Oracle® Application Server
Adapters for VSAM User's Guide
10g  Release 3 (10.1.3.4)
E13830-01

December 2008
# Contents

**Preface**........................................................................................................................................................................ ix  
  Audience ........................................................................................................................................................................... ix  
  Documentation Accessibility ............................................................................................................................................... ix  
  Related Documents ......................................................................................................................................................... x  
  Conventions ...................................................................................................................................................................... x  

**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 Installing and Configuring OracleAS Adapter for VSAM**

Preinstallation Tasks ......................................................................................................................................................... 2-1  
  IBM z/OS Hardware and Software Requirements ...................................................................................................... 2-1  
    Hardware Requirements .............................................................................................................................................. 2-1  
    Software Requirements ........................................................................................................................................... 2-2  
Installing Oracle Connect on an IBM z/OS Series Platform .............................................................................................. 2-2  
  Installation Worksheet .................................................................................................................................................. 2-2  
  Preinstallation Instructions ....................................................................................................................................... 2-4  
  Importing the Installation Kit ....................................................................................................................................... 2-4  
  Installation Instructions .............................................................................................................................................. 2-4  
  Post-Installation Instructions .................................................................................................................................. 2-8  
    Post-Installation Procedures .................................................................................................................................. 2-8  
    Starting the Daemon ................................................................................................................................................. 2-10  
    Setting Up Oracle Connect for Reentrancy .............................................................................................................. 2-10  
    Setting Up Oracle Connect to Update VSAM Data ................................................................................................. 2-10  
Updating an Existing Oracle Connect Installation with VSAM .......................................................................................... 2-11  
Installing Oracle Studio ...................................................................................................................................................... 2-12  
  Oracle Studio Requirements ........................................................................................................................................ 2-12  
  Installing Oracle Studio on Windows .......................................................................................................................... 2-12  
  Installing Oracle Studio on Linux ................................................................................................................................ 2-13  
    Installing Oracle Studio with the Wizard (SH installation) ....................................................................................... 2-13  
    Installing with a Silent Installation (RPM) ................................................................................................................... 2-13
3 Deployment and Integration with the Oracle OC4J Container

Integrating OracleAS Adapter for VSAM with OC4J ................................................................. 3-1
Configuring the J2CA 1.5 VSAM Adapter .............................................................................. 3-2
Using the CCI API to Develop Applications ........................................................................ 3-3
Configuring Multiple Adapters ............................................................................................ 3-4
Updating Configuration Information ..................................................................................... 3-4

4 BPEL Process Manager Examples

Getting Started .......................................................................................................................... 4-1
Deploy a BPEL Outbound Process .......................................................................................... 4-2
Beginning Tasks ....................................................................................................................... 4-2
Design-Time Configuration ..................................................................................................... 4-3
Create a BPEL Project for a BPEL Outbound Process .......................................................... 4-3
Create a Database Adapter to Read the Doctor Data ............................................................. 4-4
Design a BPEL Outbound Process to Read the String ......................................................... 4-5
Create a Partner Link to the Database Adapter ..................................................................... 4-6
Edit the Invoke_1 Activity ..................................................................................................... 4-7
Edit the Assign_1 Activity ..................................................................................................... 4-8
Runtime Configuration .......................................................................................................... 4-9

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
6 Troubleshooting OracleAS Adapter for VSAM

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

7 Advanced Features of OracleAS Adapter for VSAM

Configuring the Daemon for High Availability ................................. 7-1
  Adding a New Daemon Workspace Configuration ............................ 7-1
  Editing the Workspace ..................................................................... 7-2
  Configuring the Server Mode .......................................................... 7-2
Configuring a Binding Environment .................................................. 7-6
  Debug .............................................................................................. 7-7
  General ........................................................................................... 7-8
  Language ......................................................................................... 7-8
  Modeling .......................................................................................... 7-9
  ODBC .............................................................................................. 7-9
  OLEDB .............................................................................................. 7-9
  Optimizer ......................................................................................... 7-9
  Parallel Processing .......................................................................... 7-9
  Query Processor .............................................................................. 7-10
  Transaction ...................................................................................... 7-13
  Tuning ............................................................................................. 7-13
  XML .................................................................................................. 7-14
Migration Considerations ................................................................... 7-15
Security Considerations ..................................................................... 7-15
  Setting Design Time Security .......................................................... 7-16
  Setting Run Time Security .............................................................. 7-16
Transaction Support ........................................................................... 7-17

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
Preface

This guide is the primary source of user and reference information on the OracleAS CDC adapter for VSAM, which enables Oracle users to use change data capture in their integration architecture, working with the Oracle Application Server.

This document describes the features of the OracleAS CDC adapter for VSAM.

This preface covers the following topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

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

- Installing and configuring OracleAS CDC Adapters for VSAM
- Using OracleAS for developing integration solutions using change data capture
- Diagnosing errors

Note: You should understand the fundamentals of OracleAS, OC4J, the UNIX and Microsoft Windows operating system before using this guide to install or administer OracleAS Adapters for VSAM.

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.

TTY Access to Oracle Support Services
To reach AT&T Customer Assistants, dial 711 or 1.800.855.2880. An AT&T Customer Assistant will relay information between the customer and Oracle Support Services at 1.800.223.1711. Complete instructions for using the AT&T relay services are available at http://www.consumer.att.com/relay/tty/standard2.html. After the AT&T Customer Assistant contacts Oracle Support Services, an Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process.

Related Documents
For more information, see the following documents in the Oracle Other Product One Release 10.0 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:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>boldface</td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td>italic</td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
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 section provides an overview of the feature and architecture of OracleAS Adapter for VSAM.

This section contains the following topics:

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

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.
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.

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.

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 (OC4J).
- 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**

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.

**See Also:** *Oracle Application Server Adapter Concepts Guide*
Installing and Configuring OracleAS Adapter for VSAM

This section describes how to install Oracle Connect and Oracle Studio and how to configure Oracle Connect using Oracle Studio.

**Note:** In addition to the installation procedures described in this section, the J2CA 1.5 VSAM adapter must be installed with Oracle Application Server. Installing the J2CA 1.5 VSAM adapter is described in *Oracle Application Server Adapter Installation Guide*.

This section includes the following topics:

- Preinstallation Tasks
- Installing Oracle Connect on an IBM z/OS Series Platform
- Updating an Existing Oracle Connect Installation with VSAM
- Installing Oracle Studio
- Configuring Oracle Connect

**Preinstallation Tasks**

Before installing OracleAS Adapter for VSAM, ensure that your computer meets the following requirements:

- IBM z/OS Hardware and Software Requirements

**IBM z/OS Hardware and Software Requirements**

This section describes the following requirements for installing Oracle Connect on an IBM z/OS platform:

- Hardware Requirements
- Software Requirements

**Hardware Requirements**

The following table summarizes the hardware requirements for Oracle Connect.

<table>
<thead>
<tr>
<th>Hardware Component</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>An IBM zSeries computer</td>
</tr>
</tbody>
</table>
Hardware Component | Requirements
--- | ---
Memory | The minimum requirement is 4MB for each connection. A connection is defined as a connection to a server process or daemon. The actual memory requirement depends on such things as the size of the database and the number of databases accessed.
Disk Space (3380 and 3390 disks) | 150 cylinders.

Software Requirements

The following table summarizes the software requirements for Oracle Connect.

Software Component | Requirements
--- | ---
Operating System | IBM z/OS
CICS TP Monitor (when accessing VSAM data under CICS) | V4R1 or higher (recommended to use CICS V6R1 or higher). CICS EXCI support must be installed and IRCSTRT=YES must be specified in the CICS initialization parameters, so that the IRC (Inter Region Communication) starts.
You can also set the IRC to open by issuing the following command: `CEMT SET IRC OPEN`. Also the IBM group DFH$EXCI (or an equivalent user-defined group) must be installed in the CICS region - using the CEDA RDO facility.
TCP/IP | The operating system must support the TCP/IP protocol for using the Internet.
C Runtime Library | The C runtime library has all the standard C runtime programs.
Oracle Application Server | Oracle Application Server 10g (10.1.3.4).

Installing Oracle Connect on an IBM z/OS Series Platform

This section explains how to install Oracle Connect. This section includes the following:

- **Installation Worksheet**
- **Preinstallation Instructions**
- **Importing the Installation Kit**
- **Installation Instructions**
- **Post-Installation Instructions**

**Note:** If you have an Oracle Connect back-end or CDC adapter already installed on the IBM z/OS platform, then follow the instructions described in "Updating an Existing Oracle Connect Installation with VSAM" on page 2-11.

Installation Worksheet

Verify that you have all the information detailed in the following installation worksheets, so you can refer to it during the configuration process.
Table 2–1 Preinstallation Information

<table>
<thead>
<tr>
<th>Topic</th>
<th>Required Information</th>
<th>Default</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Operating system</td>
<td>-</td>
<td>z/OS</td>
</tr>
<tr>
<td>- Disk space</td>
<td>-</td>
<td>-</td>
<td>150 cylinders</td>
</tr>
<tr>
<td>- Memory</td>
<td>-</td>
<td>-</td>
<td>The minimum requirement is 4MB for each connection. A connection is defined as a connection to a server process or daemon. The actual memory requirement depends on such things as the size of the database and the number of databases accessed.</td>
</tr>
<tr>
<td>- Installation high-level qualifier</td>
<td>OCL10134</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>- Volume</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>- Unit</td>
<td>3390</td>
<td>-</td>
<td>SMS only: unit where SMS resides.</td>
</tr>
<tr>
<td>- Output class</td>
<td>A</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>- JCL job card</td>
<td>-</td>
<td>-</td>
<td>An optional card (up to 6 lines) to replace the prefix job (entered as it will appear in the job)</td>
</tr>
<tr>
<td>- ISPF load library name</td>
<td>ISP.SISPLOAD</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CICS</td>
<td>CICS EXCI load library name</td>
<td>CICS.CICS.SDFHEXCI</td>
<td>To access VSAM data under CICS</td>
</tr>
</tbody>
</table>

Table 2–2 Required Permissions

Permission
- Permission to define an APF-authorized library
- Permission to write to an active proclib, such as user.proclib
- Permission to read the CICS EXCI library (when accessing VSAM data under CICS)
- Permission to update the security manager, such as RACF
- Optionally, permission to specify an output class for Oracle Connect output

Table 2–3 Installation Checklist

<table>
<thead>
<tr>
<th>Step</th>
<th>Comment/Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>tso profile noprefix</td>
<td>Ensures that the user name is not used as part of the dataset name allocated in the next steps</td>
</tr>
<tr>
<td>Allocate dataset: {HLQ}.TRANSMIT.KIT</td>
<td>130 tracks (3390), format=FB, record length=80, block size=3120</td>
</tr>
<tr>
<td>Allocate dataset: {HLQ}.TRANSMIT.LOAD</td>
<td>500 tracks (3390), format=FB, record length=80, block size=3120</td>
</tr>
<tr>
<td>FTP files to z/OS</td>
<td>FTP using binary mode</td>
</tr>
<tr>
<td>RECEIVE INDSNAME('{HLQ}.TRANSMIT.KIT')</td>
<td>-</td>
</tr>
<tr>
<td>da('{HLQ}.TRANSMIT.LIB') UNIT(unit) VOLUME(volume)</td>
<td>-</td>
</tr>
<tr>
<td>EX {HLQ}.TRANSMIT.LIB(PREPARE)</td>
<td>Successful MAXCC is 0, 4 or 8</td>
</tr>
<tr>
<td>BUILDKIT.SRC, BUILDKIT.LOAD, and BUILDKIT.GENDemo created</td>
<td></td>
</tr>
<tr>
<td>EX {HLQ}.BUILDKIT.SRC(NAVINST)</td>
<td>Successful MAXCC is 0 or 4</td>
</tr>
</tbody>
</table>
Preinstallation Instructions

Before starting the installation, make sure that you have the following information available:

- The output class for the installation output if you do not want to use the default value A.
- If you use SMS to manage all datasets, then you cannot provide unit and volume information, which is the unit where SMS resides.

Before starting the installation, make sure that you have the following permissions:

- Permission to define an APF-authorized library
- Permission to write to an active proclib, such as user.proclib
- Permission to read the CICS EXCI library when accessing VSAM data under CICS
- Permission to update the security manager, such as RACF

---

Note: Optionally, make sure that you have permission to specify an output class for Oracle Connect output. Assigning a device which is set on HOLD prevents the loss of log information when Oracle Connect started tasks finish.

---

Oracle Connect for the IBM z/OS platform is present in the following datasets:

- OCL10134.TRANSMIT.KIT
- OCL10134.TRANSMIT.LOAD

These datasets are provided in the directory Oracle_Connect\VSAM_Legacy_Adapter.

Importing the Installation Kit

Perform the following steps to import Oracle Connect installation kit to the Mainframe:

1. Run the following command:
   
   tso profile noprefix
   
   The user name will not be used as part of the dataset name. On some systems this is the default.

2. Allocate datasets with the following space for each of the following files:
   
   - OCL10134.TRANSMIT.KIT = 130 tracks (3380 and 3390 disks)
   - OCL10134.TRANSMIT.LOAD = 500 tracks (3380 and 3390 disks)

   For each dataset: RECFM=FB and LRECL=80. The block size is 3120.

3. Using FTP, copy OCL10134.TRANSMIT.KIT and OCL10134.TRANSMIT.LOAD in binary mode to the mainframe. You can replace the OCL10134 high-level qualifier to any qualifier you want.

Installation Instructions

Perform the following steps to install Oracle Connect:

1. From within TSO, run the following command at the TSO prompt:
RECEIVE INDSNAME('nnn.TRANSMIT.KIT')

Where nnn represents the high-level qualifier you want to assign for the Oracle Connect installation. Assign the high-level qualifier you specified in step 7 of the preinstallation. The default value is OCL10134.

**Note:** You can use more than one high-level qualifier (such as ACME.DEV.OCL10134) with the following conditions:

- The total length must be less than or equal to twenty characters.
- The words transmit and buildkit cannot be used as high-level qualifiers.

2. Enter the following command when prompted for the restore parameters:

```
da('nnn.TRANSMIT.LIB') [UNIT(unit) VOLUME(volume)]
```

This extracts the nnn.TRANSMIT.LIB library from the nnn.TRANSMIT.KIT kit to the specified unit and volume. If a unit and volume are not specified, then the library is extracted to the current unit and volume.

3. Run the PREPARE member of the nnn.TRANSMIT.LIB library:

```
ex PREPARE
```

Follow the instructions in the Response column in the following table for each entry in the Screen column.

<table>
<thead>
<tr>
<th>Screen</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO YOU WANT TO USE SMS MANAGED STORAGE FOR THIS INSTALLATION Y/N [N]:</td>
<td>If you want to manage the storage using SMS, then answer Y, otherwise answer N.</td>
</tr>
<tr>
<td>ENTER THE STORCLASS FOR INSTALLATION TEMP DATASETS [ ]:</td>
<td>This prompt is displayed only if SMS is used to manage the installation (you answered Y to the first prompt). Enter the storage class</td>
</tr>
<tr>
<td>ENTER THE UNIT NAME FOR INSTALLATION TEMP DATASETS [3390]:</td>
<td>If a storage class is not specified, then enter the unit name for temporary datasets used during the installation procedure</td>
</tr>
<tr>
<td>ENTER THE VOLUME NAME FOR INSTALLATION TEMP DATASETS:</td>
<td>This prompt is displayed only if SMS is not used to manage the installation (you answered N to the first prompt). The volume name for temporary datasets used during the installation procedure</td>
</tr>
<tr>
<td>ENTER THE OUTPUT CLASS FOR INSTALLATION OUTPUT [A]:</td>
<td>Enter the output class only if you do not want the default class used (the default is A)</td>
</tr>
</tbody>
</table>
### Table 2–4 (Cont.) Installation Prepare Job Prompts and Responses

<table>
<thead>
<tr>
<th>Screen</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO YOU WANT TO USE THE DEFAULT JOB CARD Y/N [Y]</td>
<td>A job card is displayed. If you want to use a replacement card, then it must be entered as it will appear in the job. You can enter up to six lines. Enter a blank card to end input. If you do not enter a card, then the Oracle Connect default card is used.</td>
</tr>
<tr>
<td>DO YOU WANT TO PERFORM A MANUAL (M) OR AUTOMATIC (A) INSTALLATION [A]:</td>
<td>If you want to review the JCL used to install Oracle Connect, before it is submitted, then respond M for a manual installation.</td>
</tr>
<tr>
<td>PLEASE REVIEW AND SUBMIT FOR EXECUTION THE HLQ.TRANSMIT.LIB(INSTJO)</td>
<td>This prompt is displayed only if a manual installation is requested (you answered M to the previous prompt).</td>
</tr>
</tbody>
</table>

The following libraries are generated:

```
nn.BUILDKIT.LOAD
nnn.BUILDKIT.SRC
nnn.BUILDKIT.GENDemo
```

Where `nnn` is the high-level qualifiers you assigned in step 1.

4. In the `nnn.BUILDKIT.SRC` library, run the `NAVINST` member:

```
ex NAVINST
```

Follow the instructions in the Response column in the following table for each entry in the Screen column.

### Table 2–5 Installation Prompts and Responses

<table>
<thead>
<tr>
<th>Screen</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO YOU WANT TO USE SMS MANAGED STORAGE FOR THIS INSTALLATION Y/N [N]:</td>
<td>If you want to manage the storage using SMS, then answer Y, otherwise answer N.</td>
</tr>
<tr>
<td>THE SOFTWARE WILL BE INSTALLED UNDER THE HIGH LEVEL QUALIFIER THAT YOU WILL CHOOSE.</td>
<td>The high-level qualifier for the installation (referred to as <code>INSTROOT</code> throughout this guide). You can use more than one high-level qualifier (such as <code>ACME.DEV.VA10</code>). The total length must be less than or equal to twenty characters. The qualifiers can be the same as the ones used for the installation (step 1). The words &quot;transmit&quot; and &quot;buildkit&quot; cannot be used as high-level qualifiers.</td>
</tr>
<tr>
<td>ENTER THE HIGH LEVEL QUALIFIER [&quot;QUALIFIER&quot;]:</td>
<td>This prompt is displayed only if SMS is used to manage the installation (you answered Y to the first prompt). Enter the storage class</td>
</tr>
<tr>
<td>ENTER THE STORCLASS FOR TEMP DATASETS ['STORCLASS']:</td>
<td>This prompt is displayed only if SMS is not used to manage the installation (you answered N to the first prompt). The volume name for temporary datasets used during the installation procedure.</td>
</tr>
<tr>
<td>ENTER THE UNIT NAME FOR INSTALLATION TEMP DATASETS [3390]:</td>
<td>The unit name for temporary datasets used during the installation procedure.</td>
</tr>
<tr>
<td>ENTER THE VOLUME NAME FOR INSTALLATION TEMP DATASETS:</td>
<td>This prompt is displayed only if SMS is not used to manage the installation (you answered N to the first prompt). The volume name for temporary datasets used during the installation procedure</td>
</tr>
<tr>
<td>PLEASE CONFIRM (YES/NO/QUIT) [YES]:</td>
<td>Confirm the entered details</td>
</tr>
</tbody>
</table>

---

2-6  Oracle Application Server Adapters for VSAM User’s Guide
Table 2–5 (Cont.) Installation Prompts and Responses

<table>
<thead>
<tr>
<th>Screen</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTER THE OUTPUT CLASS FOR INSTALLATION OUTPUT [A]:</td>
<td>Enter the output class for Oracle Connect output. Assigning a device which is set on HOLD prevents the loss of log information when the Oracle Connect started tasks finish (the default is $A$).</td>
</tr>
<tr>
<td>DO YOU WANT TO USE THE DEFAULT JOB CARD Y/N [Y]</td>
<td>A job card is displayed. If you want to use a replacement card, then it must be entered as it will appear in the job. You can enter up to six lines. Enter a blank card to end input. If you do not enter a card, then the Oracle Connect default card is used.</td>
</tr>
<tr>
<td>ADDING AND UPDATING ORACLE CONNECT FOR VSAM CONFIGURATION ON THIS MACHINE, FROM A REMOTE ORACLE ADMINISTRATION CONSOLE, CAN ONLY BE DONE BY SOMEONE DEFINED AS AN ADMINISTRATOR FOR ORACLE CONNECT ON THIS MACHINE. ENTER A VALID USER NAME FOR AN ORACLE CONNECT ADMINISTRATOR [ALL]:</td>
<td>To manage Oracle Connect on this computer from Oracle Studio, you need to enter a user account of a user who will have administrative authorization, or press Enter to enable any user to administer Oracle Connect on this computer. The administrative rights can be changed from within Oracle Studio after the installation.</td>
</tr>
<tr>
<td>DO YOU WANT TO PERFORM A MANUAL (M) OR AUTOMATIC (A) INSTALLATION [A]:</td>
<td>If you want to review the JCL used to install Oracle Connect, before it is submitted, then respond M for a manual installation.</td>
</tr>
<tr>
<td>PLEASE REVIEW AND SUBMIT FOR EXECUTION THE DSN1 (INSTJBOR)</td>
<td>This prompt is displayed only if a manual installation is requested (you answered M to the previous prompt). DSN1 is the dataset name where INSTJBOR is located.</td>
</tr>
</tbody>
</table>

5. In the $nnn$.BUILDKIT.SRC library, run the VSAM member:

```
ex VSAM
```

Follow the instructions in the Response column in the following table for each entry in the Screen column.

Table 2–6 VSAM Adapter-Specific Installation Prompts and Responses

<table>
<thead>
<tr>
<th>Screen</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO YOU WANT ORACLE CONNECT FOR LEGACY ADAPTER TO WORK WITH VSAM UNDER CICS (YES/NO) [NO]:</td>
<td>Answer YES to this prompt if you want to access VSAM data under CICS.</td>
</tr>
<tr>
<td>ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHEXCI]:</td>
<td>If you responded YES to working with VSAM under CICS, then enter the CICS EXCI load library name only if you do not want the default.</td>
</tr>
<tr>
<td>PLEASE CONFIRM (YES/NO/QUIT) [YES]:</td>
<td>If you responded YES to working with VSAM under CICS, then confirm the entered details.</td>
</tr>
<tr>
<td>ENTER THE ISPF LOAD LIBRARY NAME [ISPSISPLOAD]:</td>
<td>Enter the ISPF load library name only if you do not want the default.</td>
</tr>
</tbody>
</table>
The installation is completed. All JCL jobs and REXX procedures are written to the
\texttt{INSTROOT.USERLIB} library. \texttt{INSTROOT} is the high-level qualifier for the installation.

**Post-Installation Instructions**

The following post-installation tasks must be done to work with Oracle Connect:

- **Post-Installation Procedures**
- **Starting the Daemon**
- **Setting Up Oracle Connect for Reentrancy**

**Post-Installation Procedures**

Perform the following procedures after completing the installation, to configure Oracle Connect.

- Allocate a dataset for \texttt{INSTROOT.DEF.BRANDBIN}, using 1 track and with\texttt{RECFM=VB} and \texttt{LRECL=256}. The block size is 6233.

  \texttt{INSTROOT} is the high-level qualifier where Oracle Connect is installed.

  Using FTP, copy the \texttt{BRANDBIN} file, in binary mode, from the \texttt{Oracle_Connect\VSAM\Legacy\Adapter} directory to the mainframe, to \texttt{INSTROOT.DEF.BRANDBIN}.

- Define the \texttt{LOADAUT} library as an APF-authorized library

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

```
*/setprog apf,add,dsn=INSTROOT.loadaut,volume=vol002*
```

Where \texttt{vol002} is the volume where you installed Oracle Connect and \texttt{INSTROOT} is the high-level qualifier where Oracle Connect is installed.

If the site uses SMS, then when defining APF-authorization in the SDSF screen, enter the following command:

```
*/setprog apf,add,dsn=INSTROOT.loadaut,SMS*
```

Ensure that the library is APF-authorized, even after an IPL (restart) of the computer.
Move the `INSTROOT.USERLIB(ATTTDAEMN)` and `INSTROOT.USERLIB(ATTSRVR)` members to any active proclib (such as `user.proclib`). `ATTTDAEMN` and `ATTSRVR` are run as started tasks.

If you decide to change the name of the `ATTSRVR` member when you move it to a general high-level qualifier, then change the name specified in the `StartupScript` parameter in the daemon configuration to the new name:

- Run `INSTROOT.USERLIB(NAVCMD)` and enter `EDIT DAEMON IRPCDINI` at the prompt.
- Change the `startupScript` parameter from `ATTSRVR` to the new name for the server:

```
<Workspace name="Navigator"
    startupScript="NEW_NAME"
    serverMode="reusable"
    ... />
```

- Exit and save the change.

Change the following line in the `ATTDAEMN` script to include the IP address and port of the IBM z/OS platform.

For example, before:

```
// PARM='-B START IRPCDINI'
```

After:

```
// PARM='-B -L ip_address:2551 START IRPCDINI'
```

Where `ip_address` is the ip address of the computer, 2551 is the default port for starting the daemon and `IRPCDINI` is the default daemon configuration.

- The `ATTDAEMN` and `ATTSRVR` started tasks need permission to use an Open Edition TCP/IP stack. The owner must be a user with OMVS segment defined and OMVS UID= 0000000000.

- In the security manager, such as RACF, define `ATTDAEMN` and `ATTSRVR` with a started task class and a general profile that enables the following:
  - Permission to issue master console commands.
  - `START` authority for the `ATTSRVR` Job.
  - Access to an Open z/OS segment (that defines access to TCP/IP OA sockets).
  - `ALTER` authority on datasets under `INSTROOT` (to access to read, write, allocate and delete datasets under `INSTROOT`).

- The installation includes a PS, `INSTROOT.DEF.GBLPARMS`, that contains global environment information. This PS is read at startup and the correct software version is used, based on the details provided in the startup task.

If you change the location of this member, then you must also change the relevant cards in the following jobs to the new locations:

- `ATTSRVR`: located in an active proclib, such as `user.proclib`)
- `ATTDAEMN`: located in an active proclib, such as `user.proclib`)
- `NAVSQL`: located in `INSTROOT.USERLIB`

- The input during the installation procedure is written to `nnn.BUILDKIT.SRC(PARS)`. You can use this file to provide the same inputs if
you rerun the installation, where \textit{nnn} is the high-level qualifier you assign for the installation.

- For information about specifying Oracle Connect as the service using port 2551 in the TCP/IP network services file, consult TCP/IP documentation.

**Starting the Daemon**

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

`'/s ATTDAEMN'`

\textit{INSTROOT} is the high-level qualifier where Oracle Connect is installed.

To submit the daemon as a job, uncomment the first two lines of the \texttt{ATTDAEMN} JCL, change the PARM line as described earlier, and run the job using the sub command. The \texttt{ATTDAEMN} JCL is similar to the following:

```plaintext
//*ATTDAEMN JOB 'RR','TTT',MSGLEVEL=(1,1),CLASS=A,
//* MSGCLASS=A,NOTIFY=&SYSUID,REGION=8M
//STEP1 EXEC PGM=IRPCD,
// PARM= '-B  START IRPCDINI'
//* PARM= '-b  -L :8883 START'
//STEPLIB DD DSN=INSTROOT.LOADAUT,DISP=SHR
//SYSPRINT DD SYSOUT=A
//GBLPARMS DD DSN=INSTROOT.DEF.GBLPARMS,DISP=SHR
// EXEC PGM=IRPCD,COND=((1,EQ,STEP1),(2,EQ,STEP1)),
// PARM= '-KATTDAEMN START ''INSTROOT.DEF.IRPCDINI'''
//STEPLIB DD DSN=INSTROOT.LOADAUT,DISP=SHR
//SYSPRINT DD SYSOUT=A
//GBLPARMS DD DSN=INSTROOT.DEF.GBLPARMS,DISP=SHR
//SYSDUMP DD DUMMY
```

**Setting Up Oracle Connect for Reentrancy**

All Oracle Connect load modules are reentrant to enable sub-tasking. Therefore, move \texttt{INSTROOT.LOAD} to the Link Pack Area (LPA).

Where \textit{INSTROOT} is the high-level qualifier where Oracle Connect is installed.

Using the LPA reduces real storage usage (because everyone shares the LPA copy) and fetch time.

---

**Note:** If you intend on using impersonation, so that you can run in a security context that is different than the context of the process that owns the server, then do the following:

- Place the \texttt{INSTROOT.LOAD(ATYSVCW)} member in an APF-authorized library outside the LPA.

- Change the \texttt{ATTSRVR} member (located in the active proclib), by adding the following to the STEPLIB list:

  ```plaintext
  // DD DSN=apf_library,DISP=SHR
  ```

  Where \textit{apf_library} is the APF-authorized library outside the LPA where the \texttt{ATYSVCW} member was moved.

---

**Setting Up Oracle Connect to Update VSAM Data**

Perform the following to set up Oracle Connect so that you can update VSAM data.
VSAM Running Under CICS To set up Oracle Connect so that you can update VSAM data from a CICS transaction, copy the UPDTRNS load module from INSTROOT.LOAD to a CICS DFHRPL library, such as CICS.USER.LOAD, and then define the UPDTRNS program under CICS using any available group such as ORA group:

CEDA DEF PROG(UPDTRNS) G(ORA) LANG(C) DA(ANY) DE(ORACLE VSAM UPDATE PROG)

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

After defining the UPDTRNS program to a group, install it as follows:

CEDA IN G(ORA)

VSAM Not Running Under CICS To enable Oracle Connect to create and delete VSAM data, run the following JCL:

// IDCSYSIN DD DSN=&&VSAM,DISP=(NEW,DELETE,DELETE),
// SPACE=(TRK,(1)),UNIT=SYSDA,
// DCB=(BLKSIZE=3200,LRECL=80,RECFM=FB)

Updating an Existing Oracle Connect Installation with VSAM

Verify that you have all the information detailed in the following installation worksheets, so you can refer to it during the configuration process.

<p>| Table 2–7 Preinstallation Information |</p>
<table>
<thead>
<tr>
<th>Topic</th>
<th>Required Information</th>
<th>Default</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS</td>
<td>CICS EXCI load library name</td>
<td>CICS.CICS.SDFHEXCI</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2–8 Required Permissions

<table>
<thead>
<tr>
<th>Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permission to read the CICS EXCI library</td>
</tr>
</tbody>
</table>

In the nnn.BUILDKIT.SRC library, run the VSAM member:

ex VSAM

Follow the instructions in the Response column in the following table for each entry in the Screen column.

<p>| Table 2–9 VSAM Adapter Installation Prompts and Responses |</p>
<table>
<thead>
<tr>
<th>Screen</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO YOU WANT ORACLE CONNECT FOR LEGACY ADAPTER TO WORK WITH VSAM UNDER CICS (YES/NO) [NO]:</td>
<td>Answer YES to this prompt if you want to access VSAM data under CICS.</td>
</tr>
<tr>
<td>ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHEXCI]:</td>
<td>If you responded YES to working with VSAM under CICS, then enter the CICS EXCI load library name only if you do not want the default.</td>
</tr>
<tr>
<td>PLEASE CONFIRM (YES/NO/QUIT) [YES]:</td>
<td>If you responded YES to working with VSAM under CICS, then confirm the entered details.</td>
</tr>
<tr>
<td>ENTER THE ISPF LOAD LIBRARY NAME [ISPSISPLOAD]:</td>
<td>Enter the ISPF load library name only if you do not want the default.</td>
</tr>
</tbody>
</table>
Installing Oracle Studio

The installation is completed. All JCL jobs and REXX procedures are written to the INSTROOT.USERLIB library. INSTROOT is the high-level qualifier for the installation.

After completing the installation, perform post-installation tasks, as described in "Post-Installation Instructions" on page 2-8, as required.

Installing Oracle Studio

This following sections explain how to install Oracle Studio.

- Oracle Studio Requirements
- Installing Oracle Studio on Windows
- Installing Oracle Studio on Linux

Note: If you have Oracle Studio version 10.1.3.4 or higher installed on your computer because you are using an Oracle AS legacy adapter or OracleAS CDC adapter, you do not need to reinstall it. If you have an older version of Oracle Studio, you must install the newest version.

Oracle Studio Requirements

The following are the hardware requirements for Oracle Studio:

- Processor: Intel or 100% compatible computer, based on a Pentium processor
- Memory: 256 MB
- Disk space: 120 MB of free disk space

You can install Oracle Studio on the following operating systems:

- Windows XP with Service Pack 2 or higher
  - Windows 2003
  - Windows Vista
- Linux OS with GTK

Installing Oracle Studio on Windows

Oracle Studio is installed with a standard install wizard. Do the following to install Oracle Studio.
Run the installation file, either using the Run option in the Windows Start menu or through Windows Explorer. Follow the instructions on the wizard screen.

Installing Oracle Studio on Linux

The Oracle Studio installation on Linux can be carried out by one of the following:

- Installing Oracle Studio with the Wizard (SH installation)
- Installing with a Silent Installation (RPM)

Installing Oracle Studio with the Wizard (SH installation)

You can install Oracle Studio by using the installation wizard. This allows you to easily carry out any standard or custom installation. Do the following for the Linux SH installation.

1. Install into a directory where you have permission.
2. Change the mode to execute mode. Type in:
   
   `chmod +x <file name>`
3. Enter the following:
   
   `./<file name>`
   
   If you are not installing to the current directory, enter the full path.

Installing with a Silent Installation (RPM)

You can use the Linux RPM method to install Oracle Studio. Do the following to carry out a silent installation on Linux.

- Enter the following:
  
  `rpm -i <file name>`
  
  You can use RPM parameters to create a custom installation.

Configuring Oracle Connect

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.

To configure Oracle Connect, refer to the following sections:

- 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.
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 (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 username 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.

Securing Access to Oracle Connect

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

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

**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:

   ![The Preferences screen](image)

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

   ![The Change Master Password screen](image)
5. Leave the **Enter current master password** field blank and type a new master password.

6. Confirm the new password.

7. Click **OK**.

**Specifying Users with Administrative Rights**

By default, only the user who was specified 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 2-14.

---

**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 who logs on with that user name has the relevant authorization.

### 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](image)


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**.
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

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 2-14.
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:
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 2–6 The New Data Source screen](image)

<table>
<thead>
<tr>
<th>CICS Application ID:</th>
<th>CICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction ID:</td>
<td>EXCI</td>
</tr>
<tr>
<td>VTAM NetName:</td>
<td>ATHQ.EDN</td>
</tr>
<tr>
<td>Program Name:</td>
<td>LORE</td>
</tr>
<tr>
<td>Trace Queue:</td>
<td></td>
</tr>
</tbody>
</table>

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

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

See Also: “VSAM Running Under CICS” on page 2-11 for details about the adapter metadata

- 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.

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 2-14.

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 2-18.

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:

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 username 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 username as the high-level qualifier, enabling you to browse and transfer files.

The Select Resources screen is shown in the following figure:
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:
19. Click Next. The Apply Filters screen is displayed.

The Apply Filters screen is shown in the following figure:

**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.
20. Apply filters to the copybooks as required.

The following table lists the available filters:

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMP_6 switch</td>
<td>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.</td>
</tr>
<tr>
<td>Compiler source</td>
<td>The compiler vendor.</td>
</tr>
<tr>
<td>Storage mode</td>
<td>The MicroFocus Integer Storage Mode. Specify either NOIBMCOMP for byte storage mode or IBMCOMP is for word storage mode.</td>
</tr>
<tr>
<td>Ignore after column 72</td>
<td>Ignores columns 73 to 80 in the COBOL copybook.</td>
</tr>
<tr>
<td>IgnoreFirst6</td>
<td>Ignores the first six columns in the COBOL copybook.</td>
</tr>
<tr>
<td>Replace hyphens (-) in record and field names with underscores (_)</td>
<td>Replaces all hyphens in either the record or field names in the metadata generated from the COBOL with underscore characters.</td>
</tr>
<tr>
<td>Prefix nested columns</td>
<td>Prefix all nested columns with the previous level heading.</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>Specifies whether to be sensitive to the search string case.</td>
</tr>
<tr>
<td>Find</td>
<td>Searches for the specified value.</td>
</tr>
<tr>
<td>Replace with</td>
<td>Replaces the value specified for Find with the value specified here</td>
</tr>
</tbody>
</table>
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:
Configuring Oracle Connect

Oracle Application Server Adapters for VSAM User's Guide

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 will be 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 needs to be resolved in order 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>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields manipulation</td>
<td>Customizing the fields definitions, using the Fields Manipulation screen. You can also access this screen by double-clicking the required table record.</td>
</tr>
<tr>
<td>Rename</td>
<td>Renaming a table. This option is used especially when more than one table is generated from the COBOL with the same name.</td>
</tr>
<tr>
<td>Set data location</td>
<td>Setting the physical location of the data file for the table.</td>
</tr>
<tr>
<td>Set table attributes</td>
<td>Setting the table attributes.</td>
</tr>
<tr>
<td>XSL manipulation</td>
<td>Specifying an XSL transformation or JDOM document that is used to transform the table definitions.</td>
</tr>
</tbody>
</table>

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:
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:
28. For each table listed, specify the logical file name for the index.

29. Click Next.

30. Specify that you want to transfer the metadata from the computer to the IBM z/OS platform 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.

**Setting Up an Oracle Connect Adapter**

To work with the Oracle Connect against the VSAM data source from an Oracle Application Server, you need to 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 2-14.
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**
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 2-14.
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 2-29.
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 that you want to access with the interaction from the VSAM data source.

---

### Table 2–12 Adapter Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectString</td>
<td>Leave this value blank.</td>
</tr>
<tr>
<td>defaultDatasource</td>
<td>The name of the data source defined in Oracle Studio that you want to access with the Database adapter. For example, Legacy.</td>
</tr>
<tr>
<td>multipleResults</td>
<td>Leave this value as true.</td>
</tr>
</tbody>
</table>

**Note:** You must specify the VSAM data source name you used to define in Oracle Studio for the defaultDatasource property.
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 that you want 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, together 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>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>passThrough</td>
<td>Defines whether the query is passed directly to the back-end database for processing or processed by the Query Processor.</td>
</tr>
<tr>
<td>Reuse compiled query</td>
<td>Defines whether the query is saved in cache for reuse.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Sets the encoding method used to return binary data in text format. You can select between the base 64 and the hexadecimal encoding methods.</td>
</tr>
<tr>
<td>Event</td>
<td>Defines whether the interaction mode is sync-send or sync-receive.</td>
</tr>
<tr>
<td>Fail on no rows return</td>
<td>Defines whether an error is returned in case no data is returned</td>
</tr>
<tr>
<td>Root element</td>
<td>Defines the root element name for records returned by the query, using the <code>&lt;root&gt; \ &lt;record&gt;</code> format.</td>
</tr>
<tr>
<td>Record element</td>
<td>Defines the record element name for records returned by the query, using the <code>&lt;root&gt; \ &lt;record&gt;</code> format.</td>
</tr>
<tr>
<td>Max. records</td>
<td>Sets the maximum number of record returned by the query.</td>
</tr>
<tr>
<td>Null string</td>
<td>Sets the string returned in place of a null value. If not specified, then the column is skipped.</td>
</tr>
</tbody>
</table>

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

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

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".

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 2-31.
To deploy and integrate OracleAS Adapter for VSAM with Oracle Application Server Containers for J2EE (OC4J), you need to configure the J2CA 1.5 VSAM adapter.

This section includes the following topics:

- Integrating OracleAS Adapter for VSAM with OC4J
- Configuring the J2CA 1.5 VSAM Adapter
- Using the CCI API to Develop Applications
- Configuring Multiple Adapters
- Updating Configuration Information

**Integrating OracleAS Adapter for VSAM with OC4J**

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). OC4J is J2EE certified and provides all the J2EE specific containers, APIs, and services. OC4J 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 OC4J container.

The contract between the OC4J client application and the resource adapter is defined by the Common Client Interface (CCI). The contract between the OC4J 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 carry out 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).


### Configuring the J2CA 1.5 VSAM Adapter

To connect to the J2CA 1.5 VSAM adapter, perform the following steps:

1. Edit the `oc4j-ra-xml` file, which is located at the following path:
   
   \root\j2ee\home\application-deployment\default\oracle\oc4j-ra-xml

   Where **root** is the Oracle Application Server root directory.

2. Set the following settings for each connection:

   ```xml
   <oc4j-connector-factories>
   <connector-factory location=" " connector-name="Oracle Legacy Adapter">
     <config-property name="userName" value=" "/>
     <config-property name="password" value=" "/>
     <config-property name="eisName" value=" "/>
     <config-property name="serverName" value=" "/>
     <config-property name="workspace" value=" "/>
     <config-property name="portNumber" value=" "/>
     <config-property name="persistentConnection" value=" "/>
     <config-property name="keepAlive" value=" "/>
     <config-property name="firewallProtocol" value=" "/>
     <config-property name="connectTimeout" value=" "/>
     <config-property name="encryptionProtocol" value=" "/>
     <config-property name="encryptionKeyName" value=" "/>
     <config-property name="encryptionKeyValue" value=" "/>
     <config-property name="fakeXa" value=" "/>
   </connector-factory>
   </oc4j-connector-factories>
   
   The following table lists the properties that must be specified, and optional properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>Specifies the JNDI location where Oracle Application Server should bind the connection factory instance for application components.</td>
</tr>
<tr>
<td>eisName</td>
<td>Sets the name of the adapter to use. The adapter is defined in the Oracle Connect server using Oracle Studio, as described in &quot;Securing Access to Oracle Connect&quot; on page 2-14.</td>
</tr>
</tbody>
</table>
Using the CCI API to Develop Applications

You can develop applications to run adapter interactions using the Common Client Interface (CCI) API.

Perform the following steps to use the CCI API with the J2CA 1.5 VSAM adapter:

1. Look up a `ConnectionFactory` for the J2CA 1.5 VSAM adapter.

### Table 3–1 (Cont.) OC4J Connection Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>serverName</code></td>
<td>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 2-14. See Also: Appendix C, &quot;Advanced Tuning of the Daemon&quot; for details about the daemon.</td>
</tr>
<tr>
<td><code>workspace</code></td>
<td>Specifies to the name of a Oracle Connect server workspace to use. The default workspace is Navigator. See Also: &quot;Workspaces&quot; for details about workspaces.</td>
</tr>
<tr>
<td><code>portNumber</code></td>
<td>Specifies the TCP/IP port where the Oracle Connect daemon is running on the server. The default port is 2551.</td>
</tr>
<tr>
<td><code>userName</code></td>
<td>Specifies a user who can access the Oracle Connect server. The user is defined in the Oracle Connect daemon configuration. See Also: &quot;Security&quot; and workspace &quot;Security&quot; for details about users allowed to access an Oracle Connect server.</td>
</tr>
<tr>
<td><code>password</code></td>
<td>Specifies a valid password for the user.</td>
</tr>
<tr>
<td><code>persistentConnection</code></td>
<td>Set to true or false. When set to true, connections can persist across multiple requests or connection context changes. It is recommended to set this property to true.</td>
</tr>
<tr>
<td><code>keepAlive</code></td>
<td>Set to true or false. When set to true, the socket used for the connection is always kept open. It is recommended to set this property to true.</td>
</tr>
<tr>
<td><code>firewallProtocol</code></td>
<td>Specifies the firewall protocol used: either none or fixedNat (the Nat protocol using a fixed address for the daemon). The default is none.</td>
</tr>
<tr>
<td><code>connectTimeout</code></td>
<td>Specifies the connection timeout in seconds. The default is 0, meaning that there is no connection timeout.</td>
</tr>
<tr>
<td><code>encryptionProtocol</code></td>
<td>Specifies the name of encryption protocol to use. The default is null. The RC4 protocol is supported.</td>
</tr>
<tr>
<td><code>encryptionKeyName</code></td>
<td>Specifies the name of the symmetric encryption key to use.</td>
</tr>
<tr>
<td><code>encryptionKeyValue</code></td>
<td>Specifies the value of the symmetric encryption key to use.</td>
</tr>
<tr>
<td><code>fakeXa</code></td>
<td>When set to true, the XA APIs are internally converted to local transaction APIs. Always set to true.</td>
</tr>
</tbody>
</table>

Using the CCI API to Develop Applications

You can develop applications to run adapter interactions using the Common Client Interface (CCI) API.

Perform the following steps to use the CCI API with the J2CA 1.5 VSAM adapter:

1. Look up a `ConnectionFactory` for the J2CA 1.5 VSAM adapter.
2. Create a **Connection** object using this **ConnectionFactory**. A **Connection** is a handle to the underlying network connection to the EIS, which is identified in the *oc4j-ra.xml* file by the `serverName` property.

3. Create a **Connection** object using this **ConnectionFactory**. Specify the interaction properties using an **AttuInteractionSpec** object. The **AttuInteractionSpec** object holds properties for driving an interaction with an EIS instance. It is used by the interaction to run the specified function on the underlying EIS. It has the following format:

```java
AttuInteractionSpec(java.lang.String name, int verb, int timeOut)
```

The following table describes the properties that can be specified:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Specifies the interaction name</td>
</tr>
<tr>
<td>verb</td>
<td>Specifies the mode for the interaction: SYNC_SEND, SYNC_SEND_RECEIVE, or SYNC_RECEIVE.</td>
</tr>
<tr>
<td>timeOut</td>
<td>Specifies the time (in milliseconds) to wait for an EIS to run the specified interaction.</td>
</tr>
</tbody>
</table>

The following is an **InteractionSpec** sample:

```java
AttuInteractionSpec isSpec = new AttuInteractionSpec("query",
javax.resource.cci.InteractionSpec.SYNC_RECEIVE, 60);
javax.resource.cci.RecordFactory rf = new AttuRecordFactory(con,
mcf.getLogger());
javax.resource.cci.MappedRecord queryRecord = rf.createMappedRecord("query"),
queryRecord.put("##text", "select * from disam:nation");
javax.resource.cci.Record oRec = interaction.execute(ispec, queryRecord);
```

4. Invoke the **execute()** method on the **interaction** to initiate a call to the EIS. Pass any data for the interaction as input and output records.

5. After the interactions have been processed, close the **Interaction** and **Connection** objects.

### Configuring Multiple Adapters

Each J2CA 1.5 VSAM adapter requires an entry in the *oc4j-ra.xml* file as described in "Configuring the J2CA 1.5 VSAM Adapter" on page 3-2.

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

### Updating Configuration Information

You can change configuration settings for a resource adapter by editing the relevant `connector-factory` entry in the *oc4j-ra.xml* file. For these changes to take effect, you need to stop and restart Oracle Application Server.
This chapter contains the following topics:

- Getting Started
- Deploy a BPEL Outbound Process

Getting Started

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

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

- Personal computer running Microsoft Windows XP or Windows Server 2003 with one GB of RAM.
- The VSAM adapter that you are using must be deployed to the Oracle Application Server.
- OracleAS VSAM Adapter must be configured for outbound processing. See Generating 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 Chapter 3, "Deployment and Integration with the Oracle OC4J Container" for more information.
- How to configure the OracleAS adapters for VSAM using Oracle Studio. See Configuring Oracle Connect for more information.

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

- **Design Time:** The OracleAS adapters for VSAM are configured with Oracle Studio as described in Configuring Oracle Connect.
- **Runtime:** After you deploy the BPEL process with JDeveloper, you can test your configurations. See Runtime Configuration.
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 resource adapters using JDeveloper. The following sections describe how to deploy a BPEL outbound process. The following steps describe the steps necessary to carry out this outbound sample.

- **Beginning Tasks**
- **Design-Time Configuration**
- **Runtime Configuration**

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

**Beginning Tasks**

Before you design the outbound process, you must carry out the following:

- Integrate the OracleAS Adapter for VSAM with OC4J. See Integrating OracleAS Adapter for VSAM with OC4J for more information.
- Configure the connection for the J2CA VSAM resource adapter. You do this by creating a connection factory. See Configuring the J2CA 1.5 VSAM Adapter for information on how to do this.
- Write a schema file. First, create an interaction called findDoctor using Oracle Studio. See Configuring an OracleAS Adapter for VSAM for Outbound Interactions for a description on how to create the findDoctor interaction.

Write a schema file called FINDDOCTORSchema.xsd to define the message schema. The following is an example of the findDoctor.xsd file:

```xml
<schema targetNamespace="http://xmlns.oracle.com/Esb/findDoctorData"
xmlns:cust="http://xmlns.oracle.com/Esb/findDoctorData"
xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="findDoctorata" type="cust:findDoctorDataType"/>
  <complexType name="findDoctorDataType">
    <sequence>
      <element name="Record1" type="string"/>
    </sequence>
  </complexType>
</schema>
```

- Create a BPEL PM Server Connection in JDeveloper.
- Create the WSDL for the interaction. You do this in JDeveloper Connections explorer before you begin to build your endpoint in JDeveloper. JDeveloper will create the WSDL automatically based on the interaction you create. In the Connections explorer, browse for the Oracle Connect server you configured in Oracle Studio (see Installing Oracle Connect on an IBM z/OS Series Platform), then double-click the findDoctor interaction to view and create the WSDL.

See Configuring an OracleAS Adapter for VSAM for Outbound Interactions for an explanation on how to create the findDoctor interaction and for an explanation on the WSDL.
Design-Time Configuration

This section describes the design-time steps necessary to deploy two BPEL outbound processes. The processes use the Web service called 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
- Create a Database Adapter to Read the Doctor Data
- Design a BPEL Outbound Process to Read the String
- Create a Partner Link to the Database Adapter
- Edit the Invoke_1 Activity
- Edit the Assign_1 Activity

Create a BPEL Project for a BPEL Outbound Process

Do the following to create a BPEL project for outbound.

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

Figure 4–1 Outbound Project Settings

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

You create a database adapter to read the DOCTOR data. SOA executes the SQL statement 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 database adapter.

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

   **Figure 4–2 Partner Link for Database Adapter**

2. Click the Adapter Wizard button. This is the third button in the WDSL Settings section of the Create Partner Link dialog box, as shown in the figure above.

3. Enter the following information in the Adapter Configuration wizard:

   a. Step 1, Adapter Type: Select Database Adapter.

   b. Step 2, Service Name: type findDoctor

   c. Step 3, Service Connection. In this step select the data you want to work with.

      Click New to open the Create Connection wizard.

      **Step 1**: Select the connection type:

      Connection Name: dbConnection1

      Connection Type: Oracle (JDBC)

      **Step 2**: Authentication.

      Enter the User Name, Password, and Role for the connection

      **Step 3**: Enter the connection information.

      Enter the driver type (thin,oci), the name of the computer where JDBC is deployed, and its port number. This information should be the same as defined in the JDBC connection factory. See Configuring the J2CA 1.5 VSAM Adapter.

      Select Service Name and enter findDoctor.

      Click Finish to go back to the Adapter Configuration wizard.

   d. Step 4, File Directories

      Directory Names are Specified as: Select Physical Path
Deploy a BPEL Outbound Process

Directory for Incoming Files (physical path): Enter the full path to the folder where you want the incoming files, for example C:\temp.

e. Step 5, File Filtering:
   Includes Files with Name Pattern: Enter *.xml

f. Step 6, File Polling:
   Polling Frequency: enter 1 then select seconds from the list.

g. Step 7, Messages:
   Schema File URL: Enter findDoctor.xsd
   Schema Element: Enter FindDoctorData

4. Click Finish.

Design a BPEL Outbound Process to Read the String

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

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

   Figure 4–3  Outbound Process

2. In the Visual Editor, connect the Receive_1 activity to the findDoctor database adapter. The Edit Receive dialog box is displayed.
3. Do the following in the Edit Receive dialog box:
   - Enter a name for the Receive activity, in this example it is `Receive_1`.
   - Make sure to create the variable. Click the first button to the right of the `Variable` field, then click `OK` in the Create Variable dialog box that is displayed.
   - Make sure that the `Create Instance` check box is selected.
   - Click `OK` to close the dialog box and accept the information.

**Create a Partner Link to the Database Adapter**

In this step, create a partner link to the database adapter you defined when you created the IMD/DB configuration in Oracle Studio. Do the following to create a partner link.

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

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

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

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

**Edit the Invoke_1 Activity**

Edit the Invoke_1 activity that you added to the Visual Editor so that it will invoke the FINDDOCTOR service. Do the following to edit the Invoke_1 activity.

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

   ![Invoke Dialog Box]

2. In the Invoke dialog box, make sure that the Partner Role Web Service Interface section has the following values:
   - Partner Link: FINDDOCTOR
   - Operation: FINDDOCTOR

3. Click the Automatically Create Input Variable button. It is the first icon to the right of the Input Variable field as shown in the figure in the previous step. The Create Variable dialog appears with the input variable.

4. The Input Variable should be Invoke_1_FINDDOCTOR_InputVariable. Click OK.

5. Click the Automatically Create Input Variable button for the Output Variable field.

6. The Output Variable should be Invoke_1_FINDDOCTOR_OutputVariable. Click OK.
Edit the Assign_1 Activity

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

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

![Assign Dialog Box](image)

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

![Create Copy](image)

3. On the From side, expand Variables then expand Read_1_SelectDoctor_InputVariable, then expand nsx:findDoctor and select Record1.

4. On the To side, expand Variables then expand Read_1_SelectDoctor_InputVariable, then expand input:findDoctor, then expand nsx:Record1 and select Record1_data.
5. Click OK. The Copy Operation tab in the Assign window updates to show the rule.

The following figure shows how the final process looks:

**Figure 4–9 Final Outbound Process**

![Diagram of Final Outbound Process]

**Runtime Configuration**

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

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

   ```xml
   <?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>
   </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
   <?xml version="1.0"?>
   <findDoctorData xmlns="http://xmlns.oracle.com/Esb/findDoctorData">
     <Record1 xmlns>"">VSAM</Record1>
   </findDoctorData>
   ```
This chapter provides an example for creating an OracleAS adapter for VSAM. In this case it describes the steps necessary to create an adapter that will get find the names of a doctor and a patient from VSAM data. To work with an OracleAS adapter for VSAM, you must also configure a BPEL endpoint connection. For information on how to create an endpoint, see BPEL Process Manager Examples.

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 will be used to access the data.

Requirements

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

- Computer with Windows operating system
- Mainframe computer with z/OS operating system
- Oracle Connect v 10.1.3.4 installed on the Mainframe computer
- Oracle Studio v 10.1.3.4 installed on the Windows computer.

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

Prepare the System

Make sure that the Hospital database is available under VSAM on the Mainframe computer.
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 with 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 will be the host name and the port number).
- **User name**: If the computer you are accessing needs an administrator password. If so enter the name of the computer’s administrator.
- **Password**: If necessary, enter the computer administrator’s password.
- **Connect via NAT with fixed IP address**: Select this if the machine uses the NAT (Network Address Translation) firewall protocol, with a fixed configuration, mapping each external IP to one internal IP, regardless of the port specified.

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

![Add Machine Dialog Box](image)

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:
Adding the VSAM Hospital Data Source

Configuring an OracleAS Adapter for VSAM for Outbound Interactions

- **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. Click **Finish**.

For detailed information on adding the VSAM 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](image)

**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 columns 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 will 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**.
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 BPEL Process Manager Examples.

Do the following to create the interactions.

- Add a Database Adapter
- Create Interactions

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:

![Add Adapter Dialog Box](image)

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 interaction, 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 information and exit Oracle Studio.
Create the findDoctor and findPatient Interactions
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 section contains the following topics:

- Troubleshooting the Daemon
- Resolving Communication Errors
- Resolving Specific Errors

Troubleshooting the Daemon

Troubleshooting the daemon and the communication between Oracle Application Server and OracleAS 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

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 6-2, 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.

**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:
Troubleshooting the Daemon

'/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

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" on page 2-14.
3. Right-click the computer and select Open Runtime Perspective.
4. In the Runtime Explorer, right-click the computer and select Shutdown Daemon.

Monitoring the Daemon During Run Time

Use the Runtime Manager perspective of Oracle Studio to monitor the daemon during 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" on page 2-14, 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.

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>
<thead>
<tr>
<th>Table 6-1 Daemon Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Option</strong></td>
</tr>
<tr>
<td>Edit Daemon Configuration</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Reload Configuration</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>View Log</td>
</tr>
<tr>
<td>View Events</td>
</tr>
</tbody>
</table>
Troubleshooting OracleAS Adapter for VSAM

### 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>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Daemon Properties</strong></td>
<td>Displays information about the computer where the daemon is running, such as the physical address and any username and password needed to access the computer.</td>
</tr>
<tr>
<td><strong>Recycle servers</strong></td>
<td>Closes all unused servers and prepares all active servers to close when the client disconnects. New connection requests are allocated with new servers.</td>
</tr>
<tr>
<td><strong>Kill servers</strong></td>
<td>Immediately closes all active and unused servers.</td>
</tr>
<tr>
<td><strong>Note:</strong> It is recommended to use this option with caution, as it may lead to data loss.</td>
<td></td>
</tr>
<tr>
<td><strong>Shutdown Daemon</strong></td>
<td>Shuts down the daemon on the computer.</td>
</tr>
<tr>
<td><strong>Rename</strong></td>
<td>Enables changing the name of the daemon displayed in the Runtime Explorer.</td>
</tr>
<tr>
<td><strong>Remove</strong></td>
<td>Removes the daemon from the Runtime Explorer.</td>
</tr>
<tr>
<td><strong>Refresh</strong></td>
<td>Refreshes the display.</td>
</tr>
</tbody>
</table>

**Table 6–2  Workspace Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Edit Workspace Configuration</strong></td>
<td>Opens the daemon editor to enable you to reconfigure the workspace. See Also: Appendix C, &quot;Advanced Tuning of the Daemon&quot; for details about the configuration settings.</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>Checks the status of the workspace whether it is available or not.</td>
</tr>
<tr>
<td><strong>Reload Configuration</strong></td>
<td>Reloads the configuration of the workspace after any changes. Any servers currently started are not affected by the changed configuration.</td>
</tr>
<tr>
<td><strong>View Log</strong></td>
<td>Displays the log for all servers for the workspace. For details see &quot;Daemon Logs&quot; on page 6-4</td>
</tr>
<tr>
<td><strong>View Events</strong></td>
<td>Displays the workspace events log.</td>
</tr>
<tr>
<td><strong>Recycle Servers</strong></td>
<td>Closes all unused servers and prepares all active servers to close when the client disconnects. New connection requests are allocated with new servers.</td>
</tr>
<tr>
<td><strong>Kill Servers</strong></td>
<td>Immediately closes all active and unused servers.</td>
</tr>
<tr>
<td><strong>Note:</strong> It is recommended to use this option with caution, as it may lead to data loss.</td>
<td></td>
</tr>
<tr>
<td><strong>Remove</strong></td>
<td>Removes the selected workspace from the Runtime Explorer.</td>
</tr>
<tr>
<td><strong>Disable</strong></td>
<td>Disables the selected workspace.</td>
</tr>
<tr>
<td><strong>Refresh</strong></td>
<td>Refreshes the display.</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>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.</td>
</tr>
<tr>
<td>View Log</td>
<td>Displays the server log. For details see &quot;Daemon Logs&quot; on page 6-4.</td>
</tr>
<tr>
<td>View Events</td>
<td>Displays the server events log.</td>
</tr>
<tr>
<td>Kill server</td>
<td>Ends the server process, regardless of its activity status. <strong>Note</strong>: It is recommended to use this option with caution, as it may lead to data loss.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refreshes the display.</td>
</tr>
</tbody>
</table>

Daemon Logs
Oracle Connect produces a number of 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.
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.

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.

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.

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>
<thead>
<tr>
<th>Scenario</th>
<th>Idle (Not Processing a Client Request)</th>
<th>Processing a Client Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit Disconnect&lt;br&gt;(client explicitly closes connection or client program terminates)</td>
<td>The server is immediately notified of the disconnect and either becomes available for use by another client or terminates (if it is not reusable).</td>
<td>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).</td>
</tr>
<tr>
<td>Abrupt Disconnect&lt;br&gt;(client closed without proper shutdown or client system hanged and communication disconnected)</td>
<td>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).</td>
<td>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).</td>
</tr>
</tbody>
</table>

To troubleshoot client/server communication problems, you need to 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.

**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.
Resolving Specific Errors

Troubleshooting OracleAS Adapter for VSAM 6-7

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

  See Also: "Setting Up Run-Time User Access to the IBM z/OS
  Platform" on page 2-17.

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
  username 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 username 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 username and
  password for the IBM z/OS platform.

  See Also: "Setting Up Run-Time User Access to the IBM z/OS
  Platform" on page 2-17.
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 username/password supplied when logging on to the daemon.
   Action: See the daemon log file for the reason that the username/password were
   not accepted.
   Action: Edit the user profile in Oracle Studio to specify a valid username and
   password for the IBM z/OS platform.

   Action: Make sure the daemon is started from an APF-authorized account that is
   allowed to check for system usernames 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.
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 oc4j-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.
**Resolving Specific Errors**

**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 username.

**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 for the reason that 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 abort. The currently active transaction is aborted as well.

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

C070: **Server failed to send reply to the client**

**Cause**: Server terminated unexpectedly.

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

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

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

**Action**: Normally, Oracle Connect automatically tries to create a new 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 crash 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.
Resolving Specific Errors

Troubleshooting OracleAS Adapter for VSAM

**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.
Advanced Features of OracleAS Adapter for VSAM

Oracle Connect includes a number of 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 section contains the following topics:

- Configuring the Daemon for High Availability
- Configuring a Binding Environment
- Migration Considerations
- Security Considerations
- Transaction Support

Configuring the Daemon for High Availability

The daemon workspace is responsible for allocating server processes to clients. You can configure a workspace to use a pool of server processes so that a server process is always available for a client request.

Use Oracle Studio to maintain the daemon and the daemon workspace parameters to control the allocation of server processes and their management in a pool.

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

Adding a New Daemon Workspace Configuration

Use Oracle Studio to add a new daemon configuration. You can set up different daemon configurations for different situations.

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 2-14.
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.

### Editing the Workspace

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

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

Use Oracle Studio to access these tabs, as follows:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.

2. In the Design perspective Configuration view, expand the **Machines** folder and then expand the machine where you want to edit the workspace.

3. Expand the **Daemons** folder. The daemon available on this computer are listed.

4. Expand the **IRPCD** daemon. The daemon workspaces are listed.

5. Right-click the workspace you are editing and select **Open**.

6. Click the tab that contains the information you want to edit. For full details of the tabs and the fields in these tabs, see **Workspaces**.

7. After editing the workspace, click **Save**.

### Configuring the Server Mode

The server mode dictates how the daemon starts up 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 already 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 already 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:
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 if you want 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**: 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 kill 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 killed. This must be done at the system level.

- **Number of prestarted servers in pool**: The number of server processes that are prestarted for this workspace when the daemon starts up. 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**: 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**: The number of sub-tasks for a server that are prestarted for this workspace when the daemon starts up. 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).

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:

![The Binding Properties tab](image)

The binding environment is divided into the following categories:

- Debug
- General
- Language
These categories are described in the following paragraphs.

### Debug

The following table lists the parameters that define debugging and logging operations:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACX trace</td>
<td>Select this for the 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.</td>
</tr>
<tr>
<td>GDB Trace</td>
<td>This parameter is not applicable for use with OracleAS Adapter for VSAM.</td>
</tr>
<tr>
<td>General trace</td>
<td>Select this to log general trace information. The default writes only error messages to the log. <strong>Note:</strong> Changing the default setting can degrade performance.</td>
</tr>
<tr>
<td>Query warnings</td>
<td>This parameter is not applicable for use with OracleAS Adapter for VSAM.</td>
</tr>
<tr>
<td>Add timestamp to traced events</td>
<td>Select this to add a timestamp on each event row in the log.</td>
</tr>
<tr>
<td>Query Processor trace</td>
<td>This parameter is not applicable for use with OracleAS Adapter for VSAM.</td>
</tr>
<tr>
<td>Binary XML Log Level</td>
<td>Select the binary XML log level from the list. The following logging levels are available:</td>
</tr>
<tr>
<td></td>
<td>- None</td>
</tr>
<tr>
<td></td>
<td>- API</td>
</tr>
<tr>
<td></td>
<td>- Info</td>
</tr>
<tr>
<td></td>
<td>- Debug</td>
</tr>
<tr>
<td>Log file</td>
<td>The high-level qualifier of the log file for messages. The following type of message are written to the log:</td>
</tr>
<tr>
<td></td>
<td>- Error messages</td>
</tr>
<tr>
<td></td>
<td>- Trace information and information about the query optimization strategy, if generalTrace is set to true.</td>
</tr>
</tbody>
</table>
General

The following table lists the parameters that define various operations the directory where temporary files are written.

<table>
<thead>
<tr>
<th>Table 7–3 General Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>NA UTIL editor</td>
<td>This parameter is not applicable for use with OracleAS Adapter for VSAM.</td>
</tr>
<tr>
<td>Temporary Dir</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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 will be represented by 00->04. All other two digits will map 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 will probably change 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 |

Language

The following table lists the parameters that define globalization support:
Modeling

The Modeling parameters are not applicable with OracleAS Adapter for VSAM.

ODBC

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

OLEDB

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

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.

Parallel Processing

This following list shows the parallel processing properties. The parallel processing properties control how parallel processes are handled in the binding.
Table 7–5  Parallel Processing Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable threads</td>
<td>Select this to disable multi-threading. If this is selected, the following properties are disabled:</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Disable threaded read ahead (QP):</strong> Select this to disable read-ahead functionality.</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Disable query read ahead (QP):</strong> Select this to disable read-ahead functionality for components using Query Processor services.</td>
</tr>
<tr>
<td></td>
<td>■ <strong>ODBC async execution</strong> property to enable ODBC asynchronous execution.</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Disable QP parallel execution:</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Hash parallelism:</strong> 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.</td>
</tr>
</tbody>
</table>

**Query Processor**

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

Table 7–6  Query Processor Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable command reuse</td>
<td>Select this to disable Query Processor caching the executed state of a query for reuse.</td>
</tr>
<tr>
<td>Disable DS property cache</td>
<td>Select this to disable caching data source properties.</td>
</tr>
<tr>
<td>Disable insert parameterization</td>
<td>Select this to disable parameterization constants in INSERT statements.</td>
</tr>
<tr>
<td>Disable metadata caching</td>
<td>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.</td>
</tr>
<tr>
<td>Disable query parametarization</td>
<td>Select this to not convert constants into parameters when accessing data sources.</td>
</tr>
<tr>
<td>Disable row mark field fetch</td>
<td>Select this for OLE DB getRows errors to be marked and reshown on every getRows, if the rowset is active.</td>
</tr>
<tr>
<td>Compile after load</td>
<td>Select this to always compiles an Oracle Connect procedure or view after it is read.</td>
</tr>
<tr>
<td>Ignore segments bind failure</td>
<td>This property determines how Oracle Connect responds when the execution of one of the segments of a segmented data source fails:</td>
</tr>
<tr>
<td></td>
<td>■ Select this to Log a message and continue execution. This is the default setting.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td>Prompt database-user password</td>
<td>Select this to configure Oracle Connect to prompt the user for security information when accessing a data source.</td>
</tr>
</tbody>
</table>
### Table 7–6 (Cont.) Query Processor Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use alternate qualifier</td>
<td>Select this to use the @ symbol instead of a colon (:) when connecting to multiple data sources.</td>
</tr>
<tr>
<td>Use table filter expression</td>
<td>Select this to enable the use of tables that have filter expressions specified in their metadata.</td>
</tr>
<tr>
<td>Write empty string as null</td>
<td>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.</td>
</tr>
<tr>
<td>Optimistic for update</td>
<td>Select this to use optimistic locking as the default locking behavior on queries with a FOR UPDATE clause.</td>
</tr>
<tr>
<td>Disable compilation cache</td>
<td>Select this to disable saving successfully compiled statements in the cache.</td>
</tr>
<tr>
<td>Maximum SQL cache</td>
<td>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.</td>
</tr>
<tr>
<td>First tree extensions</td>
<td>Enter the maximum size allowed for an SQL query after compilation. The default is 150.</td>
</tr>
<tr>
<td>Maximum columns in parsing</td>
<td>Enter the maximum number of columns that a query can reference. The default is 500.</td>
</tr>
<tr>
<td>Maximum segmented database threads</td>
<td>Enter the maximum number of open threads allowed, when working with segmented databases.</td>
</tr>
<tr>
<td>Minimum number of parameters allocated</td>
<td>Enter the minimum number of parameters that can be used in a query.</td>
</tr>
<tr>
<td>Continuous query retry interval</td>
<td>Enter the number of seconds that the query processor waits before executing a query again, when no records are returned. The default is 2.</td>
</tr>
<tr>
<td>Continuous query timeout</td>
<td>Enter the number of seconds that the query processor will continue 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 will end. Enter 0 to indicate that there is no timeout and the continuos query will not end automatically.</td>
</tr>
<tr>
<td>Continuous query prefix</td>
<td>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’.</td>
</tr>
</tbody>
</table>
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 will 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 \texttt{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 datatypes 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 \texttt{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 \texttt{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 \texttt{Expose XML fields} is selected.
Transaction

The following table lists the parameters that define transaction support:

<table>
<thead>
<tr>
<th>Table 7–7 Transaction Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td>Transaction extended logging</td>
</tr>
<tr>
<td>Commit on destroy</td>
</tr>
<tr>
<td>Disable 2PC</td>
</tr>
<tr>
<td>User commit confirm table</td>
</tr>
<tr>
<td>Transaction log file</td>
</tr>
<tr>
<td>Recovery Delay</td>
</tr>
<tr>
<td>Time limit</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Conversions</td>
</tr>
<tr>
<td>Select one:</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Tuning

The following table lists the parameters that define the tuning:

<table>
<thead>
<tr>
<th>Table 7–8 Tuning Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Dsm maximum buffer size</td>
</tr>
</tbody>
</table>
The following table lists the parameters that define XML support:

### Table 7–9 XML Category Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM maximum XML in memory</td>
<td>Specifies the maximum size of an XML document held in memory. The default is 65535 bytes.</td>
</tr>
</tbody>
</table>
| 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. |
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, any 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 2-14.
6. In the Configuration Explorer, right-click the target computer and select Import XML definitions.
7. Import the XML file to the target platform.

Security Considerations

Oracle Connect works within the confines of the platform security system. For example, on a 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML date format</td>
<td>Enter the date format to use for XML. The options are:</td>
</tr>
<tr>
<td></td>
<td>- ISO (the default): The date format is:</td>
</tr>
<tr>
<td></td>
<td>YY-MM-DDThh:mm:ss[.ss..]</td>
</tr>
<tr>
<td></td>
<td>- ODBC: The date format is: YYYY-MM-DD HH:MM:SS [.NNN...]</td>
</tr>
<tr>
<td>Replace invalid XML characters</td>
<td>Select this to replace invalid XML characters with a ‘?’ . This property is used for diagnostic and troubleshooting purposes.</td>
</tr>
<tr>
<td>XML trim char columns</td>
<td>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.</td>
</tr>
</tbody>
</table>
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 usernames 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.

**Setting Design Time Security**

Setting design time security is described in the following sections:

- Securing access to Oracle Studio is described in "Setting Password Access to Oracle Studio" on page 2-15.

- Securing rights to configure a computer in Oracle Studio is described in "Specifying Users with Administrative Rights" on page 2-16.

- Securing access to user profiles is accomplished by right-clicking the relevant user profile in Oracle Studio and selecting Change Master Password. In the dialog box that is displayed, specify a password that must be provided in the future to edit the specific user profile.

- Securing access to workspaces is accomplished by right-clicking the relevant workspace in Oracle Studio and selecting Set Authorization. In the dialog box that is displayed, specify a valid user and password that must be provided in the future to edit the specific workspace.

**Setting Run Time Security**

During 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.

### 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
```

Make sure 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.
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

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 metadata 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 Metadata View to display the Metadata view. The Metadata 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

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**

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

---

**Table A–1 (Cont.) General Tab Components**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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:  
    ```sql
    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 more than one logical table is stored in the same physical file. |
The **Columns** tab is shown in the following figure:

**Figure A–2 The Columns tab**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the column</td>
</tr>
<tr>
<td>Data type</td>
<td>The data type of the column. Selecting this field displays a drop-down box listing the possible data types.</td>
</tr>
<tr>
<td>Size</td>
<td>The size of the column</td>
</tr>
<tr>
<td>Scale</td>
<td>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.</td>
</tr>
</tbody>
</table>
Table A–2  (Cont.) Metadata Column Tab Definitions

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>The maximum number of occurrences of a group of columns that make up an array. The (+) to the left of a column indicates a group field. This type of field will have a Dimension value. Click (+) to display the group members.</td>
</tr>
<tr>
<td>Offset</td>
<td>An absolute offset for the field in a record.</td>
</tr>
</tbody>
</table>
| 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 will have a fixed offset.  
  
  The offset of a field is usually calculated dynamically by the server at runtime 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 there is a part of the buffer that you want to skip.  
  
  By selecting the check box, or by editing the offset value you pin the offset for that column. The indicated value is used at runtime 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

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert</td>
<td>Inserts a column to the table. You can insert a new column. If the table has arrays, you can add a new child column.</td>
</tr>
<tr>
<td>Up</td>
<td>Moves your selection to the column directly above where the currently selected column.</td>
</tr>
<tr>
<td>Down</td>
<td>Moves your selection to the column directly below where the currently selected column.</td>
</tr>
<tr>
<td>Rename</td>
<td>Lets you rename the selected column.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected column.</td>
</tr>
<tr>
<td>Find</td>
<td>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.</td>
</tr>
</tbody>
</table>

Column Properties
You can change the property value by clicking in the Value column. Follow these steps for displaying the column properties.

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 some of the properties available for selected columns.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>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.</td>
</tr>
<tr>
<td>Autoincrement</td>
<td>The current field is updated automatically by the data source during an <code>INSERT</code> statement and is not explicitly defined in the <code>INSERT</code> statement. The <code>INSERT</code> 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.</td>
</tr>
<tr>
<td>Comment</td>
<td>A short note or description about the column.</td>
</tr>
<tr>
<td>DB command</td>
<td>VSAM specific commands for the column The information displayed is generated automatically when the metadata is generated and should not be modified</td>
</tr>
<tr>
<td>Empty value</td>
<td>The value for the field in an insert operation, when a value is not specified.</td>
</tr>
<tr>
<td>Explicit Select</td>
<td>When <code>true</code>, the current field is not returned when you execute a <code>SELECT * FROM...</code> statement. To return this field, you must explicitly ask for it in a query, for example, <code>SELECT NATION_ID, SYSKEY FROM NATION where SYSKEY</code> 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.</td>
</tr>
<tr>
<td>Hidden</td>
<td>The current field is hidden from users. The field is not displayed when a <code>DESCRIBE</code> statement is executed on the table.</td>
</tr>
<tr>
<td>Non Selectable</td>
<td>When <code>true</code>, the current field is never returned when you execute an SQL statement. The field is displayed when a <code>DESCRIBE</code> statement is executed on the table.</td>
</tr>
<tr>
<td>Non Updateable</td>
<td>If <code>true</code>, the current field cannot be updated.</td>
</tr>
<tr>
<td>Nullable</td>
<td>This value allows the current field to contain <code>NULL</code> values.</td>
</tr>
<tr>
<td>Null value</td>
<td>The null value for the field during an insert operation, when a value is not specified.</td>
</tr>
<tr>
<td>Chapter of</td>
<td>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</td>
</tr>
<tr>
<td>OnBit</td>
<td>The position of the bit in a <code>BIT</code> field and the starting bit in a <code>BITES</code> field.</td>
</tr>
<tr>
<td>Subfield of</td>
<td>The value is generated automatically when you generate metadata from Adabas data that includes a supersdesciptor based on a subfield. A field is created to base this index on, set to the offset specified as the value of the <code>Subfield start</code> field. If no value is entered in the <code>Subfield start</code> field, the subfield is set by default to an offset of 1.</td>
</tr>
<tr>
<td>Subfield start</td>
<td>The offset within the parent field where a subfield starts.</td>
</tr>
</tbody>
</table>
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**

**Table Information**

The following table describes the fields for the top part of the tab, which defines the indexes used for the table.
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–5 Indexes Tab Components**

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

**Table A–6 Index Definition Buttons**

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert</td>
<td>Inserts an index to the table.</td>
</tr>
<tr>
<td>Rename Index</td>
<td>Lets you rename the selected index.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected index.</td>
</tr>
</tbody>
</table>

**Properties**

You can index properties for each index column. Follow these steps for displaying the properties for each index.

**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.
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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of rows</td>
<td>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.</td>
</tr>
<tr>
<td>No. of blocks</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

Table A–8  Columns group Components

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column name</td>
<td>The columns in the table.</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexes and segments</td>
<td>The indexes and segments in the table.</td>
</tr>
<tr>
<td>Cardinality</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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:
Use the Resolution group to specify the following:

**Table A–10  Type group Components**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated</td>
<td>An estimation of the amount of statistical information returned.</td>
</tr>
<tr>
<td>Estimated with rows</td>
<td>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, assuming that the value specified here is the correct value, or close to the correct value.</td>
</tr>
<tr>
<td>Exact</td>
<td>The exact statistical information returned. Note that this can be a lengthy task and can lead to disk space problems with large tables.</td>
</tr>
</tbody>
</table>

**Table A–11  Resolution group Components**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Only information about the table and indexes is collected. Information for partial indexes and columns is not collected.</td>
</tr>
<tr>
<td>All columns and indexes</td>
<td>Information about the table, indexes, partial indexes and columns is collected.</td>
</tr>
<tr>
<td>Select columns and indexes</td>
<td>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 a number of columns or indexes).</td>
</tr>
</tbody>
</table>

**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.
Advanced Tuning of the Metadata

– **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.

### 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.

### 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 that you want to edit, and select **Open**.

   The General properties editor is displayed.

The Adapter General Properties editor is shown in the following figure:
The **General** properties are listed in the following table:

**Table A–12 General tab Components**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Specifies an identifying description of the adapter.</td>
</tr>
<tr>
<td>Authentication mechanism</td>
<td>Specifies the authentication to access the adapter. The available mechanisms are:</td>
</tr>
<tr>
<td></td>
<td>■ kerbv5</td>
</tr>
<tr>
<td></td>
<td>■ none</td>
</tr>
<tr>
<td></td>
<td>■ basic password</td>
</tr>
<tr>
<td>Max request size</td>
<td>Specifies the maximum size in bytes for an XML request or reply. Larger messages are rejected with an error.</td>
</tr>
<tr>
<td>Max active connections</td>
<td>Specifies the maximum number of simultaneous connections for an adapter (per process).</td>
</tr>
<tr>
<td>Max idle timeout</td>
<td>Specifies the maximum time, in seconds, that an active connection can stay idle. After that time, the connection is closed.</td>
</tr>
<tr>
<td>Adapter Specifications</td>
<td>Specifies the adapter-specific properties for an interaction. The VSAM database adapter does not have any adapter-specific properties.</td>
</tr>
</tbody>
</table>
Adapter Metadata Interactions

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

1. In Oracle Studio Design perspective, Metadata view, expand the Adapters folder.
2. Expand the adapter with the interaction that you want 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:

![Adapter Metadata Interactions](image)

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Provides a descriptive identifier for the interaction.</td>
</tr>
</tbody>
</table>
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.

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

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

---

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 that you want 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**

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

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**

<table>
<thead>
<tr>
<th>Binary</th>
<th>Boolean</th>
<th>Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Double</td>
<td>Enum</td>
</tr>
</tbody>
</table>
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 that you want 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 .

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boolean</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Float</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>Numeric(p,s)</td>
<td>Short</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Timestamp</td>
<td></td>
</tr>
</tbody>
</table>

Table A–15  (Cont.)  Valid Data Types

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 that you want 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 .

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boolean</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Float</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>Numeric(p,s)</td>
<td>Short</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Timestamp</td>
<td></td>
</tr>
</tbody>
</table>
OracleAS Adapters for VSAM supports a number of 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 Adapters for VSAM and Oracle Application Server is performed internally by Oracle Connect.

### 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 7–10 Data Type Mapping: COBOL and Oracle Connect Back-end Adapter**

<table>
<thead>
<tr>
<th>COBOL Data Type</th>
<th>Oracle Connect Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BINARY (with fractional data)</td>
<td>string</td>
</tr>
<tr>
<td>BINARY (without fractional data)</td>
<td>int</td>
</tr>
<tr>
<td>COMP (with fractional data)</td>
<td>string</td>
</tr>
<tr>
<td>COMP (without fractional data)</td>
<td>int</td>
</tr>
<tr>
<td>COMP-2</td>
<td>double</td>
</tr>
<tr>
<td>COMP-3</td>
<td>string</td>
</tr>
<tr>
<td>COMP-4 (with fractional data)</td>
<td>string</td>
</tr>
<tr>
<td>COMP-4 (without fractional data)</td>
<td>int</td>
</tr>
<tr>
<td>COMP-5 (with fractional data)</td>
<td>string</td>
</tr>
<tr>
<td>COMP-5 (without fractional data)</td>
<td>int</td>
</tr>
<tr>
<td>COMP-X (with fractional data)</td>
<td>string</td>
</tr>
<tr>
<td>COMP-X (without fractional data)</td>
<td>int</td>
</tr>
<tr>
<td>INDEX</td>
<td>int</td>
</tr>
<tr>
<td>[SIGN [IS]] LEADING</td>
<td>string</td>
</tr>
<tr>
<td>[SIGN [IS]] LEADING SEPARATE [CHARACTER]</td>
<td>string</td>
</tr>
<tr>
<td>NATIVE-2</td>
<td>int</td>
</tr>
<tr>
<td>COBOL Data Type</td>
<td>Oracle Connect Data Type</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>NATIVE-4</td>
<td>int</td>
</tr>
<tr>
<td>NATIVE-8</td>
<td>string</td>
</tr>
<tr>
<td>PACKED-DECIMAL</td>
<td>string</td>
</tr>
<tr>
<td>POINTER</td>
<td>int</td>
</tr>
<tr>
<td>[SIGN [IS]] TRAILING</td>
<td>string</td>
</tr>
<tr>
<td>[SIGN [IS]] TRAILING SEPARATE [CHARACTER]</td>
<td>string</td>
</tr>
</tbody>
</table>
Advanced Tuning of the Daemon

The daemon configuration is managed using Oracle Studio. Daemon configuration is divided into the following groups:

- **Control**
- **Logging**
- **Security**
- **Workspaces**

**Control**

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

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design perspective Configuration view expand the **Machines** folder.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Right-click the required daemon in the Runtime Explorer and select **Edit Daemon Configuration**. The **Control** tab for the daemon is displayed in the editor.
5. After making changes to the daemon, right-click the daemon and select **Reload Configuration**.

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

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:
The following table shows the parameters that can be set in the **Control** tab:

**Table 7–11 Daemon Control Tab Components**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daemon IP Address</td>
<td>Enter the IP address of the machine(s) where the daemon is listening. If no IP address is entered, the daemon will listen on all available IP addresses.</td>
</tr>
<tr>
<td>Daemon port</td>
<td>Enter the port where the daemon is listening. If no port is entered, the daemon listens on all available ports.</td>
</tr>
<tr>
<td>Automatically recover from failure</td>
<td>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.</td>
</tr>
<tr>
<td>Maximum XML request size</td>
<td>The maximum number of bytes that the daemon handles for an XML document.</td>
</tr>
<tr>
<td>Default language</td>
<td>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.</td>
</tr>
<tr>
<td>Maximum XML in memory</td>
<td>The maximum amount of space reserved for the XML in memory.</td>
</tr>
</tbody>
</table>
Logging

Advanced Tuning of the Daemon  C-3

Table 7–11  (Cont.) Daemon Control Tab Components

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call timeout</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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. If you want 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. |

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.
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 7–4  The Logging tab**

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logging options</strong></td>
<td></td>
</tr>
<tr>
<td>Daemon log file location</td>
<td>Specifies the daemon produces its log data. The full path must be specified.</td>
</tr>
<tr>
<td>Server log filename format</td>
<td>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 will be located in the login directory of the account running Oracle Connect workstation.</td>
</tr>
<tr>
<td>Daemon operations</td>
<td>Select this to log all of the daemon operations.</td>
</tr>
</tbody>
</table>
The following tokens can appear in the log file template and will be replaced accordingly:

- %A: workspace name
- %D: date (ymmd)
- %I: instance number of the given workspace server
- %L: server account login directory
- %P: server process ID
- %T: time (hhmmss)
- %U: server account name (username)

For example, %L/server_%A%I.log may produce a log file such as /
/usr/smith/server_sales15.log.

The default log file template is %L/server_%A%I.log.

### Security

The following **Security** tab for the daemon is used to:

- Grant administration rights for the daemon.
Determine access to the computer.

The following shows how to open the Daemon Security tab:

1. From the Start menu, select Programs, Oracle, and then select Studio.
2. From the Design perspective Configuration view expand the Machines folder.
3. Right-click the computer and select Open Runtime Perspective.
4. Right-click the daemon in the Runtime Explorer and select Edit Daemon Configuration.
5. Click the Security tab.
6. After making changes to the daemon, right-click the daemon and select Reload Configuration.

---

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

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 7-5  The Daemon Security tab**

The Daemon Security tab comprises fields, as listed in the following table:
A daemon can include a number of workspaces. A workspace defines the server processes and environment that are used for the communication between the client and the server for the duration of the client request. Each workspace has its own definition. The workspace definition is divided into the following groups:

- **General**
- **Server Mode**
- **Security**

### General

Using the **General** tab, you enter general information about the features that control the operation of the workspace, such as the server type, the command procedure used to start the workspace and the binding configuration associated with this workspace.

Do the following to open the **General** tab:

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

4. Expand the Daemons node to display the workspaces in the Runtime Explorer.

5. Right-click the workspace and select **Edit Workspace Configuration**. The WS Info. tab opens.

6. After making changes to the workspace, right-click the daemon and select **Reload Configuration**.

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

   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 7–6 The General tab**

![General tab image]

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

**Table 7–14 General Tab Components**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td></td>
</tr>
</tbody>
</table>
## Table 7–14  (Cont.) General Tab Components

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workspace name</td>
<td>The name used to identify the workspace.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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.</td>
</tr>
<tr>
<td>Description</td>
<td>A description of the workspace.</td>
</tr>
<tr>
<td>Startup script</td>
<td>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.</td>
</tr>
<tr>
<td>Server type</td>
<td>This field is not applicable for use with OracleAS Adapters for VSAM.</td>
</tr>
<tr>
<td>Workspace binding name</td>
<td>This field is not applicable for use with OracleAS Adapters for VSAM.</td>
</tr>
<tr>
<td>Timeout parameters</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> Control for details about the Daemon Control section.</td>
</tr>
<tr>
<td>Client idle timeout</td>
<td>The maximum amount of time a workspace client can be idle before the connection with the server is closed.</td>
</tr>
<tr>
<td>Connect timeout</td>
<td>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.</td>
</tr>
<tr>
<td>Call timeout</td>
<td>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.</td>
</tr>
<tr>
<td>Logging and Trace Options</td>
<td></td>
</tr>
</tbody>
</table>
Specific log file format: Defines the name and location of the server log file if you want the data written 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 will be replaced accordingly:

- %A: workspace name
- %D: date (yyymmdd)
- %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 timestamp 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.
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 up a number of servers for future use, prior to 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 **Edit Daemon**.
- 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:
The **Server Mode** tab has the following fields:
Table 7–15  Server Mode Tab Components

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workspace server mode</td>
<td>Specifies the type of new server processes that the daemon starts up. The daemon supports the following server modes:</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>■ multiClient: Clients share a server process and are processed serially. This mode has low overhead because the server processes are already 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.</td>
</tr>
<tr>
<td></td>
<td>■ multiThreaded: This mode is not applicable for use with OracleAS Adapter for VSAM.</td>
</tr>
<tr>
<td></td>
<td>■ 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 already initialized. However, this server mode can use a lot of server resources as it requires as many server processes as concurrent clients.</td>
</tr>
<tr>
<td>Note:</td>
<td>The other modes can be set so that the server processes are reusable. The number of times a process can be reused is controlled by the Reuse limit field value.</td>
</tr>
<tr>
<td>Reuse limit</td>
<td>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.</td>
</tr>
<tr>
<td>Clients per server limit</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Workspaces

#### Oracle Application Server Adapters for VSAM User’s Guide

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 up. 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 killed, even if they are active. Set this field to true if you want the servers for the workspace 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 killed. 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 a number of 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 if you want to limit the number of servers that can be reused. If this is selected, the Reuse limit parameter is available.

### Table 7–15 (Cont.) Server Mode Tab Components

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server availability</td>
<td>Specifies the number of servers in a pool of servers, available to be assigned to a client.</td>
</tr>
<tr>
<td>Port range</td>
<td>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:</td>
</tr>
<tr>
<td>Use default port range</td>
<td>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.</td>
</tr>
<tr>
<td>Maximum number of server processes</td>
<td>Enter the maximum number of server processes that can run at the same time.</td>
</tr>
<tr>
<td>Limit server reuse</td>
<td>Select this if you want to limit the number of servers that can be reused. If this is selected, the Reuse limit parameter is available.</td>
</tr>
</tbody>
</table>
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 kill 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 killed. 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 up. 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 7–15 (Cont.) Server Mode Tab Components

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of spare servers</td>
<td>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.</td>
</tr>
<tr>
<td>Prestarted server pool limit</td>
<td>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 a number of 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.</td>
</tr>
</tbody>
</table>

Resource limitations

| Number of sub-tasks | The number of sub-tasks for a server that are prestarted for this workspace when the daemon starts up. In addition to setting up a pool of server processes as described earlier, you can set additional server processes as sub-tasks by specifying this parameter. If you set 10 servers and 10 prestarted sub-tasks then 100 tasks are started (10 sub-tasks for each process). |

Security

The **Security** tab lets you configure the security level for a workspace. This lets you set the security options for the workspace only. The Security tab is used to:

- Grant administration rights for the workspace
- Determine access to the workspace by a client

Do the following to open the **Security** tab:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design Perspective Configuration view, expand the **Machines** folder.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Expand the Daemons node to display the workspaces in the Runtime Explorer.
5. Right-click the workspace and select **Edit Workspace Configuration**.
6. Click the **Security** tab.
7. After making changes to the workspace, right-click the daemon and select **Reload Configuration**.
Notes:

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

- 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 7–8 The Security Tab*

The **Security** tab has the following fields:
### Table 7–16 Security Tab

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server Account</strong></td>
<td>This section defines the users (accounts) allowed to access the workspace, firewall access ports, workspace account, and anonymous login permissions.</td>
</tr>
<tr>
<td>Use specific workspace account</td>
<td>Select this if you want to define the operating system account used for the workspace.</td>
</tr>
<tr>
<td></td>
<td>If selected, enter the name of the workspace account in the <em>workspace account</em> field.</td>
</tr>
<tr>
<td></td>
<td>If not selected, the account name that was provided by the client is used.</td>
</tr>
<tr>
<td><strong>Allow anonymous clients to use this workspace</strong></td>
<td>Select this if you want to allow this workspace to be invoked without authentication.</td>
</tr>
<tr>
<td></td>
<td>If selected, enter the name of the workspace account in the <em>Server account to use with anonymous clients</em> field.</td>
</tr>
<tr>
<td><strong>Authorized Workspace users</strong></td>
<td>Indicate which users have permission to use the workspace. Select one of the following:</td>
</tr>
<tr>
<td></td>
<td>- <strong>All users</strong>: Any user who has logged on to the daemon may use the workspace.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Selected users only</strong>: Select this to allow only users (or accounts) with specific permission to use the workspace.</td>
</tr>
<tr>
<td></td>
<td>When this is selected, add the names of users (or accounts) and groups that can be use the workspace in the field below.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: If no user is specified, any user who has logged on to the daemon may use the workspace.</td>
</tr>
<tr>
<td><strong>Authorized Administrators</strong></td>
<td>Identifies the users (accounts) with administrator privileges. Select one of the following:</td>
</tr>
<tr>
<td></td>
<td>- <strong>All users</strong>: Indicates that anyone can access the workspace and change the settings.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Selected users only</strong>: Select this to allow only users (or accounts) with specific permission to be administrators.</td>
</tr>
<tr>
<td></td>
<td>When this is selected, add the names of users (or accounts) and groups that can be workspace administrators.</td>
</tr>
<tr>
<td></td>
<td>If no user is specified, any user who has logged on to the daemon may administrator this workspace.</td>
</tr>
</tbody>
</table>
Supported SQL Syntax and SQL Enhancements

OracleAS Adapter for VSAM provides basic support for standard ANSI '92 SQL along with a number of 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

Supported SQL Statements

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

Supported Operators

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

- Arithmetic Operators
- Comparison Operators

<p>| Table D–1  Arithmetic Operators |
|-------------------|----------------|
| Operator          |</p>
<table>
<thead>
<tr>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
</tr>
<tr>
<td>/</td>
</tr>
</tbody>
</table>

| Table D–2  Comparison Operators |
|-------------------|----------------|
| Operator          |
| =                 |
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

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

### Table D–4  Group Functions

<table>
<thead>
<tr>
<th>Oracle Function</th>
<th>Oracle Connect Function</th>
<th>Usage</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>Avg</td>
<td>Avg(exp)</td>
<td>Returns the average value of the expression exp</td>
</tr>
<tr>
<td>Count</td>
<td>Count</td>
<td>Count(exp)</td>
<td>Returns the count of the expression exp</td>
</tr>
</tbody>
</table>
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

Generating Hierarchical Results

A hierarchical query nests a SELECT statement as one of the columns of the rowset retrieved by a nested SELECT statement.

Use braces ({} to delimit the nesting.

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:

---

**Table D-4  (Cont.) Group Functions**

<table>
<thead>
<tr>
<th>Oracle Function</th>
<th>Oracle Connect Function</th>
<th>Usage</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>Max</td>
<td>Max(exp)</td>
<td>Returns the maximum value of the expression exp</td>
</tr>
<tr>
<td>Min</td>
<td>Min</td>
<td>Min(exp)</td>
<td>Returns the minimum value of the expression exp</td>
</tr>
<tr>
<td>Sum</td>
<td>Sum</td>
<td>Sum(exp)</td>
<td>Returns the summation of the expression exp</td>
</tr>
</tbody>
</table>

**Table D-5  Mathematical Functions**

<table>
<thead>
<tr>
<th>Oracle Function</th>
<th>Oracle Connect Function</th>
<th>Usage</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs</td>
<td>Abs</td>
<td>Abs(n)</td>
<td>Returns the absolute value of n</td>
</tr>
<tr>
<td>Ceil</td>
<td>Ceil</td>
<td>Ceil(n)</td>
<td>Returns n rounded up to the closest integer</td>
</tr>
<tr>
<td>Cos</td>
<td>Cos</td>
<td>Cos(n)</td>
<td>Returns the cosine value of n</td>
</tr>
<tr>
<td>Exp</td>
<td>Exp</td>
<td>Exp(n)</td>
<td>Returns the exponential value of n</td>
</tr>
<tr>
<td>Floor</td>
<td>Floor</td>
<td>Floor(n)</td>
<td>Returns n rounded down to the closest integer</td>
</tr>
<tr>
<td>Ln</td>
<td>Ln</td>
<td>Ln(n)</td>
<td>Returns the natural log value of n</td>
</tr>
<tr>
<td>Log</td>
<td>Log</td>
<td>Log(n)</td>
<td>Returns the log value of n</td>
</tr>
<tr>
<td>Mod</td>
<td>Mod</td>
<td>Mod(n, m)</td>
<td>Returns the integer value after dividing n by m</td>
</tr>
<tr>
<td>Nvl</td>
<td>Nvl</td>
<td>Nvl(exp1, exp2)</td>
<td>Returns exp2 when exp1 is null</td>
</tr>
<tr>
<td>Power</td>
<td>Power</td>
<td>Power(n, m)</td>
<td>Returns n to the power of m</td>
</tr>
<tr>
<td>Round</td>
<td>Round</td>
<td>Round(n, m)</td>
<td>Returns n with the fractional part rounded to m digits</td>
</tr>
<tr>
<td>Sin</td>
<td>Sin</td>
<td>Sin(n)</td>
<td>Returns the sine value of n</td>
</tr>
<tr>
<td>Sqrt</td>
<td>Sqrt</td>
<td>Sqrt(n)</td>
<td>Returns the square root of n</td>
</tr>
<tr>
<td>Tan</td>
<td>Tan</td>
<td>Tan(n)</td>
<td>Returns the tangent value of n</td>
</tr>
<tr>
<td>Trunc</td>
<td>Trunc</td>
<td>Trunc(n, m)</td>
<td>Returns the absolute value of n</td>
</tr>
</tbody>
</table>
FROM ... parent_name->child1->child2... [alias]

Or, using an alias for the parent table:
FROM ... parent_alias->child1->child2... [alias]

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.

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.

---
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
- Portuguese
- Simple Chinese
- Spanish
- Traditional Chinese
- Turkish

This appendix describes how to define the language support.

**Defining the Language and Codepage**

The language and codepage parameters are accessed from the computer where Oracle Studio is installed.

Perform the following steps to define the required language and codepage:

1. From the Start menu, select Programs, Oracle, and then select Studio.
2. In the Design perspective Configuration view, expand the Machines folder.
3. Expand the machine for which you want to set the language.
4. Expand the Bindings folder and right-click the NAV binding.
5. Select Open.
6. Expand the Language Settings and do the following:
From the **Language** list, select the NLS supported language to use in this binding. Valid values are listed in the Language Name column of the **NLS Language Codes** table.

From the **Codepage** list, select the codepage that you want to use with this language. The codepages 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 will also change. Check to be sure that you want to use the selected code page with the language you selected.

If no codepage is selected, the default codepage 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 codepage that is selected. If you have additional NLS strings available, you can manually enter them in this field.

The codepage is used by a field with a data type defined as `nlsString`. This parameter is used for a field with a codepage that is different than the machine’s codepage. This property includes values for the name of the codepage 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 codepage:

```
<misc nlsString="JA16EUC,false"/>
```

7. Save the change. New servers will use the language selected.

The following table lists the codepages:

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
<th>Windows Default</th>
<th>ASCII Platforms (Default)</th>
<th>EBCDIC Platforms (Default)</th>
<th>Alternative Codepages (EBCDIC based unless noted otherwise)</th>
</tr>
</thead>
<tbody>
<tr>
<td>English UK</td>
<td>ENUK</td>
<td>Windows-1252</td>
<td>ISO-8859-15</td>
<td>IBM1146</td>
<td>IBM285, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
</tbody>
</table>
### Table 7–17  (Cont.) NLS Language Codes

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
<th>Windows Default</th>
<th>ASCII Platforms (Default)</th>
<th>EBCDIC Platforms (Default)</th>
<th>Alternative Codepages (EBCDIC based unless noted otherwise)</th>
</tr>
</thead>
<tbody>
<tr>
<td>French</td>
<td>FRE</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1147</td>
<td>IBM297, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>Latin International</td>
<td>LAT</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1148</td>
<td>IBM500, IBM037, IBM1140, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>Spanish</td>
<td>SPA</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1145</td>
<td>IBM284, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>German</td>
<td>GER</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1141</td>
<td>IBM273, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>Portuguese</td>
<td>POR</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1140</td>
<td>IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>Italian</td>
<td>ITL</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1144</td>
<td>IBM280, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>Greek</td>
<td>GRK</td>
<td>Windows-125 3</td>
<td>ISO-8859-7</td>
<td>IBM875</td>
<td>-</td>
</tr>
<tr>
<td>Russian¹</td>
<td>RUS</td>
<td>Windows-125 1</td>
<td>ISO-8859-5</td>
<td>IBM1154</td>
<td>IBM1025</td>
</tr>
</tbody>
</table>
### Table 7–17  (Cont.) NLS Language Codes

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
<th>Windows Default</th>
<th>ASCII Platforms (Default)</th>
<th>EBCDIC Platforms (Default)</th>
<th>Alternative Codepages (EBCDIC based unless noted otherwise)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turkish</td>
<td>TUR</td>
<td>Windows-1254</td>
<td>ISO-8859-9</td>
<td>IBM1155</td>
<td>IBM1026</td>
</tr>
<tr>
<td>Hebrew</td>
<td>HEB</td>
<td>Windows-1255</td>
<td>ISO-8859-8</td>
<td>IBM424</td>
<td>IBM 862</td>
</tr>
<tr>
<td>Arabic</td>
<td>ARA</td>
<td>Windows-1256</td>
<td>ISO-8859-6</td>
<td>IBM420</td>
<td></td>
</tr>
<tr>
<td>Chinese - Simplified</td>
<td>SCHI</td>
<td>GBK</td>
<td>GBK</td>
<td>IBM935</td>
<td>-</td>
</tr>
<tr>
<td>Chinese - Traditional</td>
<td>TCHI</td>
<td>BIG5</td>
<td>BIG5</td>
<td>IBM937</td>
<td>-</td>
</tr>
<tr>
<td>Korean</td>
<td>KOR</td>
<td>MS949</td>
<td>EUC-KR</td>
<td>IBM933</td>
<td>MS949</td>
</tr>
</tbody>
</table>

1. Russian users who use ANSI 1251 Cyrillic as their Windows codepage 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\Studiol\studio.exe -nl en"
Editing XML Files in Oracle Studio

In many cases you must manually edit the metadata to configure parts of a solution or composition. Metadata is created in XML format. You define aspects of a solution by changing the values of the elements and attributes of the XML files that belong to the solution. Oracle Studio provides a graphical interface where you can define the various aspects of a solution. This interface lets you make changes easily without having to manually edit the XML file.

Preparing to Edit XML Files in Oracle Studio

You can edit XML files for the following items in Oracle Studio:

- Machines
- Bindings
- Daemons
- Users

When you open an XML file, a graphical representation of the file is opened in the editor. The editor displays the elements and attributes in the file in the first column and their corresponding values in the second column. Each entry has an icon that indicates whether the entry is an element or an attribute. Click the **Source** tab to view the file in its native format. The following figure is an example of the editor’s view of an XML file.

*Figure 7–9  XML Graphical Display*
Making Changes to the XML File

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 that you want 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) that you want to change.
5. Click in the right column next to the property you are changing and edit or add the value.
6. Save the file, then select it again in the Project Explorer and press F5 to refresh. The XML file is updated automatically.

Making Changes to the XML File

You can also make the following changes to XML files in Oracle Studio:
- Remove Objects
- Add DTD Information
- Edit Namespaces
- Add Elements and Attributes
- Replace an Element

Remove Objects

You can delete an element, attribute, or other object from the XML file.

To remove an object
1. Right-click an object from the list in the editor.
2. Select Remove.

Add DTD Information

You can add DTD information to an element or attribute.

To add DTD Information
1. Right-click an element or attribute and select Add DTD Information. The Add DTD Information dialog box opens.
2. Enter the information requested in the dialog box. The following table describes the Add DTD Information dialog box.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root element name</td>
<td>The name of the XML root element.</td>
</tr>
<tr>
<td>Public ID</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Click <strong>Browse</strong> 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.</td>
</tr>
<tr>
<td>System ID</td>
<td>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 will try to use the Public ID to locate the DTD, and if this fails, it will use the System ID to find it.</td>
</tr>
<tr>
<td></td>
<td>Click <strong>Browse</strong> to select a system ID. You can do this in two ways:</td>
</tr>
<tr>
<td></td>
<td>■ Select the file from the workbench. In this case, update the file with the import dialog box.</td>
</tr>
<tr>
<td></td>
<td>■ Select an XML catalog entry.</td>
</tr>
</tbody>
</table>

3. Save the file, then select it again in the Project Explorer and press F5 to refresh. The XML file is updated automatically.

**Edit Namespaces**

You can make changes to the namespaces associated with an element or attribute.

**To edit namespaces**

1. Right-click an element or attribute and select **Edit namespaces**. The Edit Schema Information dialog box opens.
2. Click on one of the buttons 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 7–19  New Namespace**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix</td>
<td>The prefix is added to all qualified elements and attributes in the XML file.</td>
</tr>
<tr>
<td>Namespace Name</td>
<td>The namespace of the XML file.</td>
</tr>
<tr>
<td>Location Hint</td>
<td>The location of the XML schema of the XML file. An XML Catalog ID or a URI can be entered in this field. Click <strong>Browse</strong> to search for the schema you want. You can do this in two ways:</td>
</tr>
<tr>
<td></td>
<td>- Select the schema from the workbench. In this case, update the with the import dialog box.</td>
</tr>
<tr>
<td></td>
<td>- Select an XML catalog entry. The <strong>Namespace Name</strong> and <strong>Prefix</strong> fields are be filled with the appropriate values from the schema (you must leave the fields blank for this to occur).</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: If you are creating an XML file from an XML schema, you cannot change the Namespace Name or Location Hint values.</td>
</tr>
</tbody>
</table>

**To edit a namespace**

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

You can add additional elements and attributes to the XML file.

**To add Elements and Attributes**

1. Right-click an element.
2. Select one of the following:
   - Add Attribute to add an attribute under the selected element.
   - Add Child to add another element under the selected element
   - Add Before to add another element above the selected element
   - Add After to add another element below the selected element

3. Provide a name for the element or attribute if required. You may also be able to select the element from a submenu. The element or attribute will be added to the file.
4. Save the file, then select it again in the Project Explorer and press F5 to refresh. The XML file is updated automatically.

---

**Replace an Element**

You can replace an element with another legal element.

**To replace an element**

1. Right-click an element from the list in the editor.
2. Select Replace with.
3. Select an element from the submenu. Only legal elements are available.
4. The original element is replaced with the selected element.

---

**Note:** The Oracle Studio XML editor is Context sensitive to Oracle schemas. This means that 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.
Index

A
ACX trace parameter, 7-7
add timestamp to traced events parameter, 7-7
arithmetic operators, D-1
arrays, flattening hierarchical data, D-4
Arithmetic fixed precision parameter, 7-12

B
binary XML log level parameter, 7-7

C
cache
maximum queries, 7-11
cache buffer size parameter, 7-8
code page parameter, 7-9
COM maximum XML size in memory parameter, 7-14
COM maximum XML size parameter, 7-14
COM XML transport buffer parameter, 7-14
commit on destroy, 7-13
comparison operators, D-1
continuous query
prefix, 7-11
retry interval, 7-11
timeout, 7-11
continuous query prefix parameter, 7-11
continuous query retry interval parameter, 7-11
continuous query timeout parameter, 7-11

data sources, disabling caching, 7-10
data types
atomic metadata, B-1
NLS string parameter, 7-9
data parameters
ACX trace, 7-7

d daemon
logging, C-3
security, C-5
server modes, 7-2
shutting down, 6-2
starting, 6-1
timeout, 6-6
disk space requirements
IBMmainframe, 2-2
dsm maximum buffer size parameter, 7-13
dsm middle buffer size parameter, 7-14
dsmMaxSortBufferSize parameter, 7-14

drag
debug, 7-7
e environment parameters
ACX trace, 7-7
add timestamp to traced events, 7-7
Arithmetic fixed precision, 7-12
add arithmetic operators, 7-12
cache buffer size, 7-8
code page, 7-9
COM maximum XML size, 7-14
COM maximum XML size in memory, 7-14
COM XML transport buffer, 7-14
commit on destroy, 7-13
continous query prefix, 7-11
continuous query retry interval, 7-11
continuous query timeout, 7-11
drag
debug, 7-7
disable 2PC, 7-13
disable command reuse, 7-10
disable compilation cache, 7-11
disable DS property cache, 7-10
disable insert parameterization, 7-10
disable metadata caching, 7-10
disable query parameterization, 7-10
disable row mark fail fetch, 7-10
disable SQS cache, 7-12
disable threads, 7-10
dsm maximum buffer size, 7-13
dsm maximum sort buffer size, 7-14
dsm middle buffer size, 7-14
expose XML fields, 7-12
file close on transaction, 7-14
file pool size, 7-14
file pool size per file, 7-14
first tree extensions, 7-11
GDB trace, 7-7
general trace, 7-7
hash buffer size, 7-14
hash enable RO, 7-14
hash max open files, 7-14
hash primary event size, 7-14
hash secondary event size, 7-14
ignore segment bind failure, 7-10
insert from select commit rate, 7-12
language, 7-9
log file, 7-7
maximum columns in parsing, 7-11
Maximum segmented database thread, 7-11
maximum SQL cache, 7-11
minimum number of parameters allocated, 7-11
miscellaneous, 7-8
nav utility editor, 7-8
NLS string, 7-9
odbc, 7-9
oledb, 7-9
optimistic for updates, 7-11
optimizer, 7-9
optimizer goal, 7-9
optimizer trace, 7-8
parser depth, 7-12
procedures cache size, 7-12
prompt database user password, 7-10
query processor trace, 7-7
query warnings, 7-7
queryProcessor, 7-10
recovery delay, 7-13
Replace invalid XML characters, 7-15
time limit, 7-13
token size, 7-12
trace directory, 7-8
transaction conversions parameters, 7-13
Transaction extended logging, 7-8
transaction extended logging, 7-13
transaction log file, 7-13
transactions, 7-13
temporary directory, 7-8
tuning, 7-13
use alternate qualifier, 7-11
use global file pool, 7-14
use table filter expression, 7-11
user commit confirm table, 7-13
write empty string as null, 7-11
XML, 7-14
XML date format parameter, 7-15
XML trim char column, 7-15
year 2000 policy, 7-8
error log, binary XML log level parameter, 7-7
erase log, log file parameter, 7-7
exact arithmetic, 7-12
expose XML fields parameter, 7-12
F
file close on transaction, 7-14
file pool size, 7-14
file pool size per file, 7-14
first row optimization, optimizer goal parameter, 7-9
first tree optimization, optimizer goal parameter, 7-9
floating point precision, 7-11
functions
group, D-2
string, D-2
G
GDB trace parameter, 7-7
general parameters
cache buffer size, 7-8
nav utility editor, 7-8
temporary directory, 7-8
year 2000 policy, 7-8
general trace parameter, 7-7
group functions, D-2
H
hardware requirements
IBM mainframe disk space, 2-2
IBM mainframe memory, 2-2
UNIX, 2-1
hash buffer size, 7-14
hash buffer size parameter, 7-14
hash enable RO parameter, 7-14
hash joins, 7-14
hash max open files parameter, 7-14
hash primary event size parameter, 7-14
hash secondary event size parameter, 7-14
hierarchical queries, D-4
I
IBM mainframe
installing, 2-4
post-installation, 2-8
preinstallation, 2-4
ignore segment bind failure parameter, 7-10
importing installation kit, 2-4
index cache
   dsm middle buffer size parameter, 7-14
insert from select commit rate parameter, 7-12
installation kit, importing, 2-4
installing
   IBM mainframe, 2-4
integration flow, Oracle Application Server to legacy application, 1-3

J
joins
   hash buffer size parameter, 7-14
   hash max open files parameter, 7-14

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

M
maximum columns in parsing parameter, 7-11
Maximum segmented database thread, 7-11
maximum SQL cache parameter, 7-11
memory
   dsm maximum buffer size parameter, 7-13
dsm maximum sort buffer Size parameter, 7-14
dsm middle buffer size parameter, 7-14
memory requirements
   mainframe requirements, 2-2
metadata
   atomic data types, B-1
   disabling retrieval from cache, 7-10
minimum number of parameters allocated parameter, 7-11
miscellaneous parameters
   environment, 7-8

N
nav utility editor parameter, 7-8
nav utility editor, text editor, 7-8
NLS string parameter, 7-9

O
odbc environment parameters, 7-9
operator system requirements
   UNIX, 2-2
optimizer
   environment parameters, 7-9
   optimizer goal parameter, 7-9
   trace directory parameter, 7-8
   optimizer goal parameter, 7-9
   optimizer trace parameter, 7-8
Oracle Application Server requirements, 2-2

P
parallel processing parameters
   disable threads, 7-10
parameterization of constants, disabling, 7-10
parameterization of queries, disabling, 7-10
parser depth parameter, 7-12
password, setting automatic prompt, 7-10
post-installation, IBM mainframe, 2-8
preinstallation, IBM mainframe, 2-4
procedures cache size parameter, 7-12
processor requirements
   UNIX, 2-1
prompt database user password parameter, 7-10

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

R
recovery delay, 7-13
Replace invalid XML characters parameter, 7-15
requirements
UNIX hardware requirements, 2-1
UNIX software requirements, 2-2
row optimization, optimizer goal parameter, 7-9

S
security
daemon configurations, C-5
setting automatic prompt, 7-10
segmented data sources, response upon failure, 7-10
servers
configuring modes, 7-2
reusable, 7-3
Reuse limit daemon parameter, 7-3, C-13
setting up VSAM, 2-18
SMS
See system managed storage
software requirements
Oracle Application Server, 2-2
UNIX operating system, 2-2
sort buffer, dsm maximum sort buffer size parameter, 7-14
SQL, flattening hierarchical data, D-4
stored queries, maximum in cache memory, 7-12
string functions, D-2
system managed storage, 2-4

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

U
use alternate qualifier parameter, 7-11
use global file pool, 7-14
use table filter expression parameter, 7-11
user commit confirm table, 7-13

V
VSAM, setting up, 2-18

W
Workspace server mode, C-13
write empty string as null parameter, 7-11

X
XML
environment parameters, 7-14
XML date format parameter, 7-15
XML parameters
COM maximum XML size, 7-14
COM maximum XML size in memory, 7-14
COM XML transport buffer, 7-14
Replace invalid XML characters, 7-15
XML date format parameter, 7-15
XML trim char column, 7-15
XML trim char column parameter, 7-15

Y
Y2K
See year 2000 policy parameter
year 2000 policy parameter, 7-8