer must also be configured to decipher the encrypted information using an encryption key.

Perform the following steps to set the encryption key on the server computer:

1. Select the server machine in the Oracle Studio Design perspective Configuration explorer.
2. Right-click the user profile and select **Edit User**. The User editor is displayed.
3. In the User editor, select the **Encryption Key** tab, shown in the following figure:

Figure 8–5 The Encryption Key Tab



4. Click **Add**.

The Add Encryption Key screen is displayed, as shown in the following figure:

Figure 8–6 The Add Encryption Key screen



5. Configure the encryption key parameters as follows:

- **Key name:** Enter the name associated with the encryption password and which the daemon on this machine looks up.
- **Key:** Enter the encryption key.
- **Confirm key:** Re-enter the encryption key.

6. Click **OK**.

7. In the explorer tree, expand the **Daemons** folder.

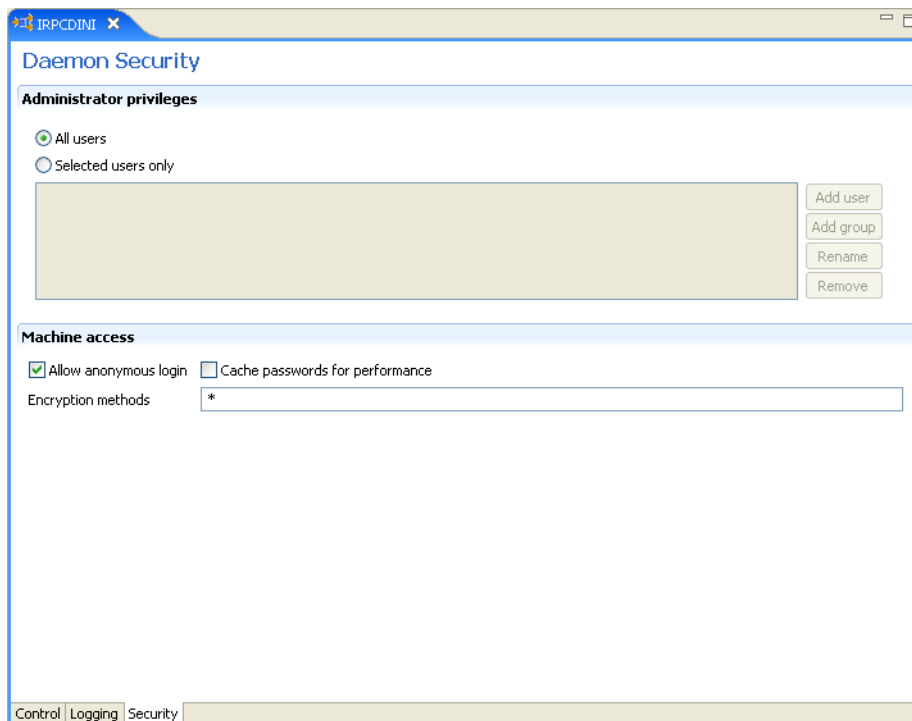
8. Right-click the daemon managing the connection and select **Open**.

9. Select the **Security** tab.

10. In the Machine access area, enter **RC4** in the **Encryption methods** field.

The Daemon Security tab is shown in the following figure:

Figure 8–7 Daemon Security Tab



Transaction Support

Note: The information in this section refers to the Tuxedo and Tuxedo Queue adapters only. the Tuxedo Gateway adapter can support two-phase commit transactions. for more information, see [Configuring the Transactional Gateway Process](#).

OracleAS Adapter for Tuxedo can participate in a distributed transaction, as the only one-phase commit resource. It supports local transactions by responding to ACX transactional verbs and issuing the corresponding ATMI transactional calls. The back-end service is responsible for the transaction semantics and implementation.

Transactional APIs included in the application, such as starting or committing a transaction, are mapped by OracleAS Adapter for Tuxedo to the corresponding Tuxedo transactional call. The adapter is also equipped with means to limit the Tuxedo transaction duration (timeout).

The following table maps the functionality of ACX transaction verbs to equivalent ATMI transaction calls.

Table 8–6 ACX-ATMI Transaction Correspondence.

ACX Transactional Verbs	ATMI Transactional Call	Comment
transactionStart	tpbegin(lTimeout, 0)	Transaction duration can be specified within the adapter definitions.
transactionCommit	tpcommit(0)	
transactionRollback	tpabort(0)	

See Also: [Advanced Tuning of the Metadata](#) for further details about transaction support configuration parameters.

Advanced Tuning of the Metadata

Oracle Studio enables you to define adapter interactions. In addition, Oracle Studio defines input and output structures used for these interactions. The interactions and input and output structures are maintained as metadata in the **Metadata** view of Oracle Studio.

Editing Adapter Interactions

Using Oracle Studio, do the following to edit the metadata in interactions for Oracle AS Adapters for Tuxedo:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. Expand the **Machines** folder, then expand the machine you are working with.
3. Expand the **Bindings** folder.
4. Expand the **NAV** binding.
5. Expand the **Adapters** folder to display the adapters list.
6. Right-click the Tuxedo adapter, and select **Show Metadata View** to open the Metadata view.

You can create and edit the adapter metadata as follows:

- **Adapter Metadata General Properties:** Enter and edit information about the adapter, such as the adapter name and the way in which you connect to the adapter. You make these changes in the Design perspective, Metadata view.
- **Adapter Metadata Interactions:** Enter details of an interaction.
- **Adapter Metadata Schema Records:** The input and output record structure for a record in the adapter definition.

Note: If you are using the Tuxedo Gateway adapter, you must restart Tuxedo after making any changes to the metadata.

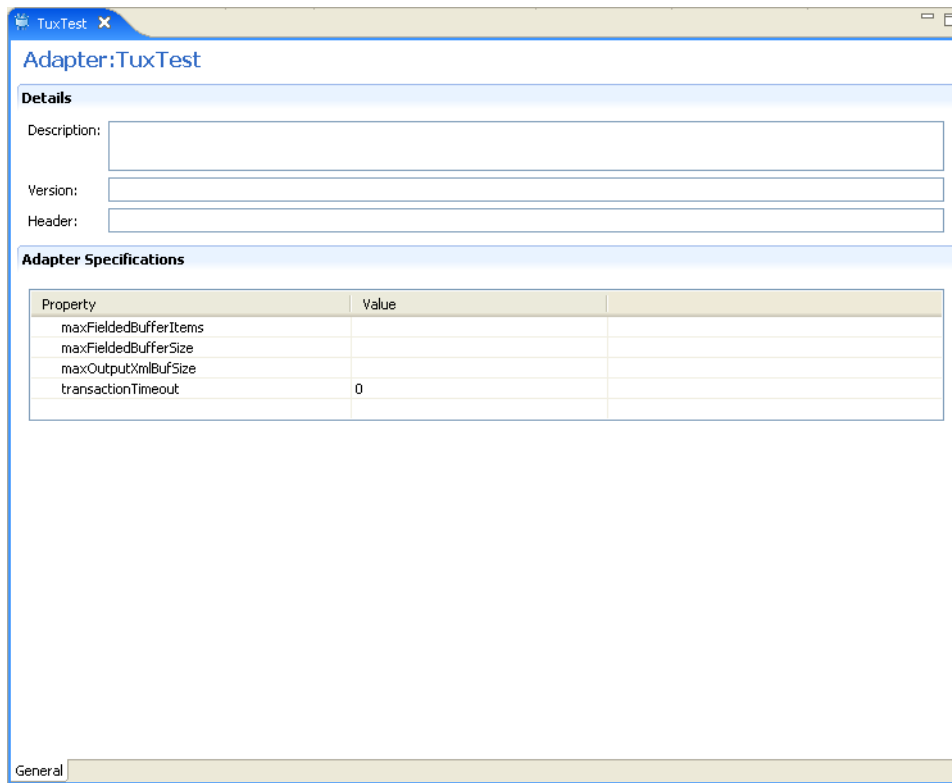
Adapter Metadata General Properties

You can enter and edit information about the adapter, such as the adapter name and the way in which you connect to the adapter. You make these changes in the Design perspective, Metadata view. The following describes how to open the Adapter General Properties editor.

1. In Oracle Studio Design perspective, Metadata view, expand the **Adapters** folder.
2. Right-click the Tuxedo adapter to edit, and select **Open**.

The General properties editor is displayed.
 The Adapter General Properties editor is shown in the following figure:

Figure A-1 Adapter Metadata General Properties



Note: If you are using the Tuxedo Gateway adapter, only the following Adapter specification properties are displayed:

- maxFieldedBufferItems
 - maxFieldedBufferSize
-
-

The following table lists the adapter general properties:

Table A-1 Adapter General Properties

Field	Description
Description	Specifies an identifying description of the adapter.
Version	The schema version.
Header	A C header file to map between the data structure and the adapter.
Authentication mechanism	Specifies the authentication to access the adapter. The available mechanisms are: <ul style="list-style-type: none"> ■ kerbv5 ■ none ■ basic password

Table A–1 (Cont.) Adapter General Properties

Field	Description
Max request size	Specifies the maximum size in bytes for an XML request or reply. Larger messages are rejected with an error.
Max active connections	Specifies the maximum number of simultaneous connections for an adapter (per process).
Max idle timeout	Specifies the maximum time, in seconds, that an active connection can stay idle. After that time, the connection is closed.
Adapter Specifications	<p>The Adapter specifications sections displays adapter-specific properties for an interaction. The properties are:</p> <ul style="list-style-type: none"> ▪ <code>maxFieldedBufferItems</code>: Sets a limit to the number of items that a fielded buffer may hold. The default value is 800. ▪ <code>maxFieldedBufferSize</code>: Limits the total size of a message buffer of type FML. The default is set to 2 Kilobytes. ▪ <code>maxOutputXmlBufSize</code>: Controls the size of memory allocated for output message buffers of type XML. The default is set to 8 Kilobytes. Note: This property is not displayed if you are using the Tuxedo Gateway adapter. ▪ <code>transactionTimeout</code>: Controls the duration of a Tuxedo transaction (in seconds). If not set (or set to 0), the Tuxedo transaction is not interrupted and lasts until explicitly terminated. This is the default. Note: This property is not displayed if you are using the Tuxedo Gateway adapter.

Adapter Metadata Interactions

The Adapter Metadata Interactions editor defines an interaction and its input and output definitions. The following describes how to open the Adapter Metadata editor.

1. In Oracle Studio Design perspective, Metadata view, expand the **Adapters** folder.
2. Expand the Tuxedo adapter with the interaction to edit.
3. Expand the **Adapter** folder.
4. Right-click the adapter you want to edit and select **Open**.

The Adapter Metadata Interactions editor is displayed.

The Adapter Metadata Interactions editor that is displayed depends on the type of Tuxedo adapter you are using. The following sections describe the two types of Interaction editors.

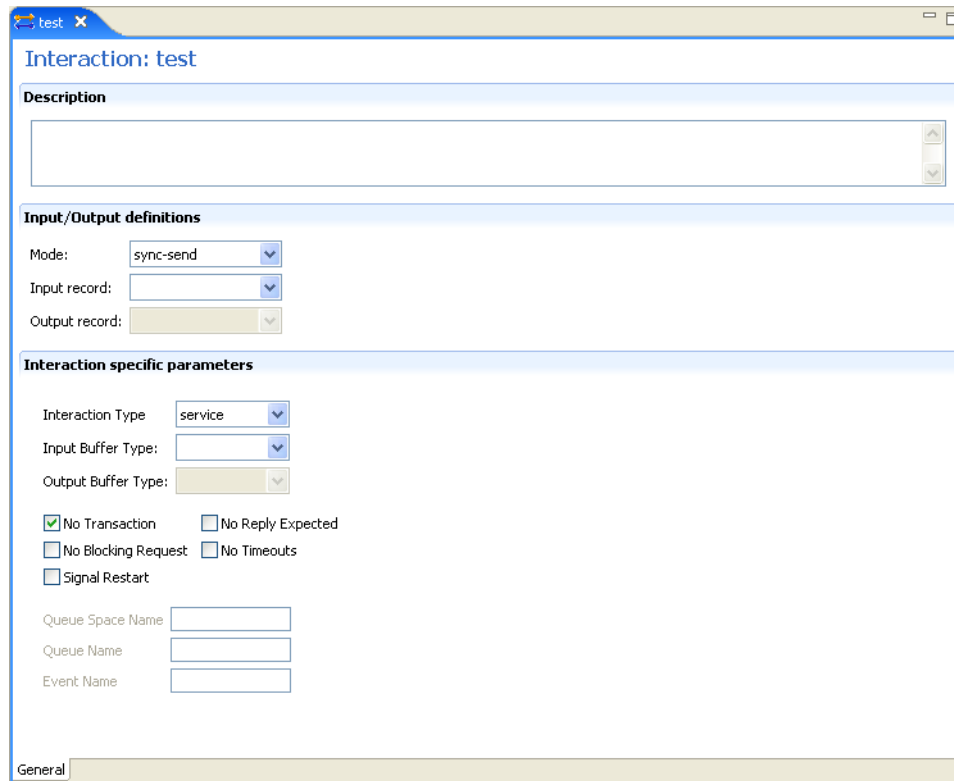
- [Adapter Metadata Interactions for the Tuxedo and Tuxedo Queue Adapters](#)
- [Adapter Metadata Interactions for the Tuxedo Gateway Adapter](#)

Adapter Metadata Interactions for the Tuxedo and Tuxedo Queue Adapters

This section describes the adapter metadata Interactions for the Tuxedo and Tuxedo Queue adapters.

The adapter metadata interactions editor for Tuxedo and Tuxedo Queue adapters is shown in the following figure:

Figure A–2 Adapter Metadata Interactions for Tuxedo and Tuxedo Queue



The following table lists the Interaction properties for the Tuxedo and Tuxedo Queue adapters:

Table A–2 Adapter Metadata Interaction Properties for Tuxedo and Tuxedo Queue

Field	Description
Interaction name	Specifies the name of the interaction.
Description	Provides a descriptive identifier for the interaction.
Mode	Determines the interaction mode. The following interaction modes are available: sync-send-receive: The interaction sends a request and expects to receive a response. sync-send: The interaction sends a request and does not expect to receive a response. sync-receive: The interaction expects to receive a response. async-send: Not applicable.
Input record	Identifies an input record.
Output record	Identifies an output record for the results of an interaction.

Table A–2 (Cont.) Adapter Metadata Interaction Properties for Tuxedo and Tuxedo

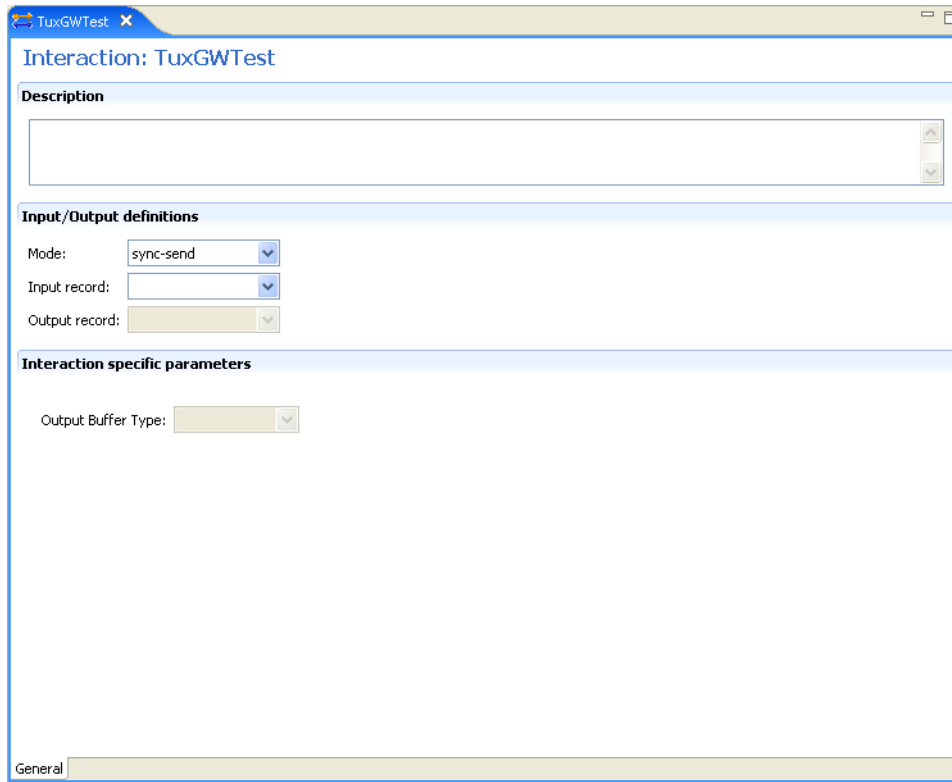
Field	Description
Interaction Specific Parameters	<p>Defines the properties and values of parameters specific to an interaction. The following properties are available:</p> <p>Interaction Type: Select one of the following:</p> <ul style="list-style-type: none"> ▪ service: Enables service interaction (default). ▪ enqueue: Stores a message on the queue that is specified by the information provided in the Queue Name box and the Queue Space Name box. ▪ post: Posts an event, whose name is specified by the Event Name box, and any related data. <p>Input Buffer Type: The type of buffer used for the input.</p> <p>Output Buffer Type: The type of buffer used for the results of an interaction.</p> <p>No Transaction: Enables a service to be executed, regardless of the transaction context.</p> <p>No Reply Expected: For future use.</p> <p>No Blocking Request: Avoids a FROM request submission if a blocking condition exists.</p> <p>No Timeouts: Ignores blocking timeouts.</p> <p>Signal Restart: If selected, whenever a signal interrupts an underlying system call, this call is reissued.</p>

Adapter Metadata Interactions for the Tuxedo Gateway Adapter

This section describes the adapter metadata interactions for the Tuxedo Gateway adapter.

The adapter metadata properties editor for the Tuxedo Gateway adapter is shown in the following figure:

Figure A–3 Adapter Metadata Interactions for Tuxedo Gateway



The following table lists the adapter metadata interaction properties for the Tuxedo Gateway adapter:

Table A–3 Adapter Metadata Interaction Properties for Tuxedo Gateway

Field	Description
Interaction name	Specifies the name of the interaction.
Description	Provides a descriptive identifier for the interaction.
Mode	Determines the interaction mode. The following interaction modes are available: sync-send-receive: The interaction sends a request and expects to receive a response. sync-send: The interaction sends a request and does not expect to receive a response. sync-receive: The interaction expects to receive a response. async-send: Not applicable.
Input record	Identifies an input record.
Output record	Identifies an output record for the results of an interaction.
Interaction Specific Parameters	Defines the properties and values of parameters specific to an interaction. The following properties are available: Output Buffer Type: The type of buffer used for the results of an interaction.

Adapter Metadata Schema Records

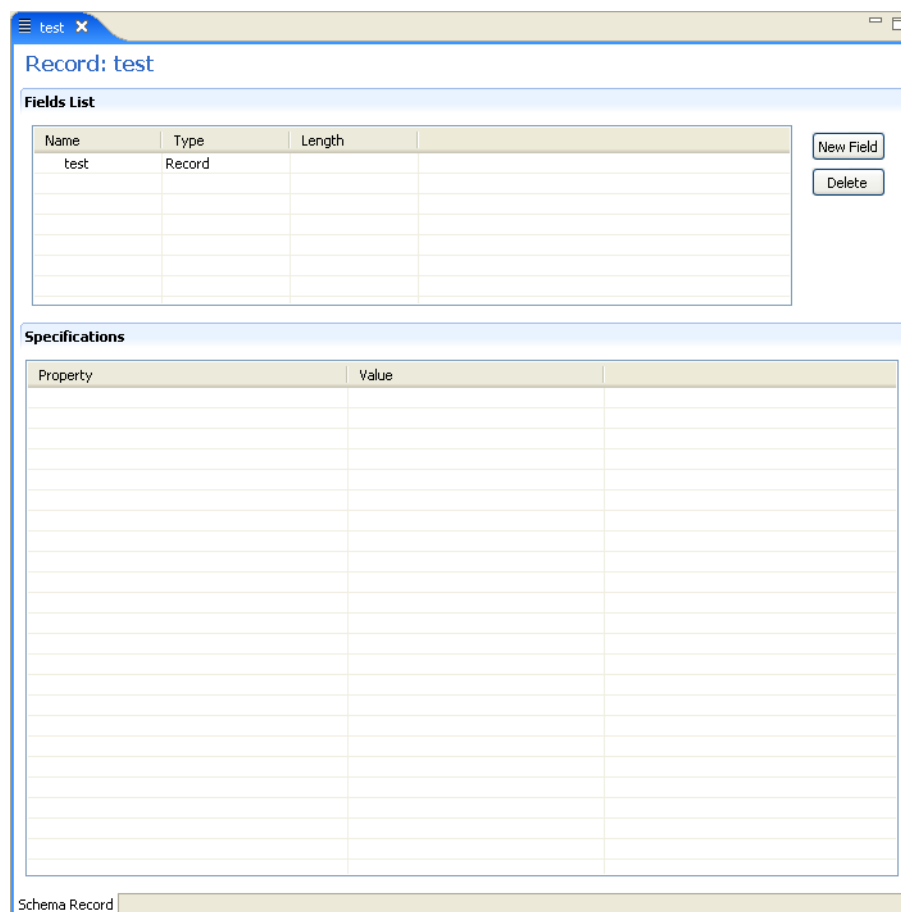
The Adapter Metadata Schema Records editor defines the general details of the input and output record structures for the interaction. The following describes how to open the Adapter Metadata Schema Records editor:

1. In Oracle Studio Design perspective, Metadata view, expand the **Adapters** folder.
2. Expand the Tuxedo adapter with the schema records to edit.
3. Expand the **Schemas**.
4. Right-click the schema you want to edit and select **Open**.

The Adapter Metadata Schema Records editor is displayed.

The Adapter Metadata Schema Records editor is shown in the following figure:

Figure A–4 Adapter Metadata Schema Records



The following table shows the adapter metadata schema record properties.

Use the Fields List area to define single data items within a record.

Table A–4 Adapter Metadata Schema Record Properties

Field	Description
Name	Specifies the name of the field.

Table A-4 (Cont.) Adapter Metadata Schema Record Properties

Field	Description
Type	The data type of the field. The following are valid data types: <ul style="list-style-type: none"> ▪ Binary ▪ Boolean ▪ Date ▪ Double ▪ Float ▪ Int ▪ Long ▪ Numeric ▪ Short ▪ String ▪ Time ▪ Timestamp ▪ XML
Length	The size of the field including a null terminator, when the data type supports null termination.

Note: Use the Specifications box to specify field properties.

Adapter Metadata XML

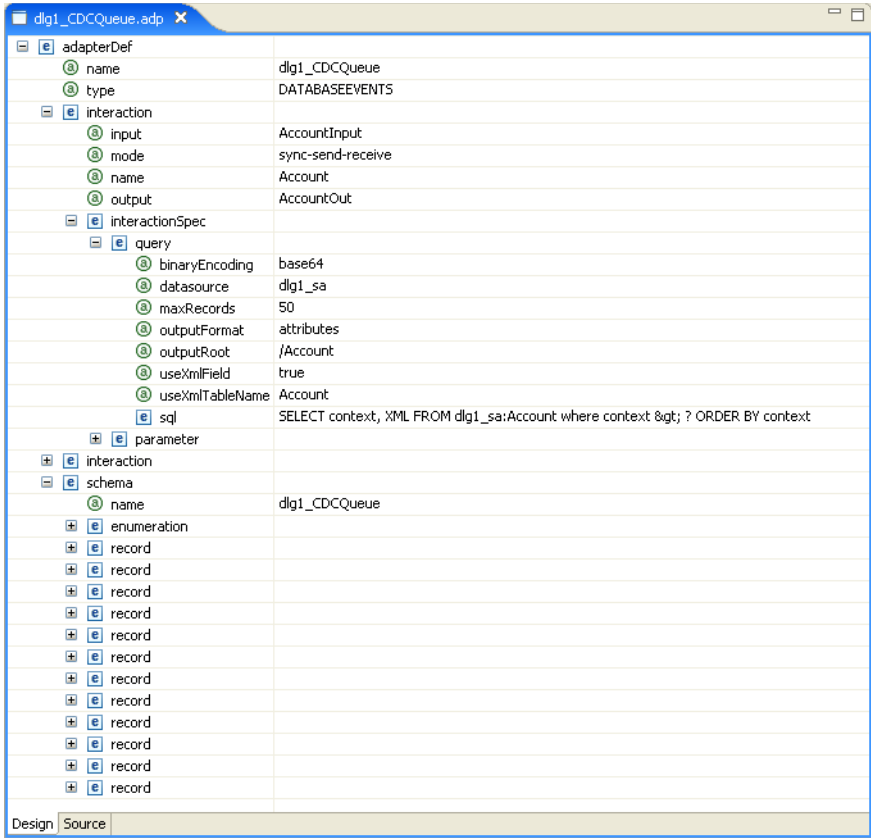
You can also edit the adapter metadata by viewing its XML schema. The following describes how to open the adapter metadata XML editor:

1. In Oracle Studio Design perspective, Metadata view, expand the **Adapters** folder.
2. Right-click the adapter to edit, and select **Open as XML**.

The adapter XML editor is displayed in the Design view.

The adapter XML editor is shown in the following figure:

Figure A-5 Adapter Metadata XML Editor



For information on how to edit the properties in the XML editor Design view, see [Editing XML Files in Oracle Studio](#).

OracleAS Adapters for Tuxedo Message Buffer Support and Data Type Support

This appendix contains the following sections:

- [OracleAS Adapters for Tuxedo Message Buffer Support](#)
- [Data Type Support](#)

OracleAS Adapters for Tuxedo Message Buffer Support

Oracle Connect provides support for all standard types of OracleAS Adapters for Tuxedo message buffers as I/O, as follows:

- **STRING:** A null terminated character array. The data type is character and its length is determined by counting the characters in the buffer until reaching the null character. It is commonly used by C programs.
- **CARRAY:** An array of un-interpreted arbitrary binary data. The application must specify the buffer length for CARRAY message buffers when used as input to ATMI functions.
- **XML:** An XML formatted data. This buffer type enables Tuxedo applications to use XML for exchanging data within and between applications. Tuxedo applications can send and receive simple XML buffers, and route them to the appropriate servers. Data dependent routing is supported for this buffer type.
- **VIEW (16-bit):** A C structure layout. This buffer is used for fixed collections of data elements, structures or records. VIEW records support integral data types such as long integer, character, and decimal. VIEW records do not support structures within structures, nor do they support arrays of structures or pointers.
- **VIEW (32-bit):** A C structure layout where 32-bit FML identifiers are used.
- **FML (16-bit):** An abstract data type, used to create, access, modify and delete fields. It is a data structure that stores tagged values. Values are typed, can be specified multiple times, and vary in length. Additionally, FML buffers support storage of multiple values for a field. The variable length format of fielded buffers enables multiple field occurrences to be stored and retrieved.
- **FML (32-bit):** An FML type where 32-bit FML identifiers are used.

Note: Synonyms such as `X_C_TYPE`, and `X_OCTET` are also recognized.

Unstructured message buffers are wrapped within a record as follows:

- A message buffer of type `STRING` is wrapped within a record containing a single field of type `string` with a fixed size.
- A message buffer of type `CARRAY` is wrapped within a record containing a single field of type `binary` with a fixed size.
- A message buffer of type `XML` is wrapped within a record containing a single field of type `XML`.

User-Defined Message Buffers

Oracle Connect does not support user-defined message buffers.

Data Type Support

This section contains the following topics:

- [Data Type Mapping](#)
- [Data Type Handling](#)
- [Header Record Structure](#)

Data Type Mapping

OracleAS Adapters for Tuxedo support several data types that are used to define metadata in Oracle Studio. The data types are mapped from the Tuxedo data types during the import procedure.

Note: The mapping of data types between OracleAS Adapters for Tuxedo and Oracle Application Server is performed internally by Oracle Connect.

Table B-1 Data Type Mapping: Tuxedo and Oracle Connect

Tuxedo Data Type	Oracle Connect Data Type
carray	binary
char	string
double	double
float	double
int	int
long	int
short	int
string	string

Data Type Handling

OracleAS Adapters for Tuxedo handle the fields contained in the `VIEW` file as follows:

- Specifying a value greater than one in the `COUNT` field of the Tuxedo field definition within a `VIEW` file, translates to `array=xx` attribute.
- Specifying a value in the `SIZE` field of Tuxedo field definition within a `VIEW` file, translates to `size=yy` attribute.

- Specifying C in the FLAG field of Tuxedo field definition within a VIEW file, imposes generation of an extra leading COUNTER field to hold the actual count value.
- Specifying L in the FLAG field of Tuxedo field definition within a VIEW file, imposes generation of an extra leading LENGTH field to hold the actual length value.

Note:

- The L flag is applicable for STRING and CARRAY data types only.
 - In the case where a COUNT value greater than one is specified for the field, the extra leading LENGTH field is generated as an array. The array entries should hold the actual length values of the corresponding field array entries.
-
-

- Specifying a value in the NULL field of Tuxedo's field definition within a VIEW file is not reflected in Oracle Connect back-end adapter record.

Tuxedo's VIEW fields restrictions are preserved.

The following is an example of a VIEW file definition:

```
VIEW emp
#TYPE      CNAME          FBNAME          COUNT  FLAG  SIZE  NULL
long       lSalary        SALARY          1      -    -    0
short      nDeptnum         DEPTNUM         1      -    -    0
short      nEmpnum          EMPLOYEE_NUMBER 1      -    -    0
short      nJobcode         JOBCODE         1      -    -    0
string     szMessageText    MESSAGE_TEXT    1      -    80   ""
string     szFirstName   FIRST_NAME      1      -    16   ""
string     szLastName    LAST_NAME       1      -    21   ""
END
```

The following is an example of a field table that defines the FML field IDs:

```
*base 100
# name          number  type  flags  comments
MESSAGE_TEXT    1      string -    -
#
DEPTNUM         100    short -    -
EMPLOYEE_NUMBER 101    short -    -
JOBCODE         102    short -    -
FIRST_NAME      103    string -    -
LAST_NAME       104    string -    -
SALARY          105    long  -    -
```

Header Record Structure

Each adapter has a HEADER record which is mapped to Tuxedo ATMI Queue Control Structure. The HEADER record is included in all records defined.

HEADER input fields (for writing and subscribing) are as listed in the following table:

Table B-2 HEADER Record Input Fields

Field	Description
long flags;	Indicates which of the values are set.
long deq_time;	Indicates absolute/relative time for dequeuing.
long priority;	Specifies the enqueueing priority.
long exp_time;	Specifies the expiration time.
long delivery_qos;	Specifies the delivery quality of service.
long reply_qos;	Specifies the reply quality of service.
long urcodes;	User-return code.
char corrid[32];	Specifies the correlation identifier used to identify the message.
char replyqueue[16];	Species the queue name for the reply message.
char failurequeue[16];	Specifies the queue name for failure messages.

HEADER output fields (for reading) are as listed in the following table:

Table B-3 HEADER Record Output Fields

Field	Description
long flags;	Indicates which of the values should be set.
long priority;	Specifies the enqueueing priority.
char msgid[32];	Specifies the ID of the message dequeued.
char corrid[32];	Specifies the correlation identifier used to identify the message.
long delivery_qos;	Specifies the delivery quality of service.
long reply_qos;	Specifies the reply quality of service.
char replyqueue;	Specifies the queue name for the reply.
char failurequeue[16];	Specifies the queue name for failure messages.
long diagnostic;	Specifies the reason for failure.
long appkey;	Specifies the application authentication client key.
long urcodes;	User-return code.
CLIENTID cltid;	Specifies the client identifier for the originating client.
char replyqueue[16];	Species the queue name for the reply message.
char failurequeue[16];	Specifies the queue name for failure messages.

Advanced Tuning of the Daemon

The daemon configuration is managed using Oracle Studio. Daemon configuration is divided into the following groups:

- [Control](#)
- [Logging](#)
- [Security](#)
- [Workspaces](#)

Control

Using the **Control** tab for the daemon, you define various daemon control options. The **Daemon Control** tab is accessed as follows:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design perspective Configuration view expand the **Machines** folder.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Right-click the required daemon in the Runtime Explorer and select **Edit Daemon Configuration**. The **Control** tab for the daemon is displayed in the editor.
5. After making changes to the daemon, right-click the daemon and select **Reload Configuration**.

Note: You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Open**.

Changes made to the daemon configuration are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager perspective.

The **Control** tab is shown in the following figure:

Figure C-1 The Control tab

The following table shows the parameters that can be set in the **Control** tab:

Table C-1 Daemon Control Tab Components

Field	Description
Daemon IP Address	Enter the IP address of the computer(s) where the daemon is listening. If no IP address is entered, the daemon listens on all available IP addresses.
Daemon port	Enter the port where the daemon is listening. If no port is entered, the daemon listens on all available ports.
Automatically recover from failure	The daemon restarts automatically if it fails for any reason (any error that causes the daemon process to terminate, such as network process lost or the CPU running the daemon crashes and the backup daemon is defined on another CPU). All available and unconnected servers are terminated and any connected servers are marked and terminated on release. Also the backup starts a backup for itself. The backup appends a new log file to the log of the original daemon, adding a line indicating that a backup daemon was started.
Maximum XML request size	The maximum number of bytes that the daemon handles for an XML document.
Default language	The language that the daemon supports. This setting is used when working with a client with a code page different from the server code page.
Maximum XML in memory	The maximum amount of space reserved for the XML in memory.

Table C-1 (Cont.) Daemon Control Tab Components

Field	Description
Call timeout	<p>The timeout period for <i>short</i> calls for all daemons. The definition of a short call is a call that should be completed in a few seconds. For example, most calls to a database such as DESCRIBE should be completed in a few seconds as opposed to call like a GETROWS call, which can take a long time. In heavily loaded or otherwise slow systems, even short calls such as calls to open a file, may take a significant amount of time. If a short call takes more than the specified time to complete, then the connection is stopped. The default value for this parameter is 60 seconds. Values of less than 60 seconds are considered to be 60 seconds.</p> <p>Specifying the timeout in a workspace overrides the value set in this field for that workspace.</p>
Connect timeout	<p>The time the client waits for a daemon server to start. If the daemon server does not start within this period, then the client is notified that the server did not respond. The value specified for this parameter serves as the default timeout for all the workspaces listed in the daemon configuration. The default value for this parameter is 60 seconds.</p> <p>Notes:</p> <ul style="list-style-type: none"> ▪ Entering the timeout in a workspace overrides the value set in this field for that workspace. ▪ Even if the XML source does not list this parameter in the workspace section, the workspace gets it using the default value. To prevent a workspace from using the default value, you must enter a value of zero for this parameter in the workspace section.
Client idle timeout	<p>The maximum amount of time any daemon client may be idle before the connection with the server is closed.</p> <p>Specifying the timeout in a Workspace overrides this setting for that workspace.</p>

Logging

Using the **Logging** tab, you define the daemon log file settings, the log file structure and the location where the log is saved. In addition, use it to define the data that is logged and traced in the file.

The following describes how to open the **Logging** tab.

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design perspective Configuration view expand the **Machines** folder.
3. Right-click a computer and select **Open Runtime Perspective**.
4. Right-click the daemon in the Runtime Explorer and select **Edit Daemon Configuration**.
5. Click the **Logging** tab.
6. After making changes to the daemon, right-click the daemon and select **Reload Configuration**.

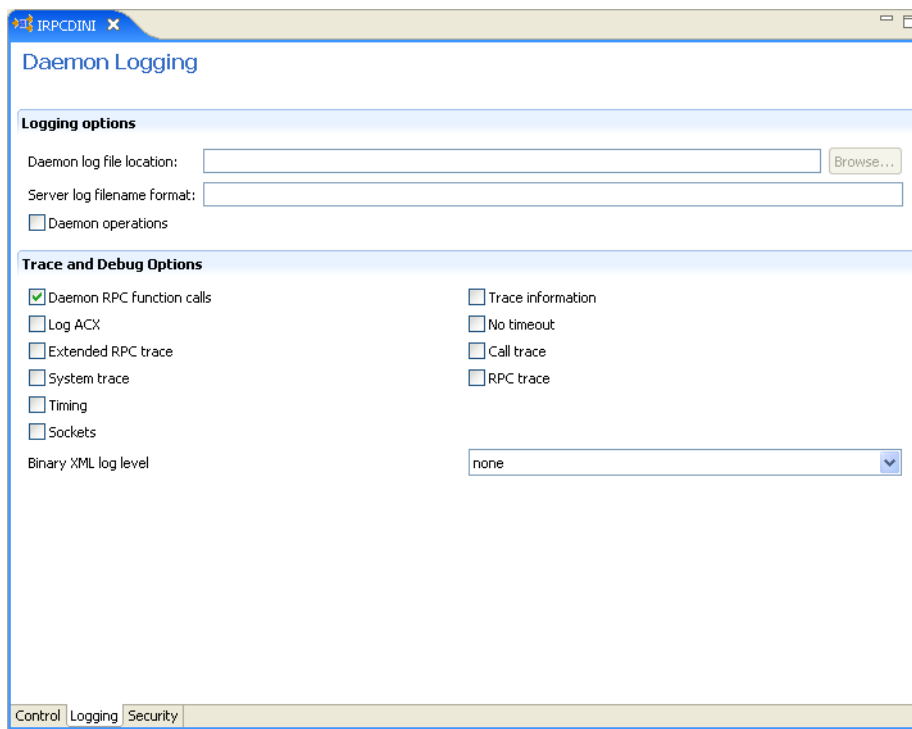
Note: You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Open**.

Changes made to the daemon configuration are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager perspective.

7. Right-click the daemon and select **Recycle Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **Logging** tab for the daemon is shown in the following figure:

Figure C-2 The Logging tab



The **Daemon Logging** tab comprises fields, as listed in the following table:

Table C-2 Daemon Logging Tab Components

Field	Description
Logging options	
Daemon log file location	Specifies the daemon produces its log data. The full path must be specified.
Server log filename format	Defines the name and location of the server log file. The field must specify the full path name. If no directory information is provided for the log file, then it is located in the login directory of the account running Oracle Connect workstation.
Daemon operations	Select this to log all of the daemon operations.

Table C-2 (Cont.) Daemon Logging Tab Components

Field	Description
Tracing and debug options	
Daemon RPC function calls	Select this to log all daemon RPC function calls.
Log ACX	Select this to log requests and processes.
Extended RPC trace	Generates a verbose message in the server log file for each low-level RPC function called. This is useful for troubleshooting the server.
System trace	Generates system-specific tracing of various operations.
Timing	Generates a time stamp for every entry to the server log file.
Sockets	Generates a message in the server log file for each socket operation.
Trace information	Select this to log low-level RPC operations.
No timeout	Disables the standard RPC timeouts, setting them to a long duration (approximately an hour) to facilitate debugging.
Call trace	Generates a message in the server log file for each RPC function called. This is useful for troubleshooting the server.
RPC trace	Enables debugging messages on the server.
Binary XML log level	Sets the binary XML log level. Your options are: <ul style="list-style-type: none"> ▪ debug ▪ none (the default) ▪ api ▪ info

The following tokens can appear in the log file template and is replaced accordingly:

- %A: workspace name
- %D: date (yymmdd)
- %I: instance number of the given workspace server
- %L: server account login directory
- %P: server process ID
- %T: time (hhmmss)
- %U: server account name (username)

For example, %L/server_%A%I.log may produce a log file such as:
/usr/smith/server_sales15.log.

The default log file template is %L/server_%A%I.log.

Security

The following **Security** tab for the daemon is used to:

- Grant administration rights for the daemon.
- Determine access to the computer.

The following shows how to open the Daemon **Security** tab:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design perspective Configuration view expand the **Machines** folder.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Right-click the daemon in the Runtime Explorer and select **Edit Daemon Configuration**.
5. Click the **Security** tab.
6. After making changes to the daemon, right-click the daemon and select **Reload Configuration**.

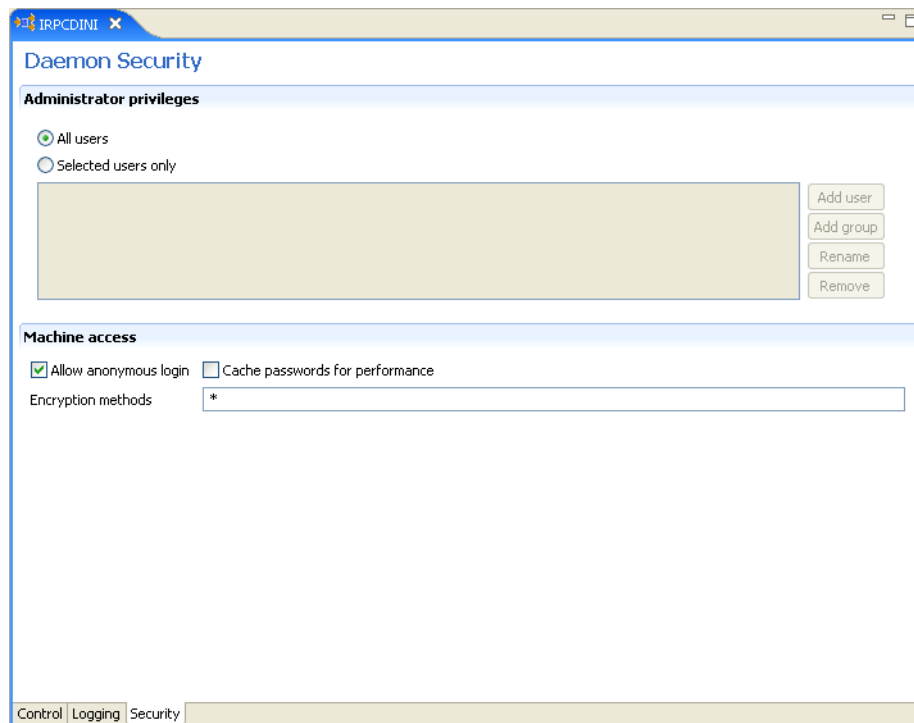
Note: You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Open**.

Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.

7. Right-click the daemon and select **Recycle servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The Daemon **Security** tab is shown in the following figure:

Figure C-3 The Daemon Security tab



The **Daemon Security** tab comprises fields, as listed in the following table:

Table C-3 Daemon Security Tab Components

Field	Description
Administrators privileges	Identifies the users (accounts) allowed to perform administrative tasks (tasks that require administrative login).
All users	Enables all users to access the daemon and change the settings.
Selected users only	Identifies the names of users (accounts) and groups that can be administrators. ¹ If a user is not specified, the account from which the daemon was started is considered the administrator. Note that the daemon does not require the user to log in to the account on the system, but to log in to the daemon using the account name and password.
Machine access	Manages access to the computer.
Allow anonymous login	Whether workspaces allow anonymous logins (without user name/password entries). For the optimal level of security, keep this option unchecked and define a user name for the Daemon Administrators parameter. If unchecked, then no workspace can have an anonymous client. If checked, then a particular workspace allows anonymous clients.
Cached password	Enables login passwords to be cached. This enhances performance by reducing login times for future connections from the same client in a session.
Encryption methods	Specifies the encryption method used to send information across the network. The default is an asterisk (*), meaning that all methods are acceptable. If an encryption method is specified, it must be used. The RC4 and DES3 protocols are currently supported.

¹ The name is prefixed with '@', to use the operating system GROUP feature.

Workspaces

A daemon can include multiple workspaces. A workspace defines the server processes and environment that are used for the communication between the client and the server for the duration of the client request. Each workspace has its own definition. The workspace definition is divided into the following groups:

- [General](#)
- [Server Mode](#)
- [Security](#)

General

Using the **General** tab, you enter general information about the features that control the operation of the workspace, such as the server type, the command procedure used to start the workspace and the binding configuration associated with this workspace.

Do the following to open the **General** tab:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design perspective Configuration view, expand the **Machines** folder.

3. Right-click a computer and select **Open Runtime Perspective**.
4. Expand the Daemons node to display the workspaces in the Runtime Explorer.
5. Right-click the workspace and select **Edit Workspace Configuration**. The **WS Info**. tab opens.
6. After making changes to the workspace, right-click the daemon and select **Reload Configuration**.

Note: You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Open**.

Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.

7. Right-click the daemon and select **Recycle Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **General** tab is shown in the following figure:

Figure C-4 The General Tab

The **General** tab comprises fields, as listed in the following table:

Table C-4 General Tab Components

Field	Description
Workspace name	The name used to identify the workspace. Note: The default configuration includes the default Navigator workspace. This workspace is automatically used if a workspace is not specified as part of the connection settings.
Description	A description of the workspace.
Startup script	The full path name of the script that starts the workspace server processes. The script specified here must always activate the <code>nav_login</code> procedure and then run the server program (<code>svc</code>). If you do not specify the directory, the startup procedure is taken from the directory where the daemon resides. Oracle Connect includes a default startup script, which it is recommended to use.
Server type	This field is not applicable for use with OracleAS Adapter for Tuxedo. <i>Do not change the default value.</i>
Workspace binding name	This field is not applicable for use with OracleAS Adapter for Tuxedo.
Timeout parameters	The time the client waits for the workspace server to start. If the workspace server does not start within this period, then the client is notified that the server did not respond. Specifying the timeout here overrides the default setting, specified in the Control section. For more information, see Control for details about the Daemon Control section.
Client idle timeout	The maximum amount of time a workspace client can be idle before the connection with the server is closed.
Connect timeout	The time the client waits for a workspace server to start. If the workspace server does not start within this period, then the client is notified that the server did not respond.
Call timeout	The timeout period for <i>short</i> calls for all daemons. The definition of a short call is a call that should be completed in a few seconds. For example, most calls to a database such as <code>DESCRIBE</code> should be completed in a few seconds as opposed to call like a <code>GETROWS</code> call, which can take a long time. In heavily loaded or otherwise slow systems, even short calls such as calls to open a file, may take a significant amount of time. If a short call takes more than the specified time to complete, then the connection is stopped. The default value for this parameter is 60 seconds. Values of less than 60 seconds are considered to be 60 seconds. Specifying the timeout in a workspace overrides the value set in this field for that workspace.

Logging and Trace Options

Table C-4 (Cont.) General Tab Components

Field	Description
Specific log file format	<p>Defines the name and location of the server log file to write the data to a file instead of SYSOUT for the server process. The parameter must specify the name and the high level qualifier.</p> <p>The following tokens can appear in the log file template and are replaced accordingly:</p> <ul style="list-style-type: none"> ■ %A: workspace name ■ %D: date (yymmdd) ■ %I: instance number of the given workspace server ■ %L: server account's login directory ■ %P: server's process ID ■ %T: time (hhmmss) ■ %U: server's account name (username) <p>Important: This is not valid if you are using the Tuxedo Gateway adapter. All log files for the Tuxedo Gateway adapter are written to the Tuxedo log.</p>
Trace options	<p>Specifies the type of tracing to be performed. The following tracing options are available:</p> <ul style="list-style-type: none"> ■ No timeout: Select this to disable the standard RPC timeouts, setting them to a long duration (approximately an hour) to facilitate debugging. ■ Call trace: Select this to generate a message in the server log file for each RPC function called. This is useful for troubleshooting the server. ■ RPC trace: Select this to enable debugging messages on the server. ■ Sockets: Select this to generate a message in the server log file for each socket operation. This is useful for troubleshooting client/server communication - providing a detailed trace of every client/server communication. ■ Extended RPC trace: Select this to generate a verbose message in the server log file for each low-level RPC function called. This is useful for troubleshooting the server. ■ System trace: Select this to generate operating system-specific tracing. ■ Timing: Select this to generate a time stamp for every entry to the server log file.
Query governing restrictions	
Max number of rows in a table that can be read	<p>Select the maximum number of table rows that are read in a query. When the number of rows read from a table exceeds the number stated the query returns an error.</p>

Table C-4 (Cont.) General Tab Components

Field	Description
Max number of rows allowed in a table before scan is rejected	<p>Select the maximum number of table rows that can be scanned. This parameter has different behavior for query optimization and execution.</p> <ul style="list-style-type: none"> ■ For query optimization, the value set is compared to the table cardinality. If the cardinality is greater than the value, the scan strategy is ignored as a possible strategy (unless it is the only available strategy). ■ For query execution, a scan is limited to the value set. When the number of rows scanned exceeds the number entered, the query returns an error.

Server Mode

The Server Mode tab lets you configure the features that control the operation of the servers started up by the workspace and allocated to clients.

For example, you can configure the workspace to start multiple servers for future use, before any client request, instead of starting each server when a request is received from a client.

Do the following to open the **Server Mode** tab:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design Perspective Configuration view, expand the **Machines** folder.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Expand the Daemons node to display the workspaces in the Runtime Explorer.
5. Right-click the workspace and select **Edit Workspace Configuration**.
6. Click the **Server Mode** tab.
7. After making changes to the workspace, right-click the daemon and select **Reload Configuration**.

Notes:

- You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Open**.
 - Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.
-
-

8. Right-click the daemon and select **Recycle Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **Server Mode** tab is shown in the following figure:

Figure C-5 The Server Mode tab

Workspace Server Mode

Server

Workspace server mode: []

Port range: From 0 To 0

Use default port range

Maximum number of server processes: 0

Limit server reuse: 50

Limit concurrent clients per server: 0

Specify server priority: 0

Use default server priority

Keep when daemon ends

Server Provisioning

Number of prestarted servers in pool: 0

Number of spare servers: 0

Prestarted server pool limit: 0

Resource limitations

Number of sub-tasks (OS/390 specific): 0

General | **Server Mode** | Security

The **Server Mode** tab has the following fields:

Table C-5 Server Mode Tab Components

Field	Description
Workspace server modeTuxedo	<p>Specifies the type of new server processes that the daemon starts. The daemon supports the following server modes:</p> <ul style="list-style-type: none"> ▪ singleClient: Each client receives a dedicated server process. The account in which a server process runs is determined either by the client login information or by the specific server workspace. <p>This mode enables servers to run under a particular user account and isolates clients from each other, as each receives its own process. However, this server mode incurs a high overhead due to process startup times and can use a lot of server resources as it requires as many server processes as concurrent clients.</p> ▪ multiClient: Clients share a server process and are processed serially. This mode has low overhead because the server processes are initialized. However, because clients share the same process, they can impact one another, especially if they issue lengthy queries. The number of clients that share a process is determined by the Clients per server limit field. <p>Note: When you use the Tuxedo Gateway adapter, this is the default selection. <i>Do not change this value.</i></p> ▪ multiThreaded: This mode is not applicable for use with OracleAS Adapter for Tuxedo. ▪ reusable: An extension of single-client mode. Once the client processing finishes, the server process does not die and can be used by another client, reducing startup times and application startup overhead. <p>This mode does not have the high overhead of single-client mode because the servers are initialized. However, this server mode can use a lot of server resources as it requires as many server processes as concurrent clients.</p> <p>Note: The other modes can be set so that the server processes are reusable. The Reuse limit field value defines the number of times a process can be reused.</p>
Reuse limit	<p>Sets the maximum number of times a particular server can be reused. A one-client server can be reused after its (single) client has disconnected. Reuse of servers enhances startup performance because it avoids the need to repeat initialization. The default for this field is none (0), indicating that server reuse is unlimited. This parameter is disabled only if the server mode value is singleClient.</p>
Clients per server limit	<p>Sets the maximum number of clients a server process for the current workspace accepts. The default for this field is none (0), indicating that the number of clients for each server is unlimited. This field is enabled only if the server mode value is multiClient or multiThreaded.</p>

Table C-5 (Cont.) Server Mode Tab Components

Field	Description
Server availability	<p data-bbox="610 268 1154 323">Specifies the number of servers in a pool of servers, available to be assigned to a client.</p> <p data-bbox="610 333 992 363">The following options are available:</p> <ul style="list-style-type: none"> <li data-bbox="610 373 1227 583">■ Initial number of servers: The number of server processes that are prestarted for this workspace when the daemon starts. When the number of available server processes drops lower than the value specified in the Minimum number field, the daemon again starts server processes until this number of available server processes is reached. The default for this field is 0. <li data-bbox="610 594 1227 804">■ Minimum number: The minimum number of server processes in the prestarted pool before the daemon resumes creating new server processes (to the value specified in the Initial number of servers field). If this field is set to a value higher than the Initial number of servers field, the daemon uses the value specified in the Initial number of servers field. The default for this field is 0. <li data-bbox="610 814 1227 1056">■ Keep when daemon ends: When a daemon is shutdown, all the servers started by that daemon are also stopped, even if they are active. Set this field to <code>true</code> for the workspace servers to remain active, even after the daemon has been shut down. If this field is set to <code>true</code>, it is the responsibility of the system operator or manager to ensure that the servers are eventually stopped. This must be done at the system level. <li data-bbox="610 1066 1227 1339">■ Set maximum number of servers: The maximum number of available server processes. Once this number is reached, no new nonactive server processes are created for the particular workspace. For example, if several server processes are released at the same time, so that there are more available server processes than specified by this field, the additional server processes higher than this value are terminated. The default for this field is zero, meaning that there is no maximum.
Port range	<p data-bbox="610 1360 1227 1465">Determines the range of ports available for this workspace when starting server processes. Use this option when you want to control the port number, so that Oracle Connect can be accessed through a firewall.</p> <p data-bbox="610 1476 1068 1505">Enter the port range in the following fields:</p> <ul style="list-style-type: none"> <li data-bbox="610 1516 1198 1545">■ From: enter the lowest numbered port in the range. <li data-bbox="610 1556 1179 1585">■ To: Enter the highest numbered port in the range.
Use default port range	<p data-bbox="610 1602 1227 1686">Select this to use the port range that is defined in the daemon. This is defined in the Port range for servers field in the daemon Control tab.</p>
Maximum number of server processes	<p data-bbox="610 1703 1208 1761">Enter the maximum number of server processes that can run at the same time.</p>
Limit server reuse	<p data-bbox="610 1772 1179 1856">Select this to limit the number of servers that can be reused. If this is selected, the Reuse limit parameter is available.</p>

Table C-5 (Cont.) Server Mode Tab Components

Field	Description
Limit concurrent clients per server	<p>If Limit server reuse is selected, in the field next to the check box, enter the maximum number of times a server can be reused. Select the maximum of clients accepted in a server process.</p> <p>A one-client server can be reused after its (single) client has disconnected. Reuse of servers enhances startup performance because it avoids the need to repeat initialization.</p> <p>This parameter is not available if the Limit server reuse parameter is not selected.</p> <p>This parameter is not available if the server mode value is singleClient.</p>
Specify Server Priority	<p>Select this to limit the number of clients that a server can accept for the current workspace process.</p> <p>If this is not selected, the number of clients is unlimited.</p> <p>If Limit concurrent clients per server is selected, in the field next to the check box, enter the maximum number of clients that a server process for the current workspace accepts. The default for this field is None, indicating that the number of clients for each server is unlimited. This field is available if the server mode value is multiClient or multiThreaded.</p> <p>Set the priority for servers. For example, a workspace for applications with online transaction processing can be assigned a higher priority than a workspace that requires only query processing. The lower the number, the higher the priority. For example, workspaces with a priority of 1 are given a higher priority than workspaces with a priority of 2.</p> <p>Note: This is unavailable if Use default server priority is selected.</p>
Use default server priority	<p>Sets the priority to 0. There is no specific priority for this workspace. Clear this check box to set a priority in the Specify server priority parameter.</p>
Keep when daemon ends	<p>Select this to stop all servers started by that daemon when a daemon is shutdown, even if they are active. Select this for the workspace servers to remain active, even after the daemon has been shut down. If selected, it is the responsibility of the system operator or manager to ensure that the servers are eventually stopped. This must be done at the system level.</p>
Server Provisioning	
Number of prestarted servers in pool	<p>Initial number of servers: The number of server processes that are prestarted for this workspace when the daemon starts. When the number of available server processes drops lower than the value specified in the Minimum number field, the daemon again starts server processes until this number of available server processes is reached. The default for this field is 0.</p>

Table C-5 (Cont.) Server Mode Tab Components

Field	Description
Number of spare servers	The minimum number of server processes in the prestarted pool before the daemon resumes creating new server processes (to the value specified in the Initial number of servers field). If this field is set to a value higher than the Initial number of servers field, the daemon uses the value specified in the Initial number of servers field. The default for this field is 0.
Prestarted server pool limit	The maximum number of available server processes. Once this number is reached, no new nonactive server processes are created for the particular workspace. For example, if several server processes are released at the same time, so that there are more available server processes than specified by this field, the additional server processes higher than this value are terminated. The default for this field is zero, meaning that there is no maximum.
Resource limitations	
Number of sub-tasks	The number of sub-tasks for a server that are prestarted for this workspace when the daemon starts. In addition to setting up a pool of server processes as described earlier, you can set additional server processes as sub-tasks by specifying this parameter. If you set 10 servers and 10 prestarted sub-tasks then 100 tasks are started (10 sub-tasks for each process).

Security

The **Security** tab lets you configure the security level for a workspace. This lets you set the security options for the workspace only. The Security tab is used to:

- Grant administration rights for the workspace
- Determine access to the workspace by a client

Do the following to open the **Security** tab:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design Perspective Configuration view, expand the **Machines** folder.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Expand the Daemons node to display the workspaces in the Runtime Explorer.
5. Right-click the workspace and select **Edit Workspace Configuration**.
6. Click the **Security** tab.
7. After making changes to the workspace, right-click the daemon and select **Reload Configuration**.

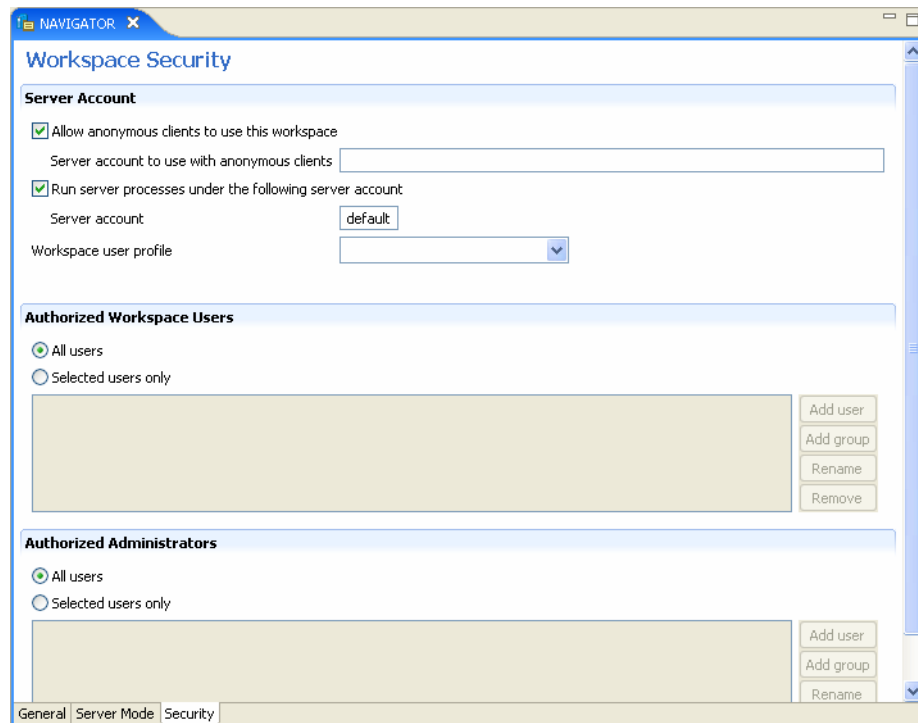
Notes:

- You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Open**.
- Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.

8. Right-click the daemon and select **Recycle Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **Security** tab is shown in the following figure:

Figure C-6 The Security tab



The **Security** tab has the following fields:

Table C-6 Security Tab

Field	Description
Server Account	This section defines the users (accounts) allowed to access the workspace, firewall access ports, workspace account, and anonymous login permissions.
Use specific workspace account	<p>Select this to define the operating system account used for the workspace.</p> <p>If selected, enter the name of the workspace account in the workspace account field.</p> <p>If not selected, the account name that was provided by the client is used.</p>
Allow anonymous clients to use this workspace	<p>Select this to allow this workspace to be invoked without authentication.</p> <p>If selected, enter the name of the workspace account in the Server account to use with anonymous clients field.</p>
Authorized Workspace users	<p>Indicate which users have permission to use the workspace. Select one of the following</p> <ul style="list-style-type: none"> ■ All users: Any user who has logged on to the daemon may use the workspace ■ Selected users only: Select this to allow only users (or accounts) with specific permission to use the workspace. When this is selected, add the names of users (or accounts) and groups that can be use the workspace in the field below. Note: If no user is specified, any user who has logged on to the daemon may use the workspace.
Authorized Administrators	<p>Identifies the users (accounts) with administrator privileges. Select one of the following:</p> <ul style="list-style-type: none"> ■ All users: Indicates that anyone can access the workspace and change the settings. ■ Selected users only: Select this to allow only users (or accounts) with specific permission to be administrators. When this is selected, add the names of users (or accounts) and groups that can be workspace administrators. If no user is specified, any user who has logged on to the daemon may administrator this workspace.

Globalization Settings

The OracleAS CDC Adapter for Tuxedo provides the globalization support for the following languages:

- Arabic
- English (the default)
- French
- German
- Greek
- Hebrew
- Italian
- Japanese
- Korean
- Portuguses
- Simple Chinese
- Spanish
- Traditional Chinese
- Turkish

This appendix describes how to define the language support.

Defining the Language and Code Page

The language and code page parameters are accessed from the computer where Oracle Studio is installed.

Perform the following steps to define the required language and code page:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. In the Design perspective **Configuration** view, expand the **Machines** folder.
3. Expand the machine for which you want to set the language.
4. Expand the Bindings folder and right-click the **NAV** binding.
5. Select **Open**.
6. Expand the **Language Settings** and do the following:

- From the **Language** list, select the NLS supported language to use in this binding. Valid values are listed in the Language Name column of [NLS Language Codes](#) table.
- From the **Codepage** list, select the code page to use with this language. The code pages available are determined by the Language that is selected. If you have additional code pages available, you can manually enter them in this field.

Note: If you change the language, the code page also changes. Use the selected code page with the language you selected.

If no code page is selected, the default code page for the selected language is used.

- From the **NLS string** list, select the NLS string for this language and code page. The NLS strings available are determined by the code page that is selected. If you have additional NLS strings available, you can manually enter them in this field.

The code page is used by a field with a data type defined as `nlsString`. This parameter is used for a field with a code page that is different than the computer's code page. This property includes values for the name of the code page and whether the character set reads from right to left (as in middle-eastern character sets).

For example, the following specifies a Japanese EUC 16-bit code page:

```
<misc nlsString="JA16EUC,false"/>
```

7. Save the change. New servers use the language selected.

The following table lists the code pages:

Table D-1 NLS Language Codes

Language Name	Language Code	Windows Default	ASCII Platforms (Default)	EBCDIC Platforms (Default)	Alternative Codepages (EBCDIC based unless noted otherwise)
English US	ENUS	Windows-1252	ISO-8859-15	IBM1140	IBM285, IBM500, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
English UK	ENUK	Windows-1252	ISO-8859-15	IBM1146	IBM285, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)

Table D-1 (Cont.) NLS Language Codes

Language Name	Language Code	Windows Default	ASCII Platforms (Default)	EBCDIC Platforms (Default)	Alternative Codepages (EBCDIC based unless noted otherwise)
French	FRE	Windows-1252	ISO-8859-15	IBM1147	IBM297, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
Latin International	LAT	Windows-1252	ISO-8859-15	IBM1148	IBM500, IBM037, IBM1140, IBM1047, ISO-8859-1 (ASCII based)
Spanish	SPA	Windows-1252	ISO-8859-15	IBM1145	IBM284, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
German	GER	Windows-1252	ISO-8859-15	IBM1141	IBM273, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
Portuguese	POR	Windows-1252	ISO-8859-15	IBM1140	IBM037, IBM500, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
Italian	ITL	Windows-1252	ISO-8859-15	IBM1144	IBM280, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
Greek	GRK	Windows-1253	ISO-8859-7	IBM875	-
Russian ¹	RUS	Windows-1251	ISO-8859-5	IBM1154	IBM1025

Table D-1 (Cont.) NLS Language Codes

Language Name	Language Code	Windows Default	ASCII Platforms (Default)	EBCDIC Platforms (Default)	Alternative Codepages (EBCDIC based unless noted otherwise)
Turkish ²	TUR	Windows-1254	ISO-8859-9	IBM1155	IBM1026
Hebrew	HEB	Windows-1255	ISO-8859-8	IBM424	IBM 862
Arabic	ARA	Windows-1256	ISO-8859-6	IBM420	
Japanese	JPN	SJIS	SJIS	IBM939	EUC-JP=EUC, VMS-JP
Chinese - Simplified	SCHI	GBK	GBK	IBM935	-
Chinese - Traditional	TCHI	BIG5	BIG5	IBM937	-
Korean	KOR	MS949	EUC-KR	IBM933	MS949

¹ Russian users who use ANSI 1251 Cyrillic as their Windows code page must edit the `RUS.TXT` file and compile it to `RUS.CP` using the `NAV_UTIL CODEPAGE`.

² To work with solutions in Oracle Studio, when using Turkish, add the `-nl en` switch to the Target path in the Oracle Studio shortcut properties. For example: `"C:\Program Files\Oracle\Studio1\studio.exe -nl en"`

Editing XML Files in Oracle Studio

In many cases you must manually edit the metadata to configure parts of a solution or composition. Metadata is created in XML format. You define aspects of a solution by changing the values of the elements and attributes of the XML files that belong to the solution. Oracle Studio provides a graphical interface where you can define the various aspects of a solution. This interface lets you make changes easily without having to manually edit the XML file.

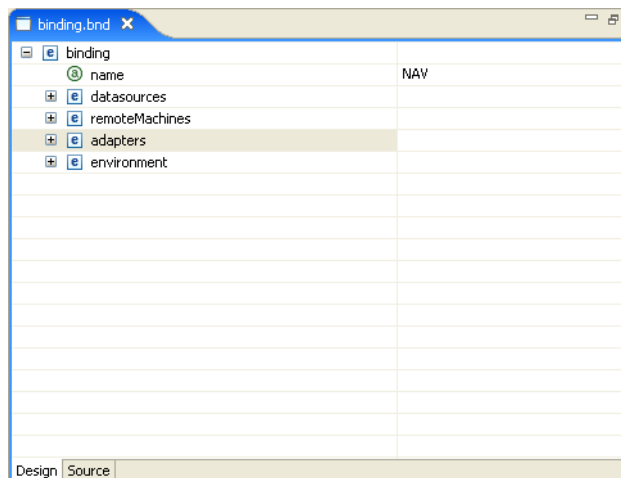
Preparing to Edit XML Files in Oracle Studio

You can edit XML files for the following items in Oracle Studio:

- Machines
- Bindings.
- Daemons
- Users

When you open an XML file, a graphical representation of the file is opened in the editor. The editor displays the elements and attributes in the file in the first column and their corresponding values in the second column. Each entry has an icon that indicates whether the entry is an element or an attribute. Click the **Source** tab to view the file in its native format. The following figure is an example of the editor's view of an XML file.

Figure E-1 XML Graphical Display



To edit an XML file in Oracle Studio

1. In the Design perspective, open the Navigator view.
2. In the Navigator view, find the item with the XML file to edit. This can be a machine, binding, daemon, or user.
3. Right-click the item and select **Open as XML**. A graphical list of the file's elements and attributes opens in the editor.
4. Find the element or attribute (property) to change.
5. Click in the right column next to the property you are changing and edit or add the value.
6. Save the file, then select it again in the Project Explorer and press F5 to refresh.
The XML file is updated automatically.

Making Changes to the XML File

You can also make the following changes to XML files in Oracle Studio:

- [Remove Objects](#)
- [Add DTD Information](#)
- [Edit Namespaces](#)
- [Add Elements and Attributes](#)
- [Replace an Element](#)

Remove Objects

You can delete an element, attribute, or other object from the XML file.

To remove an object

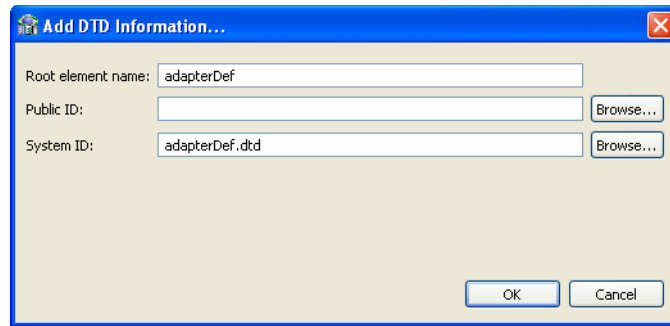
1. Right-click an object from the list in the editor.
2. Select **Remove**.

Add DTD Information

You can add DTD information to an element or attribute.

To add DTD Information

1. Right-click an element or attribute and select **Add DTD Information**. The Add DTD Information dialog box opens.

Figure E–2 Add DTD Information Dialog Box

2. Enter the information requested in the dialog box. The following table describes the Add DTD Information dialog box.

Table E–1 Add DTD Information

Field	Description
Root element name	The name of the XML root element.
Public ID	<p>The value in this field is the Public Identifier. It is used to associate the XML file (using an XML catalog entry) with a DTD file by providing a hint to the XML processor.</p> <p>Click Browse to select an XML catalog entry from a list. An XML Catalog entry contains two parts, a Key (which represents a DTD or XML schema) and a URI (which contains information about a DTD or XML schema's location). Select the catalog entry you want to associate with your XML file.</p>
System ID	<p>The value in this field is the DTD the XML file is associated with. You can change the DTD the file is associated with by editing this field. The XML processor tries to use the Public ID to locate the DTD, and if this fails, it uses the System ID to find it.</p> <p>Click Browse to select a system ID. You can this in two ways:</p> <ul style="list-style-type: none"> ■ Select the file from the workbench. In this case, update the with the import dialog box. ■ Select an XML catalog entry.

3. Save the file, then select it again in the Project Explorer and press F5 to refresh. The XML file is updated automatically.

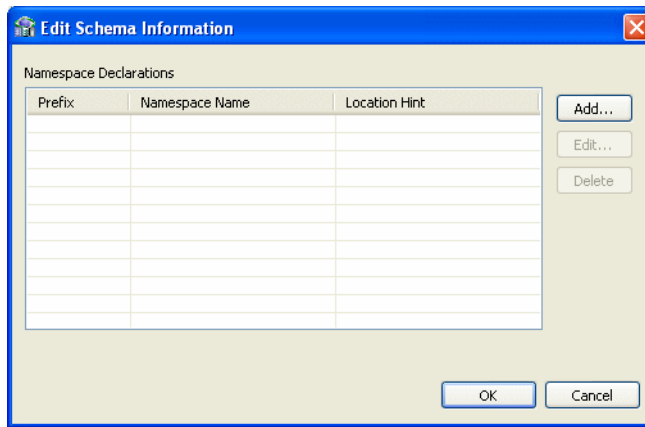
Edit Namespaces

You can make changes to the namespaces associated with an element or attribute.

To edit namespaces

1. Right-click an element or attribute and select **Edit namespaces**. The Edit Schema Information dialog box opens.

Figure E-3 Edit Schema Information



2. Click a button to make any changes to this information.

To add a new namespace

1. From the Schema Information dialog box, click **Add**.
2. The Add Namespace Definitions dialog box opens. Select one of the following:
 - **Select from registered namespaces.** This selection is available when the dialog box opens. Select from the list of registered namespaces and then click OK. If no registered namespaces are available, the list is empty.
 - **Specify new namespace.** Enter the information described in the following table:

Table E-2 New Namespace

Field	Description
Prefix	The prefix is added to all qualified elements and attributes in the XML file.
Namespace Name	The namespace of the XML file.
Location Hint	<p>The location of the XML schema of the XML file. An XML Catalog ID or a URI can be entered in this field.</p> <p>Click Browse to search for the schema you want You can this in two ways:</p> <ul style="list-style-type: none"> ■ Select the schema from the workbench. In this case, update the with the import dialog box. ■ Select an XML catalog entry. <p>The Namespace Name and Prefix fields are be filled with the appropriate values from the schema (you must leave the fields blank for this to occur).</p> <p>Note: If you are creating an XML file from an XML schema, you cannot change the Namespace Name or Location Hint values.</p>

To edit a namespace

1. From the Schema Information dialog box, click **Edit**.
2. Enter the information in the fields.

Add Elements and Attributes

You can add additional elements and attributes to the XML file.

To add Elements and Attributes

1. Right-click an element.
2. Select one of the following:
 - **Add Attribute** to add an attribute under the selected element.
 - **Add Child** to add another element under the selected element
 - **Add Before** to add another element above the selected element
 - **Add After** to add another element below the selected element

Note: The Oracle Studio XML editor is Context sensitive to Oracle schemas. In this case, when adding elements and attributes to an XML file with an Oracle schema, you can select an element or attribute from a list of the possible values (depending on the schema definition). This list is available as a submenu.

3. Provide a name for the element or attribute if required. You may also select the element from a submenu. The element or attribute is added to the file.
4. Save the file, then select it again in the Project Explorer and press F5 to refresh. The XML file is updated automatically.

Replace an Element

You can replace an element with another legal element.

To replace an element

1. Right-click an element from the list in the editor.
2. Select **Replace with**.
3. Select an element from the submenu. Only legal elements are available.
4. The original element is replaced with the selected element.

Index

A

ACX trace parameter, 8-6
add timestamp to traced events parameter, 8-6

B

binary XML log level parameter, 8-6

C

cache buffer size parameter, 8-8
code page parameter, 8-8
COM maximum XML size in memory parameter, 8-9
COM maximum XML size parameter, 8-9
COM XML transport buffer parameter, 8-9
Configuration, 2-1
 OracleAS adapters, 2-1
 Tuxedo OracleAS adapter, 2-6
configuration
 security, 2-2
configuring JCA configuration files for inbound interactions
 Tuxedo queue adapter, 4-4
configuring JCA configuration files for outbound, 4-3
creating JCA configuration files, 4-1
 Tuxedo adapter, 4-1
 Tuxedo gateway adapter, 4-2
 Tuxedo queue adapter, 4-1

D

daemon
 logging, C-3
 security, C-5
 server modes, 8-2
 shutting down, 7-2
 starting, 7-1
 timeout, 7-6
data types
 atomic metadata, B-2
 NLS string parameter, 8-8
debug parameters
 ACX trace, 8-6
 add timestamp to traced events, 8-6

environment, 8-6
GDB trace, 8-6
general trace, 8-6
library XML log level, 8-6
log file, 8-7
optimizer trace, 8-7
query processor trace, 8-6
query warnings, 8-6
trace directory, 8-7
 Transaction extended logging, 8-7
defining records
 Tuxedo Queue OracleAS adapter, 2-23

E

environment parameters
 ACX trace, 8-6
 add timestamp to traced events, 8-6
 cache buffer size, 8-8
 code page, 8-8
 COM maximum XML size, 8-9
 COM maximum XML size in memory, 8-9
 COM XML transport buffer, 8-9
 debug, 8-6
 GDB trace, 8-6
 general trace, 8-6
 language, 8-8
 log file, 8-7
 miscellaneous, 8-7
 nav utility editor, 8-7
 NLS string, 8-8
 odbc, 8-8
 oledb, 8-8
 optimizer, 8-8
 optimizer trace, 8-7
 parallel processing, 8-9
 query processor trace, 8-6
 query warnings, 8-6
 queryProcessor, 8-8
 Replace invalid XML characters, 8-9
 temporary directory, 8-7
 trace directory, 8-7
 Transaction extended logging, 8-7
 transactions, 8-9
 tuning, 8-9
 XML, 8-9

- XML date format parameter, 8-9
- XML trim char column, 8-9
 - year 2000 policy, 8-7
- error log, binary XML log level parameter, 8-6
- error log, log file parameter, 8-7

G

- GDB trace parameter, 8-6
- general parameters
 - cache buffer size, 8-8
 - nav utility editor, 8-7
 - temporary directory, 8-7
 - year 2000 policy, 8-7
- general trace parameter, 8-6

I

- Importing metadata
 - Tuxedo OracleAS adapter, 2-7
- inbound configuration
 - Tuxedo gateway (synchronous), 3-4
 - Tuxedo queue, 3-3

J

- JCA configuration files
 - configuring outbound interactions, 4-3
 - configuring inbound interactions
 - Tuxedo queue, 4-4
 - creating, 4-1
 - Tuxedo adapter, 4-1
 - Tuxedo gateway adapter, 4-2
 - Tuxedo queue adapter, 4-1

L

- language parameter, 8-8
- language parameters
 - code page, 8-8
 - environment, 8-8
 - language, 8-8
 - NLS string, 8-8
- log file parameter, 8-7
- log files
 - binary XML log level parameter, 8-6
 - daemon options, C-3
 - log file parameter, 8-7
- logging
 - daemon configurations, C-3
 - optimizer strategy, 8-7

M

- metadata, atomic data types, B-2
- Microsoft software requirements, 8-15
- miscellaneous parameters
 - environment, 8-7

N

- nav utility editor parameter, 8-7
- nav utility editor, text editor, 8-7
- NLS string parameter, 8-8

O

- odbc environment parameters, 8-8
- oledb environment parameters, 8-8
- operating system requirements
 - PC, 8-15
- optimizer
 - environment parameters, 8-8
 - trace directory parameter, 8-7
- optimizer trace parameter, 8-7
- OracleAS adapter configuration, 2-1
- OracleAS adapter setup, 2-5
- outbound configuration, 3-2

P

- parallel processing environment parameters, 8-9

Q

- query optimizer
 - logging strategy, 8-7
 - trace directory parameter, 8-7
- query processor trace parameter, 8-6
- query warnings parameter, 8-6
- queryProcessor environment parameters, 8-8

R

- Replace invalid XML characters parameter, 8-9

S

- Security
 - administrative rights, 2-3
 - password access, 2-2
- security, 2-2
 - daemon configurations, C-5
- servers
 - configuring modes, 8-2
 - reusable, 8-3
 - Reuse limit daemon parameter, C-13
 - ReuseLimit daemon parameter, 8-3
- software requirements
 - Microsoft, 8-15
 - PC operating system, 8-15

T

- temporary directory parameter, 8-7
- temporary files, 8-7
- timeout
 - client idle, 7-6
 - daemon, 7-6
- trace directory parameter, 8-7

- trace information, logging, 8-6
- Transaction extended logging parameter, 8-7
- transactions environment parameters, 8-9
- tuning environment parameters, 8-9
- Tuxedo adapter configuration, 3-2
- Tuxedo adapter integration, 4-1
- Tuxedo gateway adapter configuration, 3-4
- Tuxedo Gateway OracleAS adapter, 2-25
 - adding to the system, 2-26
 - importing metadata, 2-27
 - properties, 2-27
- Tuxedo UBB configuration file, 2-35, 2-36
- Tuxedo queue adapter configuration, 3-3
- Tuxedo Queue OracleAS adapter setup, 2-19
- Tuxedo Queue unstructured records, 2-23
- TuxedoAS adapter configuration, 2-6
 - metadata import, 2-7

W

- WebLogic integration
 - WebLogic
 - integration, 3-1
- Workspace server mode, C-13

X

- XML
 - environment parameters, 8-9
- XML date format parameter, 8-9
- XML parameters
 - COM maximum XML size, 8-9
 - COM maximum XML size in memory, 8-9
 - COM XML transport buffer, 8-9
 - Replace invalid XML characters, 8-9
 - XML date format parameter, 8-9
 - XML trim char column, 8-9
- XML trim char column parameter, 8-9

Y

- Y2K
 - See* year 2000 policy parameter
- year 2000 policy parameter, 8-7

