Oracle® Application Server

Adapters for VSAM User's Guide

12*c* Release 1 (12.1.3)

E67143-01

Novemver 2015

ORACLE®

Oracle Application Server Adapters for VSAM User's Guide, 12*c* Release 1 (12.1.3)

E67143-01

Copyright © 2005, 2011, 2015 Oracle and/or its affiliates. All rights reserved.

Primary Author: David Goldman

Contributor: Sami Zeitoun, Itamar Ankorion

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 vii

1 Introduction

OracleAS Adapter for VSAM Overview 1-1
VSAM Data 1-2
Accessing VSAM Data 1-2
OracleAS Adapter for VSAM Architecture 1-2
Integration Flow from Oracle Application Server to the Legacy Application 1-3

2 Configuring OracleAS Adapter for VSAM

Setting Up the IBM z/OS Platform 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
Setting Up Run-Time User Access to the IBM z/OS Platform 2-5
Modeling Interactions for OracleAS Adapter for VSAM 2-5
Setting Up the VSAM Data Source 2-6
Importing Metadata for the VSAM Data Source 2-8
Setting Up an Oracle Connect Adapter 2-17
Generating Outbound Interactions 2-19
Viewing the XML Schema 2-22
Creating XML Schemas 2-22

3 Integrating OracleAS Adapters for VSAM with WebLogic

Overview of Integrating OracleAS Adapter for VSAM with WebLogic 3-1
Configuring the VSAM Adapter 3-2
High Availability 3-3

4 Integrating OracleAS Adapters for VSAM

Configuring the OracleAS Adapter for VSAM 4-1
Creating JCA Configuration Files 4-1
Configuring the Binding for Outbound Applications 4-2

5 Configuring an OracleAS Adapter for VSAM for Outbound Interactions

Creating Outbound Interactions with the OracleAS Adapter for VSAM 5-1
Requirements 5-1
Adding the VSAM Hospital Data Source 5-1
Prepare the System 5-1
Set up Machine Access to Oracle Connect 5-2
Add a VSAM Data Source 5-2
Import the Metadata 5-3
Create the findDoctor and findPatient Interactions 5-4
Add a Database Adapter 5-4
Create Interactions 5-4
Create the JCA Configuration Files 5-5

6 Creating the Project in JDeveloper Examples

Getting Started 6-1
Prerequisites 6-1
Beginning Tasks 6-2
Deploy a BPEL Outbound Process 6-2
Design-Time Configuration 6-2
Create a BPEL Project for a BPEL Outbound Process 6-3
Use a Third-Party Adapter to Read the Doctor Data 6-4
Design a BPEL Outbound Process 6-6
Edit the Invoke_1 Activity 6-7
Deploy the Project 6-8
Run-time Configuration 6-8
Deploy a Mediator Process 6-8
Design-Time Tasks 6-8
Create a Mediator Project for a Mediator Outbound Process 6-9
Use a Third-Party Adapter to Read the Doctor Data 6-10
Design an Outbound Mediator 6-12
Create Mapping Rules and Map the Files 6-13
Deploy the Project 6-14

7 Troubleshooting OracleAS Adapter for VSAM

Troubleshooting the Daemon 7-1
Starting the Daemon 7-1
Task: Starting the Daemon 7-2
Shutting Down the 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 7-5

Resolving Communication Errors 7-6
Resolving Specific Errors 7-6

8 Advanced Features of OracleAS Adapter for VSAM

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 8-6
Debug 8-7
General 8-8
Language 8-8
Modeling 8-9
ODBC 8-9
OLEDB 8-9
Optimizer 8-9
Parallel Processing 8-10
Query Processor 8-10
Transaction 8-13
Tuning 8-13
XML 8-14
Migration Considerations 8-15
Security Considerations 8-15
Setting Design Time Security 8-16
Setting Run Time Security 8-16
Transaction Support 8-17

A Advanced Tuning of the Metadata

Metadata for VSAM A-1
General Tab A-1
Columns Tab A-3
Column Definition Section A-4
Column Properties A-5
Indexes Tab A-7
Table Information A-7
Properties A-8
Statistics Tab A-9
Generating Statistics A-10
Advanced Tab A-11
Adapter Metadata A-13
Adapter Metadata General Properties A-13
Adapter Metadata Interactions A-15
Interaction Advanced Tab A-16
Adapter Metadata Schema Records A-18
Adapter Metadata XML A-20

B Back-end Adapter Data Type Support

Data Type Mapping B-1

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 Supported SQL Syntax and SQL Enhancements

Supported SQL Statements D-1
Supported Operators D-1
Supported Functions D-2
SQL Enhancements D-3
Generating Hierarchical Results D-3
Accessing Hierarchical Data Using SQL D-3
Flattening Hierarchical Data Using SQL D-4
Using an Alias D-4

E Globalization Settings

Defining the Language and Code Page E-1

F Editing XML Files in Oracle Studio

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

Index

Preface

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

This document describes the features of OracleAS Adapter for VSAM.

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 VSAM
- Diagnosing errors
- Using OracleAS to access VSAM 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 VSAM.

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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

Oracle Application Server connects to a Virtual Sequential Access Method (VSAM) system through the Oracle Application Server Adapter for VSAM (OracleAS Adapter for VSAM). The OracleAS Adapter for VSAM provides connectivity and runs interactions on a VSAM system. This chapter provides an overview of the feature and architecture of OracleAS Adapter for VSAM.

This chapter contains the following topics:

- OracleAS Adapter for VSAM Overview
- OracleAS Adapter for VSAM Architecture

1.1 OracleAS Adapter for VSAM Overview

The OracleAS Adapter for VSAM models VSAM as a simple database with an SQL front end. The adapter is native to the data source and IBM OS/390 or z/OS system, providing direct access that results in improved performance.

OracleAS Adapter for VSAM includes the following features:

- Captures and maintains a metadata schema for VSAM by importing COBOL copybooks and transforming them into mapping definitions for Oracle Connect for VSAM on the IBM z/OS system and by associating the data structures with specific physical files.
- Full-fledged query processor and optimizer that runs SQL against the VSAM data with full support for VSAM data types such as arrays.
- Uses an enterprise application integration (EAI) model. Users of a requesting application can model the interactions that they want implemented, specifying what each interaction does, when it occurs and the inputs and outputs expected for each interaction.
- Maps data structures for facilitating access to VSAM data from within Oracle Application Server.
- Implements interactions as parameterized SQL with the parameters forming the input record, and with the output (in cases where there is an output) aggregated into an XML document forming the output record.
- Enables you to access VSAM data managed by CICS or directly.
- Does support global transactions under CICS and can fully participate in a distributed transaction.

1.1.1 VSAM Data

VSAM is an IBM disk file storage scheme first used in S/370 and VS (Virtual Storage). VSAM comprises the following three access methods:

- Keyed Sequential Data Set (KSDS)-indexed file
- Relative Record Data Set (RRDS)-relative file
- Entry Sequenced Data Set (ESDS)-sequential file

These VSAM access methods are supported by the adapter

VSAM has no built-in metadata schema although a VSAM file does include information about its type (KSDS, RRDS, or ESDS) and index definition (KSDS only). Typically, VSAM is used by COBOL programs, either standalone (also known as batch programs) or CICS transactions. The VSAM record definitions are typically maintained within the COBOL code as COBOL copybooks.

1.1.1.1 Accessing VSAM Data

OracleAS Adapter for VSAM enables access to VSAM data under CICS or directly. If the VSAM cluster is managed by CICS, you can still access VSAM data directly to improve performance when the VSAM data is managed by CICS but access to it is read only, and there is no risk that changes are buffered by CICS while the read is performed.

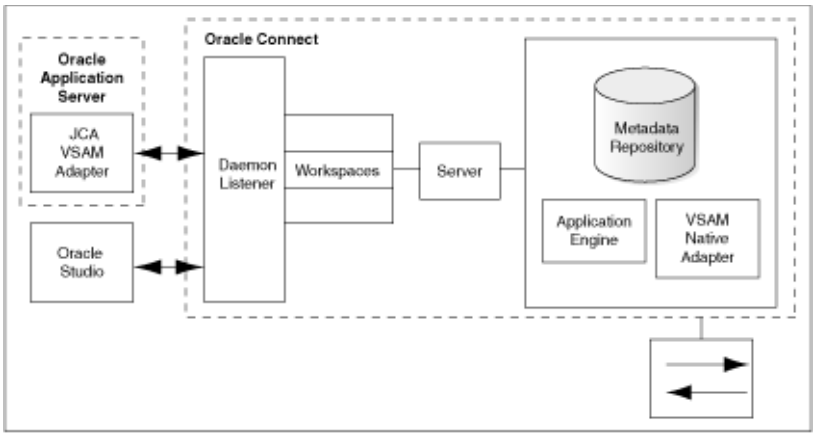
1.2 OracleAS Adapter for VSAM Architecture

OracleAS Adapter for VSAM includes the following components:

- J2CA 1.5 VSAM adapter: The J2CA VSAM 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 VSAM 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 or Linux computer. Oracle Studio uses perspectives that enables you to generate specific information necessary to model OracleAS Adapter for VSAM.

The following figure illustrates the components of OracleAS Adapter for VSAM:

Figure 1–1 The Adapter Architecture



See Also: *Oracle Application Server Adapter Concepts Guide*

1.2.1 Integration Flow from Oracle Application Server to the Legacy Application

The J2CA 1.5 VSAM adapter converts the J2CA interaction invocation received from Oracle Application Server 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 VSAM 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 VSAM 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.

2

Configuring OracleAS Adapter for VSAM

This chapter describes how to configure Oracle Connect using Oracle Studio.

All modeling of Oracle Connect is performed using Oracle Studio. To use Oracle Studio, you first configure it to enable access to the IBM z/OS platform where the VSAM data resides.

This chapter contains the following topics:

- Setting Up the IBM z/OS Platform in Oracle Studio
- Securing Access to Oracle Connect
- Modeling Interactions for OracleAS Adapter for VSAM
- Viewing the XML Schema
- Creating XML Schemas

Note: These tasks assume you have permission to access the IBM z/OS platform and that the Oracle Connect daemon is running on this computer.

Check with the system administrator to ensure these requirements are fulfilled.

2.1 Setting Up the IBM z/OS Platform in Oracle Studio

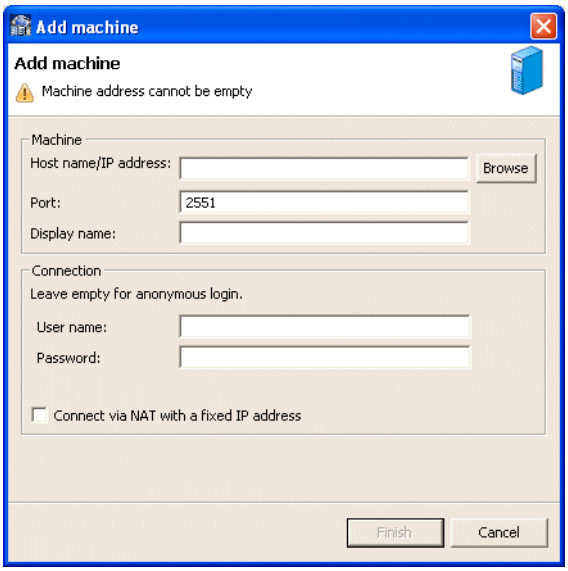
Using Oracle Studio, perform the following steps to configure the IBM z/OS platform:

1. From the **Start** menu, select **Programs**, **Oracle**, and then select **Studio**. Oracle Studio opens.
2. Right-click **Machines** in the Configuration Explorer and select **Add Machine**. The Add Machine screen is displayed. For more information, see Figure 2–1).
3. Enter the name of the computer you want to connect to, or click **Browse** to select the computer from the list of computers that is displayed and which use the default port `2551`.
4. Specify the user name and password of the user who was specified as the administrator when Oracle Connect was installed.

Note: Selecting **Anonymous connection** enables anyone having access to the computer to be an administrator, if this was defined for the computer.

The Add Machine screen is shown in the following figure:

Figure 2–1 The Add Machine screen



5. Click **Finish**.

   The computer is displayed in the Configuration Explorer.

2.2 Securing Access to Oracle Connect

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

During modeling 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:

- Setting Up Run-Time User Access to the IBM z/OS Platform

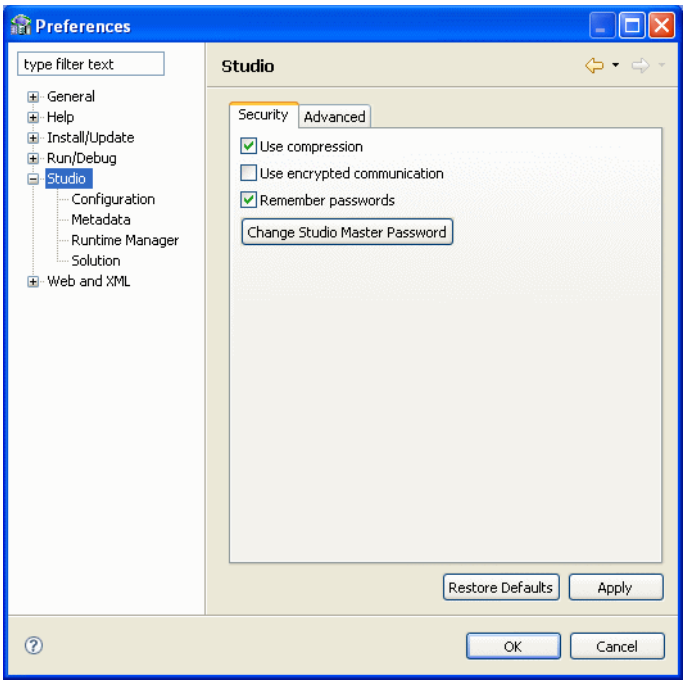
2.2.1 Setting Password Access to Oracle Studio

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 the password:

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

Figure 2–2 The Preferences screen



4. Click **Change Studio Master Password**. The Change Master Password screen is displayed as shown in the following figure:

Figure 2–3 The Change Master Password screen

5. Leave the **Enter current master password** field blank and type a new master password.
6. Confirm the new password.
7. Click **OK**.

2.2.2 Specifying Users with Administrative Rights

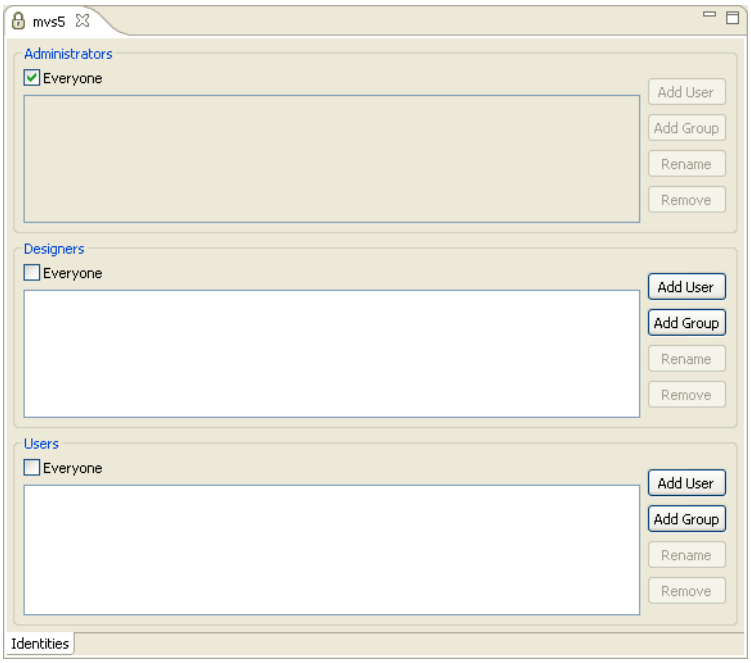
By default, only the user who was specified during the installation as an administrator has the authorization to modify settings on that computer from Oracle Studio. This user can then authorize other users to make changes or to view the definitions for a selected computer. Adding a computer to Oracle Studio is described in "Setting Up the IBM z/OS Platform in Oracle Studio" on page 2-1.

Note: The default during installation is to enable all users to be administrators.

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

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

Figure 2–4 The Administration Authorization Identities tab



The screen has the following sections:

Administrators: Administrators can view and modify all the definitions in Oracle Studio for the selected computer. 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 the Bindings and Users nodes 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 computer in Oracle Studio for the selected computer. Regular users cannot modify any of the definitions.

3. Add users or groups of users by clicking **Add User** or **Add Group** for the relevant sections.

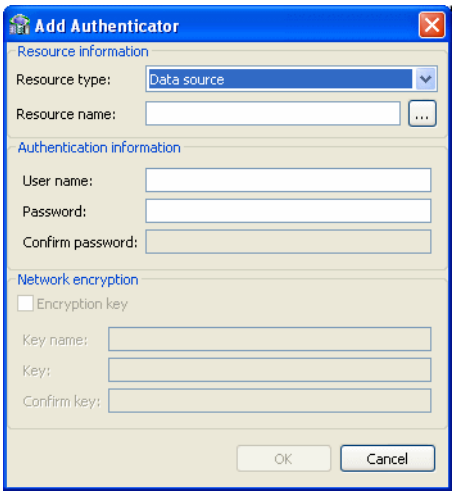
   The user or group that is added must be recognized as a valid user or group for the computer. Once a name has been added to a section, only the user or group that logs on with that user name has the relevant authorization.

2.2.3 Setting Up Run-Time User Access to the IBM z/OS Platform

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 at run time, when anonymous access is not allowed.

1. From the **Start** menu, select, **Programs**, **Oracle**, and then select **Studio**. Oracle Studio opens.
2. From the Design perspective, Configuration view, expand the **Machines** folder, then expand the machine where you want to set the user name and password.
3. Expand **Users**.
4. Right-click **NAV** and select **Edit User.** The NAV user profile editor is displayed.
5. In the User editor, click **Add**. The Add Authenticator screen is displayed as shown in the following figure:

Figure 2–5 The Add Authenticator screen



6. Select **Remote Machine** from the Resource Type list.
7. Enter the name of the IBM z/OS computer defined in Oracle Studio.
8. Enter the name and password used to access the computer and confirm the password.
9. Click **OK**.

2.3 Modeling Interactions for OracleAS Adapter for VSAM

Modeling interactions for OracleAS Adapter for VSAM involves defining an Oracle Connect back-end adapter, using Oracle Studio. All the definitions specified in Oracle Studio are written to the IBM z/OS platform.

This section contains the following:

- Setting Up the VSAM Data Source
- Importing Metadata for the VSAM Data Source
- Setting Up an Oracle Connect Adapter
- Generating Outbound Interactions

2.3.1 Setting Up the VSAM Data Source

Oracle Connect requires you to specify the VSAM data source as the first step in setting up the adapter.

Perform the following steps to setup the VSAM data source:

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 defined in "Setting Up the IBM z/OS Platform 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. Right-click **Data sources** and select **New Data source**.

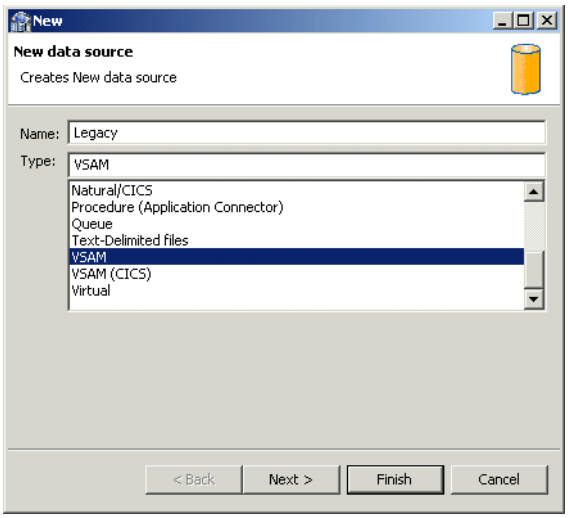
   The New Data Source screen is displayed.
7. Enter a name for the VSAM data source. The name can contain letters and numbers and the underscore character only.
8. Select the data source type from the `Type` list, as follows:
   - If you are accessing VSAM data under CICS, then select **VSAM (CICS)**.
   - If you are accessing VSAM data directly, then select **VSAM**.

   Note: Only use the VSAM option to connect directly to the VSAM data in the following circumstances:

   - The VSAM records are not managed by CICS.
   - The VSAM records are required for read only purposes and changes to the data buffered by CICS while reading the data are not expected.

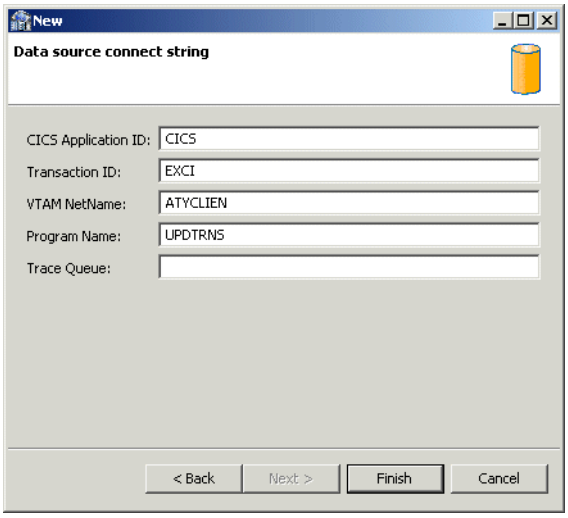
   The New Data Source screen is shown in the following figure:

Figure 2–6 The New Data Source screen



9. Click **Next**. The Data Source Connect String screen is displayed.
10. Enter the Data source connect string. If you select **VSAM (CICS)**, then the following screen is displayed:

Figure 2–7 The Data Source Connect String screen



Where:

- CICS Application ID: The VTAM applid of the CICS target system. The default value is `CICS`. This parameter is used when updating VSAM data. You can determine this value by activating the `CEMT` transaction on the target CICS system. On the bottom right corner of the screen appears the legend `APPLID=`*`target_system`*.
- Transaction ID: The mirror transaction within CICS that receives control through `MRO`, which transfers the transaction from the Oracle Connect for VSAM environment to CICS. The default value is `EXCI`.
- VTAM NetName: The `VTAM` netname of the specific connection being used by `EXCI` (and `MRO`) to relay the program call to the CICS target system. For example, if you issue to `CEMT` the following command:

```
CEMT INQ CONN
```

Then you see that the netname is `BATCHCLI` (this is the default connection supplied by IBM upon the installation of CICS) on the display screen. The default value is `ATYCLIEN`.

- Program Name: The `UPDTRNS` program that is supplied by Oracle Connect for VSAM to enable updating VSAM data.

  For more information, see the *OracleAS Legacy Adapters Installation Guide*.

- Trace Queue: The name of queue for output which is defined under CICS when tracing the output of the UPDTRNS program. When not defined, the default CICS queue is used.

If you select **VSAM**, then the Data Source Connect String screen is displayed, where you provide the following connection string properties:

- Data HLQ: The high-level qualifier where the data files are located. If a value is not specified in this field, then data files are written to the DEF high-level qualifier where Oracle Connect for VSAM is installed.
- Disk Volume name: The high-level qualifier (volume) where the data resides.

11. Click **Finish**.

The new data source is displayed in the Configuration Explorer.

2.3.2 Importing Metadata for the VSAM Data Source

Oracle Connect requires metadata describing the VSAM data source records and the fields in these records. Use the Import Metadata procedure in Oracle Studio to import metadata for the VSAM data source from COBOL copybooks, which describe the data.

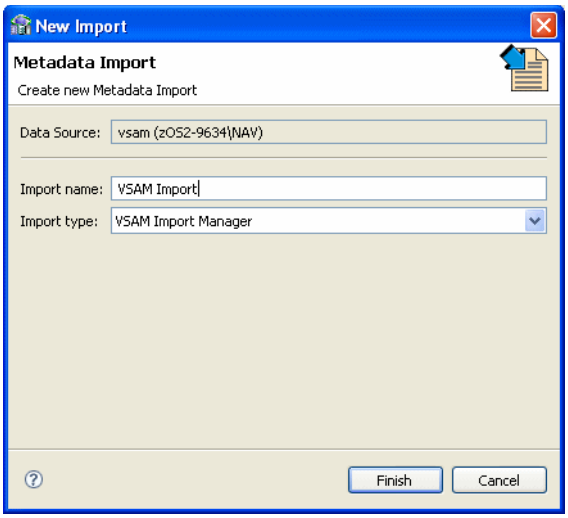
Perform the following steps to import metadata for the VSAM data source, as follows:

1. From the **Start** menu, select **Programs**, **Oracle**, and then select **Studio**.
2. In the Design Perspective Configuration view, expand the **Machines** folder.
3. Expand the machine defined in "Setting Up the IBM z/OS Platform 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
6. Expand the **Data sources** folder.
7. Right-click the VSAM data source defined in "Setting Up the VSAM Data Source" on page 2-6.
8. Select **Show in Metadata View** to open the Metadata tab, with the VSAM data source displayed under the data sources list.
9. Right-click the VSAM data source and select **New Import**.

   The New Import screen is displayed.

10. Enter a name for the import. The name can contain letters and numbers and the underscore character only.
11. Select the import type from the **Import Type** list, as shown in the following figure:

Figure 2–8 The Metadata Import screen



Note: The same New Import screen is displayed for both VSAM imports (VSAM under CICS and VSAM direct) except for the `Import type` field value: either `VSAM Under CICS Import Manager` or `VSAM Import Manager`, respectively)

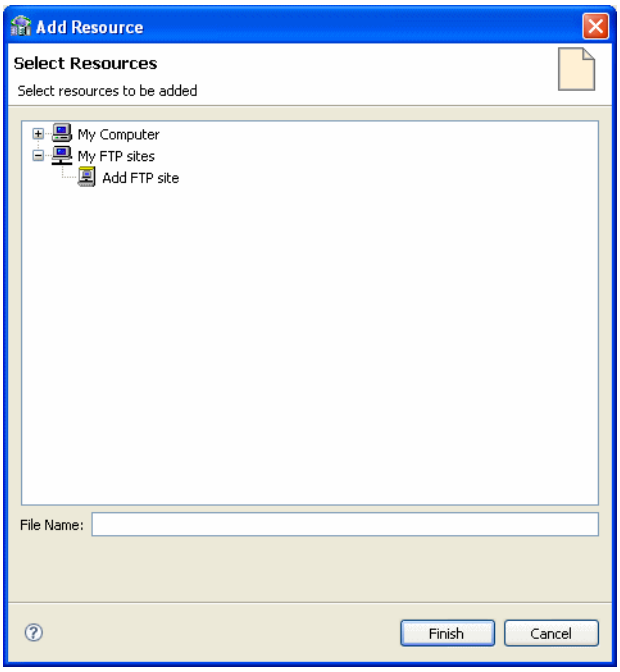
12. Click **Finish**. The Metadata Import wizard opens.
13. Click **Add**.

    The Select Resources screen is displayed, which provides the option to select files from the local computer or copy the files from another computer.
14. If the files are on another computer, then right-click My FTP Sites and select Add. Optionally, double-click **Add FTP site**. The Add FTP Site screen is displayed.
15. Enter the server name or IP address where the COBOL copybooks reside and enter a valid user name and password to access the computer (if anonymous access is used, then click the Anonymous connection check-box) then click **OK**. The FTP site is added to the list of available sites.

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

    The Select Resources screen is shown in the following figure:

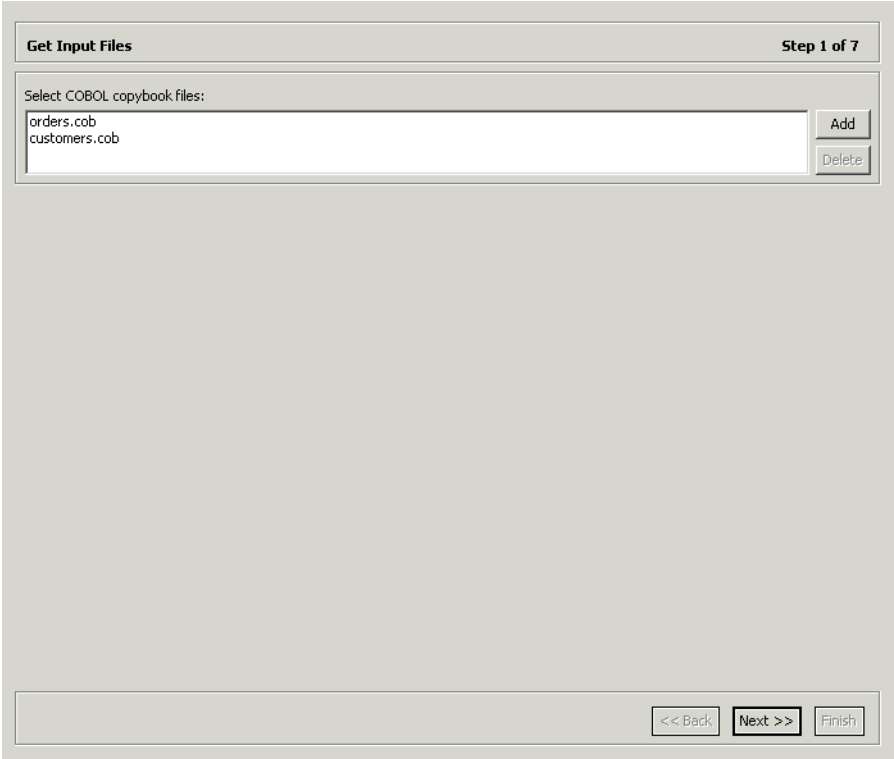
Figure 2–9 The Select Resources screen



16. Right-click the computer and select Set Transfer Type. Enter the transfer type (ASCII or BINARY) and click OK.
17. Expand the node of the added site and locate the necessary COBOL files. To change the high-level qualifier, right-click the computer and select Change Root Directory. Enter the high-level qualifier enclosed in quotes, and click **OK**.
18. Select the 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 2–10 The Get Input Files screen



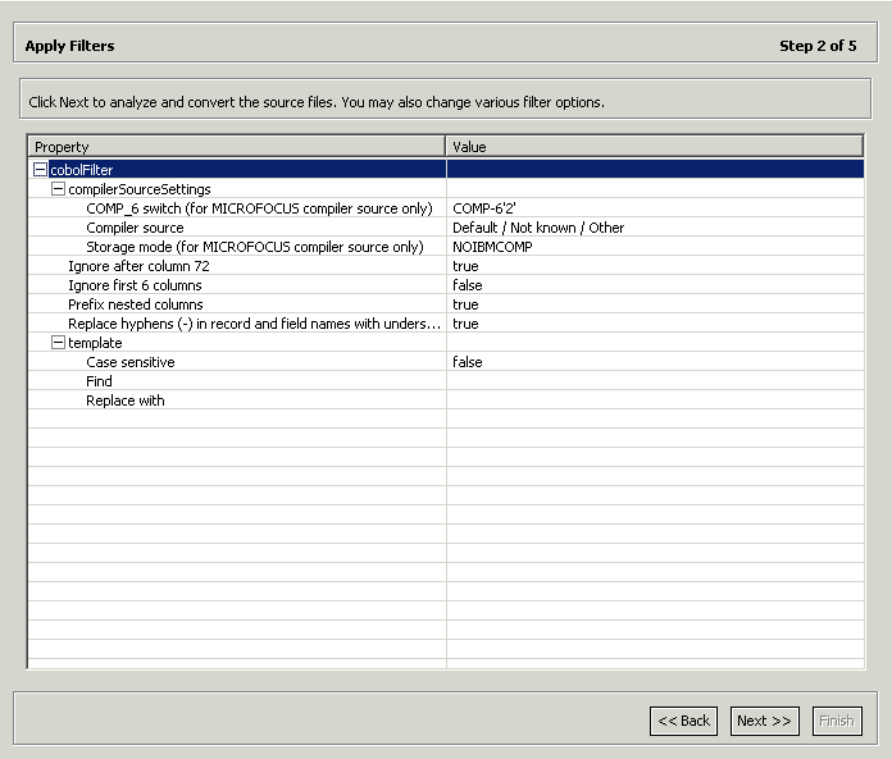
Note: You can import the metadata from one COBOL copybook and later add to this metadata by repeating the import procedure using different COBOL copybooks.

The format of the COBOL copybooks must be identical. That is, you cannot import a COBOL copybook that uses the first six columns with a COBOL copybook that ignores the first six columns. In this type of case you must repeat the import procedure.

19. Click **Next**. The Apply Filters screen is displayed.

The Apply Filters screen is shown in the following figure:

Figure 2–11 The Apply Filters screen



20. Apply filters to the copybooks as required.

The following table lists the avaiable filters:

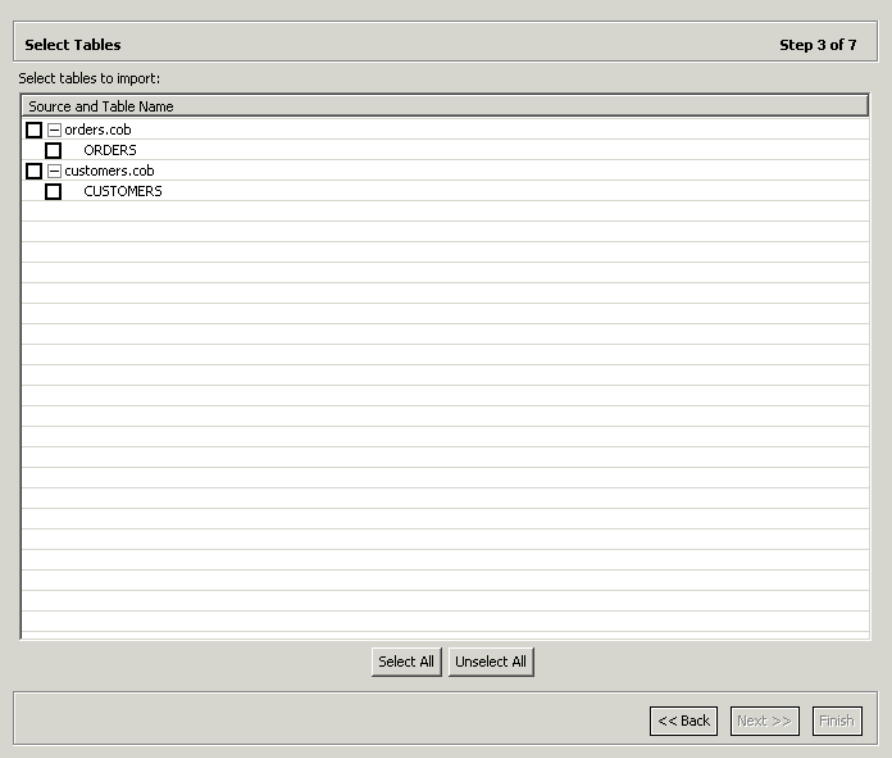
Table 2–1 Available Filters

Filter	Description
COMP_6 switch	The MicroFocus COMP-6 compiler directive. Specify either COMP-6'1' to treat COMP-6 as a COMP data type or COMP-6'2' to treat COMP-6 as a COMP-3 data type.
Compiler source	The compiler vendor.
Storage mode	The MicroFocus Integer Storage Mode. Specify either NOIBMCOMP for byte storage mode or IBMCOMP is for word storage mode.
Ignore after column 72	Ignores columns 73 to 80 in the COBOL copybook.
IgnoreFirst6	Ignores the first six columns in the COBOL copybook.
Replace hyphens (-) in record and field names with underscores (_)	Replaces all hyphens in either the record or field names in the metadata generated from the COBOL with underscore characters.
Prefix nested columns	Prefix all nested columns with the previous level heading.
Case sensitive	Specifies whether to be sensitive to the search string case.
Find	Searches for the specified value.
Replace with	Replaces the value specified for Find with the value specified here

21. Click **Next**.

    The Select Tables screen is displayed, showing the records that are identified in the COBOL copybooks, as shown in the following figure:

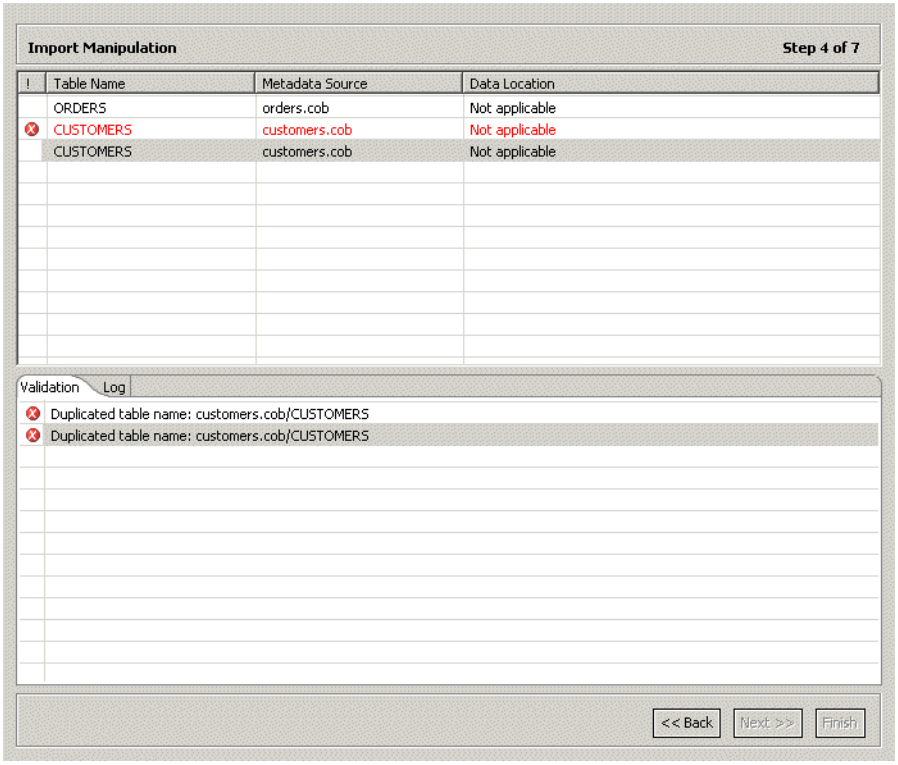
Figure 2–12 The Select Tables screen



22. Select the required tables or click Select All, then click Next.

    The Import Manipulation screen is displayed as shown in the following figure:

Figure 2–13 The Import Manipulation screen



This screen enables you to perform the following operations:

- Resolve table names, where tables with identical names are generated from different COBOL copybooks specified during the import.
- Specify the physical location for the data.
- Specify table attributes.
- Manipulate the fields generated from the COBOL, as follows:
  - Merging sequential fields into one for simple fields.
  - Resolving variants by either marking a selector field or specifying that only one case of the variant is relevant.
  - Adding, deleting, hiding, or renaming fields.
  - Changing a data type.
  - Setting a field size and scale.
  - Setting a field as nullable.
  - Changing the order of the fields.
  - Selecting a counter field for fields with dimensions (arrays). You can select the counter for the array from a list of potential fields.
  - Setting column wise normalization for fields with dimensions (arrays). You can create new fields instead of the array field where the number of generated fields are determined by the array dimension.
  - Creating arrays and setting the array dimensions.

The **Validation** tab at the lower area of the screen displays information about what must be resolved to validate the tables and fields generated from the COBOL. The Log tab displays a log of what has been performed (such as renaming a table or specifying a data location).

23. To manipulate table metadata, right-click the table record, and select the necessary operation. The following table lists the available options:

Table 2–2 Table Manipulation options

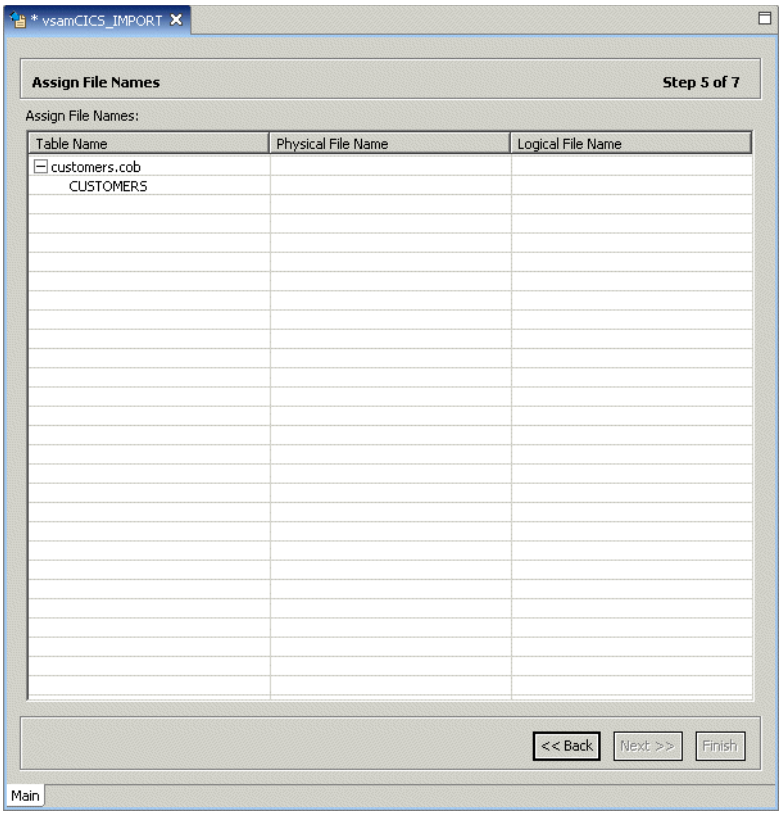
Option	Description
Fields manipulation	Customizing the fields definitions, using the Fields Manipulation screen. You can also access this screen by double-clicking the required table record.
Rename	Renaming a table. This option is used especially when multiple tables are generated from the COBOL with the same name.
Set data location	Setting the physical location of the data file for the table.
Set table attributes	Setting the table attributes.
XSL manipulation	Specifying an XSL transformation or JDOM document that is used to transform the table definitions.

24. Once all the validation error issues have been resolved, the Import Manipulation screen is displayed with no error indicators.

25. Click Next. The next step depends whether the VSAM Under CICS Import Manager or the VSAM Import Manager is used.

    If the VSAM Under CICS Import Manager is used, then the Assign File Names screen is displayed, as shown in the following figure:

Figure 2–14 The Assign File Names screen



In this screen you specify the physical file name, including the high-level qualifiers and the logical file name, for each record listed.

If the VSAM Import Manager is used, then the Create VSAM Indexes screen is displayed.

26. Click **Next** to retrieve index information.

If the VSAM Under CICS Import Manager is used and this step fails for any reason (such as the IBM z/OS platform is temporarily not accessible), a warning message is issued and you continue to the next step. Click **Continue** in the message box to continue with the next step.

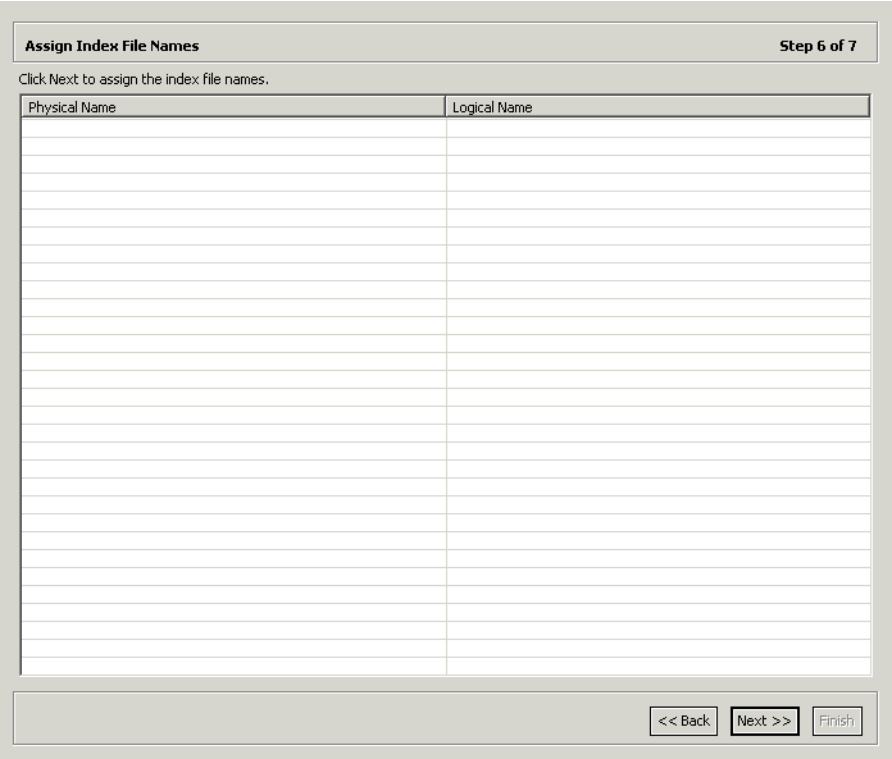
Note: If the indexes are not generated, then define manually in the Metadata tab of Oracle Studio Design perspective. For details, refer to "Metadata for VSAM" on page A-1.

---

27. The next step (assigning index file names) applies only if the VSAM under CICS Import Manager is used.

The Assign Index FIle Names screen is shown in the following figure:

Figure 2–15 The Assign Index File Names screen



28. For each table listed, specify the logical file name for the index.
29. Click **Next**.
30. For **Do you want to transfer the metadata to the server?** click **Yes** to transfer the metadata from the your computer to the IBM z/OS computer and click **Finish**.

The metadata is imported based on the options specified and it is stored on the IBM z/OS platform. An XML representation of the metadata is generated. This XML file can be viewed by expanding the Output node.

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

> **See Also:** "Metadata for VSAM" on page A-1 for details about the data source metadata.

2.3.3 Setting Up an Oracle Connect Adapter

To work with the Oracle Connect against the VSAM data source from an Oracle Application Server, you must set up an adapter definition on the IBM z/OS platform to handle the interactions to and from the VSAM data.

Perform the following steps to setup the 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 "Setting Up the IBM z/OS Platform in Oracle Studio" on page 2-1.
4. Expand the **Bindings**.

5. Expand the **NAV** binding.
6. Right-click **Adapters** and select **New Adapter** to open the New Adapter wizard.
7. Enter a name for the back-end adapter.

> **Note:** The word *event* is a reserved word and cannot be used when naming an adapter.

8. Select **Database** as the back-end adapter type from the `Type` list. The Database adapter enables accessing the VSAM data source from Oracle Application Server.
9. Select Events to create an event queue for the adapter.
10. Click **Finish**. The back-end 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.

11. Select the **Properties** tab and change any of the properties for the adapter, as required.

    The **Properties** tab is shown in the following figure:

Figure 2–16 The Adapter Properties tab

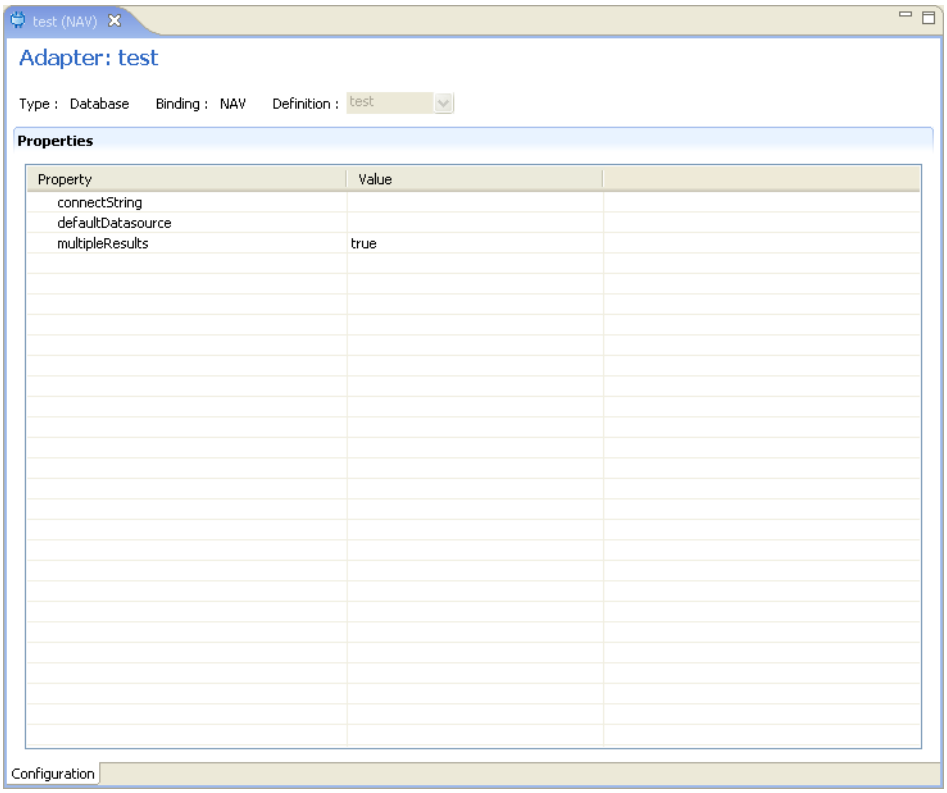


Table 2–3 Adapter Properties

Property	Description
connectString	Leave this value blank.
defaultDatasource	The name of the data source defined in Oracle Studio to access with the Database adapter. For example, `Legacy`.
multipleResults	Leave this value as `true`.

Note: You must specify the VSAM data source name you used to define in Oracle Studio for the defaultDatasoure property.

2.3.4 Generating Outbound Interactions

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

Use the Metadata Import wizard in Oracle Studio to generate interaction metadata, 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.
3. Expand the machine defined in "Setting Up the IBM z/OS Platform 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.
6. Expand the **Adapters** folder.
7. Right-click the Database back-end adapter defined in "Setting Up an Oracle Connect Adapter" on page 2-17.
8. Select **Show in Metadata View** to open the Metadata tab, with the database back-end adapter displayed under the adapters list.
9. Right-click the Interactions node and select **New** to open the New Interaction wizard. The wizard opens with the following options displayed:
   - Automatic: Four interactions are generated for each VSAM table, enabling you to run the `SELECT`, `INSERT`, `UPDATE`, and `DELETE` commands.
   - Manual: One interaction is generated, based on the type of SQL selected: Database Query (a `SELECT` statement) or Database Modification (an `INSERT`, `UPDATE` or `DELETE` statement).

   Note: VSAM does not support the stored procedure option.

10. Select how you want to generate interactions (Automatic or Manual).

    If you select Automatic generation, then perform the following steps:

    a. Click **Next**. The Select Tables screen is displayed, enabling you to add tables to access with the interaction from the VSAM data source.

b. Click **Add** to add tables. The data sources that have been defined and all the tables, for each data source, that have had metadata defined for them are displayed.

c. Select the tables to access with the interaction and click the right-pointing arrow to move the selected tables to the right-hand pane.

d. Click **Finish**. The selected tables are displayed.

e. Click **Finish**. Four interactions are generated for each table selected with the record structures to support the interactions and the responses from the VSAM data source.

f. Click **Yes** to complete the task. The interactions and the record structures that relate to the interactions are displayed in the Metadata tab.

If you select Manual generation, then perform the following steps:

a. Select the type of SQL (query or modification) for the interaction and click Next. The Interaction Name screen is displayed.

b. Enter a name for the interaction, and select **Create new query**.

Note: The option to use a previously. saved query is not applicable.

---

c. Click **Next**. The Define Interaction screen is displayed, enabling you to build the SQL query.

Note: If Database query was selected in step a, then the Define Interaction screen is displayed, enabling you to build a SELECT statement only, as indicated in the Query type field. If the Database Modification option was selected, then this field enables you to select the required SQL modification statement from a list (`INSERT`, `UPDATE`, or `DELETE`).

The `SELECT` or Modification query is built as follows:

- Selecting Tables: In the left-hand pane, expand the data source node where the required table resides and select the required table. Drag and drop it to the first available row in the Tables tab in the right-hand pane.
- Selecting Columns: Click the Columns tab in the right-hand pane. In the left-hand pane, expand the data source and the table containing the required column. Select the required column and drag and drop it into the Columns tab in the right-hand pane.
- Joining columns from different tables: When a column from a different table is selected, the Join Tables wizard is displayed. Expand the table node in the left-hand pane, select the required column and click the right-pointing arrow. Click Next to set the operator and logical parameters for each column/segment as required, and then click Finish to close the wizard.
- Adding conditions in a `WHERE` clause: Select the column you are setting the WHERE clause for, and drag and drop it into the Where tab in the right-hand pane. Set the operator, value and logical parameters as required.

- Grouping columns: Select the required columns and drag and drop them into the Group tab in the right-hand pane.
- Filtering results using a HAVING clause: The HAVING clause provides conditions for grouping columns. Select the required column and drag and drop it into the Having tab in the right-hand pane. Set the operator, value and logical parameters as required.
- Sorting results: Select the column whose results you want to sort and drag and drop it into the Sort tab in the right-hand pane. Set the sort order as required.

d. Click **Next**. The Interaction Parameters screen is displayed, enabling you to specify input parameters for the interaction. The following parameters are specified:

Table 2–4 Interaction Input Parameters

Parameter	Description
passThrough	Defines whether the query is passed directly to the back-end database for processing or processed by the Query Processor.
Reuse compiled query	Defines whether the query is saved in cache for reuse.
Encoding	Sets the encoding method used to return binary data in text format. You can select between the base 64 and the hexadecimal encoding methods.
Event	Defines whether the interaction mode is sync-send or sync-receive.
Fail on no rows return	Defines whether an error is returned in case no data is returned
Root element	Defines the root element name for records returned by the query, using the <root> \ <record> format.
Record element	Defines the record element name for records returned by the query, using the <root> \ <record> format.
Max. records	Sets the maximum number of record returned by the query.
Null string	Sets the string returned for a null value. If not specified, then the column is skipped.

e. Click **Next**. The Interaction Parameters screen is displayed, enabling you to specify parameters for the interaction. The following parameters are specified:

Table 2–5 Interaction Parameters

Parameter	Description
Name	The name of the parameter.
Type	The type of parameter (such as string, number, binary).
Nullable	The nullable value (true or false).
Default	The default value for the parameter.
Context Field	This field is not applicable.
Bind to Sqls	This field is not applicable.

f. Click **Finish** to generate the interaction, including the record schema required to support the interaction input and output.

See Also: "Adapter Metadata" on page A-13 for details about the data source metadata

2.4 Viewing the XML Schema

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 F, "Editing XML Files in Oracle Studio".

2.5 Creating XML Schemas

The XML schema describing the adapter interactions and the input and output records for these interactions is created automatically during the import procedure, as described in "Generating Outbound Interactions" on page 2-19.

3

Integrating OracleAS Adapters for VSAM with WebLogic

To deploy and integrate OracleAS Adapter for VSAM with Oracle Application Server Containers for J2EE (WebLogic), you must configure the J2CA 1.5 VSAM adapter.

This chapter includes the following topics:

- Overview of Integrating OracleAS Adapter for VSAM with WebLogic
- Configuring the VSAM Adapter

3.1 Overview of Integrating OracleAS Adapter for VSAM with WebLogic

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

J2CA defines standard Java interfaces for simplifying the integration of applications with the EIS. The Oracle adapters are deployed as a resource adapter 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. Security management provides authentication, authorization, and secure communication between the J2EE server and the EIS.

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 (like security and transaction).

> **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*.

3.2 Configuring the VSAM Adapter

Deploy the Oracle Connect RAR into the WebLogic server.

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

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

Table 3–1 Connection Factory Properties

Property	Description
`location`	Specifies the JNDI location where Oracle Application Server should bind the connection factory instance for application components.
`eisName`	Sets the name of the adapter to use. The adapter is defined in the Oracle Connect server using Oracle Studio, as described in Securing Access to Oracle Connect on page 2-2.
`serverName`	Sets the TCP/IP address or host name where the Oracle Connect daemon is running. The daemon runs on the computer specified at the beginning of Setting Up the IBM z/OS Platform in Oracle Studio on page 2-1. For more information, see Appendix C, "Advanced Tuning of the Daemon" for details about the daemon.
`workspace`	Specifies to the name of a Oracle Connect server workspace to use. The default workspace is Navigator. For more informatin, see "Workspaces" for details about workspaces.
`portNumber`	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 "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 none or fixedNat (the Nat protocol using a fixed address for the daemon). The default is `none`.
`connectTimeout` (optional)	Specifies the connection timeout in seconds. The default is 0, meaning that there is no connection timeout.
`encryptionProtocol` (optional)	Specifies the name of encryption protocol to use. The default is `null`. The RC4 protocol is supported.

Table 3–1 (Cont.) Connection Factory Properties

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` (optional)	When set to `true`, the XA APIs are internally converted to local transaction APIs. Always set to `true`.

3.3 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.

4

Integrating OracleAS Adapters for VSAM

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

1. Configuring OracleAS Adapter for VSAM
2. Integrating OracleAS Adapters for VSAM with WebLogic
3. Configuring the OracleAS Adapter for VSAM

See Also: *Oracle Application Server Adapter Concepts Guide*.

4.1 Configuring the OracleAS Adapter for VSAM

This section includes the following topics:

- Creating JCA Configuration Files
- Configuring the Binding for Outbound Applications

4.1.1 Creating JCA Configuration Files

Perform the following steps to create a binding file and a WSDL file for the VSAM database adapter, using Oracle Studio:

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 Setting Up the IBM z/OS Platform 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. Expand the **Adapters** folder and right-click the database adapter you defined in Setting Up an Oracle Connect Adapter on page 2-17 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. For more information on the binding file, see Configuring the Binding for Outbound Applications.

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

   The View Genereated 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 the dialog box.

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

4.1.2 Configuring the Binding for Outbound Applications

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 a binding file for outbound applications:

```
<?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.AttuInteractionSpec">
```

```
            <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.AttuInteractionSpec">
            <property name="FunctionName" value="sub"/>

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

5

Configuring an OracleAS Adapter for VSAM for Outbound Interactions

This chapter provides an example for creating an OracleAS adapter for VSAM. To work with an OracleAS adapter for VSAM, 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.

5.1 Creating Outbound Interactions with the OracleAS Adapter for VSAM

To work with the OracleAS Adapter for VSAM and create outbound interactions you must create a VSAM data source, import the metadata from VSAM, create a database adapter, then create interactions that are used to access the data.

5.1.1 Requirements

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

- Computer with Windows operating system
- Mainframe computer with z/OS operating system
- Oracle Connect 12.1.3 installed on the Mainframe computer
- Oracle Studio 12.1.3 installed on the Windows computer.

5.2 Adding the VSAM Hospital Data Source

To create an VSAM data source you must do the following:

- Prepare the System
- Set up Machine Access to Oracle Connect
- Add a VSAM Data Source
- Import the Metadata

5.2.1 Prepare the System

Make sure that the Hospital database is available under VSAM on the Mainframe computer.

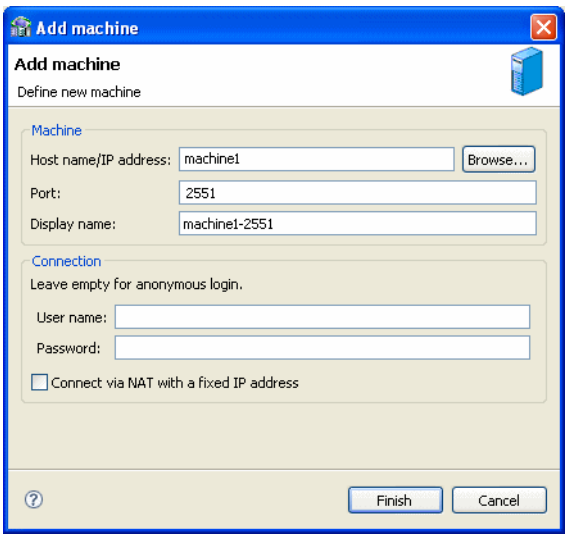
5.2.2 Set up Machine Access to Oracle Connect

You begin by configuring access to the Mainframe computer with your VSAM data. You configure this access with Oracle Connect. Open Oracle connect and follow the directions Setting Up the IBM z/OS Platform in Oracle Studio. Enter the following information in the Add machine dialog box:

- **Host name/IP address**: Enter the name or IP address for the Mainframe computer whith the Hospital database. 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 o f 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



5.2.3 Add a VSAM Data Source

After you add the computer configuration to Oracle Studio, you can add your VSAM data source to the Configuration view. To add a VSAM data source, do the following:

1. Expand the computer you just added and then expand the **NAV** binding.
2. Right-click the **Data sources** folder and then select **New data source**.
3. In the New data source dialog box, type **Hospital** in the **Name** field.
4. In the Type field, select **VSAM**, and then click **Next**.
5. Enter the following information in the Data source connect string page:

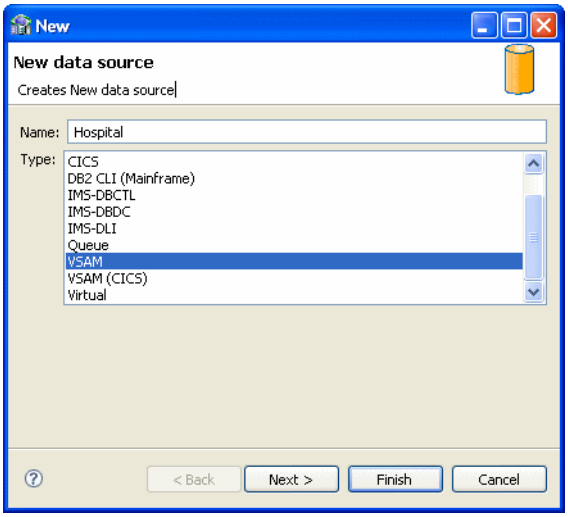
- **Data HLQ**:You can leave this blank so that the data files are written to the DEF high-level qualifier where Oracle Connect for VSAM is installed.
- **Disk Volume name**: The High Level Qualifier (HLQ) with the data.

6. Cick **Finish**.

For detailed information on adding theVSAM data source, see Setting Up the VSAM Data Source.

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

Figure 5–2 Add Data Source Dialog Box



5.2.4 Import the Metadata

You now import the metadata from the VSAM data into the data source. For information on how to import data, see Importing Metadata for the VSAM Data Source.

In this example you should import the metadata for the `Doctor` and `Patient` coluumns of the Hospital database. This database is supplied as an example with Oracle Connect and should be installed when you install Oracle Connect on your Mainframe computer.

You need the following information for the metadata import:

- In the **Get Input Files** step, add the files, `hospital.cob`.
- In the **Apply Filters** step, just click **Next**.
- In the **Select Tables** step, select DOCTOR and PATIENT, then click **Next**.
- In the **Import Manipulation** step, click **Next**.
- In the **Create VSAM Indexes** step, click **Next**.
- In the **Metadata Model Selection** step, select Default value for all tables, then click **Next**.
- In the **Import Metadata** step, select **Yes** and then click **Finish**.

5.3 Create the findDoctor and findPatient Interactions

You use a database 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 Creating the Project in JDeveloper Examples.

Do the following to create the interactions.

- Add a Database Adapter
- Create Interactions

5.3.1 Add a Database Adapter

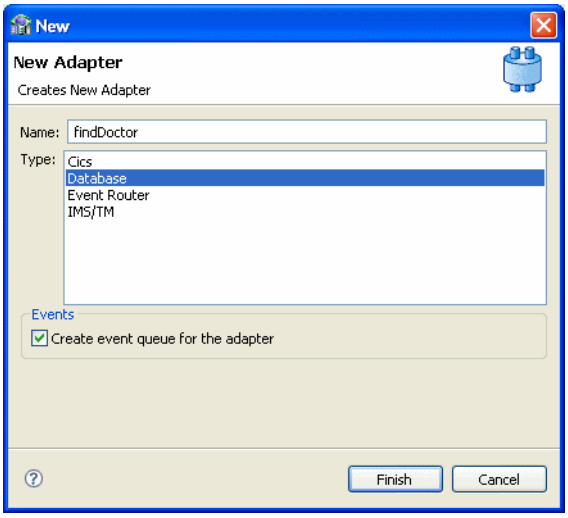
You now create a database adapter, which lets you create interactions to get the information you are seeking. To create a database adapter, do the following

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**: findDoctor
   - **Type**: Database
   - Select **Create event queue for the adapter**.

For a detailed explanation on how to set up a database adapter, see Setting Up an Oracle Connect Adapter.

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

Figure 5–3 Add Adapter Dialog Box



5.3.2 Create Interactions

You create interactions to work with the data under the Oracle Application Server. You use the interactions that you create when you create your endpoint using JDeveloper. In this example, you should create an interaction called `findDoctor`. To create an intearction, do the following:

1. Expand the computer you just added and then expand the **NAV** binding.
2. Right-click the adapter you are using and select Show in **Metadata**.
3. Right-click Interactions and select **New**.
4. Enter the following information for in the wizard, then click **Finish**.
   - **Name**: findDoctor
   - **Type**: Database
   - Select **Create event queue for the adapter**.
- In **Creation Mode**: select **Manual**, and **Database query**.
- Interaction Name: `findDoctor`.
- Select **Create New Query**
- Create a Select statement find a doctor in the database.

For detailed information on how to create an interaction, see Generating Outbound Interactions.

Save all of your infomation and exit Oracle Studio.

5.3.3 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 Creating JCA Configuration Files. To create the files for the `findDoctor` adapter, do the following:

1. From the Cofiguration pane in Oracle Studio, expand the NAV binding and the Adaper folder.
2. Right-click the findDoctor 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.

6

Creating the Project in JDeveloper Examples

This chapter contains the following topics:

- Getting Started
- Deploy a BPEL Outbound Process
- Deploy a Mediator Process

6.1 Getting Started

This topic describes the components necessary to work with the samples and how to prepare your system. It has the following topics:

- Prerequisites
- Beginning Tasks

6.1.1 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 VSAM adapter that you are using must be deployed to the Oracle Application Server.
- OracleAS CICS Adapter must be configured for outbound processing. See Configuring an OracleAS Adapter for VSAM for Outbound Interactions.
- Oracle JDeveloper must be installed.
- Legacy adapter must be deployed.
- Oracle Studio with the necessary machines loaded. See Setting Up the IBM z/OS Platform in Oracle Studio.

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

- How to create J2CA connections. See Setting Up an Oracle Connect Adapter for more information.
- How to configure the OracleAS adapters for VSAM using Oracle Studio. See Setting Up an Oracle Connect Adapter for more information.

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

- **Design Time**: The OracleAS adapters for CICS are configured with Oracle Studio as described in Setting Up an Oracle Connect Adapter.
- **Run time**: After you deploy the BPEL process with JDeveloper, you can test your configurations. See Run-time Configuration.

6.1.2 Beginning Tasks

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

- Perform the tasks in Configuring an OracleAS Adapter for VSAM for Outbound Interactions to create the CICS adapter, import the metadada, and create the interactions.
- Configure the connection for the J2CA CICS resource adapter. You do this by creating a connection factory.

  For more information, see Modeling Interactions for OracleAS Adapter for VSAM.
- Create the WSDL and binding files 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 Create the JCA Configuration Files for information on creating these files.

6.2 Deploy a BPEL Outbound Process

This section describes how to create a BPEL outbound process that connects to a OracleAS adapter for VSAM. For information on how to configure a VSAM adapter for outbound, see Modeling Interactions for OracleAS Adapter for VSAM.

To configure a process with the BPEL process manager, you use JDeveloper. This section describes how to create a BPEL outbound process for VSAM reource adapters using JDeveloper. The following sections describe how to deploy a BPEL outbound process. The following sections describe the steps necessary to perform this outbound sample.

- Design-Time Configuration
- Run-time Configuration

Before you design the outbound process, integrate and connect your OracleAS adapter for VSAM with the BPEL process manager.

6.2.1 Design-Time Configuration

This section describes the design-time steps necessary to deploy two BPEL outbound processes. The processes use the Web service called FINDDOCTOR. This service finds a list of doctors from VSAM data. The following are the steps required to create the outbound process.

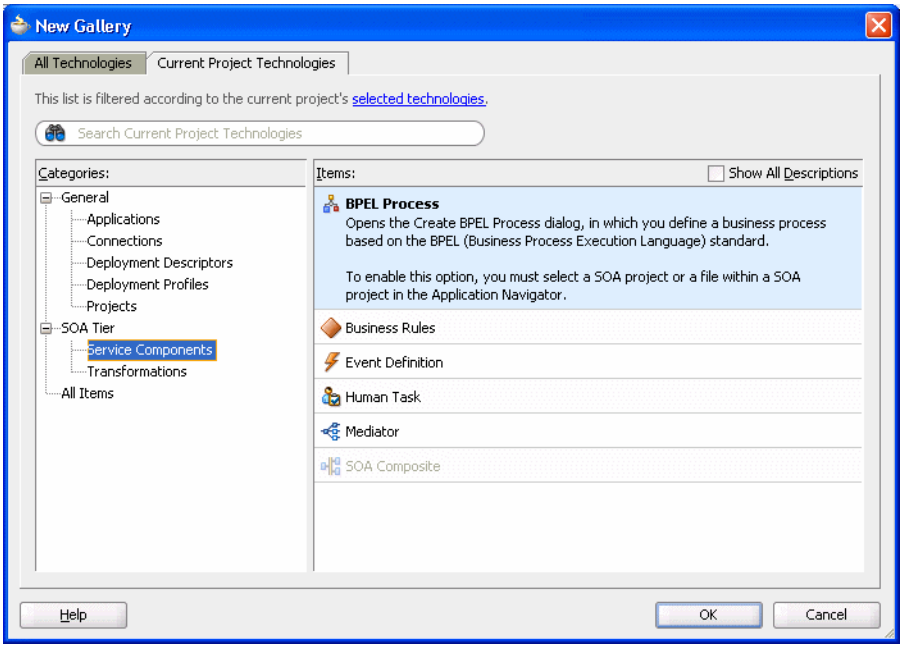
- Create a BPEL Project for a BPEL Outbound Process
- Use a Third-Party Adapter to Read the Doctor Data
- Design a BPEL Outbound Process
- Edit the Invoke_1 Activity
- Deploy the Project

6.2.1.1 Create a BPEL Project for a BPEL Outbound Process

Perform the following to create a BPEL project for outbound.

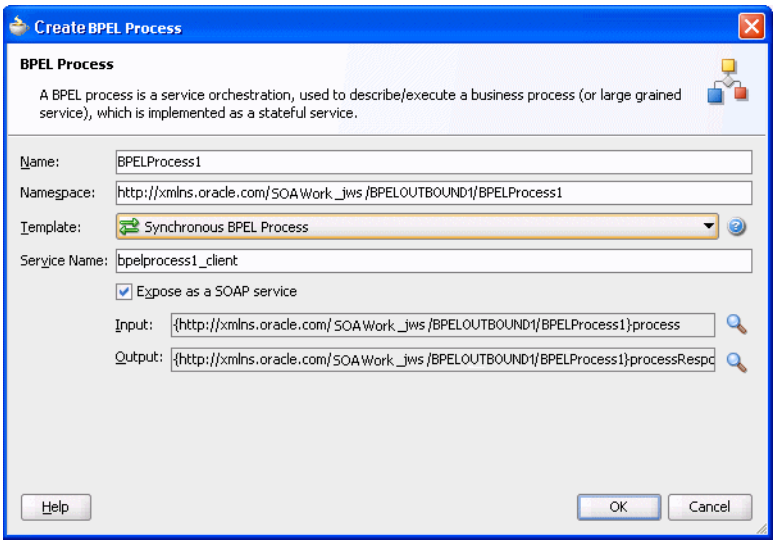
1. In JDeveloper, create a new SOA project.
2. Rright-click the SOA project and select **New**. The New Gallery is displayed as shown in Figure 6–1.

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** and click **OK**. The 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 example **BPELOUTBOUND1**.
- Enter the **Directory** where you want to save the project.
- Select a Synchronous BPEL Process **Template**.

6. Click **OK**. The BPEL Process is dislpayed in the visual editor.

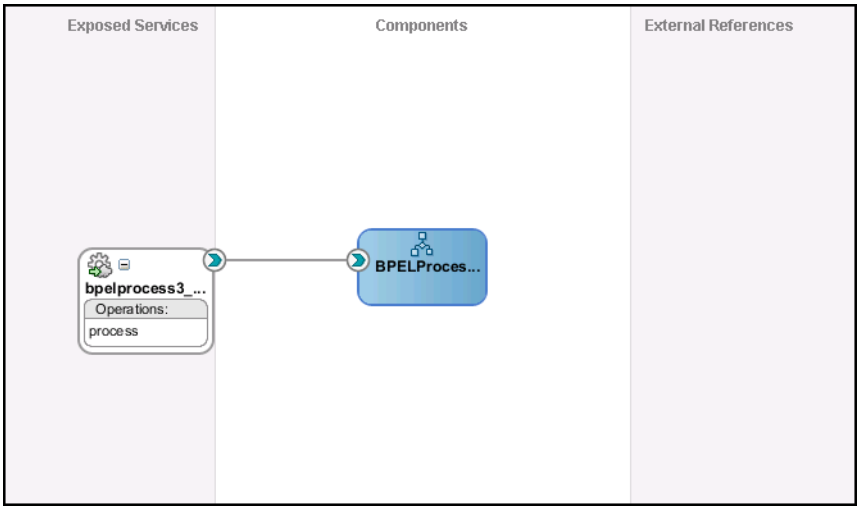
   Next Use a Third-Party Adapter to Read the Doctor Data.

6.2.1.2 Use a Third-Party Adapter to Read the Doctor Data

You must use a third-party adapter to read the DOCTOR data. SOA executes the SQL statment defined in the interaction and passes it to a resource adapter. The resource adapter then processes the information and returns the data. Perform the following to create the third-party adapter.

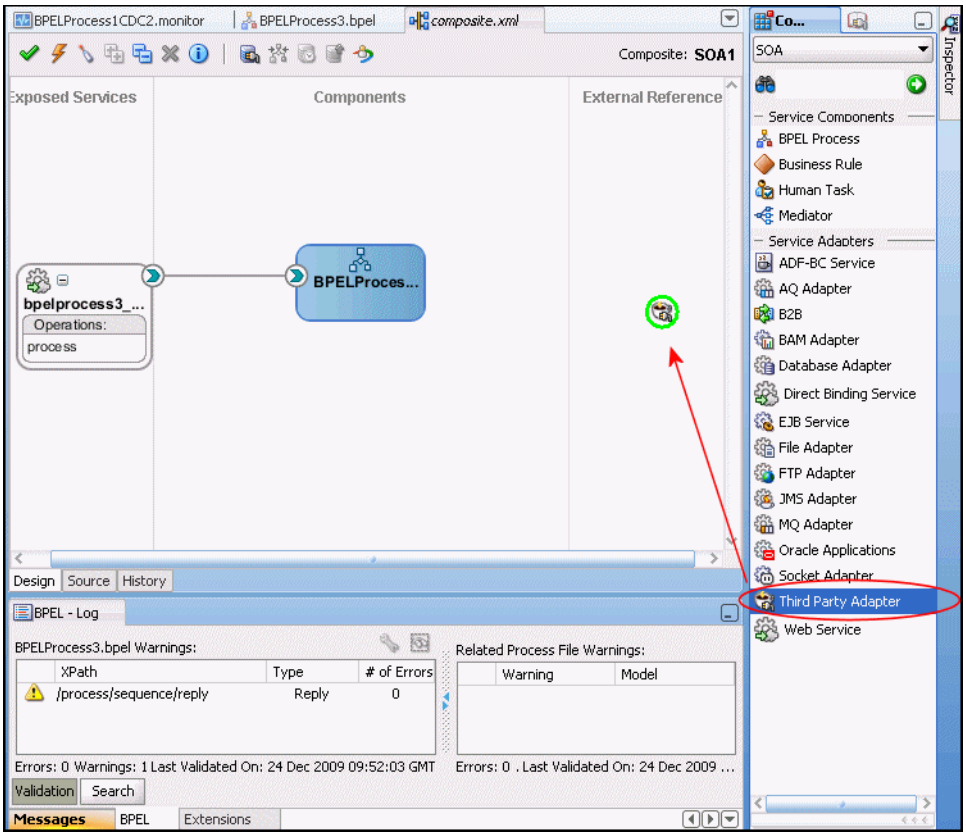
1. From the **Projects** area, on the left side, double-click the `composite.xml`. At the bottom of the visual editor, click the **Design** tab, if necessary. Figure 6–3 shows the Composite View in the visual editor:

Figure 6–3 Composite View



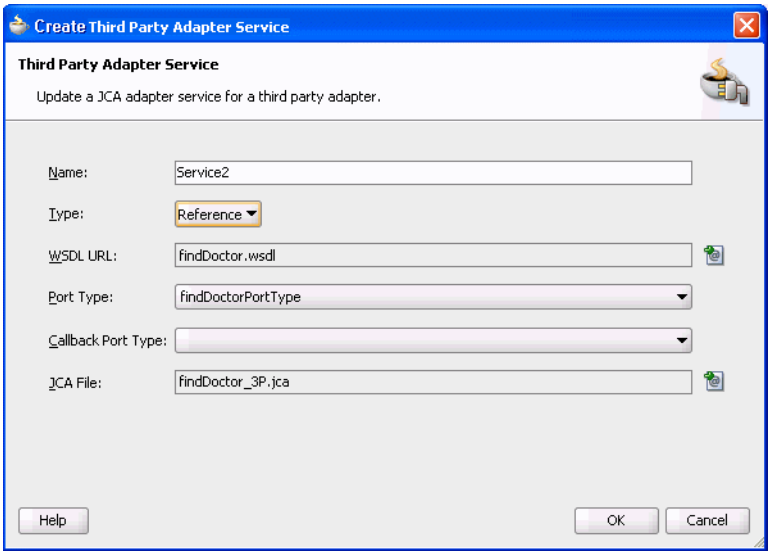
2. Drag a third-party adapter into **External References** lane of the visual editor, as shown in Figure 6–4.

Figure 6–4 Third-Party Adapter



When you drag the third-pary adapter into JDeveloper the Create Third Party Adapter Service screen is displayed as shown in figure Figure 6–5.

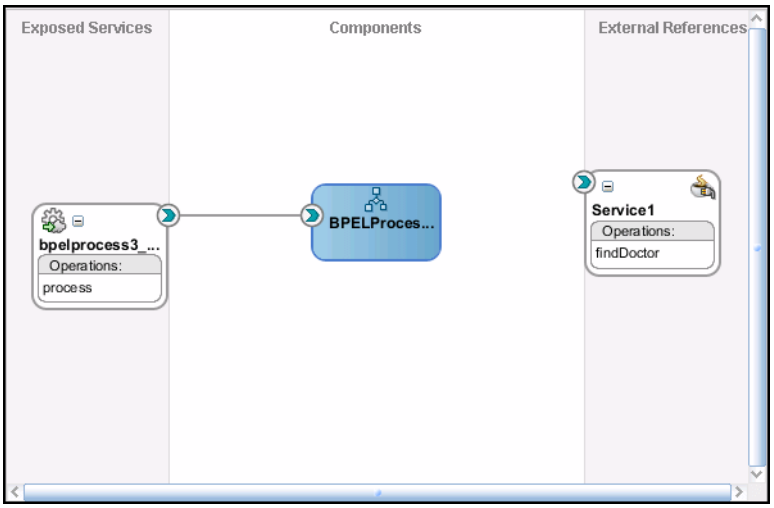
Figure 6–5 Create Third-Party Adapter Service



3. Click the button next to the **WSDL URL** field and browse to find the `findDoctor.wsdl` file that you created in Create the JCA Configuration Files.
4. In the **Type** list, select **Reference**.

5. Click the button next to the **JCA File** field and browse to find the `findDoctor.jca` file that you created in Create the JCA Configuration Files.
6. Click **OK** to close the dialog box. You should now have a composite design as shown in Figure 6–6.

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



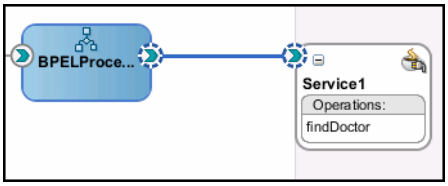
Continue with the Design a BPEL Outbound Process step.

6.2.1.3 Design a BPEL Outbound Process

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

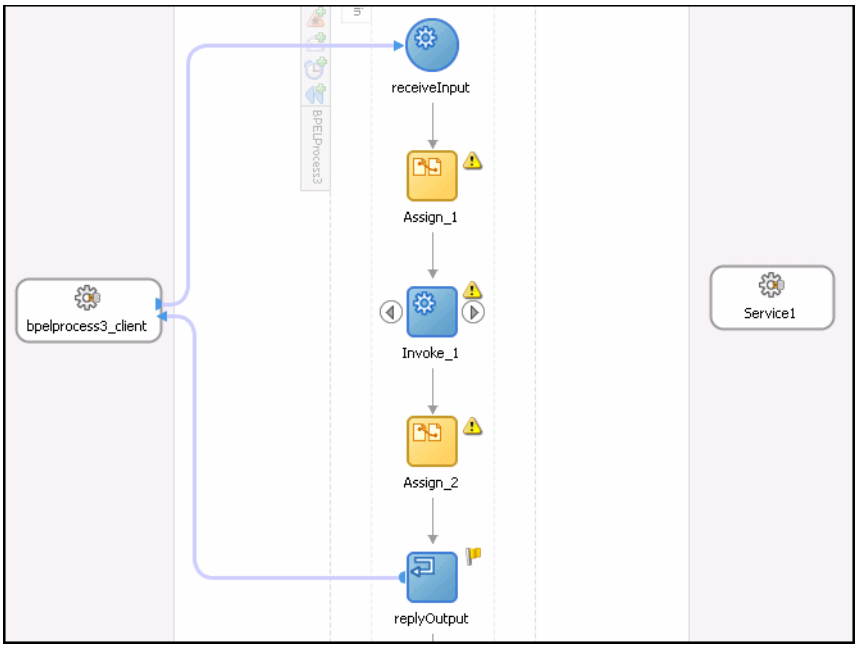
1. In the Composite, connect the BPEL process to the third-pary adapter as shown in Figure 6–7.

Figure 6–7 Connect to Third-Party Adapter



2. Double-click the BPEL process.
3. From the Component Palette, BPEL Process Activities section, drag an Assign, Invoke, and another Invoke activity into the editor for the process as shown in Figure 6–8.

Figure 6–8 Outbound Process



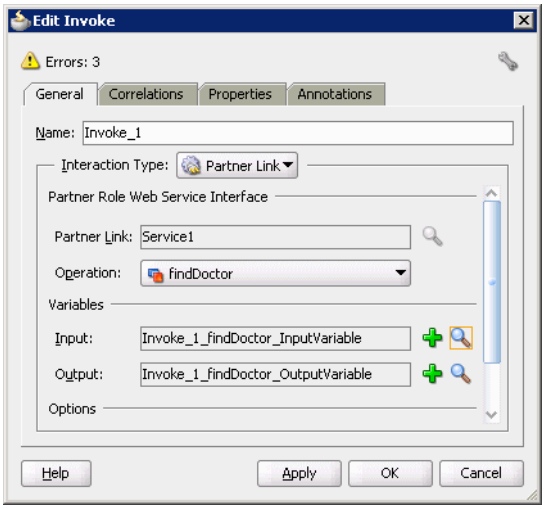
4. Continue with the Edit the Invoke_1 Activity.

6.2.1.4 Edit the Invoke_1 Activity

You must now edit the Invoke activity so that it invokes the `findDoctor` interaction.

1. In the Visual Editor, connect the `Invoke_1` activity to the third-party adapter. The Edit Invoke screen is displayed s shown in Figure 6–9.

Figure 6–9 Edit Invoke



2. In the Invoke dialog box, ensure that the Partner Role Web Service Interface section has the following values:
   - **Operation**: `FINDDOCTOR`
3. Click the plus sign next to the **Input** field and select the `Invoke_1_findDoctor_InputVariable`.

4. Click the plus sign next to the Output field and select the `Invooke_1_findDoctor_OutputVariable`.
5. Click **OK** then save then Deploy the Project.

6.2.1.5 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.

6.2.2 Run-time 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 `FindDoctor.xml` as shown in the following example:

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

<Record1>VSAM</Record1>
</CU:FindDoctorData>
```

2. Save the file to the `C:/temp` folder that was defined for the first adapter. The file is deleted after the SOA server reads the file and completes the process.
3. Open the `C:\tempOut` folder that was defined in the second adapter. A file with the following contents is copied to this folder.

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

6.3 Deploy a Mediator Process

This section describes how to create a Mediator outbound process that connects to a OracleAS adapter for CICS.

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

- Design-Time Tasks

6.3.1 Design-Time Tasks

This section describes the design-time steps necessary to deploy a Mediator outbound process. The processes use the Web service called FINDDOCTOR. This service finds a list of doctors from the data. The following are the steps required to create the outbound process.

- Create a Mediator Project for a Mediator Outbound Process
- Use a Third-Party Adapter to Read the Doctor Data
- Design an Outbound Mediator

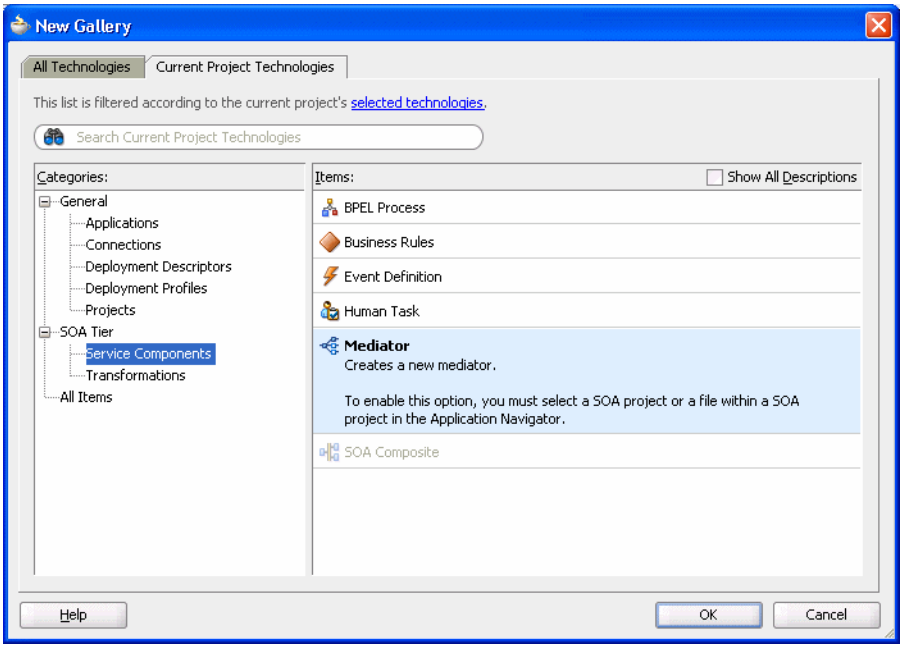
- Create Mapping Rules and Map the Files
- Deploy the Project

6.3.1.1 Create a Mediator Project for a Mediator Outbound Process

Do the following to create a Mediator project for outbound.

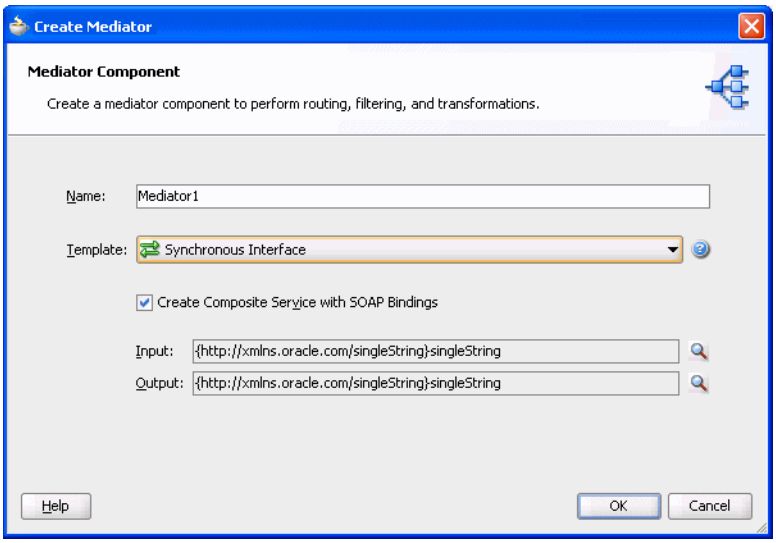
1. In JDeveloper, create a new SOA project.
2. Right-click the SOA project and select **New**. The New Gallery is displayed as shown in Figure 6–10.

Figure 6–10 New Gallery



3. In the **Categories** list, expand **SOA Tier** and select **Service Components**.
4. From the **Items** list, select **Mediator** and click **OK**. The Creator Mediator screen is displayed as shown in Figure 6–11.

Figure 6–11 Mediator



5. Enter the following in the Create Mediator screen:
   - Enter a **Name**, for example **MEDOUTBOUND1**.
   - Select a Synchronous Mediator **Template**.
6. Click **OK**. The Mediator is dislpayed in the visual editor.

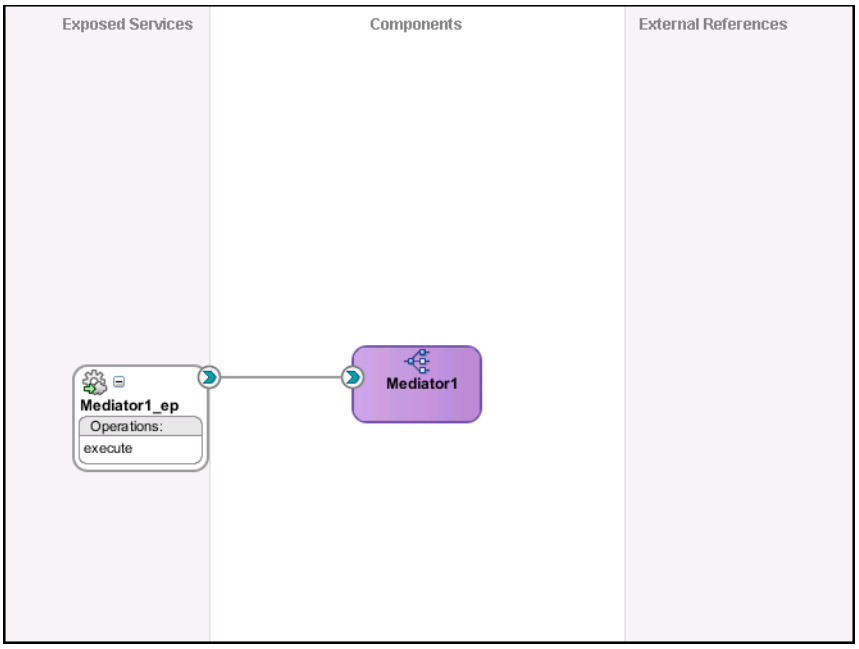
   Next Use a Third-Party Adapter to Read the Doctor Data.

6.3.1.2 Use a Third-Party Adapter to Read the Doctor Data

You must use a third-party adapter to read the DOCTOR data. SOA exectes the SQL statment defined in the interaction and passes it to a resource adapter. The resource adapter then processes the information and returns the data. Do the following to create the third-party adapter.

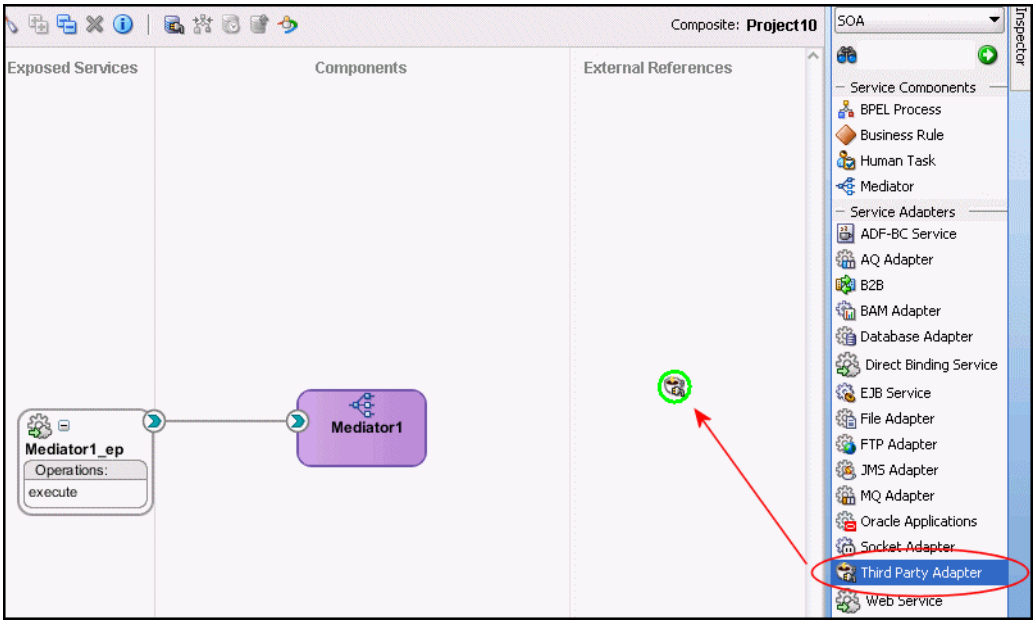
1. From the **Projects** area, on the left side, double-click the `composite.xml`. At the bottom of the visual editor, click the **Design** tab, if necessary. Figure 6–12 shows the Composite View in the visual editor:

Figure 6–12 Composite View



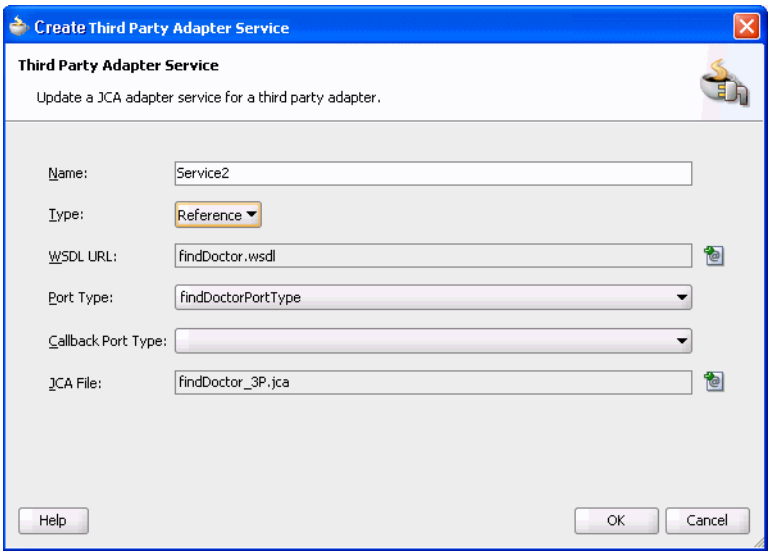
2. Drag a third-party adapter into **External References** lane of the visual editor, as shown in Figure 6–13.

Figure 6–13 Third-Party Adapter



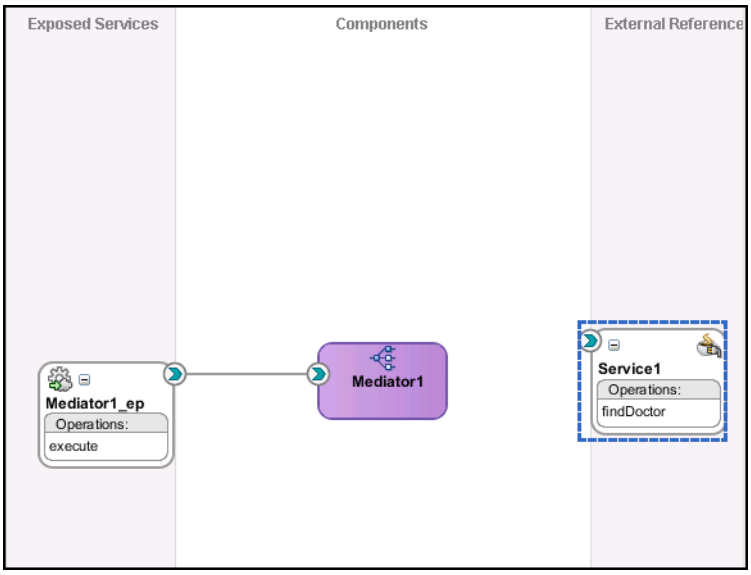
When you drag the third-pary adapter into JDeveloper the Create Third Party Adapter Service screen is displayed as shown in figure Figure 6–14.

Figure 6–14 Create Third-Party Adapter Service



3. Click the button next to the **WSDL URL** field and browse to find the `findDoctor.wsdl` file that you created in Create the JCA Configuration Files.
4. In the **Type** list, select **Reference**.
5. Click the button next to the **JCA File** field and browse to find the `findDoctor.jca` file that you created in Create the JCA Configuration Files.
6. Click **OK** to close the dialog box. You should now have a composite design as shown in Figure 6–15.

Figure 6–15 Ready to Create Mediator (Composite View)



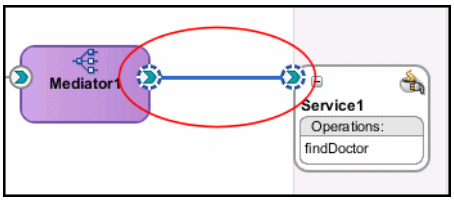
Continue with the Design an Outbound Mediator step.

6.3.1.3 Design an Outbound Mediator

In this step, you design the outbound Mediator. Perform the following to design the Mediator outbound process:

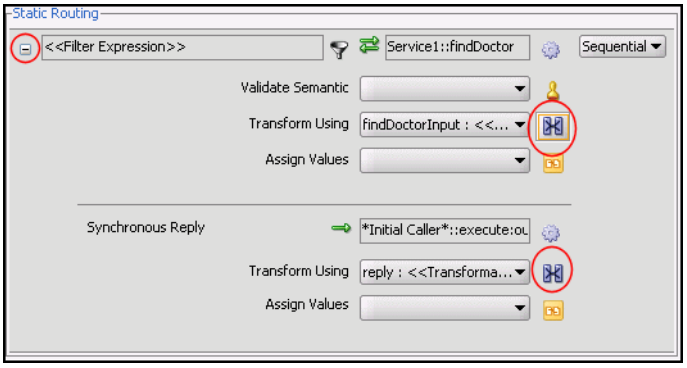
1. In the Composite, connect the Mediator to the third-party adapter as shown in Figure 6–16.

Figure 6–16 Connect to Third-Party Adapter



2. Double-click the Mediator process.
3. Expand the Filter Expression as shown in Figure 6–17.

Figure 6–17 Filter Expression



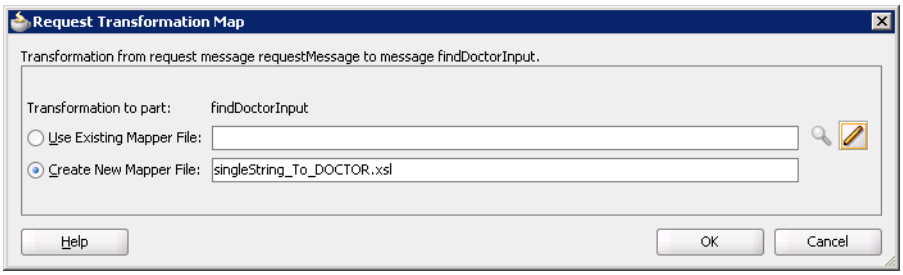
4. Continue with Create Mapping Rules and Map the Files.

6.3.1.4 Create Mapping Rules and Map the Files

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

1. In the upper section of the Filter Expression screen, click the icon next to the **Transform using** field as shown in Figure 6–17. The Request Transformation Map screen is displayed as shown in Figure 6–18.

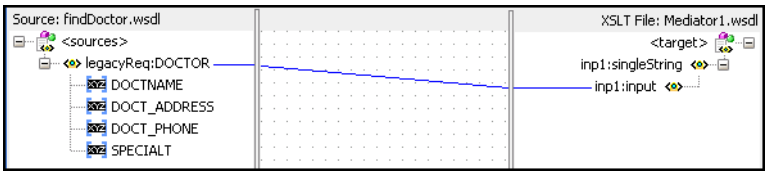
Figure 6–18 Request Transformation Map



2. In the Request Transformation Map, select **Create New Mapper** file. Use the default value, which should be `singleString_To_Doctor.xsl`.
3. Click **OK**.
4. In the visual editor, click to return to the **mediator.mplan** tab.

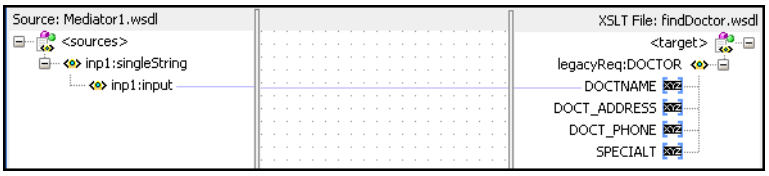
5. In the bottom section of the Filter Expression screen, click the icon next to the Transform using field as shown in Figure 6–17. The Request Transformation Map screen is displayed.
6. In the Request Transformation Map, select **Create New Mapper** file. Use the default value, which should be `Doctor_To_singleString`.xsl.
7. Click the `singleString_To_Doctor` tab. Connect the `Doctor` element fromthe `source:findDoctor.wsdl` to the `inp1:input` element in the `Mediatorx.wsdl` as shown in Figure 6–19.

Figure 6–19 Map Doctor to Single String



8. Click the `Doctor_To_singleString` tab. Connect the `inp1:input` element fromthe `source:Mediatorx.wsdl` to the `DOCTOR` attribute in the `findDoctor.wsdl` as shown in Figure 6–20.

Figure 6–20 Map Single String to Doctor



9. Save and then Deploy the Project.

6.3.1.5 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.

7

Troubleshooting OracleAS Adapter for VSAM

Troubleshooting Oracle AS Adapter for VSAM 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

7.1 Troubleshooting the Daemon

Troubleshooting the daemon and the communication between Oracle Application Server and OracleAS Adapter for VSAM 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.

> **See Also:** Appendix C, "Advanced Tuning of the Daemon" for details about the configuration settings.

This section contains the following topics:

- Starting the Daemon
- Shutting Down the Daemon
- Monitoring the Daemon During Run Time
- Daemon Logs

7.1.1 Starting the Daemon

The daemon is started when OracleAS Adapter for VSAM is installed. In case you have shut down the daemon, as described in "Shutting Down the Daemon" on page 7-2, you can restart the daemon as described in the following task.

Note: The daemon is started on the IBM z/OS platform. It cannot be started remotely using Oracle Studio.

7.1.1.1 Task: Starting the Daemon

Activate *INSTROOT*.USERLIB(ATTDAEMN) as a started task to invoke the daemon. For example, in the SDSF screen, enter the following command:

```
'/s ATTDAEMN'
```

Where *INSTROOT* is the high-level qualifier where Oracle Connect is installed.

> **See Also:** Starting the Daemon for details about the ATTDAEMN JCL

7.1.2 Shutting Down the Daemon

To shut down the daemon use Oracle Studio, as follows:

1. From the **Start** menu, select **Start**, **Programs**, **Oracle**, and then select **Studio**.
2. Select the computer defined in Setting Up the IBM z/OS Platform 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**.

7.1.3 Monitoring the Daemon During Run Time

Use the Runtime Manager perspective of Oracle Studio to monitor the daemon during run time.

Perform the following steps:

1. From the **Start** menu, select **Start**, **Programs**, **Oracle**, and then select **Studio**.
2. Right-click the computer defined in Setting Up the IBM z/OS Platform in Oracle Studio, and select **Open Runtime Perspective**.

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

7.1.3.1 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, which enables you to reconfigure the daemon. For more information, see Appendix C, "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.
Reload Configuration	Reloads the configuration after any changes. Servers currently started are not affected by the changed configuration. For more information, see Appendix C, "Advanced Tuning of the Daemon" for details about the configuration settings.

Table 7–1 (Cont.) Daemon Options

Option	Description
View Log	Displays the daemon log. For more information, see "Daemon Logs" on page 7-4.
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 user name and password needed to access 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.
Shutdown Daemon	Shuts down the daemon on the computer.
Rename	Enables changing the name of the daemon displayed in the Runtime Explorer.
Remove	Removes the daemon from the Runtime Explorer.
Refresh	Refreshes the display.

7.1.3.2 Workspace Options

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

The following table lists the available options:

Table 7–2 Workspace Options

Option	Description
Edit Workspace Configuration	Opens the daemon editor to enable you to reconfigure the workspace. For more information, see Appendix C, "Advanced Tuning of the Daemon" for details about the configuration settings.
Status	Checks the status of the workspace whether it is available or not.
Reload Configuration	Reloads the configuration of the workspace after any changes. Servers currently started are not affected by the changed configuration.
View Log	Displays the log for all servers for the workspace. For more information, see "Daemon Logs" on page 7-4
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.
Remove	Removes the selected workspace from the Runtime Explorer.

Table 7–2 (Cont.) Workspace Options

Option	Description
Disable	Disables the selected workspace.
Refresh	Refreshes the display.

7.1.3.3 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

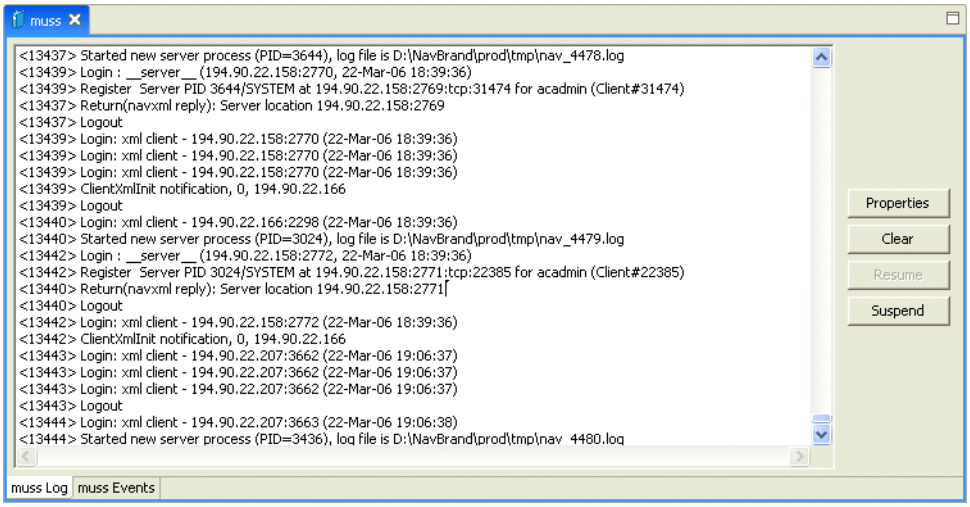
Option	Description
Status	Checks the status of the server. The information about the server includes the server mode and the number of active client sessions for the server.
View Log	Displays the server log. For more information, see "Daemon Logs" on page 7-4.
View Events	Displays the server events log.
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.

7.1.4 Daemon Logs

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

- Daemon
- Workspace
- Server process

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



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

7.1.4.1 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: "Logging" on page C-3.

7.1.4.2 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 daemon configuration.

See Also: "Logging" on page C-3.

7.1.4.3 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: "Logging" on page C-3.

7.2 Resolving Communication Errors

When Oracle Studio disconnects from the IBM z/OS 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 IBM z/OS computer.

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. Oracle Application Server Adapter for VSAM uses TPC/IP for internal intercomputer 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.

7.3 Resolving Specific Errors

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

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 IBM z/OS platform. 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 stop 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 stop 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 user name and password.

Action: Edit the user profile in Oracle Studio to specify a valid user name and password for the IBM z/OS platform.

See Also: Setting Up Run-Time User Access to the IBM z/OS Platform on page 2-5.

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 user name and password.

See Also: "Security" on page C-5.

C00F: Anonymous client logins are not allowed.

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

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

See Also: "Security" on page C-5.

Action: Edit the user profile in Oracle Studio to specify a valid user name and password for the IBM z/OS platform.

> **See Also:** Setting Up Run-Time User Access to the IBM z/OS Platform on page 2-5.

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 WS Info. tab of the daemon configuration.

> **See Also:** "General" on page C-7.

C012: Invalid username/password.

Cause: Invalid user name/password supplied when logging on to the daemon.

Action: See the daemon log file because the user name/password were not accepted.

Action: Edit the user profile in Oracle Studio to specify a valid user name and password for the IBM z/OS platform.

> **See Also:** Setting Up Run-Time User Access to the IBM z/OS Platform on page 2-5.

Action: Ensure that the daemon is started from an APF-authorized account that is allowed to check for system user names and passwords.

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 `Number of sub-tasks` in the WS Server section of the daemon configuration.

> **See Also:** "Server Mode" on page C-11.

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 logs 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.

To set impersonation: APF authorize all the steplibs in the server script on a z/OS computer. For example:

```
setprog... ada622-volume adavol
       CICS.CICS.SDFHEXCI - p390dx
       INSTROOT.load - 111111
       INSTROOT.loadaut - 111111
```

INSTROOT is the high level qualifier where Oracle Connect is installed.

In the **Security** tab of the Navigator workspace, under the daemon node in the Configuration Explorer, select **Use specific workspace account** and clear the **Workspace account** field of all values.

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 WS Info. section of the daemon configuration.

See Also: "General" on page C-7.

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 WS Server section of the daemon configuration.

See Also: "Server Mode" on page C-11.

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 reregister 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: "Logging" on page C-3.

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 z/OS platform. 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 WS Info. section of the daemon configuration.

See Also: "General" on page C-7.

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 IBM z/OS platform. 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 username given is not allowed to use the requested server. Use an authorized user name.

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.

C06B: Disconnecting from server

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

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 a server program failure caused the connection to end. The currently active transaction is also stopped.

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 computer should be fixed before connection can be resumed.

Action: In case of a server failure 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 WS Security sections in the daemon configuration.

> **See Also:** "Security" on page C-5 or workspace "Security" on page C-16.

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 setting are OK, 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 the nonkeep 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.

8

Advanced Features of OracleAS Adapter for VSAM

Oracle Connect includes several tuning parameters that can improve performance. Specifically, the daemon can be configured to optimize communication between the IBM z/OS platform 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

8.1 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 the daemon and the daemon workspace parameters to control the allocation of server processes and their management in a pool.

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

8.1.1 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.

Perform the following steps to add a new daemon workspace configuration:

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 computer defined in Setting Up the IBM z/OS Platform in Oracle Studio on page 2-1.
4. Expand the **Daemons** node. The daemon configurations available on this computer are listed.

5. Right-click **IRPCD** and select New Workspace. The New Daemon Workspace screen is displayed.
6. Specify a name for the new workspace and optionally add a description.
7. Specify whether you want it to have default settings or to 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 workspace is displayed under the IRPCD daemon node.

8.1.2 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**.

8.1.3 Configuring the Server Mode

The server mode dictates 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).

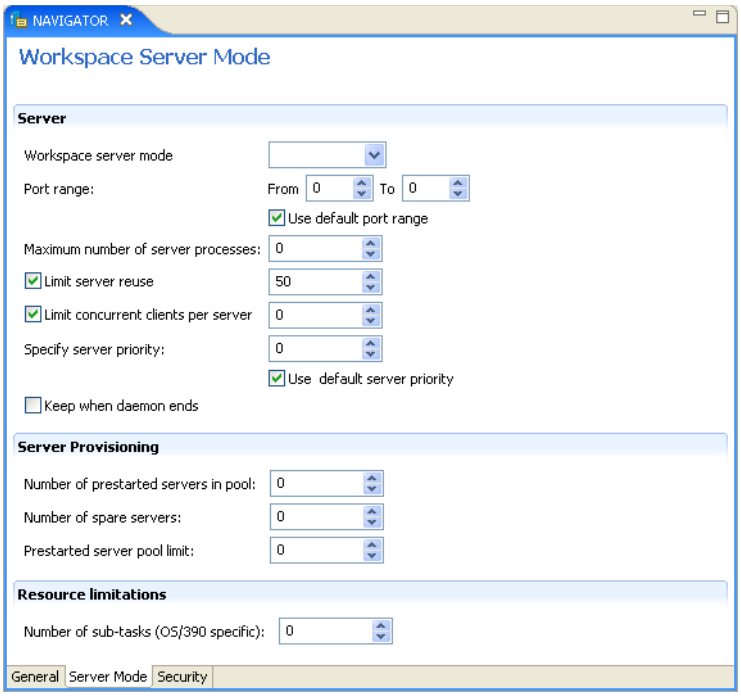
- **reusable**: This is an extension of the single client mode. Once the client processing finishes, the server process does not die and can be used by another client, reducing startup times and application startup overhead.

  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 value for the `Reuse limit` field 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 **Server Mode** tab.

- **Port Range**: Select the range for specific firewall ports through which you access the workspace. 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.

  Enter the port range in the following fields:

  - **From**: Enter the highest numbered port in the range
  - **To**: Enter the lowest numbered port in the range
- **Use Default Port Range**: Select this to use the port range that is defined in the daemon.
- **Maximum number of server processes**: Enter the maximum number of server processes that can run at the same time.
- **Limit server reuse**: Select this to limit the number of servers that can be reused. If this is selected, the **Reuse limit** parameter is available.

  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.

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.

This parameter is not available if the Limit server reuse parameter is not selected.

This parameter is not available if the server mode value is **singleClient**.

- **Limit Concurrent clients per server**: Select this to limit the number of clients that a server can accept for the current workspace process.

  If this is not selected, the number of clients is unlimited.

  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**.

- **Specify Server Priority**: Enter the server priority. 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`.

  Note: This is unavailable if **Use default server priority** is selected.

- **Use default server priority**: Select this to set 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.

- **Keep when daemon ends**: Select this to stop all servers started by that daemon when a daemon is shutdown, even if they are active. Select this if you want the servers for the workspace 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.

- **Number of prestarted servers in pool**: Enter 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.

- **Number of spare servers**: Enter 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 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.

- **Number of sub-tasks**: Enter 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).

8.2 Configuring a Binding Environment

Each binding configuration includes the following information:

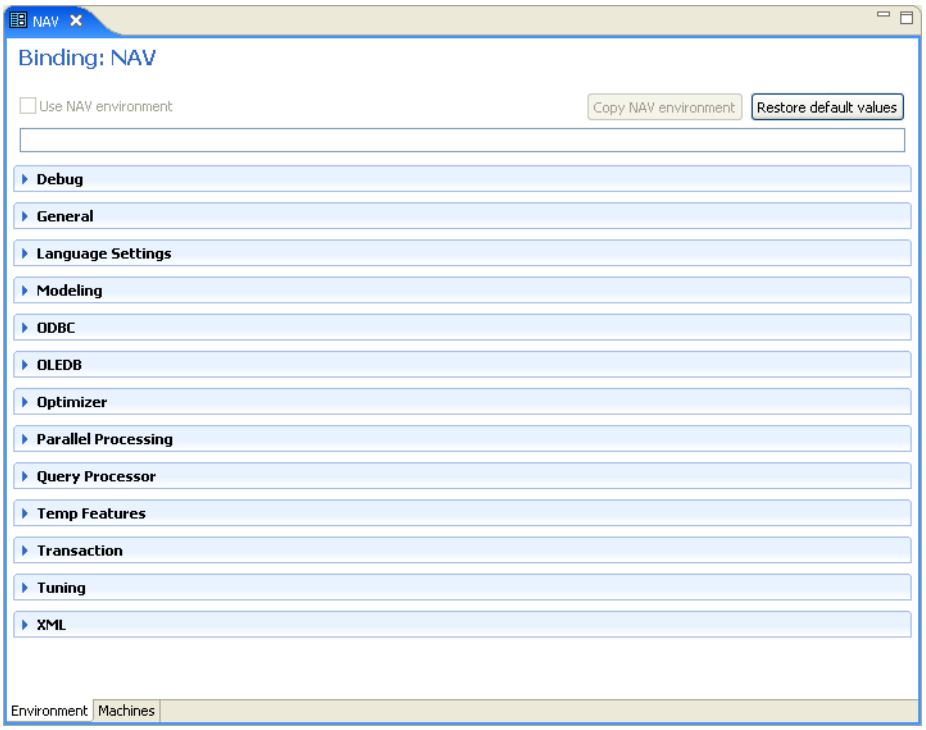
- Environment settings, which are used to configure the environment used by any of the adapters defined in the binding.
- Application adapters on the current computer.

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

1. From the **Start** menu, select **Programs**, **Oracle**, and then select **Studio**.
2. In the Configuration Explorer, expand the node of the required computer.
3. Expand the **Bindings** node. The binding configurations available on this computer are listed.
4. Right-click **NAV** and select Edit Binding.
5. In the Properties tab, edit the environment settings as needed. To edit an environment setting, expand the property category and select the value to edit.

   The binding Properties tab is shown in the following figure:

Figure 8–2 The Binding Properties tab



The binding environment is divided into the following categories:

- Debug
- General

- Language
- Modeling
- ODBC
- OLEDB
- Optimizer
- Parallel Processing
- Parallel Processing
- Query Processor
- Transaction
- Tuning
- XML

These categories are described in the following paragraphs.

8.2.1 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 VSAM.
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 VSAM.
Add timestamp to traced events	Select this to add a time stamp on each event row in the log.
Query Processor trace	This parameter is not applicable for use with OracleAS Adapter for VSAM.
Binary XML Log Level	Select the binary XML log level from the list. The following logging levels are available: ■ None ■ API ■ Info ■ Debug
Log file	The high-level qualifier of the log file for messages. The following type of message are written to the log: ■ Error messages ■ Trace information and information about the query optimization strategy, if `generalTrace` is set to `true`.

Table 8–2 (Cont.) Debug Parameters

Parameter	Description
Trace Directory	This parameter is not applicable for use with OracleAS Adapter for VSAM.
Optimizer trace	This parameter is not applicable for use with OracleAS Adapter for VSAM.
Transaction extended logging	Select this for the transaction manager to write additional information about transactions to the log.

8.2.2 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 VSAM.
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	Determines the way two-digit years are converted into four-digit years. When the `year2000Policy` 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: Fixed Base Year: `year2000Policy` is set to a value greater than, or equal to 1900. In this case, the value of `year2000Policy` is the first four-digit year after 1900 that can be represented by a two-digit year. For example, if `year2000Policy` is set to 1905, the years 2000->2004 are represented by 00->04. All other two digits are mapped to 19xx. 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. The user probably changes the base date only after ensuring that these old dates have been deleted from the data source. Sliding Base Year: `year2000Policy` is set to a positive value less than 100. In this case, the value of `year2000Policy` 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.
Cache buffer size	Enter the number of bytes to be used for a memory buffer on a client machine, which is used by the Oracle Connect client/server to store read-ahead data. The default is `200000`.

8.2.3 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 code page is selected based on the value specified for this parameter. For more information, see Appendix E, "Globalization Settings".
Code Page	For use with globalization support to identify the code page for the workspace. For more information, see Appendix E, "Globalization Settings".
NLS String	Specifies the code page used by a field whose data type is defined as `nlsString`. Use this for a field whose code page is other than that of the computer code page. This parameter includes the following values: ■ The name of the code page. ■ Whether the character set reads from right to left (as in middle eastern character sets). The default is `false`.

8.2.4 Modeling

The **Modeling** parameters are not applicale with OracleAS Adapter for VSAM.

8.2.5 ODBC

The **ODBC** parameters are not applicable for use with OracleAS Adapter for VSAM.

8.2.6 OLEDB

The **OLEDB** parameters are not applicable for use with OracleAS Adapter for VSAM.

8.2.7 Optimizer

The following parameters enable you to customize the performance:

Optimizer goal: The optimization policy. Select one of the following from the list to the right:

- **none** (default): All row optimization is used.
- **first**: First row optimization is performed based on the assumption that the results produced by the query are used as the rows are retrieved. The query optimizer uses a strategy that retrieves the first rows as fast as possible, which might result in a slower overall time to retrieve all the rows.
- **all**: Optimization is performed based on the assumption that the results produced by the query are used after all the rows have been retrieved. The query optimizer uses a strategy that retrieves all the rows as fast as possible, which might result in a slower time to retrieve the first few rows.

Aggregate queries automatically use all row optimization, regardless of the value of this parameter.

All other optimizer parameters are not applicable for use with the OracleAS Adapter for VSAM.

8.2.8 Parallel Processing

The following list shows the parallel processing properties. The parallel processing properties control how parallel processes are handled in the binding.

Table 8–5 Parallel Processing Parameters

Parameter	Description
Disable threads	Select this to disable multi-threading. If this is selected, the following properties are disabled: ■ **Disable threaded read ahead (QP)**: Select this to disable read-ahead functionality. ■ **Disable query read ahead (QP)**: Select this to disable read-ahead functionality for components using Query Processor services. ■ **ODBC async execution** Select this property to enable ODBC asynchronous execution ■ **Disable QP parallel execution**: Select this to disable parallel processing for query execution. This option is available only if both **Disable threaded read ahead (QP)** and **Disable query ready ahead (QP)** are *not* selected. ■ **Hash parallelism**: Select this to read both sides of hash joins at the same time. By default, this property is selected. If you do not want this behavior, clear the check box.

8.2.9 Query Processor

The following table lists the parameters that enable you to fine tune how queries are processed:

Table 8–6 Query Processor Parameters

Parameter	Description
Disable command reuse	Select this to disable Query Processor caching the executed state of a query for reuse.
Disable DS property cache	Select this to disable caching data source properties.
Disable insert parameterization	Select this to disable parameterization constants in INSERT statements.
Disable metadata caching	Select this to disable caching object metadata. If this is selected, the object metadata is taken from the from the original data source instead of the cache.
Disable query parametarization	Select this to not convert constants into parameters when accessing data sources.
Disable row mark field fetch	Select this for OLE DB `getRows` errors to be marked and reshown on every `getRows`, if the rowset is active.
Compile after load	Select this to compile an Oracle Connect procedure or view after it is read.

Table 8–6 (Cont.) Query Processor Parameters

Parameter	Description
Ignore segments bind failure	This property determines how Oracle Connect responds when the execution of one segment of a segmented data source fails: ■ Select this to Log a message and continue execution. This is the default setting. ■ Clear the check box to Log a message and stop execution. By default, this property is selected. If you want to stop execution after sending a message, clear this check box.
Prompt database-user password	Select this to configure Oracle Connect to prompt the user for security information when accessing a data source.
Use alternate qualifier	Select this to use the @ symbol instead of a colon (:) when connecting to multiple data sources.
Use table filter expression	Select this to enable the use of tables that have filter expressions specified in their metadata.
Write empty string as null	Select this to replace empty strings in a SET clause of an UPDATE statement or in a VALUES list of an INSERT statement with null values.
Optimistic for update	Select this to use optimistic locking as the default locking behavior on queries with a FOR UPDATE clause.
Disable compilation cache	Select this to disable saving successfully compiled statements in the cache.
Maximum SQL cache	Enter the maximum number of SQL queries that can be stored in cache memory. This property's value is ignored if **Disable compilation cache** is selected. The default is 3.
First tree extensions	Enter the maximum size allowed for an SQL query after compilation. The default is 150.
Maximum columns in parsing	Enter the maximum number of columns that a query can reference. The default is 500.
Maximum segmented database threads	Enter the maximum number of open threads allowed, when working with segmented databases.
Minimum number of parameters allocated	Enter the minimum number of parameters that can be used in a query.
Continuous query retry interval	Enter the number of seconds that the query processor waits before executing a query again, when no records are returned. The default is 2.
Continuous query timeout	Enter the number of seconds that the query processor continues to issue queries, when no records are returned. The default is 3600 (one hour), which indicates that after an hour without new messages the continuous query ends. Enter 0 to indicate that there is no timeout and the continuous query does not end automatically.
Continuous query prefix	Enter a prefix to replace the $$ prefix that is used to identify the continuous query special columns. For example, if you enter ##, then the continuous query alias is '##StreamPosition' and the control command alias is '##ControlCommand'.

Table 8–6 (Cont.) Query Processor Parameters

Parameter	Description
Arithmetic fixed precision	Enter an integer determine the precision scale factor for floating decimal position. The default is 0, which indicates that the exact arithmetic function is not used. When the value is set to a small positive integer, the special precise floating point arithmetic is used in the query processor. The value determines the precision scale factor (for example, a value of 2 indicates two digits decimal precision). Setting this parameter can be done at a workspace level and it affects all queries running in that workspace with no change to the query or to the underlying data source. The query processor ADD(), SUBTRACT() and SUM() functions that currently use double arithmetic for both floating and decimal types use this logic. When the value is set to the default, 0, the exact arithmetic function is not used. This property is used to set the Exact Arithmetic function. The `qpArithmeticFixedPrecision` property is an integer value that determines the fixed precision the Oracle connect query processor uses for precise floating point arithmetic. It is used to create an accurate result when using the SUM function. Because floating point data types are not accurate their results over time does not correspond to the expected arithmetic sum. In other words, in the floating point representation, values such as 0.7 cannot be represented precisely. If there are eight precision digits, there is usually imprecision in the least significant digit so the number is actually approximately 0.699999995. The`qpArithmeticFixedPrecision` property corrects this imprecision by using an exact floating point.
Parser depth	The maximum depth of the expression tree. The default is 500
Token size	Enter the maximum length of a string in an SQL query. The minimum value is 64. The default value is 350.
Insert from select commit rate	Enter the commit rate to use when executing an INSERT-FROM-SELECT operation. If a value more than 0 is entered, a commit is performed automatically after inserting the indicated number of rows. For example, if the value is 5,a commit is performed every time 5 rows are inserted.
Disable SQS cache	Select this to always read compiled Oracle Connect procedures and views from a disk. In this case, they are not saved in the cache.
Procedures cache size	Enter the number of stored queries created with a CREATE PROCEDURE statement that can be kept in cache memory. This property's value is ignored if **Disable SQS cache** size is selected.
Expose XML fields	Expose XML fields: Select this to display data returned for a query as XML, representing the true structure of the result. This is useful when querying a data source table that contains arrays or variants. ■ **XML field name**: Enter the name used in a query to indicate that the data is returned as XML, instead of the keyword XML. This is available only if **Expose XML fields** is selected.

8.2.10 Transaction

The following table lists the parameters that define transaction support:

Table 8–7 Transaction Parameters

Parameter	Description
Transaction extended logging	Select this to write extended information about transactions to the transaction manager log files.
Commit on destroy	Select this to commit all single-phase commit transactions opened for a data source, if a connection closes while the transaction is still open.
Disable 2PC	Select this to disable two-phase commit capabilities, even in drivers that support two phase commit.
User commit confirm table	This parameter is not applicable for use with OracleAS Adapter for VSAM.
Transaction log file	The high-level qualifier and name of the log file that logs activity when using transactions. The `logfile` parameter can also include the keyword `NORRS` after a comma (so that the format is `log,NORRS`) when RRS is not running on the z/OS platform.
Recovery Delay	The number of minutes from the start of a transaction before any recovery operation on that transaction can be attempted. The default is `15 minutes`.
Time limit	Enter the time to wait for a transaction to complete before an error is returned. This parameter is also used when performing a RECOVERY, and it then indicates the number of minutes to wait before a forced activity can be performed, since the last transaction activity.
Conversions Select one:	
	No conversion: Select this if you want all transactions to remain as sent. This is selected by default **Convert all to distributed**: Select this to convert all simple transactions into distributed transactions. **Convert all to simple**: Select this to convert all distributed transactions into simple transactions.

8.2.11 Tuning

The following table lists the parameters that define the tuning:

Table 8–8 Tuning Parameters

Parameter	Description
Dsm maximum buffer size	Enter the maximum size of a cache memory. This cache is used when memory is required on a temporary basis (as when Oracle Connect sorts data for a query output, for a subquery, or for aggregate queries). This cache size is not used for hash joins and lookup joins. The default value is `1000000` bytes. For more information, see the `hashBufferSize` parameter.

Table 8–8 (Cont.) Tuning Parameters

Parameter	Description
Dsm maximum Sort buffer size	Enter the maximum size of the sort buffers. Use this parameter instead of `dsmMaxBufferSize` for sorts only. The default value is `1000000` bytes.
Dsm middle buffer size	Enter the maximum size of the index cache. This cache is not used for hash joins and lookup joins. The default value is `100000` bytes.
File pool size	Enter the maximum number of files that can be opened in the file pool. The default is 10.
File pool size per file	Enter the size of the file in the pool. The default is 3.
File close on transaction	Select this if you want the File Pool to close when a transaction is committed.
Use global file pool	Select this to use a global file pool. When the workspace server mode parameter is set to multiClient or reusable, this parameter also indicates whether the file pool closes upon the client disconnection.
Hash buffer size	Enter the amount of cache memory that is available for each hash join or lookup join. The default is `1000000` bytes.
Hash max open files	Enter the maximum number of files that a query can open simultaneously for use when performing hash joins. The number assigned to this parameter must not exceed the system maximum. The default isset to `90` files. **Note**: The hash join optimization strategy results in multiple files being opened to perform the join. The larger the table size, the more files are opened. By adjusting this parameter you can disable hash joins on very large tables, while allowing hash joins for small tables.
Hash primary extent size	Enter the primary extent size.
Hash secondary extent size	Enter the secondary extent size.
Hash enable RO	Select this for the QP to store the first hash bucket in memory instead of a sequential file.

8.2.12 XML

The following table lists the parameters that define XML support:

Table 8–9 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.

Table 8–9 (Cont.) XML Category Parameters

Parameter	Description
XML date format	Enter the date format to use for XML. The options are: ■ `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.

8.3 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.

To migrate an adapter configuration using Oracle Studio, perform the following steps:

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 where the XML representation of the computer and its complete configuration is stored.
4. Edit any paths in the XML definition to the paths required on the target platform. For example, the setting for the serverLogFile parameter may require changing, depending on the platform.
5. Set up the target platform in Oracle Studio in the same way you set up the source platform, as described in Setting Up the IBM z/OS Platform in Oracle Studio on page 2-1.
6. In the Configuration Explorer, right-click the target computer and select Import XML definitions.
7. Import the XML file to the target platform.

8.4 Security Considerations

Oracle Connect works within the confines of the platform security system. For example, on an z/OS computer with RACF installed, and with the workspace server mode set to multi-tasking, a `RACROUTE VERIFY` is performed for each task in the address space, according to the client connection.

In addition, Oracle Connect provides the following security:

- 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, hardcoded key is used (this key is hardcoded 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 hardcoded key).

- Credentials: Passwords and user names exchanged over the network are encrypted using a pre-defined, hardcoded, 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.
- **Run time**: Security used to access VSAM, including controlling the daemon for the access.

8.4.1 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 on page 2-2.
- Securing rights to configure a computer in Oracle Studio is described in Specifying Users with Administrative Rights on page 2-3.
- 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.

8.4.2 Setting Run Time Security

During run time, 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 `userName` and `password` properties in the J2CA 1.5 VSAM 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 WS Security tab in Oracle Studio. For details on the Workspace **Security** tab, see "Security" on page C-16.

> **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 WS Security tab in Oracle Studio. For details of the Workspace **Security** tab, see to "Security" on page C-16.
- 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" on page C-5.

> **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 Workspace "Security" on page C-16.

8.5 Transaction Support

OracleAS Adapter for VSAM supports global transactions and can fully participate in a distributed transaction if you access the VSAM data under CICS.

> **Note:** To use OracleAS Adapter for VSAM under CICS with global transactions, you must have RRS installed and configured.

To work with global transactions, select the **Convert all to distributed** in the Transaction environment property section.

To use global transactions capability to access data on the z/OS computer, define every library in the ATTSRVR JCL as an APF-authorized library.

To define a DSN as APF-authorized, in the SDSF screen, enter the following command:

```
"/setprog apf,add,dsn=instroot.library,volume=ac002"
```

Where ac002 is the volume where you installed Oracle Connect and *INSTROOT* is the high level qualifier where Oracle Connect is installed.

If the Oracle Connect installation volume is managed by SMS, when defining APF-authorization, then enter the following command in the SDSF screen:

```
"/setprog apf,add,dsn=instroot.library,SMS"
```

Ensure that the library is APF-authorized, even after an IPL (restart) of the computer.

If RRS is not running, then OracleAS Adapter for VSAM can participate in a distributed transaction, as the only one-phase commit resource, if the Transaction environment property `logFile` includes the keyword `NORRS`.

Note: If a log file is not specified, then the format for the `logFile` parameter when RRS is not running is: `,NORRS`.

A

Advanced Tuning of the Metadata

Oracle Studio enables you to define outbound 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 by Oracle Studio in the **Metadata** tab of the Design perspective.

This appendix contains the following sections:

- Metadata for VSAM
- Adapter Metadata

A.1 Metadata for VSAM

Using Oracle Studio, perform the following steps to maintain the metadata for VSAM:

1. From the **Start** menu, select **Programs**, **Oracle**, and then select **Studio**.
2. Expand the **Machines** folder, then expand the machine with the metada you are working with.
3. Expand the **Bindings** folder.
4. Expand the **NAV** binding.
5. Expand the **Data sources** folder to display the data sources.
6. Right-click the VSAM data source and select **Show Metada View** to display the **Metadata** view.

   The **Metedata** tab is shown in the following figure:
7. Right-click the table you want to view in the Metadata Explorer and select **Edit**.

The metadata editor opens, displaying the **General** tab, with general table details. The following tabs are used to view and edit the metadata:

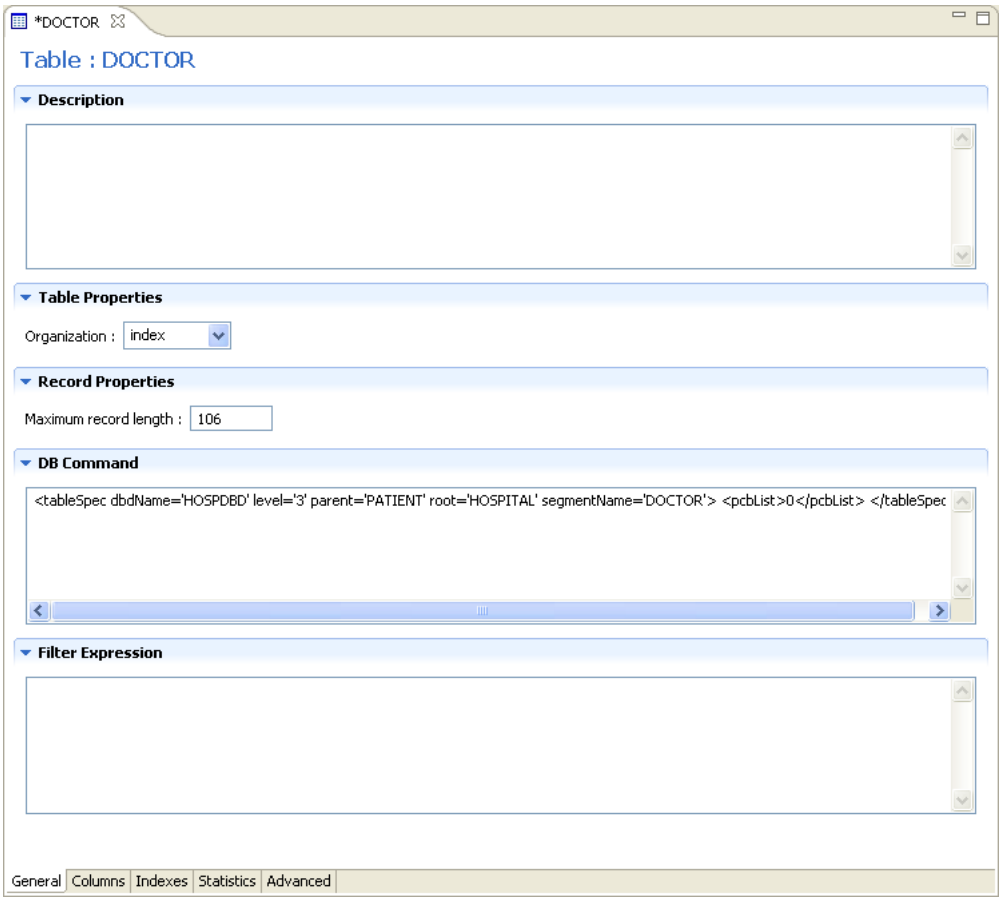
- General Tab
- Columns Tab
- Indexes Tab
- Statistics Tab
- Advanced Tab

A.1.1 General Tab

Use the **General** tab to maintain information about the whole table, such as the table name and the way the table is organized.

The **General** tab is shown in the following figure:

Figure A–1 The General tab



The **General** tab comprises fields, as listed in the following table:

Table A–1 General Tab Components

Field	Description
Description	An optional description of the table.
Table Properties	
Data file location	The name of the file that contains the table. You must enter the full path and include the file extension for the file. For example, `D:\COBOL\orders.cob`. You can click Browse and browse to find and enter the location of the table file. Note: Do not enter the file extension for DIASM or CIASM files.

Table A–1 (Cont.) General Tab Components

Field	Description
Organization	Select how the record represented by the table is organized. The options that are displayed depend on the record. The following options are available: ■ Index ■ Sequential ■ Relative: Used with RRDS files. Access to a specific record number of a relative file is performed by using a pseudo column to specify the record position. The hash symbol (#) is used to specify a pseudo column. For example: `SELECT * FROM colleges WHERE # = 6`
Record format	Specifies how the record, represented by the table, is formatted. The options that are displayed depend on the record. The VSAM records have a fixed format.
Maximum record length	The maximum allowable size of a record (in bytes).
DB Command	VSAM specific commands are displayed in the DB Command section of the **General** tab. The information displayed is generated automatically when the metadata is generated and should not be modified.
Filter Expression	A `WHERE` clause. This clause is added to every query accessed using this metadata. The filter is specified without the `WHERE` keyword. Specify a filter when multiple logical tables are stored in the same physical file.

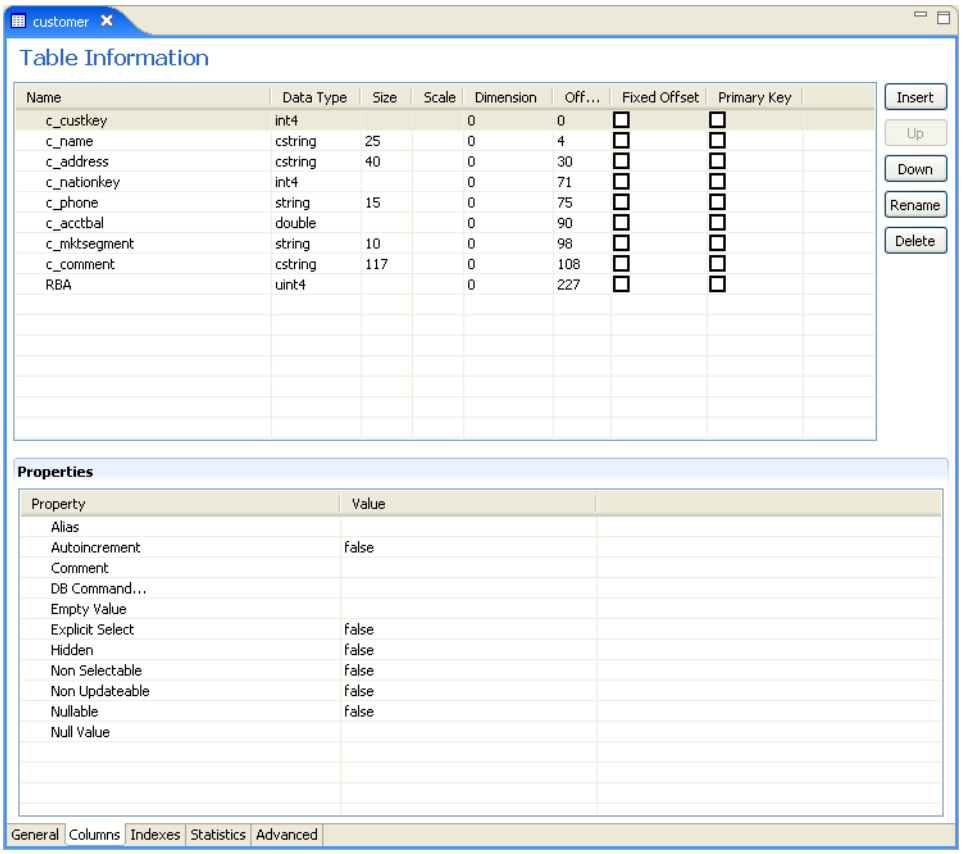
A.1.2 Columns Tab

Use the **Columns** tab to specify metadata that describe the table columns. The tab is divided into the following:

- Column Definition Section
- Column Properties

The **Columns** tab is shown in the following figure:

Figure A–2 The Columns tab



A.1.2.1 Column Definition Section

The top section of this tab lets you define the columns in the source data. You can click in any row (which represents a column in the data base table) to edit the information. The following table describes this section.

Table A–2 Metadata Column Tab Definitions

Field name	Description
Name	The name of the column
Data type	The data type of the column. Selecting this field displays a list of the possible data types.
Size	The size of the column
Scale	The information entered in this field depends on the data type: For decimal data types, this is the number of digits to the right of the decimal place. This number must not be greater than the number of digits. The default value is **0**. For scaled data types, this is the total number of digits. The number must be negative.

Table A–2 (Cont.) Metadata Column Tab Definitions

Field name	Description
Dimension	The maximum number of occurrences of a group of columns that are part of an array. The (+) to the left of a column indicates a group field. This type of field has a Dimension value. Click (+) to display the group members.
Offset	An absolute offset for the field in a record.
Fixed offset	This column lets you determine whether to calculate the offset. There are two options: ■ Calc offset: If you clear this check box, the absolute offset for each of the columns is calculated. ■ Fixed offset: When you select this check box, you have a fixed offset. The offset of a field is usually calculated dynamically by the server at run time according the offset and size of the proceeding column. Select the check box in this column to override this calculation and specify a fixed offset at design time. This can happen if you skip part of the buffer. By selecting the check box, or by editing the offset value you pin the offset for that column. The indicated value is used at run time for the column instead of a calculated value. Note that the offset of following columns that do not have a fixed offset are calculated from this fixed position.
Primary Key	Select this to indicate that this column is a primary key.

The buttons on the right side of the tab are used to manipulate the data in this section of the tab. The following table describes how you can move around in this section.

Table A–3 Definition Section Buttons

Button	Description
Insert	Inserts a column to the table. You can insert a new column. If the table has arrays, you can add a new child column.
Up	Moves your selection to the column directly above where the currently selected column.
Down	Moves your selection to the column directly below where the currently selected column
Rename	Lets you rename the selected column.
Delete	Deletes the selected column.
Find	Click this button to open a list of all columns in the database. Select a column and click **OK** to select it in the table.

A.1.2.2 Column Properties

You can change the property value by clicking in the **Value** column. To display the column properties, select a column from the Column Definition (top) section.

The properties for the column are displayed at the bottom of the tab.

The following table shows the properties available for selected columns.

Table A–4 Metadata Properties

Property	Description
Alias	A name used to replace the default virtual table name for an array. Virtual table names are created by adding the array name to the record name. When an array includes another array the name of the nested array is the name of the record and the parent array and the nested array. When the default generated virtual table name is too long, use an Alias to replace the long name.
Autoincrement	The current field is updated automatically by the data source during an `INSERT` statement and is not explicitly defined in the `INSERT` statement. The `INSERT` statement should include an explicit list of values. This attribute is used for fields such as an order number field whose value is incremental each time a new order is entered to the data source.
Comment	A short note or description about the column.
DB command	VSAM specific commands for the column The information displayed is generated automatically when the metadata is generated and should not be modified
Empty value	The value for the field in an insert operation, when a value is not specified.
Explicit Select	When `true`, the current field is not returned when you execute a `SELECT * FROM...` statement. To return this field, you must explicitly ask for it in a query, for example, `SELECT NATION_ID, SYSKEY FROM NATION` where `SYSKEY` is a field defined with Explicit Select. You cannot use an asterisk (*) in a query where you want to retrieve a field defined with the Explicit Select value.
Hidden	The current field is hidden from users. The field is not displayed when a `DESCRIBE` statement is executed on the table.
Non Selectable	When `true`, the current field is never returned when you execute an SQL statement. The field is displayed when a `DESCRIBE` statement is executed on the table.
Non Updateable	If `true`, the current field cannot be updated.
Nullable	This value allows the current field to contain `NULL` values.
Null value	The null value for the field during an insert operation, when a value is not specified.
Chapter of	This property shows that the set member field is a chapter of an owner field. A value for this property must be used when accessing a set member as a chapter in an ADO application. This property is used for DBMS metadata
OnBit	The position of the bit in a `BIT` field and the starting bit in a `BITS` field.
Subfield of	The value is generated automatically when you generate metadata from Adabas data that includes a superdescriptor based on a subfield. A field is created to base this index on, set to the offset specified as the value of the `Subfield start` field. If no value is entered in the `Subfield start` field, the subfield is set by default to an offset of 1.
Subfield start	The offset within the parent field where a subfield starts.

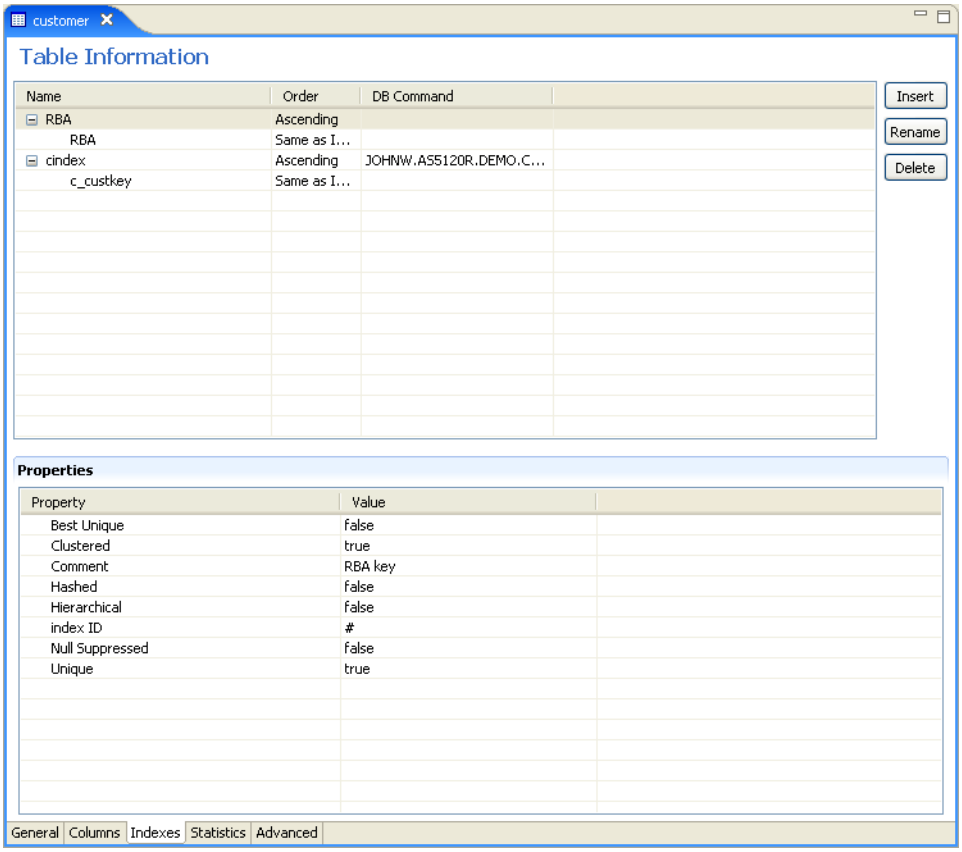
A.1.3 Indexes Tab

Use the **Indexes** tab to specify metadata describing the indexes of a table.

> **Note:** The Indexes tab contains information only if the **Organization** field in the Table tab is set to `Index`.

The **Indexes** tab is shown in the following figure:

Figure A–3 The Indexes tab



This tab has two sections. The first section lets you define the index keys for the columns in the table. The bottom of the tab lists the properties for each of the columns at the top.

The following sections describe the **Indexes** tab:

- Table Information
- Properties

A.1.3.1 Table Information

The following table describes the fields for the top part of the tab, which defines the indexes used for the table.

Table A–5 Indexes Tab Components

Field	Description
Name	The names of existing indexes for the current table.
Order	The ordering of the rows retrieved by the index.
DB Command	VSAM-specific commands for the index. The information displayed is generated automatically when the metadata is generated and should not be modified.

The buttons on the right side of the tab are used to manipulate the data in this section of the tab. The following table describes how you can move around in this section.

Table A–6 Index Definition Buttons

Button	Description
Insert	Inserts an index to the table.
Rename Index	Lets you rename the selected index.
Delete	Deletes the selected index.

A.1.3.2 Properties

You can index properties for each index column. To display the index properties, select a column from the Index Definitions (top) section.

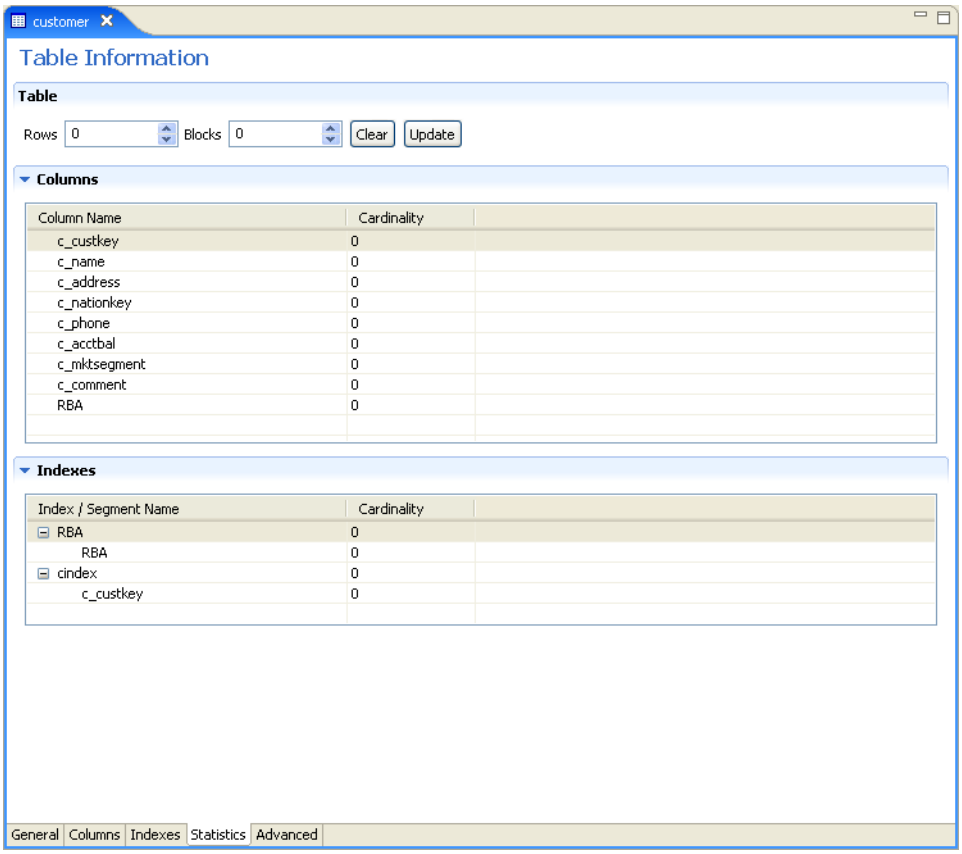
The properties for the column are displayed at the bottom of the tab.

This properties displayed at the bottom of the tab describe the index or segment. The properties available depend on the data source.

A.1.4 Statistics Tab

Use the **Statistics** tab to update statistics for a table. The **Statistics** tab is shown in the following figure:

Figure A–4 The Statistics tab



The **Statistics** tab comprises fields as listed in the following tables:

Table A–7 Statistics tab Components

Field	Description
No. of rows	The approximate number of rows in the table. If the value is `-1`, then the number of rows in the table is unknown (a value was not supplied and the update statistics utility was not run to update the value). A value of `0` indicates that this table is empty.
No. of blocks	The approximate number of blocks in the table. Note: If neither the number of rows nor the number of blocks is specified for a table, queries over the table might be executed in a nonoptimal manner.

Use the Columns group box to specify cardinality for each of the columns in the table:

Table A–8 Columns group Components

Field	Description
Column name	The columns in the table.

Table A–8 (Cont.) Columns group Components

Field	Description
Cardinality	The number of distinct values for the column. If the value is `-1`, then the number of distinct values for the column is unknown (a value was not supplied and the update statistics utility was not run to update the value). A value of `0` indicates that there are no distinct values for the column.

Use the Indexes group box to specify cardinality for the columns in each of the indexes in the table:

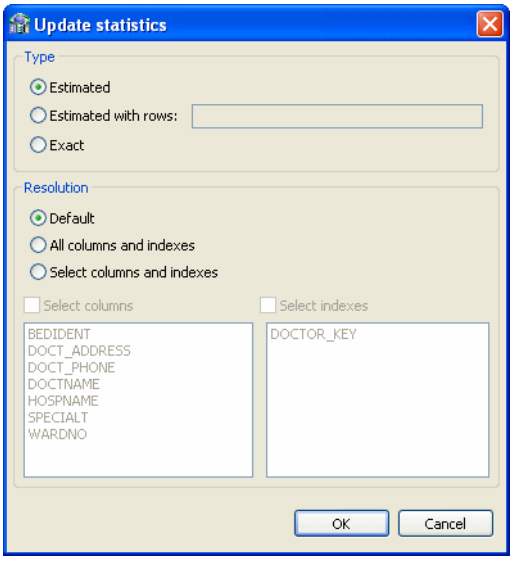
Table A–9 Indexes group Components

Field	Description
Indexes and segments	The indexes and segments in the table.
Cardinality	The number of distinct key values in the index. If the value is `-1`, then the number of distinct key values in the index is unknown (a value was not supplied and the update statistics utility was not run to update the value). A value of `0` indicates that there are no distinct key values in the index.

A.1.4.1 Generating Statistics

Click **Update** in the **Statistics** tab to generate updated statistics for the table. The Update Statistics screen is displayed, as shown in the following figure:

Figure A–5 The Update Statistics screen



The following tables list the fields that are used to update statistics:

Use the Type group to specify the following:

Table A–10 Type group Components

Field	Description
Estimated	An estimation of the amount of statistical information returned.
Estimated with rows	An estimation of the amount of statistical information returned. The estimate includes an estimation of the number of rows in the table. Specify the number in the text box. This number is used to shorten the time to produce the statistics if the value specified here is the correct value, or close to the correct value.
Exact	The exact statistical information returned. Note that this can be a lengthy task and can lead to disk space problems with large tables.

Use the Resolution group to specify the following:

Table A–11 Resolution group Components

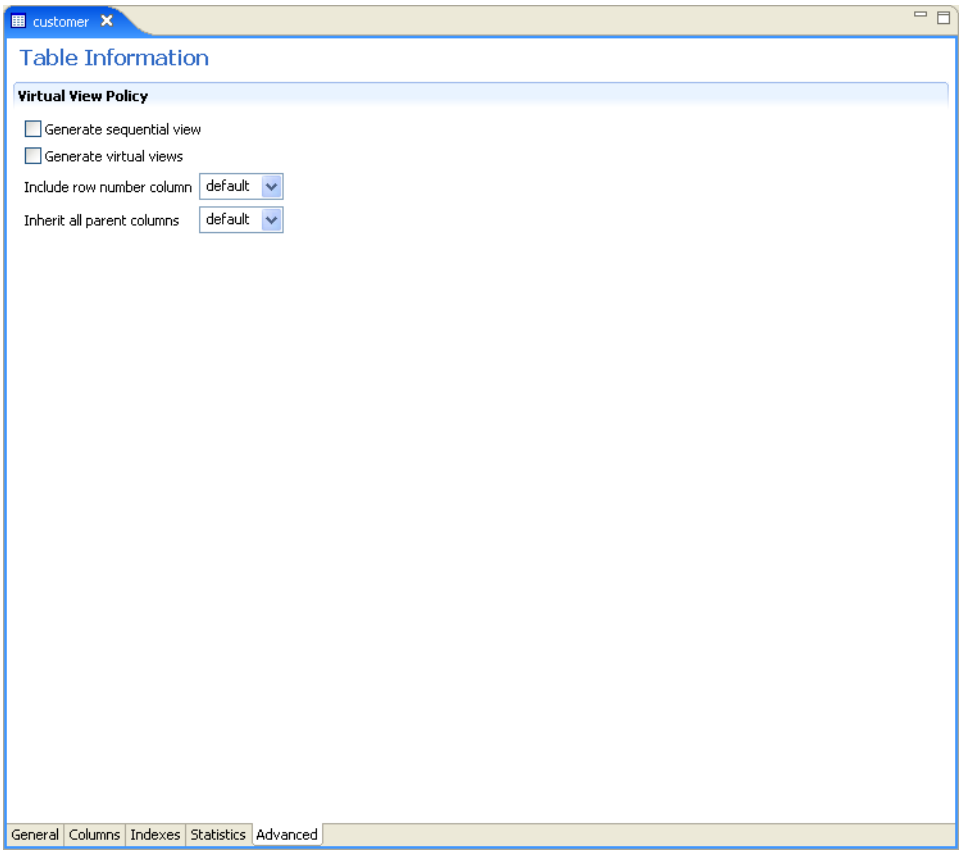
Field	Description
Default	Only information about the table and indexes is collected. Information for partial indexes and columns is not collected.
All columns and indexes	Information about the table, indexes, partial indexes and columns is collected.
Select columns and indexes	Enables you to select the columns and indexes for which you want to collect statistics. In the enabled list of columns or indexes, left click those columns you want included (you can use shift-click and control-click to select multiple columns or indexes).

A.1.5 Advanced Tab

The **Advanced** tab lets you enter information about the virtual view policy for arrays. These parameters are valid only if you are using virtual array views. The

configurations made in this editor are for the selected table, only. The same parameters are configured on the data source level in the data source editor.

Figure A–6 Data Source Metadata Advanced Tab



Enter the following information in this tab:

- **Generate sequential view**: Select this to map non-relational files to a single table.
- **Generate virtual views**: Select this to have individual tables created for each array in the non-relational file.
- **Include row number column**: Select one of the following:
  - **true**: Select **true**, to include a column that specifies the row number in the virtual or sequential view. This is true for this table only, even in the data source is not configured to include the row number column.
  - **false**: Select **false**, to not include a column that specifies the row number in the virtual or sequential view for this table even if the data source is configured to include the row number column.
  - **default**: Select **default** to use the default data source behavior for this parameter.
- **Inherit all parent columns**: Select one of the following:
  - **true**: Select **true**, for virtual views to include all the columns in the parent record. This is true for this table only, even in the data source is not configured to include all of the parent record columns.

- **false**: Select **false**, so virtual views do not include the columns in the parent record for this table even if the data source is configured to include all of the parent record columns.
- **default**: Select **default** to use the default data source behavior for this parameter.

A.2 Adapter Metadata

Use Oracle Studio to maintain the metadata for your database adapter.

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 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. The interaction Advanced tab is displayed for some adapters only, such as the Database adapter and includes more details about the interaction.
   - Adapter Metadata Schema Records: The input and output record structure for a record in the adapter definition.

A.2.1 Adapter Metadata General Properties

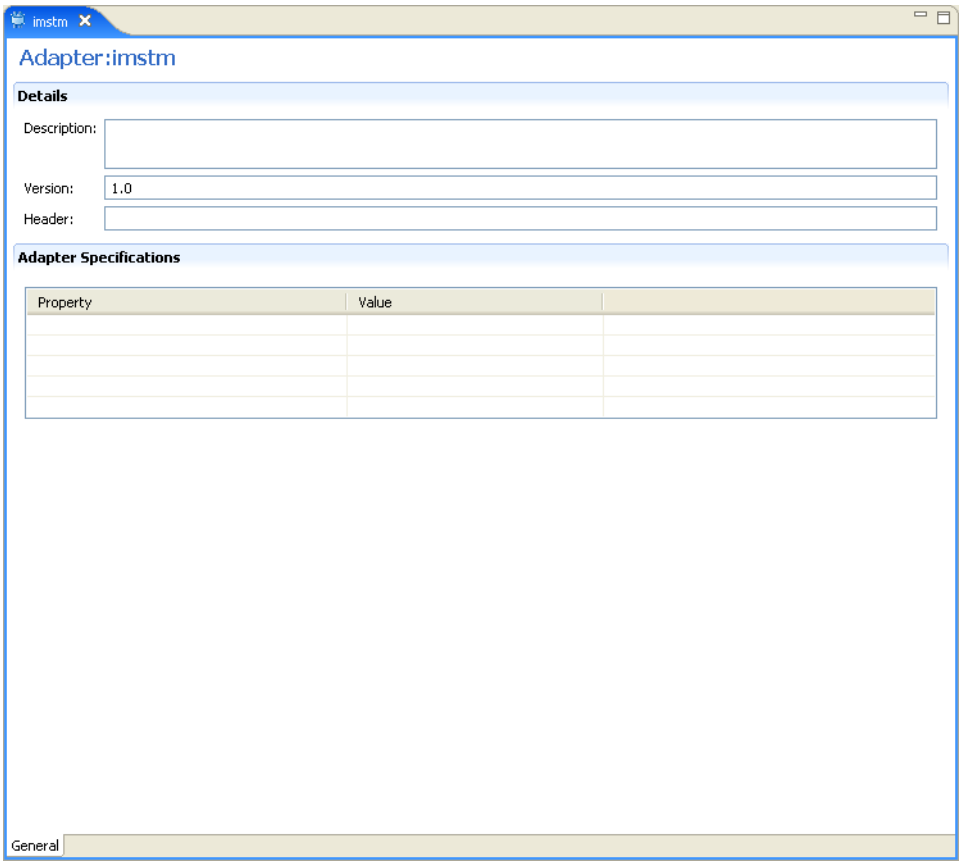
You can enter and edit information about the adapter, such as the adapter name and the way in which you connect to the adapter. You make these changes in the Design perspective, Metadata view. The following describes how to open the Adapter General Properties editor.

1. In Oracle Studio Design perspective, Metadata view, expand the **Adapters** folder.
2. Right-click the 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–7 Adapter Metadata General Properties



The **General** properties are listed in the following table:

Table A–12 General tab Components

Field	Description
Description	Specifies an identifying description of the adapter.
Authentication mechanism	Specifies the authentication to access the adapter. The available mechanisms are: ■ kerbv5 ■ none ■ basic password
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	Specifies the adapter-specific properties for an interaction. The VSAM database adapter does not have any adapter-specific properties.

A.2.2 Adapter Metadata Interactions

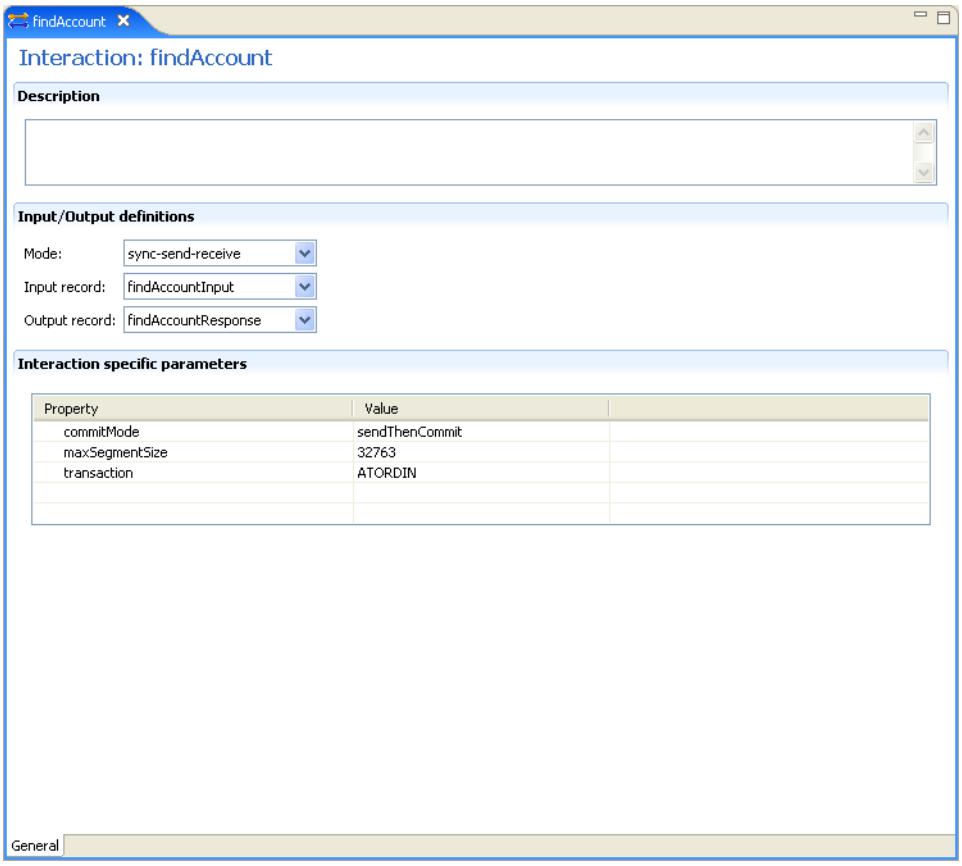
The Adapter Metedata Interactions editor defines an interaction and its input and output definitions. The following describes how to open the Adapter Metadata editor.

1. In Oracle Studio Design perspective, Metadata view, expand the **Adapters** folder.
2. Expand the 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 is shown in the following figure:

Figure A–8 Adapter Metadata Interactions



The Adapter Metadata Interaction properties are listed in the following table:

Table A–13 Interaction General Editor Components

Field	Description
Description	Provides a descriptive identifier for the interaction.

Table A–13 (Cont.) Interaction General Editor Components

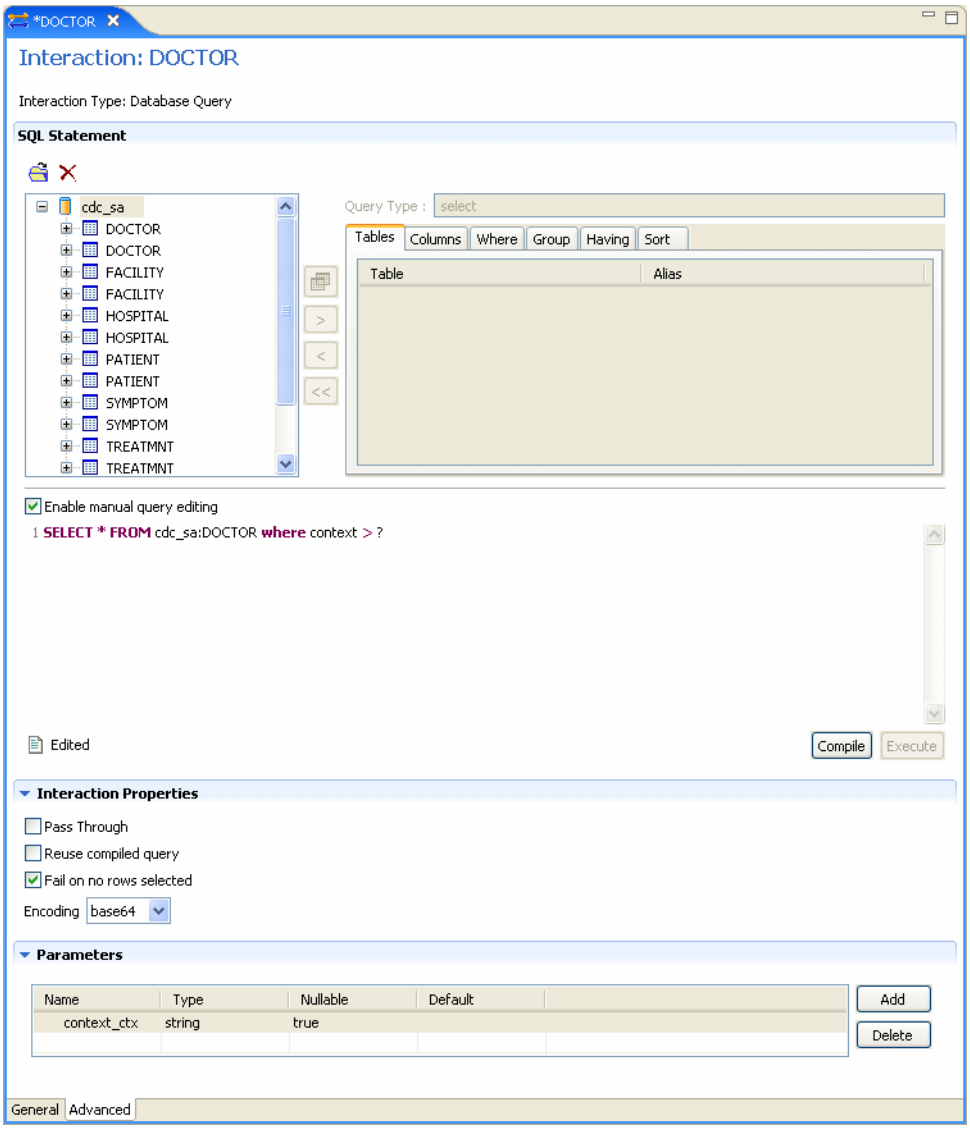
Field	Description
Mode	Determines the interaction mode. The following interaction modes are available: ■ sync-send-receive: The interaction sends a request and expects to receive a response. ■ sync-send: The interaction sends a request and does not expect to receive a response. ■ sync-receive: The interaction expects to receive a response.
Input record	Identifies an input record.
Output record	Identifies an output record for the results of an interaction.
Interaction Specific Parameters	Specific properties for the interaction. When an Interaction **Advanced** tab is used, this section is not displayed.

A.2.2.1 Interaction Advanced Tab

In the Adapter Metadata Interactions editor, click **Advanced** at the bottom of the editor to open the Interaction **Advanced** tab. Use this to enter advanced details for the interaction or to create interaction manually.

The following figure shows the Interactions **Advanced** tab.

Figure A–9 Interaction Advanced Tab



This tab has three sections.

The **SQL Statement** section lets you build any valid SQL statement. Use the tabs to select the tables and statement types and build a statement. When you select **Enable manual query editing**, you can manually enter a valid SQL statement at the bottom of the screen.

The **Interaction Properties** section lets you select any of the following:

- **Pass Through**: Select this to pass a query directly to the VSAM data.
- **Reuse compiled query**: Select this to save query objects created in the previous execution to the cache. This allows the objects to be reused.
- **Fail on no rows returned**: Select this if you want the system to return an error if no rows are selected.
- **Encoding**: Select one of the following from the list:
  - **base64**: Select this for base 64 encoding

  - **hex**: Select this for hexadecimal encoding

The **Parameters** section lets you create parameters to use for the interaction. To create a parameter, click **Add**. Enter a name in the dialog box and click **OK**. The parameter is entered in the list. You can edit the following properties for each parameter you create. Click in the corresponding cell to edit the properties.

- **Name**: The name of the parameter. This is automatically entered when you create a new parameter. You can click in the cell to change this parameter.
- **Type**: The type of parameter. Select one of the following types from the list:
  - **string**
  - **number**
  - **timestamp**
  - **binary**
  - **xml**
- **Nullable**: Select True or False to determine whether the parameter can be nullable.
- **Default**: Enter a default value for the parameter, which is used if the parameter attribute is missing in the input record.

Notes:

- If a field is not nullable and a default value is not supplied in the schema part of the Adapter Definition, an error occurs if the parameter attribute is missing in the input record.
- The parameters must be entered in the same order as they are used in the SQL statement.

A.2.3 Adapter Metadata Schema Records

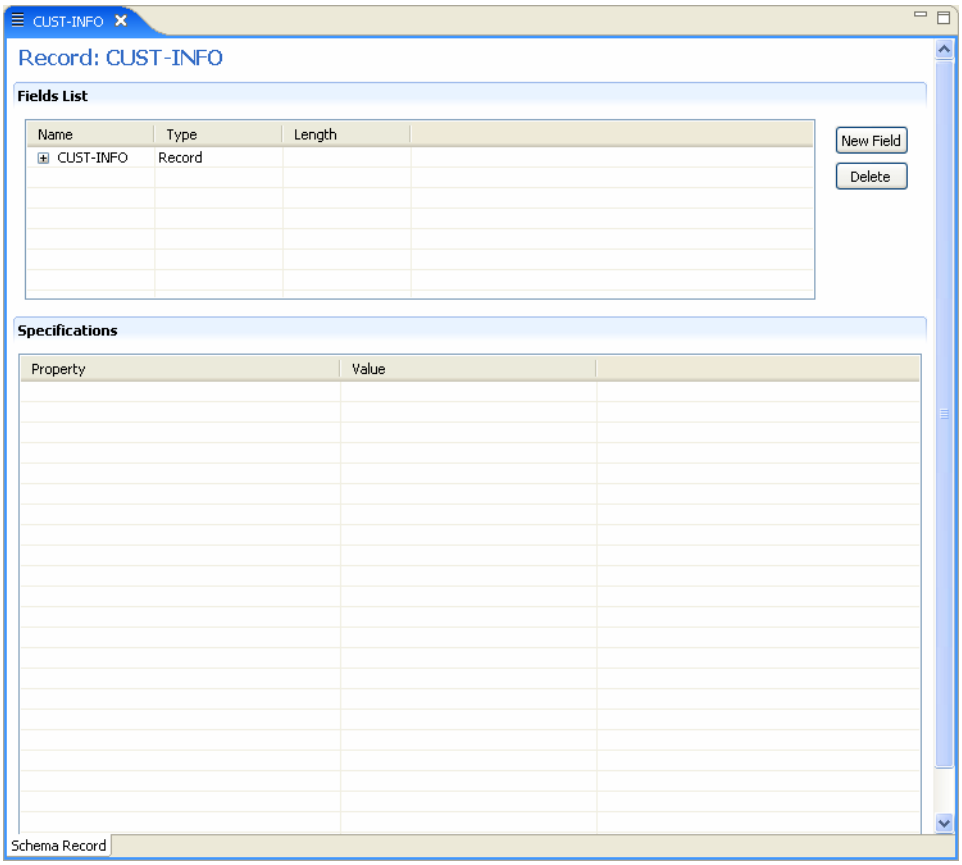
The Adapter Metadata Schema Records editor defines the general details of the input and output record structures for the interaction.The following describes how to open the Adapter Metadata Schema Records editor:

1. In Oracle Studio Design perspective, Metadata view, expand the **Adapters** folder.
2. Expand the 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–10 Adapter Metadata Schema Records



The Adapter Metadata Schema Records properties are listed in the following tables:

Table A–14 Schema Record Tab

Field	Description
Fields list	Defines the single data item within a record. This section has a table with the following three columns: ▪ **Name**: The name of the field ▪ **Type**: The data type of the field. For a list of the valid data types, see the Valid Data Types table. ▪ **Length**: The size of the field including a null terminator, when the data type supports null termination (such as the string data type).
Specifications	Defines specific field properties. To display the properties, select the specific field in the Fields list.

The following table describes the valid data types that can be used when defining these specifications in the Schema Record editor.

Table A–15 Valid Data Types

Binary	Boolean	Byte
Date	Double	Enum

Table A–15 (Cont.) Valid Data Types

Binary	Boolean	Byte
Float	Int	Long
Numeric[(p[,s])]	Short	String
Time	Timestamp	

A.2.4 Adapter Metadata XML

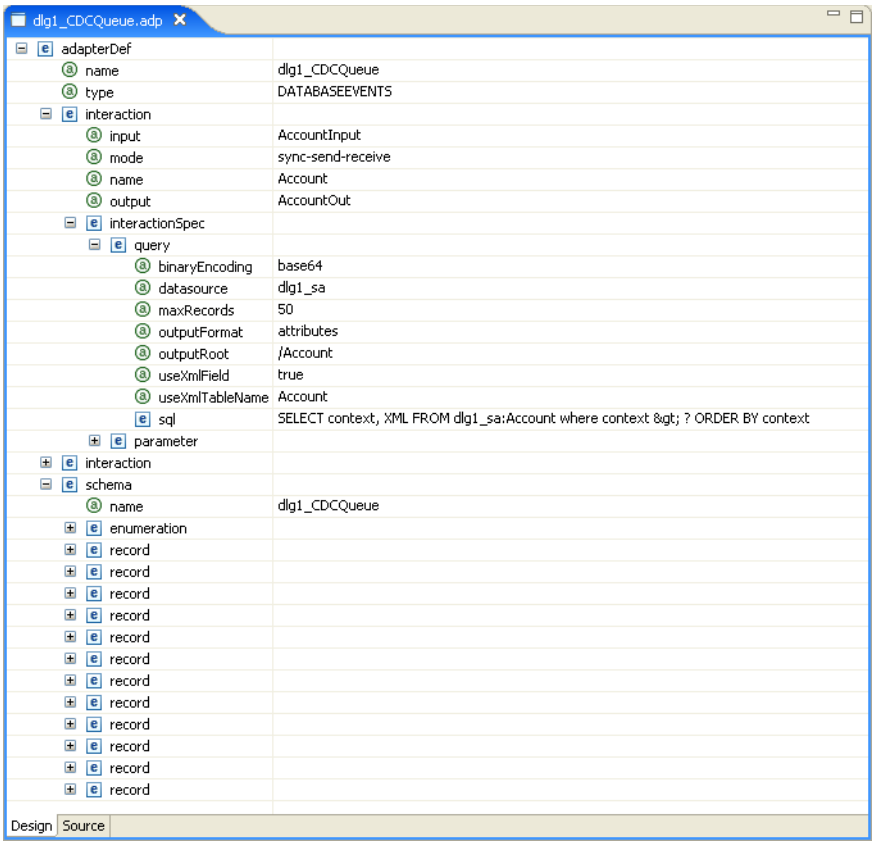
You can also edit the adapter metadata by viewing its XML schema. The following describes how to open the adapter metadata XML editor:

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–11 Adapter Metadata XML Editor



For information on how to edit the properties in the XML editor Design view, see .

B

Back-end Adapter Data Type Support

OracleAS Adapter for VSAM supports several data types that are used when defining metadata in Oracle Studio. The data types are mapped from the COBOL data types during the import procedure.

> **Note:** The mapping of data types between OracleAS Adapter for VSAM and Oracle Application Server is performed internally by Oracle Connect.

B.1 Data Type Mapping

The COBOL data type `COMP`, in the table is an abbreviation for, and synonymous with, COMPUTATIONAL. Square brackets ([ ]) denote optional qualifiers for some COBOL compilers, which may not be allowed for other COBOL compilers.

Table B–1 Data Type Mapping: COBOL and Oracle Connect Back-end Adapter

COBOL Data Type	Oracle Connect Data Type
`BINARY` (with fractional data)	`string`
`BINARY` (without fractional data)	`int`
`COMP` (with fractional data)	`string`
`COMP` (without fractional data)	`int`
`COMP-2`	`double`
`COMP-3`	`string`
`COMP-4` (with fractional data)	`string`
`COMP-4` (without fractional data)	`int`
`COMP-5` (with fractional data)	`string`
`COMP-5` (without fractional data)	`int`
`COMP-X` (with fractional data)	`string`
`COMP-X` (without fractional data)	`int`
`INDEX`	`int`
`[SIGN [IS]] LEADING`	`string`
`[SIGN [IS]] LEADING SEPARATE [CHARACTER]`	`string`
`NATIVE-2`	`int`

Table B–1 (Cont.) Data Type Mapping: COBOL and Oracle Connect Back-end Adapter

COBOL Data Type	Oracle Connect Data Type
NATIVE-4	int
NATIVE-8	string
PACKED-DECIMAL	string
POINTER	int
[SIGN [IS]] TRAILING	string
[SIGN [IS]] TRAILING SEPARATE [CHARACTER]	string

C

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

C.1 Control

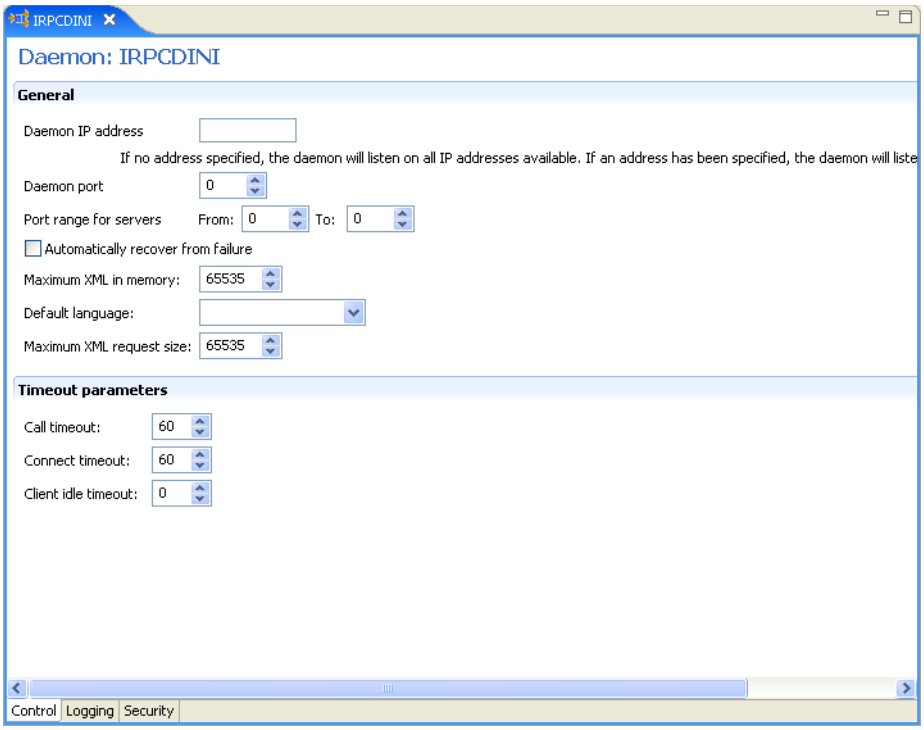
Using the **Control** tab for the daemon, you define various daemon control options. The **Daemon Control** tab is accessed as follows:

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	The timeout period for *short* 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. Specifying the timeout in a workspace overrides the value set in this field for that workspace.
Connect timeout	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. **Notes**: ■ 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	The maximum amount of time any daemon client may be idle before the connection with the server is closed. Specifying the timeout in a Workspace overrides this setting for that workspace.

C.2 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**.

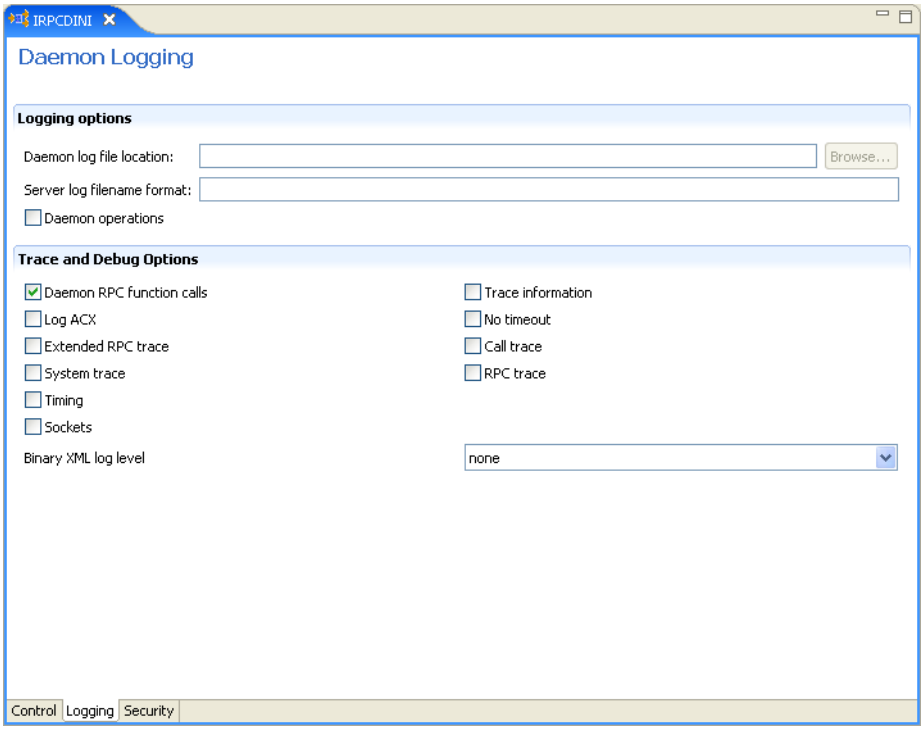
Note: You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Open**.

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: ■ 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`.

C.3 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:

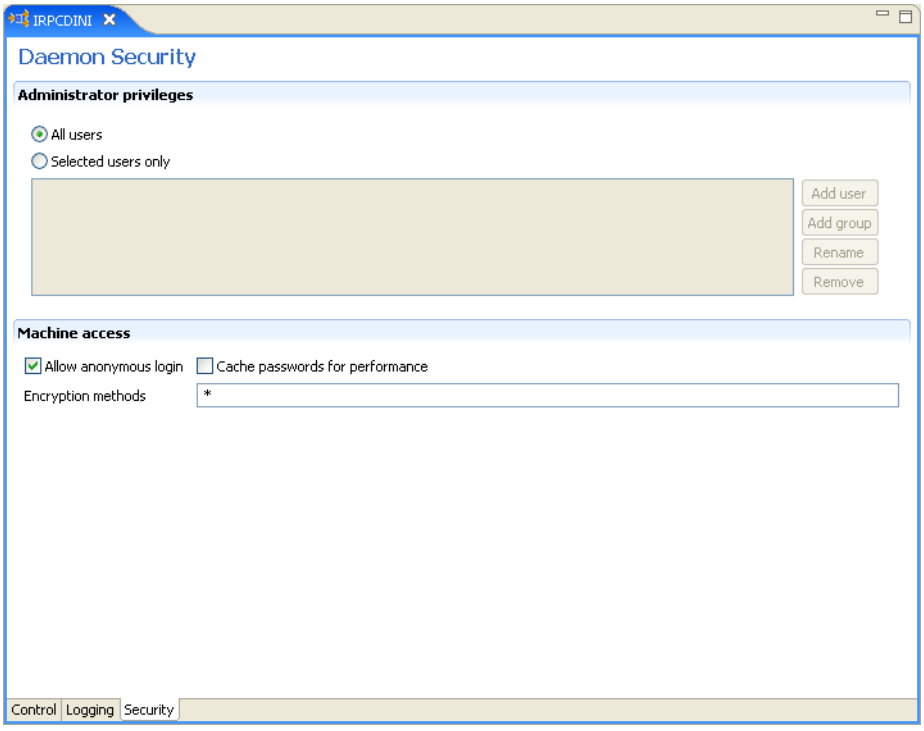
1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design perspective Configuration view expand the **Machines** folder.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Right-click the daemon in the Runtime Explorer and select **Edit 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.

C.4 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

C.4.1 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.

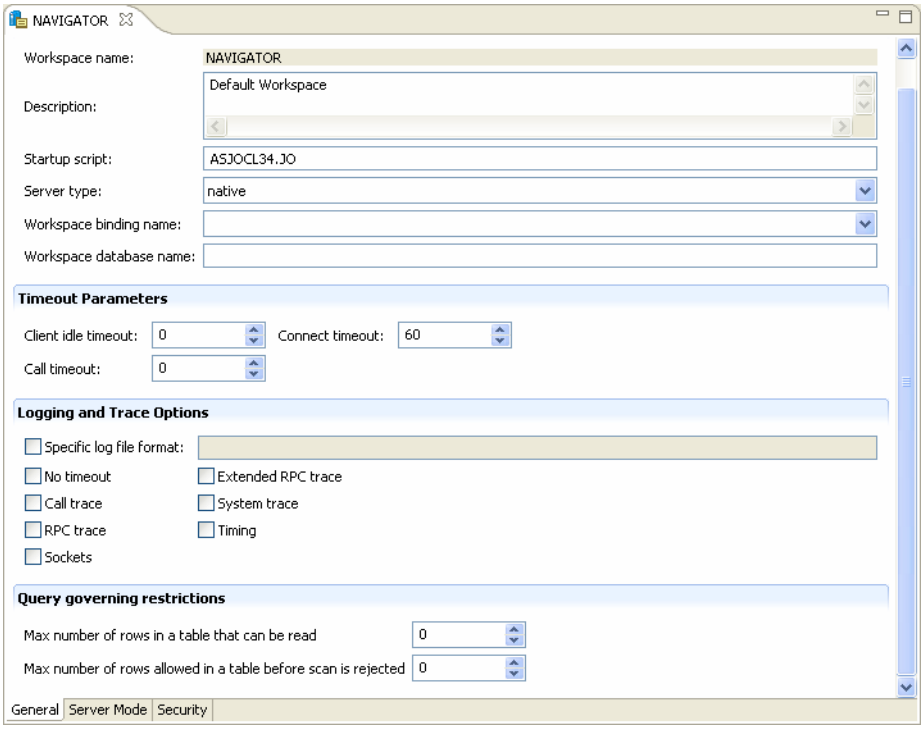
3. Right-click a computer and select **Open Runtime Perspective**.
4. Expand the Daemons node to display the workspaces in the Runtime Explorer.
5. Right-click the workspace and select **Edit Workspace Configuration**. The **WS Info**. tab opens.
6. After making changes to the workspace, right-click 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 `nav_login` procedure and then run the server program (svc). 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 VSAM. *Do not change the default value*.
Workspace binding name	This field is not applicable for use with OracleAS Adapter for VSAM.
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 *short* 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. 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	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. The following tokens can appear in the log file template and are replaced accordingly: ■ %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)
Trace options	Specifies the type of tracing to be performed. The following tracing options are available: ■ **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	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.
Max number of rows allowed in a table before scan is rejected	Select the maximum number of table rows that can be scanned. This parameter has different behavior for query optimization and execution. ■ 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.

C.4.2 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**.

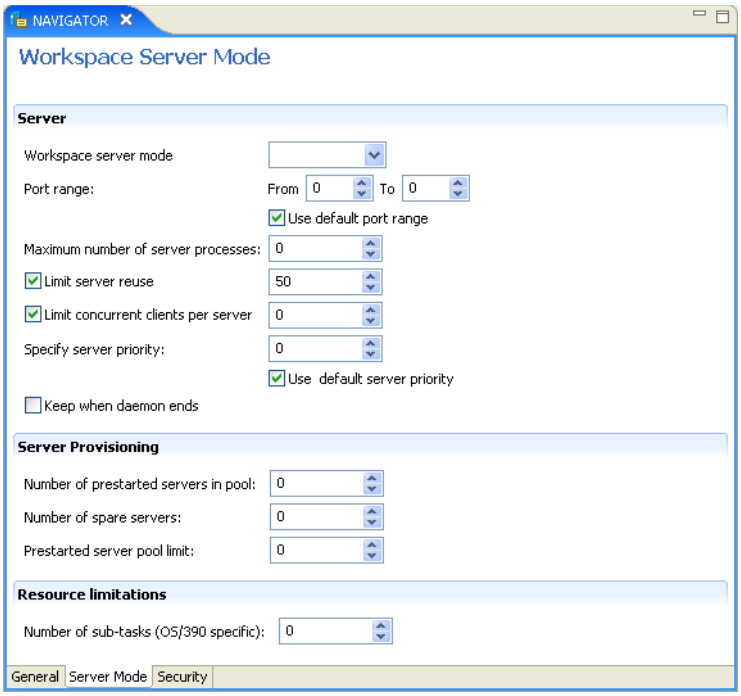
Notes:

- 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



The **Server Mode** tab has the following fields:

Table C–5 Server Mode Tab Components

Field	Description
Workspace server modeVSAM	Specifies the type of new server processes that the daemon starts. 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, 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. ■ **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. ■ **multiThreaded**: This mode is not applicable for use with OracleAS Adapter for VSAM. ■ **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. 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. **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.
Reuse limit	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**.
Clients per server limit	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**.

Table C–5 (Cont.) Server Mode Tab Components

Field	Description
Server availability	Specifies the number of servers in a pool of servers, available to be assigned to a client. The following options are available: ■ **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. ■ **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. ■ **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 `true` for the workspace servers to remain active, even after the daemon has been shut down. If this field is set to `true`, 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. ■ **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	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. Enter the port range in the following fields: ■ **From**: enter the lowest numbered port in the range. ■ **To**: Enter the highest numbered port in the range.
Use default port range	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.
Maximum number of server processes	Enter the maximum number of server processes that can run at the same time.
Limit server reuse	Select this to limit the number of servers that can be reused. If this is selected, the Reuse limit parameter is available.

Table C–5 (Cont.) Server Mode Tab Components

Field	Description
	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. 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. This parameter is not available if the Limit server reuse parameter is not selected. This parameter is not available if the server mode value is **singleClient**.
Limit concurrent clients per server	Select this to limit the number of clients that a server can accept for the current workspace process. If this is not selected, the number of clients is unlimited. 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**.
Specify Server Priority	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`. **Note**: This is unavailable if **Use default server priority** is selected.
Use default server priority	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.
Keep when daemon ends	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.
Server Provisioning	
Number of prestarted servers in pool	**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.

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).

C.4.3 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**.

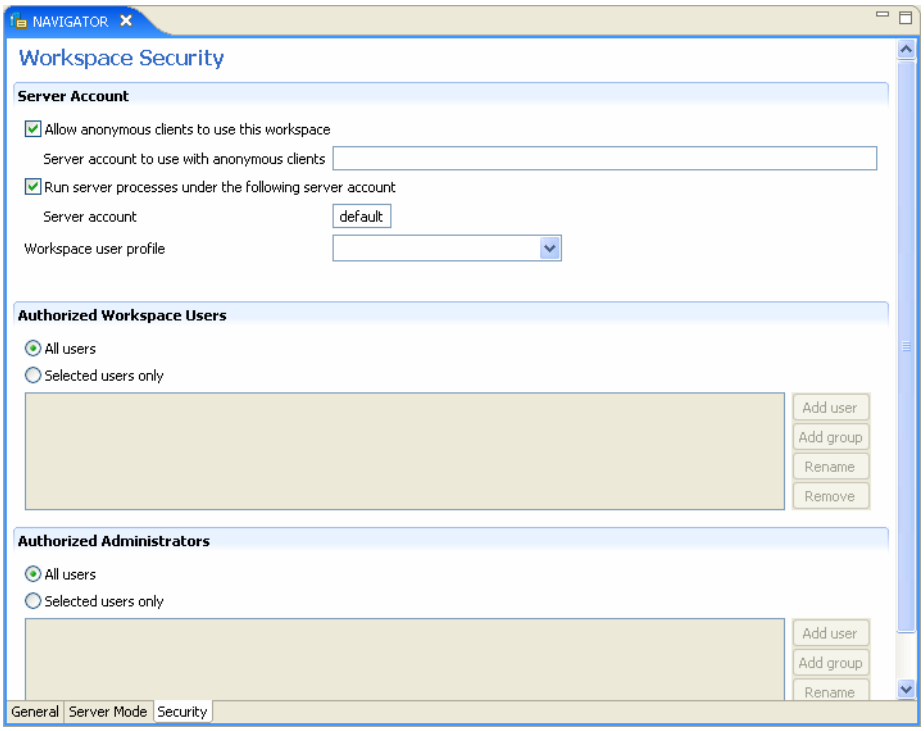
Notes:

- 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	Select this to define the operating system account used for the workspace. If selected, enter the name of the workspace account in the **workspace account** field. If not selected, the account name that was provided by the client is used.
Allow anonymous clients to use this workspace	Select this to allow this workspace to be invoked without authentication. If selected, enter the name of the workspace account in the **Server account to use with anonymous clients** field.
Authorized Workspace users	Indicate which users have permission to use the workspace. Select one of the following ■ **All user**s: 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	Identifies the users (accounts) with administrator privileges. Select one of the following: ■ **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.

D

Supported SQL Syntax and SQL Enhancements

OracleAS Adapter for VSAM provides basic support for standard ANSI '92 SQL along with several enhancements, all of which can be used when specifying adapter interactions.

This appendix contains the following sections:

- Supported SQL Statements
- Supported Operators
- Supported Functions
- SQL Enhancements

D.1 Supported SQL Statements

OracleAS Adapter for VSAM supports `SELECT`, `DELETE`, `INSERT`, and `UPDATE` statements, when specified in interactions.

D.2 Supported Operators

OracleAS Adapter for VSAM enables the use of the following operators in SQL statements:

- Arithmetic Operators
- Comparison Operators

Table D–1 Arithmetic Operators

Operator
+
-
*
/

Table D–2 Comparison Operators

Operator
=

Table D–2 (Cont.) Comparison Operators

Operator
>
<
>=
<=
<>, !=, ^=
IS NOT NULL
IS NULL

D.3 Supported Functions

OracleAS Adapter for VSAM enables the use of the following functions in SQL statements:

- String Functions
- Group Functions
- Mathematical Functions

Table D–3 String Functions

Oracle Function	Oracle Connect Function	Usage	Comment
||	||	*str1* || *str2*	Returns a string consisting of *str1* concatenated with *str2*
Ascii	Ascii	Ascii(*c*)	Returns the ASCII value of *c*
Chr	Chr	Chr(*ascii*)	Returns the character corresponding to the given ASCII value
Instr	Position	Position(*str2*, *str1*)	Returns an index in *str2* to the first occurrence of *str1* in *str2*
Length	Length	Length(*str*)	Returns the number of bytes of *str*
Lower	Lower	Lower(*str*)	Returns *str* in lowercase
Lpad	Lpad	Lpad(*str*, *n*)	Returns *str* with *n* leading blanks
Ltrim	Ltrim	Ltrim(*str*)	Returns *str* with leading blanks removed
Rpad	Rpad	Rpad(*str*, *n*)	Returns *str* with *n* trailing blanks
Rtrim	Rtrim	Rtrim(*str*)	Returns *str* with trailing blanks removed
Substr	Substr	Substr(*str*, *n* [, *m*])	Returns a substring of *str*, starting with the *n*th character and *m* characters in length, or until the end of the string if *m* is not supplied
Upper	Upper	Upper(*str*)	Returns *str* in uppercase

Table D–4 Group Functions

Oracle Function	Oracle Connect Function	Usage	Comment
Avg	Avg	Avg(*exp*)	Returns the average value of the expression *exp*
Count	Count	Count(*exp*)	Returns the count of the expression *exp*

Table D–4 (Cont.) Group Functions

Oracle Function	Oracle Connect Function	Usage	Comment
Max	Max	Max(*exp*)	Returns the maximum value of the expression *exp*
Min	Min	Min(*exp*)	Returns the minimum value of the expression *exp*
Sum	Sum	Sum(*exp*)	Returns the summation of the expression *exp*

Table D–5 Mathematical Functions

Oracle Function	Oracle Connect Function	Usage	Comment
Abs	Abs	Abs(*n*)	Returns the absolute value of *n*
Ceil	Ceil	Ceil(*n*)	Returns *n* rounded up to the closest integer
Cos	Cos	Cos(*n*)	Returns the cosine value of *n*
Exp	Exp	Exp(*n*)	Returns the exponential value of *n*
Floor	Floor	Floor(*n*)	Returns *n* rounded down to the closest integer
Ln	Ln	Ln(*n*)	Returns the natural log value of *n*
Log	Log	Log(*n*)	Returns the log value of *n*
Mod	Mod	Mod(*n*, *m*)	Returns the integer value after dividing *n* by *m*
Nvl	Nvl	Nvl(*exp1*, *exp2*)	Returns *exp2* when *exp1* is null
Power	Power	Power(*n*, *m*)	Returns *n* to the power of *m*
Round	Round	Round(*n*, *m*)	Returns *n* with the fractional part rounded to *m* digits
Sin	Sin	Sin(*n*)	Returns the sine value of *n*
Sqrt	Sqrt	Sqrt(*n*)	Returns the square root of *n*
Tan	Tan	Tan(*n*)	Returns the tangent value of *n*
Trunc	Trunc	Trunc(*n*, *m*)	Returns the absolute value of *n*

D.4 SQL Enhancements

You can incorporate the following SQL enhancements into the adapter interactions to handle hierarchical data in VSAM.

- Generating Hierarchical Results
- Accessing Hierarchical Data Using SQL
- Flattening Hierarchical Data Using SQL

D.4.1 Generating Hierarchical Results

A hierarchical query nests a `SELECT` statement as a column of the rowset retrieved by a nested `SELECT` statement.

Use braces ({}) to delimit the nesting.

D.4.2 Accessing Hierarchical Data Using SQL

Data stored hierarchically in a VSAM data source can be referenced by using a hyphen followed by a right arrow (->) to denote the parent child relationship in the source:

FROM ... *parent_name->child1->child2*... [*alias*]

Or, using an alias for the parent table:

FROM ... *parent_alias->child1->child2*... [*alias*]

D.4.3 Flattening Hierarchical Data Using SQL

You can produce a flattened view of hierarchical data by embedding a `SELECT` statement inside the list of columns to be retrieved by another `SELECT` statement. You use parentheses to delimit the nesting. The nested `SELECT` statement can reference a child rowset (using the parent->child syntax) only in its `FROM` clause.

D.4.3.1 Using an Alias

To list the hierarchical data with the parent data only, you must use an alias for the child data.

Note: Without an alias the query lists, for each parent row, all of the children of all of the parent rows.

E

Globalization Settings

The OracleAS CDC Adapter for VSAM 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.

E.1 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 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 E–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 E–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 E–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"`

F

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.

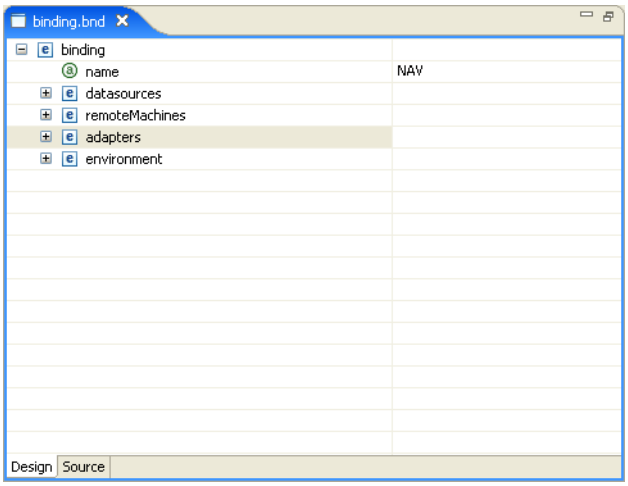
F.1 Preparing to Edit XML Files in Oracle Studio

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 F–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.

F.2 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

F.2.1 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**.

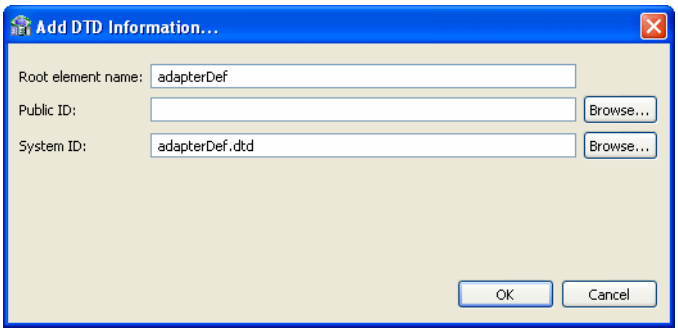
F.2.2 Add DTD Information

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 F–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 F–1 Add DTD Information

Field	Description
Root element name	The name of the XML root element.
Public ID	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. 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.
System ID	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. Click **Browse** to select a system ID. You can this in two ways: ■ 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.

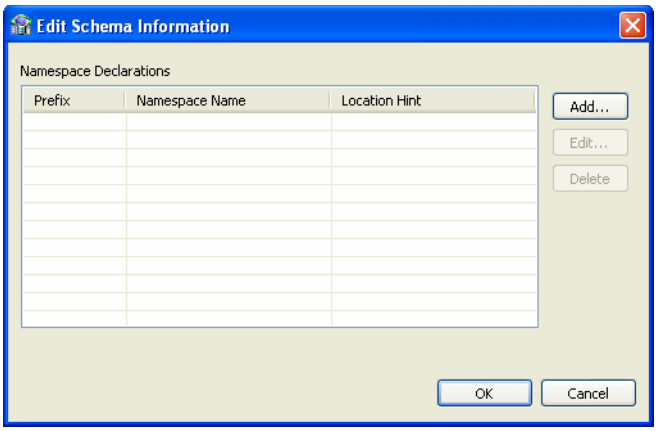
F.2.3 Edit Namespaces

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 F–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 F–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	The location of the XML schema of the XML file. An XML Catalog ID or a URI can be entered in this field. Click **Browse** to search for the schema you want You can this in two ways: ▪ Select the schema from the workbench. In this case, update the with the import dialog box. ▪ Select an XML catalog entry. 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). **Note**: If you are creating an XML file from an XML schema, you cannot change the Namespace Name or Location Hint values.

To edit a namespace

1. From the Schema Information dialog box, click **Edit**.
2. Enter the information in the fields.

F.2.4 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.

F.2.5 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, 8-7
add timestamp to traced events parameter, 8-7
arithmetic operators, D-1
arrays, flattening hierarchical data, D-4
Artithmetic fixed precision parameter, 8-12

B

binary XML log level parameter, 8-7

C

cache
  maximum queries, 8-11
cache buffer size parameter, 8-8
code page parameter, 8-9
COM maximum XML size in memory parameter, 8-14
COM maximum XML size parameter, 8-14
COM XML transport buffer parameter, 8-14
commit on destroy, 8-13
comparison operators, D-1
configuring JCA configuration files for outbound, 4-2
continuous query
  prefix, 8-11
  retry interval, 8-11
  timeout, 8-11
continuous query prefix parameter, 8-11
continuous query retry interval parameter, 8-11
continuous query timeout parameter, 8-11
creating JCA configuration files
  JCA configuration files
    creating, 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 sources, disabling caching, 8-10
data types
  atomic metadata, B-1
  NLS string parameter, 8-9
debug parameters
  ACX trace, 8-7
  add timestamp to traced events, 8-7
  environment, 8-7
  GDB trace, 8-7
  general trace, 8-7
  lbinary XML log level, 8-7
  log file, 8-7
  optimizer trace, 8-8
  query processor trace, 8-7
  query warnings, 8-7
  trace directory, 8-8
  Transaction extended logging, 8-8
disable 2PC, 8-13
disable command reuse parameter, 8-10
disable compilation cache parameter, 8-11
disable DS property cache parameter, 8-10
disable insert parameterization parameter, 8-10
disable metadata caching parameter, 8-10
disable query parameterization parameter, 8-10
disable row mark fail fetch parameter, 8-10
disable SQS cache parameter, 8-12
disable threads parameter, 8-10
disabling
  parameterization of constants, 8-10
  parameterization of queries, 8-10
dsm maximum buffer size parameter, 8-13
dsm middle buffer size parameter, 8-14
dsmMaxSortBufferSize parameter, 8-14

E

environment parameters
  ACX trace, 8-7
  add timestamp to traced events, 8-7
  Artithmetic fixed precision, 8-12
  cache buffer size, 8-8
  code page, 8-9
  COM maximum XML size, 8-14
  COM maximum XML size in memory, 8-14
  COM XML transport buffer, 8-14
  commit on destroy, 8-13
  continuous query prefix, 8-11

continuous query retry interval, 8-11
continuous query timeout, 8-11
debug, 8-7
disable 2PC, 8-13
disable command reuse, 8-10
disable compilation cache, 8-11
disable DS property cache, 8-10
disable insert parameterization, 8-10
disable metadata caching, 8-10
disable query parameterization, 8-10
disable row mark fail fetch, 8-10
disable SQS cache, 8-12
disable threads, 8-10
dsm maximum buffer size, 8-13
dsm maximum sort buffer Size, 8-14
dsm middle buffer size, 8-14
expose XML fields, 8-12
file close on transaction, 8-14
file pool size, 8-14
file pool size per file, 8-14
first tree extensions, 8-11
GDB trace, 8-7
general trace, 8-7
hash buffer size, 8-14
hash enable RO, 8-14
hash max open files, 8-14
hash primary event size, 8-14
hash secondary event size, 8-14
ignore segment bind failure, 8-11
insert from select commit rate, 8-12
language, 8-9
log file, 8-7
maximum columns in parsing, 8-11
Maximum segmented database thread, 8-11
maximum SQL cache, 8-11
minimum number of parameters allocated, 8-11
miscellaneous, 8-8
nav utility editor, 8-8
NLS string, 8-9
odbc, 8-9
oledb, 8-9
optimistic for updates, 8-11
optimizer, 8-9
optimizer goal, 8-9
optimizer trace, 8-8
parser depth, 8-12
procedures cache size, 8-12
prompt database user pasword, 8-11
query processor trace, 8-7
query warnings, 8-7
queryProcessor, 8-10
recovery delay, 8-13
Replace invalid XML characters, 8-15
time limit, 8-13
token size, 8-12
trace directory, 8-8
transaction conversions parameters, 8-13
Transaction extended logging, 8-8
transaction extended logging, 8-13
transaction log file, 8-13
transactions, 8-13
ttemporary directory, 8-8
tuning, 8-13
use alternate qualifier, 8-11
use global file pool, 8-14
use table filter expression, 8-11
user commit confirm table, 8-13
write empty string as null, 8-11
XML, 8-14
XML date format parameter, 8-15
XML trim char column, 8-15
year 2000 policy, 8-8
error log, binary XML log level parameter, 8-7
error log, log file parameter, 8-7
exact arithmetic, 8-12
expose XML fields parameter, 8-12

F

file close on transaction, 8-14
file pool size, 8-14
file pool size per file, 8-14
first row optimization, optimizer goal parameter, 8-9
first tree extensions parameter, 8-11
floating point precision, 8-12
functions
group, D-2
string, D-2

G

GDB trace parameter, 8-7
general parameters
cache buffer size, 8-8
nav utility editor, 8-8
temporary directory, 8-8
year 2000 policy, 8-8
general trace parameter, 8-7
group functions, D-2

H

hash buffer size, 8-14
hash buffer size parameter, 8-14
hash enable RO parameter, 8-14
hash joins, 8-14
hash max open files parameter, 8-14
hash primary event size parameter, 8-14
hash secondary event size parameter, 8-14
hierarchical queries, D-4

I

ignore segment bind failure parameter, 8-11
index cache
dsm middle buffer size parameter, 8-14
insert from select commit rate parameter, 8-12
integration flow, Oracle Application Server to legacy application, 1-3

J

JCA configuration files
  configuring outbound interactions, 4-2
joins
  hash buffer size parameter, 8-14
  hash max open files parameter, 8-14

L

language parameter, 8-9
language parameters
  code page, 8-9
  environment, 8-9
  language, 8-9
  NLS string, 8-9
locking, optimistic, 8-11
log file parameter, 8-7
log files
  binary XML log level parameter, 8-7
  daemon options, C-3
  log file parameter, 8-7
logging
  daemon configurations, C-3
  optimizer strategy, 8-8

M

maximum columns in parsing parameter, 8-11
Maximum segmented database thread, 8-11
maximum SQL cache parameter, 8-11
memory
  dsm maximum buffer size parameter, 8-13
  dsm maximum sort buffer Size parameter, 8-14
  dsm middle buffer size parameter, 8-14
metadata
  atomic data types, B-1
  disabling retrieval from cache, 8-10
minimum number of parameters allocated
    parameter, 8-11
miscellaneous parameters
  environment, 8-8

N

nav utility editor parameter, 8-8
nav utility editor, text editor, 8-8
NLS string parameter, 8-9

O

odbc environment parameters, 8-9
oledb environment parameters, 8-9
optimistic for updates parameter, 8-11
optimizer
  environment parameters, 8-9
  optimizer goal parameter, 8-9
  trace directory parameter, 8-8
optimizer goal parameter, 8-9
optimizer trace parameter, 8-8

P

parallel processing parameters
  disable threads, 8-10
parameterization of constants, disabling, 8-10
parameterization of queries, disabling, 8-10
parser depth parameter, 8-12
password, setting automatic prompt, 8-11
procedures cache size parameter, 8-12
prompt database user pasword parameter, 8-11

Q

queries
  disabling reuse, 8-10
  disabling saved compilation, 8-11
  enabling retrieval after failure, 8-10
  maximum cached in memory, 8-11
  maximum length of string, 8-12
  maximum size after compilation, 8-11
query optimizer
  logging strategy, 8-8
  optimizer goal parameter, 8-9
  trace directory parameter, 8-8
query processor parameters
  Artithmetic fixed precision, 8-12
  continuous query prefix, 8-11
  continuous query retry interval, 8-11
  continuous query timeout, 8-11
  disable command reuse, 8-10
  disable compilation cache, 8-11
  disable DS property cache, 8-10
  disable insert parameterization, 8-10
  disable metadata caching, 8-10
  disable query parameterization, 8-10
  disable row mark fail fetch, 8-10
  disable SQS cache, 8-12
  emptyStringISNull parameter, 8-11
  expose XML fields, 8-12
  first tree extensions, 8-11
  ignore segment bind failure, 8-11
  insert from select commit rate, 8-12
  maximum columns in parsing, 8-11
  Maximum segmented database thread
      parameter, 8-11
  maximum SQL cache, 8-11
  minimum number of parameters allocated, 8-11
  optimistic for updates, 8-11
  parser depth, 8-12
  procedures cache size, 8-12
  prompt database user pasword, 8-11
  token size, 8-12
  use alternate qualifier, 8-11
  use table filter expression, 8-11
query processor trace parameter, 8-7
query warnings parameter, 8-7
queryProcessor environment parameters, 8-10

R

recovery delay, 8-13

Replace invalid XML characters parameter, 8-15
row optimization, optimizer goal parameter, 8-9

S

security
  daemon configurations, C-5
  setting automatic prompt, 8-11
segmented data sources, response upon failure, 8-11
servers
  configuring modes, 8-2
  reusable, 8-3
  Reuse limit daemon parameter, 8-3, C-13
setting up VSAM, 2-6
sort buffer, dsm maximum sort buffer size
    parameter, 8-14
SQL, flattening hierarchical data, D-4
stored queries, maximum in cache memory, 8-12
string functions, D-2

T

temporary directory parameter, 8-8
temporary files, 8-8
threading model, 8-11
time limit parameter, 8-13
timeout
  client idle, 7-6
  daemon, 7-6
token size parameter, 8-12
trace directory parameter, 8-8
trace information, logging, 8-7
transaction conversions parameters, 8-13
transaction extended logging, 8-13
Transaction extended logging parameter, 8-8
transaction log file, 8-13
transactions environment parameters, 8-13
transactions parameters
  commit on destroy, 8-13
  disable 2PC, 8-13
  recovery delay, 8-13
  time limit, 8-13
  transaction conversions parameters, 8-13
  transaction extended logging, 8-13
  transaction log file, 8-13
  user commit confirm table, 8-13
tuning environment parameters, 8-13
tuning parameters
  dsm maximum buffer size, 8-13
  dsm maximum sort buffer Size, 8-14
  dsm middle buffer size, 8-14
  file close on transaction, 8-14
  file pool size, 8-14
  file pool size per file, 8-14
  hash buffer size, 8-14
  hash enable RO, 8-14
  hash max open files, 8-14
  hash primary event size, 8-14
  hash secondary event size, 8-14
  use global file pool, 8-14

U

use alternate qualifier parameter, 8-11
use global file pool, 8-14
use table filter expression parameter, 8-11
user commit confirm table, 8-13

V

VSAM adapter integration, 3-2, 4-1
VSAM, setting up, 2-6

W

WebLogic
  integration, 3-1
WebLogic integration, 3-1
Workspace server mode, C-13
write empty string as null parameter, 8-11

X

XML
  environment parameters, 8-14
XML date format parameter, 8-15
XML parameters
  COM maximum XML size, 8-14
  COM maximum XML size in memory, 8-14
  COM XML transport buffer, 8-14
  Replace invalid XML characters, 8-15
  XML date format parameter, 8-15
  XML trim char column, 8-15
XML trim char column parameter, 8-15

Y

Y2K
  See year 2000 policy parameter
year 2000 policy parameter, 8-8