Oracle® Application Server Integration

Adapter for VSAM Installation and User's Guide

10*g* (9.0.4)

Part No. B10457-01

September 2003

ORACLE®

Oracle Application Server Integration Adapter for VSAM Installation and User's Guide, 10*g* (9.0.4)

Part No. B10457-01

Copyright © 2003 Oracle Corporation. All rights reserved.

Primary Author: Julian Weiss

Contributors: Seshu Adunuthula, Deanna Bradshaw, Yishai Hadas, Dror Harari, Arvind Jain, Maneesh Joshi, Mark Kennedy, Adeeb Massad, Bo Stern, Costi Zaboura

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle Store, Oracle8*i*, Oracle9*i*, SQL*Plus, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments xiii

Preface xv

Audience xvi
Organization xvi
Related Documentation xviii
Conventions xviii
Documentation Accessibility xxiii

1 Overview of the Oracle Application Server Integration Adapter for VSAM

Overview of Application Integration 1-2
Overview of VSAM and Oracle Application Server Integration Adapter for VSAM 1-2
VSAM Data 1-3
Accessing VSAM Data Under CICS 1-4
Accessing VSAM Data Directly 1-4
Architecture of the Oracle Application Server Integration Adapter for VSAM 1-4
Inbound Interactions 1-6
Integrating Oracle Application Server ProcessConnect with the Adapter for VSAM 1-6
The Modeling Phase 1-6
Runtime 1-7
Modeling the VSAM Adapter Using Oracle Studio 1-7
The Configuration Perspective 1-7
The Metadata Perspective 1-7

The Metadata Import Perspective 1-8
The Runtime Manager Perspective 1-8

2 Installing the Oracle Application Server Integration Adapter for VSAM

Before You Begin 2-2
Determining the Hardware and Software Requirements for the OS/390 2-2
Hardware Requirements 2-2
Software Requirements 2-2
Determining the Hardware and Software Requirements for Windows 2-3
Hardware Requirements 2-3
Software Requirements 2-4
Installing Oracle Connect for VSAM on an OS/390 Platform 2-4
Installation Worksheet 2-5
Preinstallation Instructions 2-6
Installing the Kit on the Mainframe 2-7
Installation Instructions 2-7
Postinstallation Instructions 2-12
General Postinstallation Procedures 2-12
Starting the Daemon 2-14
Setting Up Oracle Connect for VSAM for Reentrancy 2-15
Setting Up Oracle Connect for VSAM to Update VSAM Data 2-15
VSAM Running Under CICS 2-16
VSAM Not Running Under CICS 2-16
Updating Oracle Connect with VSAM 2-16
Installing Oracle Studio 2-19
Installing Oracle Studio from the CD-ROM 2-19
Configuring Oracle Connect for VSAM 2-19
Task: Setting Up the IBM OS/390 Machine 2-20
Securing Access to Oracle Connect for VSAM 2-22
Setting Password Access to Oracle Studio 2-23
Task: Setting Up a Master Password to Access Oracle Studio 2-23
Specifying Users with Administrative Rights 2-24
Task: Setting Administrative Rights to a Machine 2-24
Setting Up Runtime User Access to the IBM OS/390 Machine 2-26

3 Modeling Interactions for Oracle Application Server Integration Adapter for VSAM

Setting Up the VSAM Data Source 3-2
Task: Adding the VSAM Data Source Definition 3-2
Importing Metadata for the VSAM Data Source 3-5
Task 1: Defining an Import Procedure 3-6
Task 2: Generating Metadata for the VSAM Data 3-7
Setting Up an Oracle Connect for the VSAM Backend Adapter 3-22
Task: Setting Up the Adapter 3-22
Generating Outbound Interactions 3-24
Task 1: Setting Up the Metadata Perspective 3-25
Task 2: Generating Outbound Interactions 3-25
Task 3: Modifying the Generated Outbound Interactions 3-30
Generating Inbound Interactions 3-33
Task 1: Defining an Import Procedure 3-34
Task 2: Generating Inbound Interactions 3-37

4 Using Interactions in a Business Process

Adding the VSAM Adapter in Oracle Application Server ProcessConnect 4-2
Setting Up the Delivery Channel 4-2
Task: Setting Up the Delivery Channel 4-2
Modifying an Adapter in Oracle Studio 4-4
Comparing a ProcessConnect XSD to an Oracle Connect Record Schema 4-4

5 Oracle Application Server Integration Adapter for VSAM Features and Restrictions

Accessing VSAM Using SQL 5-2
Generating Hierarchical Results 5-2
Accessing Hierarchical Data Using SQL 5-3
Flattening Hierarchical Data Using SQL 5-3
Sending Inbound Interactions Using a CICS Transaction 5-4
Task 1: Setting Up the IBM OS/390 Machine 5-4
Task 2: Setting Up a Call to the CICS Transaction 5-6
High Availability with Oracle Connect 5-11

Known Restrictions 5-11
VSAM Data Source Restrictions 5-11
VSAM Under CICS 5-11
VSAM Not Under CICS 5-11
Array Updates 5-12

6 Troubleshooting the Oracle Application Server Integration Adapter for VSAM

Design-Time Troubleshooting 6-2
Specifying the Wrong Adapter Name 6-2
Specifying the Wrong Workspace Name 6-2
Specifying the Wrong Machine 6-3
Specifying the Wrong Port Number 6-3
When the Connection with the OS/390 Machine Is Down 6-4
Troubleshooting the Daemon 6-4
Starting the Daemon 6-5
Task: Starting the Daemon 6-6
Shutting Down the Daemon 6-6
Monitoring the Daemon During Runtime 6-6
Daemon (Machine) Options 6-6
Workspace Options 6-7
Server Options 6-8
Daemon Logs 6-8
The Daemon Log Monitor 6-9
The Workspace Log Monitor 6-10
The Server Log Monitor 6-11
Resolving Communication Errors 6-12
Resolving Specific Errors 6-13

7 Configuring the OS/390 Working Environment

Configuring the Daemon for High Availability 7-2
Task: Adding a New Daemon Workspace Configuration 7-2
Task: Editing the Workspace 7-2
Task: Configuring the Server Mode 7-3
Configuring a Binding Environment 7-5

Task: Setting a Binding Environment 7-6
comm Category 7-6
debug Category 7-7
miscellaneous Category 7-7
odbc Category 7-9
oledb Category 7-9
optimizer Category 7-9
queryProcessor Category 7-10
transactions Category 7-11
tuning Category 7-11

A The Oracle Connect VSAM Metadata Definitions

Metadata for the VSAM Data Source A-2
General Tab A-3
Columns Tab A-5
Indexes Tab A-8
Statistics Tab A-10
Generating Statistics A-11
Source Tab A-12
Interaction Metadata for the Adapter A-12
General Tab A-13
Interaction General Tab A-15
Interaction Advanced Tab A-17
Schema General Tab A-18
Schema Record Tab A-19
Source Tab A-21

B Datatype Support

Datatype Mapping for the VSAM Data Source B-2
VSAM Data Source Metadata Datatypes B-6
Datatype Mapping for Oracle Connect for VSAM B-11

C Supported SQL Syntax and SQL Enhancements

Supported SQL Statements C-2

Supported Operators C-2
Supported Functions C-3
SQL Enhancements C-4
Generating Hierarchical Results C-5
Accessing Hierarchical Data Using SQL C-5
Flattening Hierarchical Data Using SQL C-5
Using an Alias C-5

D Daemon Configuration Settings

Daemon Control D-2
Daemon Logging D-4
Daemon Security D-7
Workspaces D-9
WS Info. D-10
WS Server D-12
WS Logging D-16
WS Security D-18
WS Governing D-22

E COBOL APIs to Applications

Connection APIs E-2
The Connect Function E-2
The Disconnect Function E-4
Transaction APIs E-4
The Set Autocommit Function E-5
The Transaction Commit Function E-5
The Transaction Rollback Function E-6
The Execute Function E-6
The Get Event Function E-7
The Ping Function E-7
The Get Error Function E-8

F National Language Settings

Defining the Language and Codepage F-2

Index

List of Tables

2–1	Preinstallation Information	2-5
2–2	Required Permissions	2-5
2–3	Installation Checklist	2-6
2–4	Installation Prepare Job Prompts and Responses	2-8
2–5	Installation Prompts and Responses	2-10
2–6	VSAM Adapter-Specific Installation Prompts and Responses	2-11
2–7	Preinstallation Information	2-16
2–8	Required Permissions	2-16
2–9	VSAM Adapter Installation Prompts and Responses	2-17
6–1	Log Monitor Toolbar Buttons	6-12
6–2	Scenarios When a Client Is Disconnected	6-13
B–1	Datatype Mapping: COBOL and VSAM Data Source	B-2
B–2	VSAM Metadata Datatypes	B-6
B–3	Datatype Mapping: COBOL and Oracle Connect for VSAM	B-11
C–1	Arithmetic Operators	C-2
C–2	Comparison Operators	C-2
C–3	String Functions	C-3
C–4	Group Functions	C-3
C–5	Mathematical Functions	C-4
E–1	ACXACNCT Parameters	E-3
E–2	ACXADSCO Parameters	E-4
E–3	ACXASCMT Parameters	E-5
E–4	ACXACMIT Parameters	E-6
E–5	ACXARBCK Parameters	E-6
E–6	ACXAEXEC Parameters	E-6
E–7	ACXAGTEV Parameters	E-7
E–8	ACXAPING Parameters	E-7
E–9	ACXAGTER Parameters	E-8
F–1	Supported Codepages	F-3

Send Us Your Comments

Oracle Application Server Integration Adapter for VSAM Installation and User's Guide, 10*g* (9.0.4)

Part No. B10457-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appserverdocs_us@oracle.com
- FAX: (650) 506-7407 Attn: Oracle Application Server Documentation Manager
- Postal service:
  Oracle Corporation
  Oracle Application Server Documentation
  500 Oracle Parkway, Mailstop 2op3
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This guide is the primary source of user and reference information on the Oracle Application Server Integration Adapter for VSAM, which enables client applications to access Virtual Storage Access Method (VSAM) data through the Sun 2JEE Connector Architecture (JCA) API using Structured Query Language (SQL).

This document describes the features of Oracle Application Server Integration Adapter for VSAM software that apply to the IBM OS/390 and z/OS Series, Windows NT, Windows 2000, Windows XP, and Windows Server 2003 operating systems.

This preface covers the following topics:

- Audience
- Organization
- Related Documentation
- Conventions
- Documentation Accessibility

Audience

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

- Installing and configuring the Oracle Application Server Integration Adapter for VSAM
- Diagnosing errors
- Using Oracle Application Server ProcessConnect to access VSAM data

---

Note: You should understand the fundamentals of Oracle Application Server ProcessConnect and the IBM OS/390 and z/OS series and Microsoft Windows operating system before using this guide to install or administer the Oracle Application Server Integration Adapter for VSAM.

Organization

This document contains:

Chapter 1, "Overview of the Oracle Application Server Integration Adapter for VSAM"

This chapter introduces the Oracle Application Server Integration Adapter for VSAM.

Chapter 2, "Installing the Oracle Application Server Integration Adapter for VSAM"

This chapter explains how to install Oracle Connect for VSAM on an IBM OS/390 or z/OS series operating system and Oracle Studio on a Windows platform.

Chapter 3, "Modeling Interactions for Oracle Application Server Integration Adapter for VSAM"

This chapter describes how to model interactions in Oracle Studio.

Chapter 4, "Using Interactions in a Business Process"

This chapter describes how to include Oracle Connect for VSAM interactions in a business process in Oracle Application Server ProcessConnect.

Chapter 5, "Oracle Application Server Integration Adapter for VSAM Features and Restrictions"

This chapter describes specific features of the Oracle Application Server Integration Adapter for VSAM.

Chapter 6, "Troubleshooting the Oracle Application Server Integration Adapter for VSAM"

This chapter describes how to troubleshoot problems both during modeling and runtime.

Chapter 7, "Configuring the OS/390 Working Environment"

This chapter describes how to tune the working environment to improve performance.

Appendix A, "The Oracle Connect VSAM Metadata Definitions"

This appendix describes the metadata for a VSAM data source and an Oracle Connect for VSAM backend adapter definition.

Appendix B, "Datatype Support"

This appendix describes how COBOL datatypes are converted to Oracle Connect for VSAM datatypes and Oracle datatypes.

Appendix C, "Supported SQL Syntax and SQL Enhancements"

This appendix describes the supported SQL syntax that can be used with the Oracle Application Server Integration Adapter for VSAM and SQL enhancements that are not part of standard ANSI '92 SQL.

Appendix D, "Daemon Configuration Settings"

This appendix describes daemon configuration used to control client/server communication.

Appendix E, "COBOL APIs to Applications"

This appendix describes how to access the Oracle Application Server Integration Adapter for VSAM from a COBOL program.

Appendix F, "National Language Settings"

This appendix describes how to set up Oracle Connect for VSAM to support different languages.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Application Server ProcessConnect User's Guide*

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

`http://otn.oracle.com/membership/`

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

`http://otn.oracle.com/documentation/`

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text
- Conventions in Code Examples
- Conventions for Windows Operating Systems

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle9i Database Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.

Convention	Meaning	Example
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to open SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods.
lowercase italic monospace (fixed-width) font	Lowercase italic monospace font represents placeholders or variables.	You can specify the *parallel_clause*. Run *old_release*.SQL where *old_release* refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
`[ ]`	Brackets enclose one or more optional items. Do not enter the brackets.	`DECIMAL (digits [ , precision ])`
`{ }`	Braces enclose two or more items, one of which is required. Do not enter the braces.	`{ENABLE | DISABLE}`
`|`	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	`{ENABLE | DISABLE}` `[COMPRESS | NOCOMPRESS]`

`...`	Horizontal ellipsis points indicate either:
	■ That we have omitted parts of the code that are not directly related to the example
	■ That you can repeat a portion of the code
		`CREATE TABLE ... AS subquery;`
		
		`SELECT col1, col2, ... , coln FROM employees;`

. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	`SQL> SELECT NAME FROM V$DATAFILE;`
		`NAME`
		`----------`
		`/fs1/dbs/tbs_01.dbf`
		`/fs1/dbs/tbs_02.dbf`
		`.`
		`.`
		`.`
		`/fs1/dbs/tbs_09.dbf`
		`9 rows selected.`

Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	`acctbal NUMBER(11,2);` `acct CONSTANT NUMBER(4) := 3;`
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	`CONNECT SYSTEM/system_password` `DB_NAME = database_name`

`UPPERCASE`	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	`SELECT last_name, employee_id FROM employees;`
		`SELECT * FROM USER_TABLES;`
		`DROP TABLE hr.employees;`

Convention	Meaning	Example

`lowercase`	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.
	
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
		`SELECT last_name, employee_id FROM employees;`
		`sqlplus hr/hr`
		`CREATE USER mjones IDENTIFIED BY ty3MU9;`

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (|), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	`c:\winnt"\"system32 is the same as C:\WINNT\SYSTEM32`
`C:\>`	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual.	`C:\oracle\oradata>`

Convention	Meaning	Example
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	`C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\"` `C:\>imp SYSTEM/`*`password`*` FROMUSER=scott TABLES=(emp, dept)`
HOME_NAME	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	`C:\> net start Oracle`*`HOME_NAME`*`TNSListener`
ORACLE_HOME and *ORACLE_BASE*	In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory. For Windows NT, the default location was `C:\orant`. This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle`. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\ora`*`nn`*, where *nn* is the latest release number. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. Refer to *Oracle9i Database Getting Started for Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE*`\`*ORACLE_HOME*`\rdbms\admin` directory.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. 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. Standards will continue to evolve over time, and Oracle Corporation 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 additional information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, 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, JAWS 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 Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

1

Overview of the Oracle Application Server Integration Adapter for VSAM

This chapter introduces the Oracle Application Server Integration Adapter for VSAM. The Oracle Application Server Integration Adapter for VSAM is a plug-in component of Oracle Application Server ProcessConnect, enabling Oracle Application Server ProcessConnect to access VSAM data on IBM OS/390 or z/OS series operating systems.

This chapter contains the following sections:

- Overview of Application Integration
- Overview of VSAM and Oracle Application Server Integration Adapter for VSAM
- Integrating Oracle Application Server ProcessConnect with the Adapter for VSAM
- Modeling the VSAM Adapter Using Oracle Studio

Overview of Application Integration

When business applications need to communicate with each other, the output from one application is not always understandable by the other application. Application adapters enable different applications to interoperate.

Adapters for application integration address a mismatch between the integrating interface of a source or target application and the integration infrastructure, such as Oracle Application Server ProcessConnect. Simple adapters only exchange data using an API, but complex adapters can also do transformations (such as SQL to XML), as well as deal with exception conditions. Therefore, adapters are a combination of modeling tools and runtime software that act as "glue" to link applications, considered as sources or targets (or both), to the integration infrastructure that transports, translates, transforms, and routes data between systems. On the one hand, an adapter deals with a group of touchpoints (one or more entry/exit points, collectively an interface) for a source or target; on the other hand, an adapter links to the integration infrastructure. This means that adapters generally have two interfaces - one that is native to the source or target application and the other that is native to the integrating infrastructure. Adapters perform a variety of key tasks, including recognizing events, collecting and transforming data, and exchanging data with the integration infrastructure. They also handle exception conditions, and can often dynamically (or with minor reconfiguration changes) accommodate new revisions of back-end applications.

Each task performed by the adapter is defined as an interaction. Interactions that are initiated by the source or target application are referred to as inbound interactions. Interactions that are initiated by the integrating infrastructure are referred to as outbound interactions.

Overview of VSAM and Oracle Application Server Integration Adapter for VSAM

The Oracle Application Server Integration Adapter for VSAM is a plug-in component of Oracle Application Server ProcessConnect.

The Oracle Application Server Integration Adapter for VSAM models Virtual Sequential Access Method (VSAM) as a simple database with an SQL front end. The adapter is native to the data source and OS/390, providing direct access that results in improved performance.

VSAM Data

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

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

All of the VSAM access methods (KSDS, RRDS and ESDS) 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 its 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.

The Oracle Application Server Integration Adapter for VSAM includes the following features:

The Oracle Application Server Integration Adapter for VSAM 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 OS/390 machine and by associating the data structures with specific physical files.

The Oracle Application Server Integration Adapter for VSAM has a full-fledged query processor and optimizer that executes SQL against the VSAM data with full support for VSAM datatypes such as arrays.

The Oracle Application Server Integration Adapter for VSAM 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. Moreover, the Oracle Application Server Integration Adapter for VSAM maps data structures, facilitating access to VSAM data from within Oracle Application Server ProcessConnect.

Since VSAM is viewed as a database, the Oracle Application Server Integration Adapter for VSAM implements outbound 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.

VSAM does not have a concept of procedures or triggers. Applications that use VSAM may need to interact with Oracle Application Server ProcessConnect by

sending it a native event. For inbound interactions the VSAM application uses a special API to send a native event to a queue running on the mainframe and the adapter subscribes to that queue for inbound event handling. Typically, each application associated with an Oracle Connect for VSAM will subscribe to a dedicated queue.

The Oracle Application Server Integration Adapter for VSAM includes the options to access VSAM data managed by CICS or directly.

Accessing VSAM Data Under CICS

When VSAM records are defined under CICS, all access to the data is managed by CICS. The Oracle Application Server Integration Adapter for VSAM includes an option to access the VSAM data under CICS.

Accessing VSAM Data Directly

When VSAM records are not defined under CICS, the Oracle Application Server Integration Adapter for VSAM includes an option to access the VSAM data directly. This option can also be used to improve performance when the VSAM records are managed by CICS but access to them is for read only, and there is no risk that there are changes buffered by CICS while the read is performed.

Architecture of the Oracle Application Server Integration Adapter for VSAM

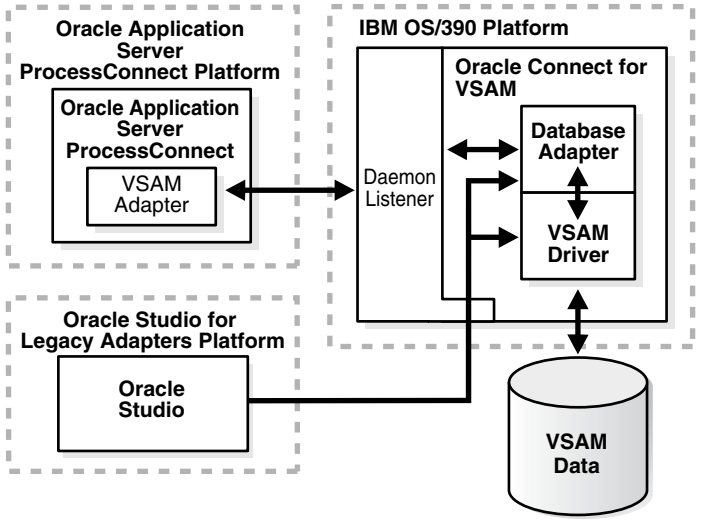
The Oracle Application Server Integration Adapter for VSAM includes the following components:

- The VSAM adapter, which resides in Oracle Application Server ProcessConnect
- Oracle Connect for VSAM, which resides on the OS/390 machine
- A GUI-based configuration tool, called Oracle Studio for Legacy Adapters, which resides on a Windows platform.

> **Note:** Throughout this manual Oracle Studio for Legacy Adapters is referred to as Oracle Studio.

The components of the Oracle Application Server Integration Adapter for VSAM interact with each other, as shown in Figure 1–1 on page 1-5.

Figure 1–1 Architecture of the Oracle Application Server Integration Adapter for VSAM



The VSAM adapter enables communication with Oracle Connect for VSAM on the OS/390 platform.

Oracle Connect for VSAM is comprised of three main components:

- **Daemon (a listener)** - Responsible for managing and allocating server processes for client requests coming from Oracle Application Server ProcessConnect. Server processes can be started per client or kept in a pool for reuse. The daemon resides in a single address space and is executed as a started task.

  See Also:

  - "Troubleshooting the Daemon" on page 6-4
  - "Configuring the Daemon for High Availability" on page 7-2
  - Appendix D, "Daemon Configuration Settings"

- **Database adapter** - Provides the interaction definitions and input and output record schemas required to access the VSAM data
- **VSAM driver** - Accesses the VSAM data based on database adapter interactions

Oracle Studio is used to model the adapter interactions and manage the back-end component.

Inbound Interactions

When VSAM runs under CICS, it can interact with Oracle Application Server ProcessConnect by sending it a native event. The native event is sent to a queue running on the mainframe and the adapter subscribes to that queue for inbound event handling. Typically, each application associated with Oracle Connect for VSAM will subscribe to a dedicated queue.

The Oracle Application Server Integration Adapter for VSAM provides the following ways to send an inbound interaction:

- Invoking a special CICS transaction
- Using an API from a COBOL program

Integrating Oracle Application Server ProcessConnect with the Adapter for VSAM

Oracle Application Server Integration Adapter for VSAM is fully integrated with Oracle Application Server ProcessConnect, both during the modeling and runtime phases.

The Modeling Phase

The Oracle Application Server Integration Adapter for VSAM is integrated with Oracle Application Server ProcessConnect as follows:

- Modeling the Oracle Application Server Integration Adapter for VSAM is performed using Oracle Studio. This step includes modeling both the metadata for the VSAM data source and modeling inbound and outbound interactions as required.

  See Also: "Modeling the VSAM Adapter Using Oracle Studio" on page 1-7

- After the Oracle Application Server Integration Adapter for VSAM has been modeled in Oracle Studio, the integration between the adapter and the application in Oracle Application Server ProcessConnect is modeled. Within Oracle Application Server ProcessConnect you define a delivery channel for the adapter and then add interactions from the list modeled using Oracle Studio.

  See Also: *Oracle Application Server ProcessConnect User's Guide*

Runtime

At runtime, the interactions specified in the Oracle Application Server ProcessConnect application are executed.

For outbound interactions, the interactions for the Oracle Application Server Integration Adapter for VSAM are handled by Oracle Connect for VSAM on the OS/390 machine. The VSAM data source is queried and the results of this query are returned to Oracle Application Server ProcessConnect.

For inbound interactions, the interactions are initiated on the OS/390 machine and sent to Oracle Application Server ProcessConnect.

Modeling the VSAM Adapter Using Oracle Studio

Oracle Studio is used to model and manage the Oracle Application Server Integration Adapter for VSAM.

Oracle Studio uses perspectives enabling you to focus specific information necessary to model the Oracle Application Server Integration Adapter for VSAM.

The following perspectives are available:

- The Configuration Perspective
- The Metadata Perspective
- The Metadata Import Perspective
- The Runtime Manager Perspective

The Configuration Perspective

Use the Configuration perspective to define access to the IBM OS/390 machine and to VSAM data on the machine.

The Metadata Perspective

Use the Metadata perspective to model the following:

- The Oracle Application Server Integration Adapter for VSAM interactions and the expected inputs and output
- The VSAM data source records and the fields in these records

The Metadata Import Perspective

Assuming that metadata for the VSAM data source is available in a COBOL copybook, use the Metadata Import perspective to import this information for use by Oracle Connect for VSAM to create outbound interactions.

The Runtime Manager Perspective

Use the Runtime Manager perspective to troubleshoot Oracle Connect for VSAM by monitoring daemon activity on the OS/390 machine.

2

Installing the Oracle Application Server Integration Adapter for VSAM

This chapter explains how to install Oracle Connect for VSAM and Oracle Studio software from the distribution CD-ROM.

> **Note:** In addition to the installation procedures described in this chapter, the VSAM adapter component of the Oracle Application Server Integration Adapter for VSAM software must be installed with Oracle Application Server ProcessConnect. Installing the VSAM adapter component is described in the *Oracle Application Server ProcessConnect Installation Guide*.

This chapter includes the following sections:

- Before You Begin
- Installing Oracle Connect for VSAM on an OS/390 Platform
- Updating Oracle Connect with VSAM
- Installing Oracle Studio
- Configuring Oracle Connect for VSAM
- Securing Access to Oracle Connect for VSAM

Before You Begin

Confirm that you have met all hardware and software requirements. Refer to the following sections:

- Determining the Hardware and Software Requirements for the OS/390
- Determining the Hardware and Software Requirements for Windows

Determining the Hardware and Software Requirements for the OS/390

This section describes the following requirements for installing Oracle Connect for VSAM:

- Hardware Requirements
- Software Requirements

Hardware Requirements

The following table summarizes the hardware requirements for Oracle Connect for VSAM.

Hardware Component	Requirements
Processor	IBM S/390 machine
Memory	The minimum requirement is 1MB per user. The actual memory requirement depends on the size of the database and the number of databases accessed.
CD-ROM Drive	An internal or external CD-ROM drive
Disk Space (3380 and 3390 disks)	150 cylinders

Software Requirements

The system software configuration described in this section is supported by Oracle Corporation as long as the underlying system software products are supported by their respective vendors. Verify the latest support status with your system software vendors.

The following table summarizes the software requirements for Oracle Connect for VSAM.

Software Component	Requirements
Operating System	IBM OS/390 V2R5 and up or z/OS series
Oracle Application Server ProcessConnect	Oracle Application Server 10*g* (9.0.4) Oracle Application Server ProcessConnect can reside on any supported platform

Note: The CICS TP Monitor, V4R1, is required when using Oracle Connect for VSAM under CICS. In this case, 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.

Determining the Hardware and Software Requirements for Windows

This section describes the following requirements for installing Oracle Studio:

- Hardware Requirements
- Software Requirements

Hardware Requirements

The following table summarizes the hardware requirements for Oracle Studio.

Hardware Components	Requirements
Processor	An Intel or 100% compatible personal computer (PC), based on a Pentium processor
Memory	128 MB of RAM
CD-ROM Drive	An internal or external CD-ROM drive
Disk Space	75MB of free disk space

Software Requirements

The system software configuration described in this section is supported by Oracle Corporation as long as the underlying system software products are supported by their respective vendors. Verify the latest support status with your system software vendors.

The following table summarizes the software requirements for Oracle Studio.

Software Components	Requirements
Operating System	Microsoft Windows NT Server Version 4.0 with service pack 5 or higher, Microsoft Windows 2000 with service pack 2 or higher, or Microsoft Windows XP
Microsoft	Network transport protocol software, TCP/IP, included with Microsoft Windows

Installing Oracle Connect for VSAM on an OS/390 Platform

Note: If you have an Oracle Connect legacy adapter already installed on the OS/390 machine, follow the instructions described in "Updating Oracle Connect with VSAM" on page 2-16.

The other legacy adapters that run on an OS/390 machine are:

- Oracle Application Server Integration Adapter for CICS
- Oracle Application Server Integration Adapter for IMS/DB
- Oracle Application Server Integration Adapter for IMS/TM

This section explains how to install Oracle Connect for VSAM software from the distribution CD-ROM. This section includes the following:

- Installation Worksheet
- Preinstallation Instructions
- Installing the Kit on the Mainframe
- Installation Instructions
- Postinstallation Instructions

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

Topic	Required Information	Default	Comment
General	Operating system	-	OS/390 V2R5 and up (including z/OS)
-	Disk space	-	150 cylinders
-	Memory	-	Minimum 1MB per user
-	Installation high-level qualifier	OCL904	-
-	Volume	-	-
-	Unit	3390	SMS only: unit where SMS resides.
-	Output class	A	-
-	JCL job card	-	An optional card (up to 6 lines) to replace the prefix job (entered as it will appear in the job)
-	ISPF load library name	ISP.SISPLOAD	-
CICS	CICS EXCI load library name	CICS.CICS.SDFHEXCL	To access VSAM data under CICS

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 for VSAM output

Table 2–3 Installation Checklist

Step	Comment/Outputs
tso profile prefix	Ensures that the user name is not used as part of the dataset name allocated in the next steps
Allocate dataset: {HLQ}.TRANSMIT.KIT	130 tracks (3390), format=FB, record length=80, block size=3120
Allocate dataset: {HLQ}.TRANSMIT.LOAD	420 tracks (3390), format=FB, record length=80, block size=3120
FTP files to OS/390 (or z/OS)	FTP using binary mode
RECEIVE INDSNAME('{HLQ}.TRANSMIT.KIT')	-
da('{HLQ}.TRANSMIT.LIB') UNIT(unit) VOLUME(volume)	-
EX {HLQ}.TRANSMIT.LIB(PREPARE)	Successful MAXCC is 0, 4 or 8 BUILDKIT.SRC and BUILDKIT.LOAD created
EX {HLQ}.BUILDKIT.SRC(NAVINST)	Successful MAXCC is 0 or 4

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, which is A
- If you use SMS to manage all datasets (you cannot provide unit and volume information), 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 the Oracle Connect for VSAM output. Assigning a device which is set on HOLD prevents the loss of log information when the Oracle Connect for VSAM started tasks finish.

Oracle Connect for VSAM software for the IBM OS/390 machine is contained in the following datasets:

- `OCL904.TRANSMIT.KIT`
- `OCL904.TRANSMIT.LOAD`

These datasets are provided on a CD-ROM in the directory `Oracle_Connect\VSAM_Legacy_Adapter`.

Installing the Kit on the Mainframe

1. On the mainframe, run the following command:

   ```
   tso profile noprefix
   ```

2. The user name will not be used as part of the dataset name. On some systems this is the default.
3. On the mainframe, allocate datasets with the following space for each of these files:
   - `OCL904.TRANSMIT.KIT` = 130 tracks (3380 and 3390 disks)
   - `OCL904.TRANSMIT.LOAD` = 420 tracks (3380 and 3390 disks)

   For each dataset: `RECFM=FB` and `LRECL=80`. The block size is 3120.
4. Using FTP, copy `OCL904.TRANSMIT.KIT` and `OCL904.TRANSMIT.LOAD` in binary mode from the installation CD to the mainframe. You can replace the `OCL904` high-level qualifier to any qualifier you want.

Installation Instructions

Complete the following steps to install Oracle Connect for VSAM:

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 for VSAM installation. Assign the high-level qualifier you specified in step 7 of the preinstallation. The default value is `OCL904`.

Note: You can use more than one high-level qualifier (such as `ACME.DEV.OCL904`) 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 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 the library is extracted to the current unit and volume.

3. Execute the `PREPARE` member of the *nnn*`.TRANSMIT.LIB` library:

   ```
   ex PREPARE
   ```

   Follow the instructions in the Response column in Table 2–4 for each entry in the Screen column.

Table 2–4 Installation Prepare Job Prompts and Responses

Screen	Response
DO YOU WANT TO USE SMS MANAGED STORAGE FOR THIS INSTALLATION Y/N [N] :	If you want to manage the storage using SMS, answer Y, otherwise answer N.
ENTER THE STORCLASS FOR INSTALLATION TEMP DATASETS [ ] :	This prompt is displayed only if SMS is used to manage the installation (you answered Y to the first prompt). Enter the storage class
ENTER THE UNIT NAME FOR INSTALLATION TEMP DATASETS [3390] :	If a storage class is not specified, enter the unit name for temporary datasets used during the installation procedure
ENTER THE VOLUME NAME FOR INSTALLATION TEMP DATASETS :	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

Table 2–4 (Cont.) Installation Prepare Job Prompts and Responses

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

The following libraries are generated:

nnn.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, execute the NAVINST member:

```
ex NAVINST
```

Follow the instructions in the Response column in Table 2–5 for each entry in the Screen column.

Table 2–5 Installation Prompts and Responses

Screen	Response
DO YOU WANT TO USE SMS MANAGED STORAGE FOR THIS INSTALLATION Y/N [N] :	If you want to manage the storage using SMS, answer Y, otherwise answer N.
THE SOFTWARE WILL BE INSTALLED UNDER THE HIGH LEVEL QUALIFIER THAT YOU WILL CHOOSE. ENTER THE HIGH LEVEL QUALIFIER ["QUALIFIER"] :	The high-level qualifier for the installation (referred to as *INSTROOT* throughout this guide) You can use more than one high-level qualifier (such as `ACME.DEV.VA10`). 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 "transmit" and "buildkit" cannot be used as high-level qualifiers.
ENTER THE STORCLASS FOR TEMP DATASETS ['STORCLASS'] :	This prompt is displayed only if SMS is used to manage the installation (you answered Y to the first prompt). Enter the storage class
ENTER THE UNIT NAME FOR INSTALLATION TEMP DATASETS [3390] :	The unit name for temporary datasets used during the installation procedure.
ENTER THE VOLUME NAME FOR INSTALLATION TEMP DATASETS :	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
PLEASE CONFIRM (YES/NO/QUIT) [YES] :	Confirm the entered details
ENTER THE OUTPUT CLASS FOR INSTALLATION OUTPUT [A] :	Enter the output class for Oracle Connect for VSAM output. Assigning a device which is set on `HOLD` prevents the loss of log information when Oracle Connect for VSAM started tasks finish (the default is A).
DO YOU WANT TO USE THE DEFAULT JOB CARD Y/N [Y]	A job card is displayed. If you want to use a replacement card, it must be entered as it will appear in the job. You can enter up to six lines. Enter a blank line to end input. If you do not enter a card, the Oracle Connect for VSAM default card is used.

Table 2–5 (Cont.) Installation Prompts and Responses

Screen	Response
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]:	In order to manage Oracle Connect on this machine 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 machine. The administrative rights can be changed from within Oracle Studio after the installation.
DO YOU WANT TO PERFORM A MANUAL (M) OR AUTOMATIC (A) INSTALLATION [A] :	If you want to review the JCL used to install Oracle Connect for VSAM, before it is submitted, respond M for a manual installation.
PLEASE REVIEW AND SUBMIT FOR EXECUTION THE DSN1 (INSTJBOR)	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.

5. In the *nnn*.BUILDKIT.SRC library, execute the VSAM member:

```
ex VSAM
```

Follow the instructions in the Response column in Table 2–6 for each entry in the Screen column.

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

Screen	Response
DO YOU WANT ORACLE CONNECT FOR LEGACY ADAPTER TO WORK WITH VSAM UNDER CICS (YES/NO) [NO] :	If you want to access VSAM data under CICS, using the Oracle Application Server Integration Adapter for VSAM, respond YES.
ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHEXCI] :	If you responded YES to working under CICS, enter the CICS EXCI load library name only if you do not want the default.
PLEASE CONFIRM (YES/NO/QUIT) [YES] :	If you responded YES to working under CICS, confirm the entered details.
ENTER THE ISPF LOAD LIBRARY NAME [ISP.SISPLOAD] :	Enter the high-level qualifier and name of the ISPF load library

Table 2–6 (Cont.) VSAM Adapter-Specific Installation Prompts and Responses

Screen	Response
PLEASE CONFIRM (YES/NO/QUIT) [YES] :	Confirm the entered details
ENTER THE OUTPUT CLASS FOR INSTALLATION OUTPUT [A] :	Enter the output class for Oracle Connect for VSAM output. Assigning a device which is set on `HOLD` prevents the loss of log information when the Oracle Connect for VSAM started tasks finish (the default is A).
DO YOU WANT TO USE THE DEFAULT JOB CARD Y/N [Y]	A job card is displayed. If you want to use a replacement card, it must be entered as it will appear in the job. You can enter up to six lines. Enter a blank line to end input. If you do not enter a card, the Oracle Connect for VSAM default card is used.

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.

Postinstallation Instructions

The following postinstallation tasks must be done to work with Oracle Connect for VSAM:

- General Postinstallation Procedures
- Starting the Daemon
- Setting Up Oracle Connect for VSAM for Reentrancy
- Setting Up Oracle Connect for VSAM to Update VSAM Data

General Postinstallation Procedures

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

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

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

  Using FTP, copy the `BRANDBIN` file, in binary mode, from the `Oracle Connect\VSAM Legacy Adapter` directory in the installation CD to the mainframe, to *`INSTROOT`*`.DEF.BRANDBIN`.

- Define the LOADAUT library as an APF-authorized library

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

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

  where vol002 is the volume where you installed Oracle Connect for VSAM and *INSTROOT* is the high-level qualifier where Oracle Connect for VSAM is installed.

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

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

  Make sure that the library is APF-authorized, even after an IPL (reboot) of the machine.

- Move the *NSTROOT*.USERLIB(ATTDAEMN) and *NSTROOT*.USERLIB(ATTSRVR) members to any active proclib (such as user.proclib). ATTDAEMN 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, 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 of the OS/390 machine.

  For example, before:

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

After:

```
// PARM='-B -L ip_address:2252 START IRPCDINI'
```

where *ip_address* is ip address of the machine, 2552 is the default port for starting the daemon and `IRPCDINI` is the default daemon configuration.

- In the security manager, define `ATTDAEMN` and `ATTSRVR` with a started task class and a general profile that enables access to an Open OS/390 segment (that defines access to TCP/IP OA sockets)
- 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, 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 nnn is the high-level qualifier you assign for the installation.
- For information about specifying Oracle Connect for VSAM as the service using port 2552 in the TCP/IP network services file, consult TCP/IP documentation.

Starting the Daemon

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

```
'/s ATTDAEMN'
```

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

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

```
//*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 VSAM for Reentrancy

To set up Oracle Connect for VSAM to be reentrant, move *INSTROOT*.LOAD to the Link Pack Area (LPA).

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

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

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

- Place the *INSTROOT*.LOAD(ATYSVCW) member in an APF-authorized library outside the LPA.
- Change the ATTSRVR member (located in the active proclib), by adding the following to the STEPLIB list:

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

Where *apf_library* is the APF-authorized library outside the LPA where the ATYSCVW member was moved.

Setting Up Oracle Connect for VSAM to Update VSAM Data

Perform the following to set up Oracle Connect for VSAM so that you can update VSAM data.

VSAM Running Under CICS To set up Oracle Connect for VSAM 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 for VSAM 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 for VSAM 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 Oracle Connect 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.

Table 2–7 Preinstallation Information

Topic	Required Information	Default	Comment
CICS	CICS EXCI load library name	CICS.CICS.SDFHEXCL	To access IMS/DB or VSAM data under CICS

Table 2–8 Required Permissions

Permission
Permission to read the CICS EXCI library (when accessing IMS/DB or VSAM data under CICS)

In the *nnn*.BUILDKIT.SRC library, execute the CUSTOCL member:

```
ex CUSTOCL
```

Follow the instructions in the Response column in Table 2–9 for each entry in the Screen column.

Table 2–9 VSAM Adapter Installation Prompts and Responses

Screen	Response
DO YOU WANT ORACLE CONNECT FOR LEGACY ADAPTER TO WORK WITH IMS/DB (YES/NO) [YES] :	Answer YES to this prompt if you have Oracle Connect for IMS/DB already installed.
ENTER DBD LIBRARY NAME [IMS.DBDLIB] :	If you responded YES to working with IMS/DB, enter the library where database definition (DBD) files are located.
ENTER PSB LIBRARY NAME [IMS.PSBLIB] :	If you responded YES to working with IMS/DB, enter the library where Program Specification Blocks (PSBs) are located.
ENTER YOUR PSB NAME [HOSPPSB] :	If you responded YES to working with IMS/DB, enter the name of the PSB file to use.
PLEASE CONFIRM (YES/NO/QUIT) [YES] :	If you responded YES to working with IMS/DB, confirm the entered details.
DO YOU WANT ORACLE CONNECT FOR LEGACY ADAPTER TO WORK WITH IMS/DB UNDER CICS (YES/NO) [NO] :	If you want to access IMS/DB data under CICS, using the Oracle Application Server Integration Adapter for IMS/DB, respond YES.
ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHEXCI] :	If you responded YES to working with IMS/DB under CICS, enter the CICS EXCI load library name only if you do not want the default.
PLEASE CONFIRM (YES/NO/QUIT) [YES] :	If you responded YES to working with IMS/DB under CICS, confirm the entered details.
DO YOU WANT ORACLE CONNECT FOR LEGACY ADAPTER TO WORK WITH CICS APP ADAPTER (YES/NO) [YES] :	Answer YES to this prompt if you have Oracle Connect for CICS already installed.
ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHEXCI] :	If you responded YES to working with the CICS adapter, enter the CICS EXCI load library name only if you do not want the default.
PLEASE CONFIRM (YES/NO/QUIT) [YES] :	If you responded YES to working with the CICS adapter, confirm the entered details.
THE VSAM DRIVER IS INSTALLED AUTOMATICALLY. DO YOU ALSO WANT ORACLE CONNECT FOR LEGACY ADAPTER TO WORK WITH VSAM UNDER CICS (YES/NO) [NO] :	Answer YES to this prompt if you want to access VSAM data under CICS, using the Oracle Application Server Integration Adapter for VSAM.

Table 2–9 (Cont.) VSAM Adapter Installation Prompts and Responses

Screen	Response
ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHEXCI] :	If you responded YES to working with VSAM under CICS, enter the CICS EXCI load library name only if you do not want the default.
PLEASE CONFIRM (YES/NO/QUIT) [YES] :	If you responded YES to working with VSAM under CICS, confirm the entered details.
ENTER THE ISPF LOAD LIBRARY NAME [ISP.SISPLOAD] :	Enter the ISPF load library name only if you do not want the default.
PLEASE CONFIRM (YES/NO/QUIT) [YES] :	Confirm the entered details
ENTER THE OUTPUT CLASS FOR INSTALLATION OUTPUT [A] :	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).
DO YOU WANT TO USE THE DEFAULT JOB CARD Y/N [Y]	A job card is displayed. If you want to use a replacement card, it must be entered as it will appear in the job. You can enter up to six lines. Enter a blank line to end input. If you do not enter a card, the Oracle Connect default card is used.

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 postinstallation tasks, as described in "Postinstallation Instructions" on page 2-12, as required.

Installing Oracle Studio

Note: If you have Oracle Studio already installed because you are also using another legacy adapter, you do not need to reinstall it.

The other legacy adapters are:

- Oracle Application Server Integration Adapter for CICS
- Oracle Application Server Integration Adapter for IMS/DB
- Oracle Application Server Integration Adapter for IMS/TM
- Oracle Application Server Integration Adapter for Tuxedo

This section explains how to install Oracle Studio software from the distribution CD-ROM.

Installing Oracle Studio from the CD-ROM

Assuming that the CD-ROM drive is D:, the installation file is located in the `D:\Oracle_Studio` directory. Install the software from the CD-ROM by running the self-extracting executable installation file, `OSL904-win32.exe`, either using the Run option in the Windows Start menu or through Windows Explorer.

Caution: If you are installing Oracle Studio on a Windows XP machine, you cannot specify a logical drive as the Destination folder for the installation.

Configuring Oracle Connect for VSAM

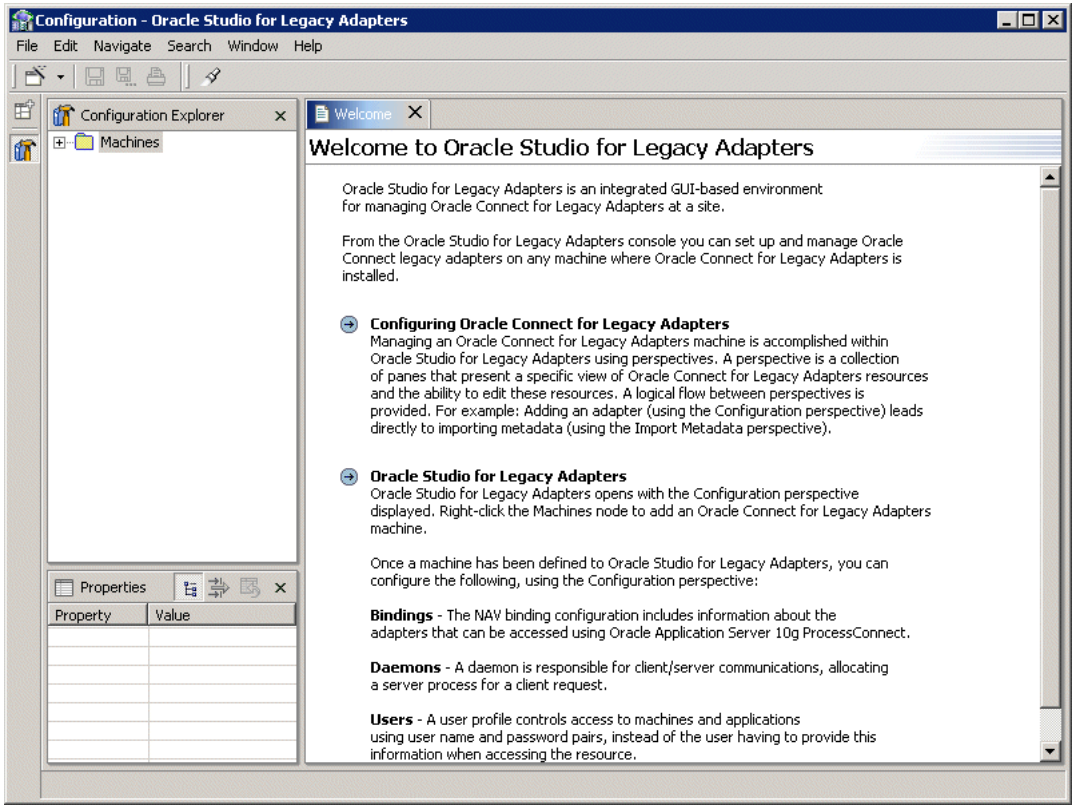
All modeling of Oracle Connect for VSAM is performed using Oracle Studio. To use Oracle Studio you first configure it to enable access to the IBM OS/390 machine.

Note: The following tasks assume you have permission to access the IBM OS/390 machine where the VSAM data resides and that Oracle Connect for VSAM daemon is running on this machine.

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

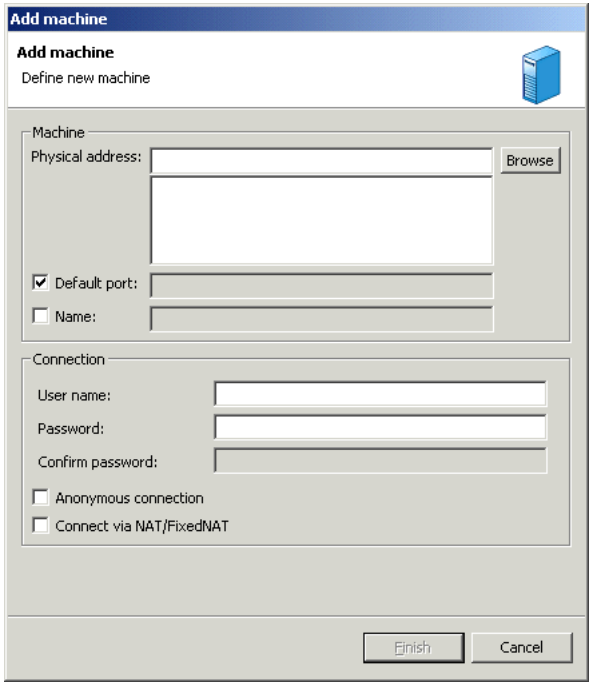
Task: Setting Up the IBM OS/390 Machine

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**. Oracle Studio opens, displaying the Configuration perspective.



2. Right-click **Machines** in the **Configuration Explorer** and choose **Add Machine** from the popup menu.

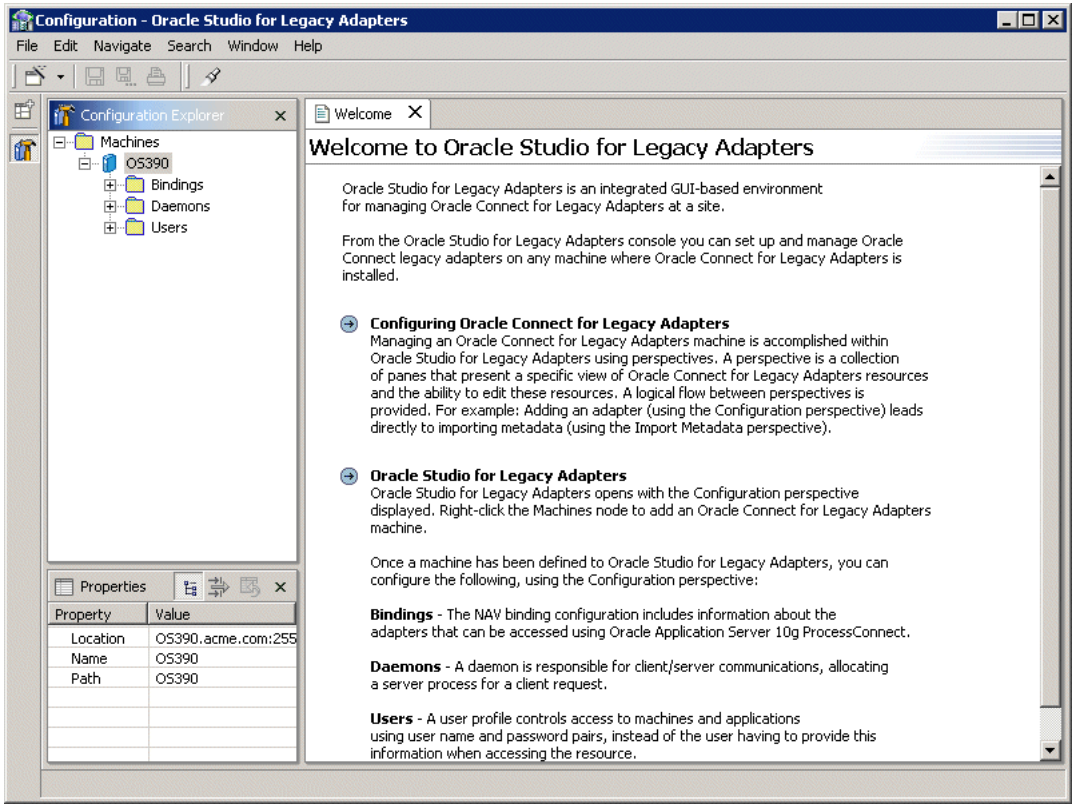
   The Add machine window appears.



3. In the Add Machine window, uncheck the Default port check box and enter 2552 as the port number.
4. Enter the name of the machine you want to connect to, or click **Browse** to choose the machine from the list of machines that is displayed.
5. Specify the username and password of the user who was specified as the administrator when Oracle Connect for VSAM was installed.

> **Note:** Checking the **Anonymous connection** checkbox enables anyone having access to the machine to be an administrator, if this was defined for the machine.

6. Click **Finish**. The machine is displayed in the **Configuration Explorer**.



Securing Access to Oracle Connect for VSAM

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

During modeling the following security mechanisms can be applied:

- Setting Password Access to Oracle Studio
- Specifying Users with Administrative Rights

At runtime client access to Oracle Connect for VSAM is provided by the user profile:

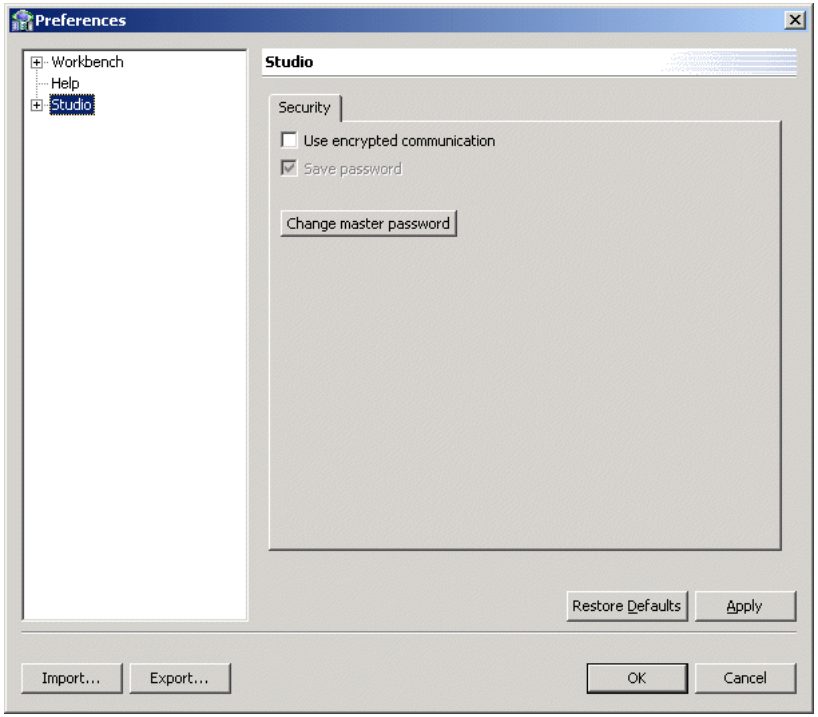
- Setting Up Runtime User Access to the IBM OS/390 Machine

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.

Task: Setting Up a Master Password to Access Oracle Studio

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**. Oracle Studio opens.
2. Choose **Window > Preferences** from the menu bar. to open the Preferences window.
3. Select the **Studio** node.



Note: The **Save password** checkbox is for future use.

4. Click **Change master password** to open the Change master password window.

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

Specifying Users with Administrative Rights

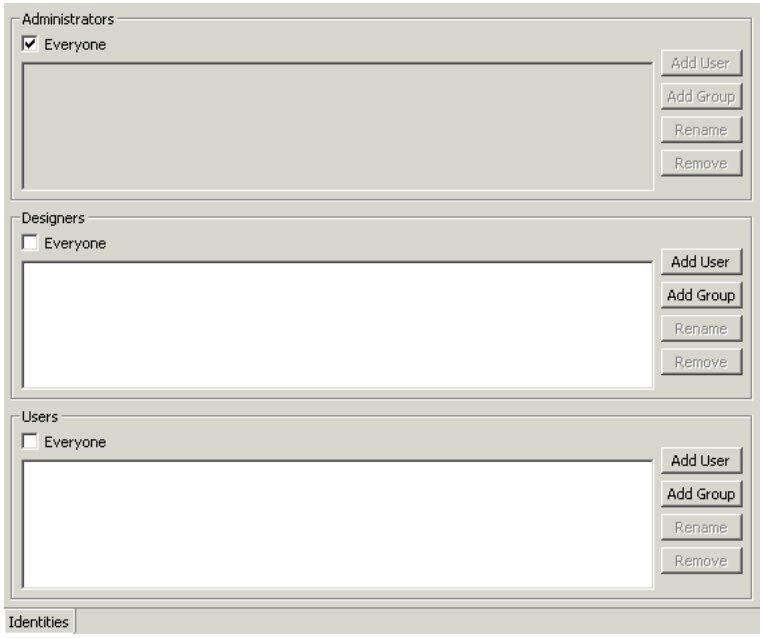
By default, only the user who was specified during the installation as an administrator has the authorization to modify settings on that machine from Oracle Studio. This user can then authorize other users to make changes or to view the definitions for a selected machine. Adding a machine to Oracle Studio is described in "Task: Setting Up the IBM OS/390 Machine" on page 2-20.

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

Task: Setting Administrative Rights to a Machine

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle** > **Studio** > **Studio**. Oracle Studio opens, displaying the Configuration perspective.
2. Right-click the machine in the **Configuration Explorer** and choose **Administration Authorization** from the popup menu.

   The Administration Authorization window opens.



The window has three sections:

Administrators - Administrators can view and modify all the definitions in Oracle Studio for the selected machine. On initial entry to Oracle Studio, every user is defined as a system administrator.

Designers - Designers can view all the definitions for the machine in Oracle Studio and can modify any of the definitions under the **Bindings** and **Users** nodes for the selected machine. 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 machine in Oracle Studio for the selected machine. 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 machine.

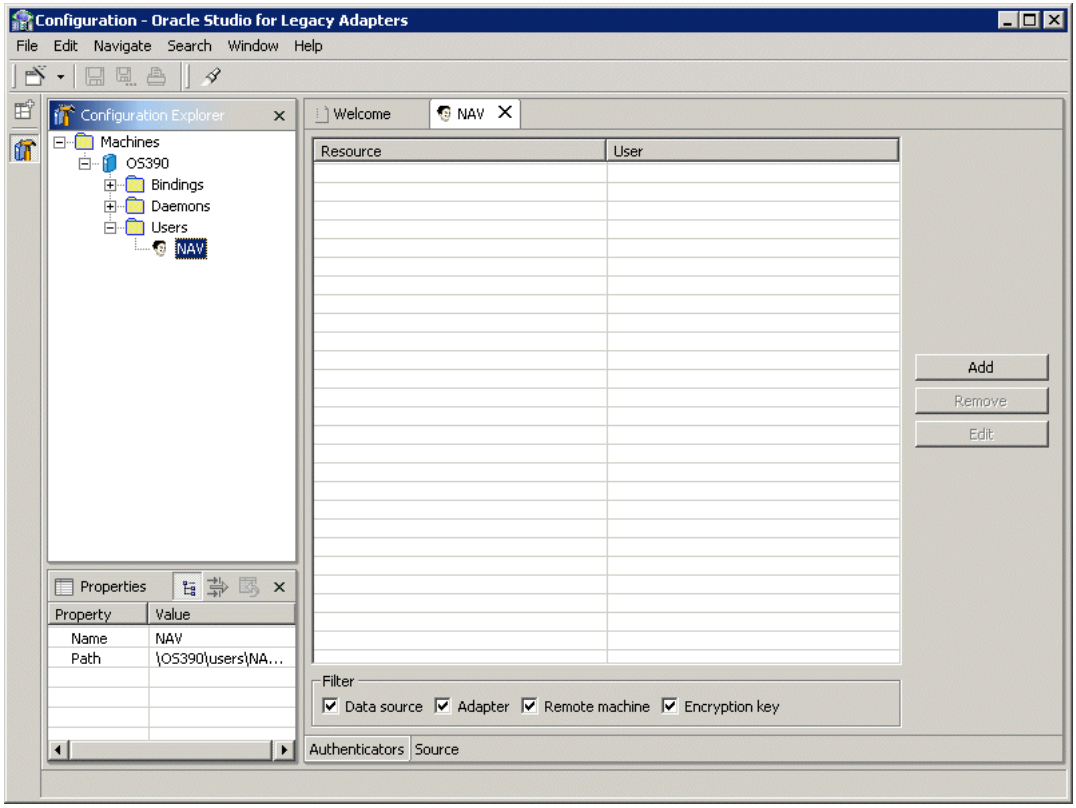
   Once a name has been added to a section, only the user or group who logs on with that user name has the relevant authorization.

Setting Up Runtime User Access to the IBM OS/390 Machine

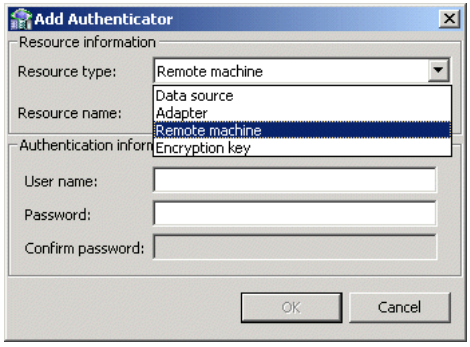
At runtime, client access to Oracle Connect for VSAM is provided by the user profile. A user profile contains name and password pairs that are used to access a machine, data source or application at runtime, when anonymous access is not allowed.

1. In the **Configuration Explorer**, click the + next to the machine node of the machine for which you want to set the user name and password.
2. Click the + next to the **Users** node.
3. Right-click the **NAV** user profile and choose **Edit User**.

Caution: You must not delete the NAV user profile since it also contains Oracle Connect for VSAM system settings.



4. In the User editor, click **Add** to open the Add authenticator window.
5. Select the **Remote Machine** resource type.



6. Enter the name of the IBM OS/390 machine defined in the previous task.
7. Enter the name and password used to access the machine and confirm the password.
8. Click **OK**.

3

Modeling Interactions for Oracle Application Server Integration Adapter for VSAM

Modeling interactions for Oracle Application Server Integration Adapter for VSAM involves defining the data source and an Oracle Connect for VSAM backend adapter using Oracle Studio. All the definitions specified in Oracle Studio are written to the OS/390.

This chapter contains the following sections:

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

Setting Up the VSAM Data Source

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

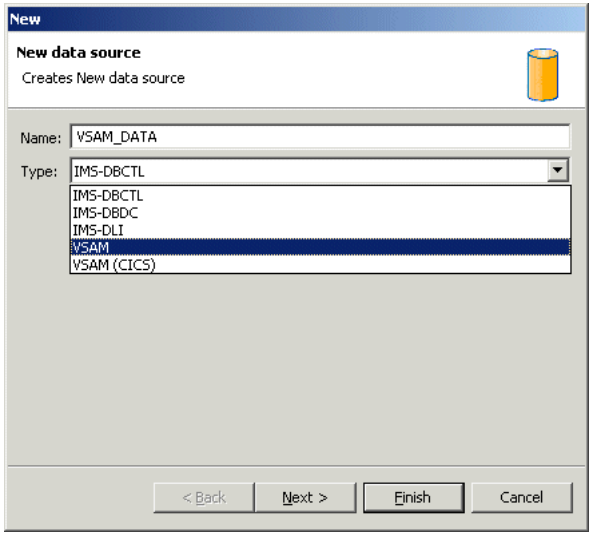
Task: Adding the VSAM Data Source Definition

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. In the Configuration Explorer, click the + next to the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Click the + next to the **Bindings** node. The binding configurations available on this machine are listed.
4. Click the + next to the **NAV** binding node. The NAV binding configuration includes branches for data sources and adapters that are located on the machine.

   Caution: You must not delete the NAV binding configuration since it also contains Oracle Connect for VSAM system settings.

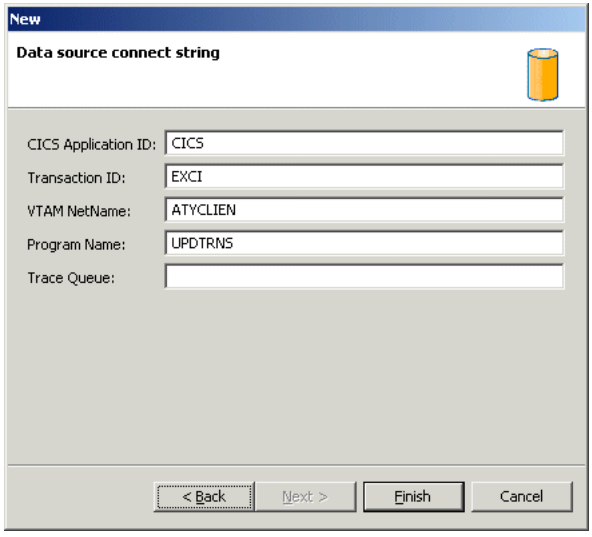
5. Right-click **Data sources** in the **Configuration Explorer** and choose **New data source** from the popup menu, to open the New data source wizard.
6. In the **New** window, enter a name for the VSAM data source. The name can contain letters and numbers and the underscore character only.
7. Change the `Type` field as follows:
   - If you are accessing VSAM data under CICS, select **VSAM (CICS)**
   - If you are accessing VSAM data directly, 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.



8. Click **Next**.
9. Enter the Data source connect string, as follows:

   If you chose **VSAM (CICS)**, the following window opens:



where:

CICS Application ID - The VTAM applid of the CICS target system. The default value is CICS. This parameter is used when updating VSAM data. You can determine this value by activating the CEMT transaction on the target CICS system. On the bottom right corner of the screen appears the legend APPLID=*target_system*.

Transaction ID - The mirror transaction within CICS that receives control through MRO, which transfers the transaction from the Oracle Connect for VSAM environment to CICS. The default value is EXCI.

VTAM NetName - The VTAM netname of the specific connection being used by EXCI (and MRO) to relay the program call to the CICS target system. For example, if you issue to CEMT the following command:

```
CEMT INQ CONN
```

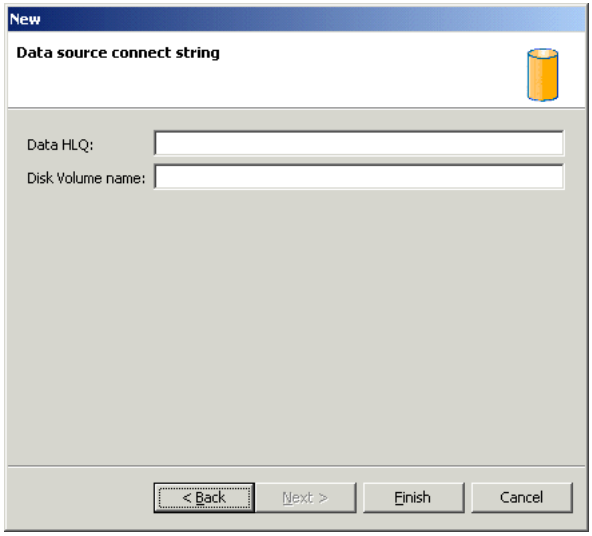
you see on the display screen that the netname is BATCHCLI (this is the default connection supplied by IBM upon the installation of CICS). The default value is ATYCLIEN.

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 2-16 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 chose **VSAM**, the following window opens:



where:

Data HLQ - The high-level qualifier where the data files are located. If a value is not specified in this field, 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 (volume) where the data resides.

10. Click **Finish**.

The data source you just defined now appears in the **Configuration Explorer** list.

Importing Metadata for the VSAM Data Source

Oracle Connect for VSAM requires metadata describing the VSAM data source records and the fields in these records.

Use the Import Metadata perspective in Oracle Studio to import metadata for the VSAM data source from COBOL copybooks, which describe the data. To import the metadata involves the following tasks:

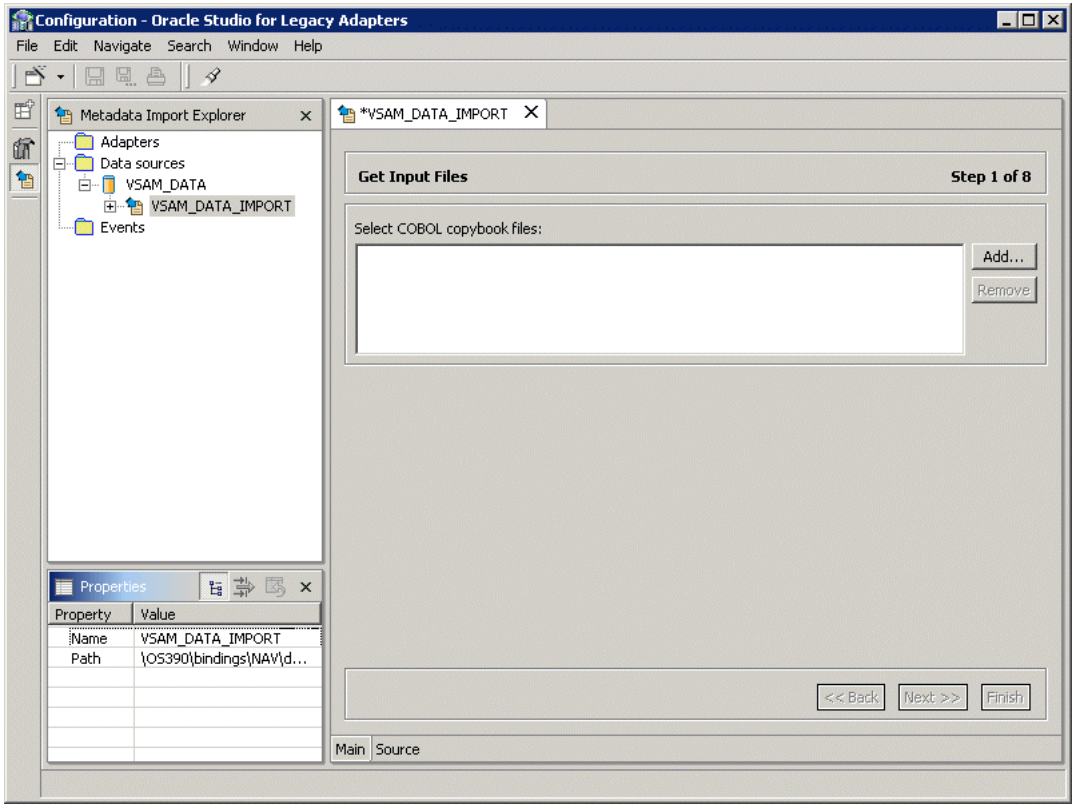
- Task 1: Defining an Import Procedure
- Task 2: Generating Metadata for the VSAM Data

Task 1: Defining an Import Procedure

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. In the Configuration Explorer, click the + next to the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Click the + next to the **Bindings** node. The binding configurations available on this machine are listed.
4. Click the + next to the **NAV** binding node.
5. Click the + next to the **Data sources** node.
6. Right-click the VSAM data source defined in "Setting Up the VSAM Data Source" on page 3-2.
7. Choose **Import** from the popup menu, to open the Metadata Import perspective, with the VSAM data source displayed under the data sources list in the Metadata Import Explorer.
8. Right-click the VSAM data source and choose **New Import** from the popup menu, to open the New Metadata Import wizard.
9. Enter a name for the import. The name can contain letters and numbers and the underscore character only.

   Note: The New import window is the same for both the VSAM imports (VSAM under CICS and direct), with the exception of the `Import Type` value.

10. Click **Finish**. The metadata import wizard opens in Oracle Studio.

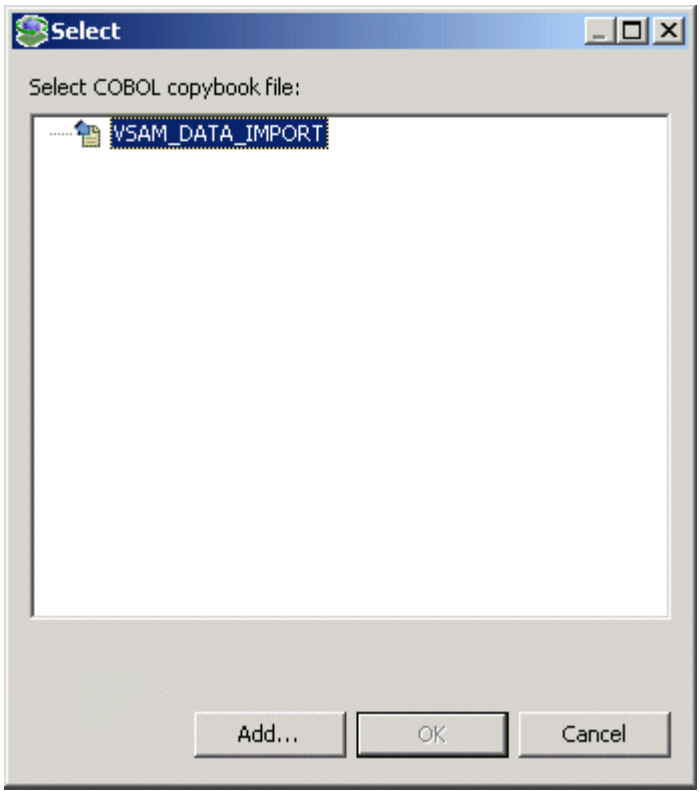


Task 2: Generating Metadata for the VSAM Data

After defining an import as described in the preceding task, the metadata import wizard opens in Oracle Studio. COBOL copybooks are used to create the metadata.

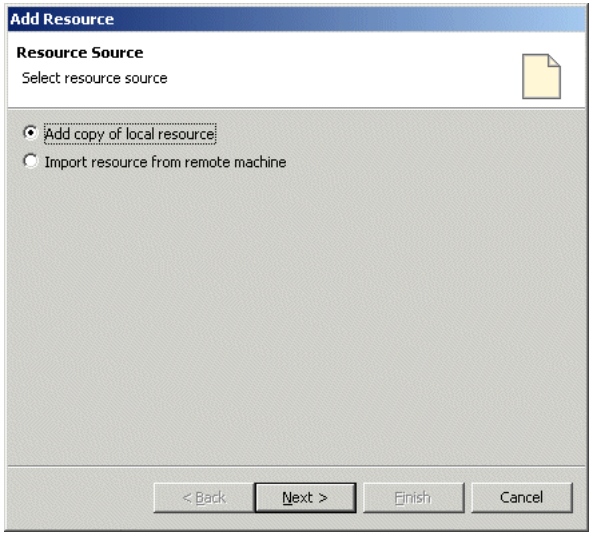
1. Click **Add** to add COBOL copybooks.

   A Select window opens. This window lists any files selected.

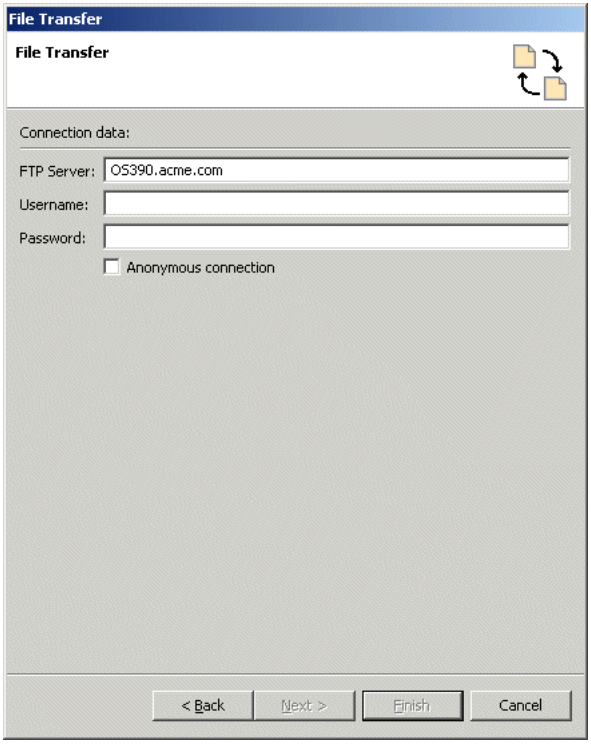


2. Click **Add**.

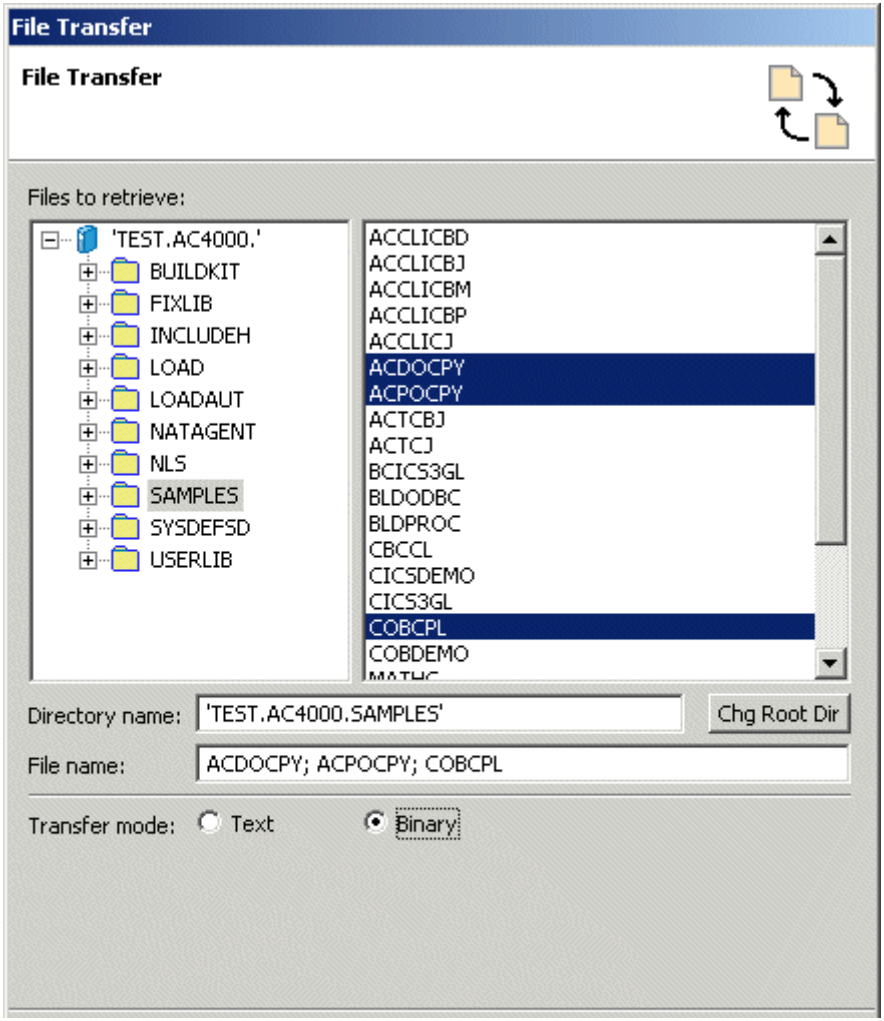
   The Add Resource window opens, which provides the option to select files from the local machine or FTP the files from another machine.



3. If the files are on another machine, select the **Import resource from remote machine** option to start a file transfer process from the machine.
4. Click **Next** to open the File Transfer wizard.



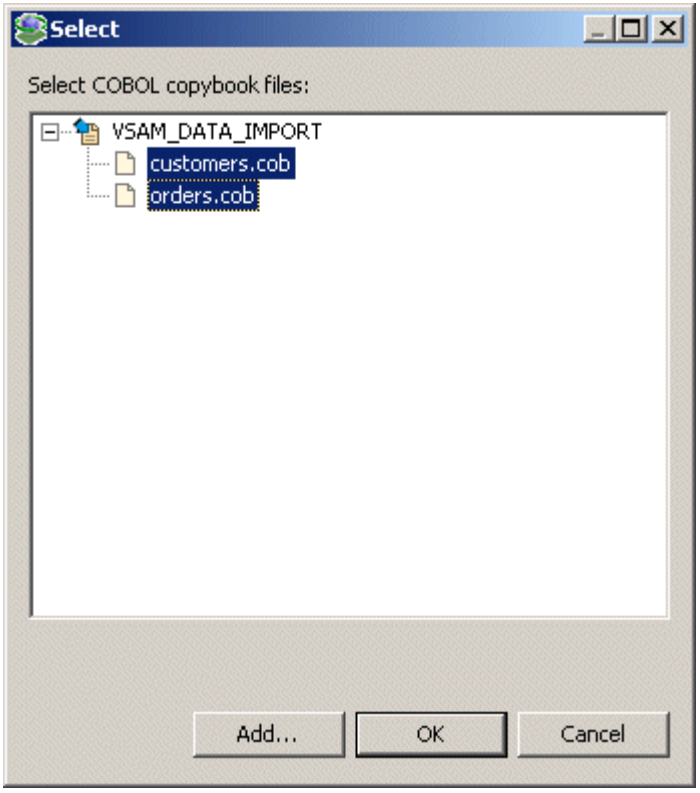
5. Enter the server name where the COBOL copybooks reside and, if not using anonymous access, enter a username and password to access the machine.
6. Click **Next**. The machine is accessed and the directory structure displayed, enabling you to browse to the files to be transferred.



You access the OS/390 machine at the high-level qualifier for the user name. To change the high-level qualifier, enter the name of the high-level qualifier in the Directory name field and click the **Chg Root Dir** button.

7. Navigate to the location where the files reside and select the files to be transferred and the transfer method (**Text** or **Binary**).
8. Click **Next**. A summary screen is displayed with details of what is going to be transferred.

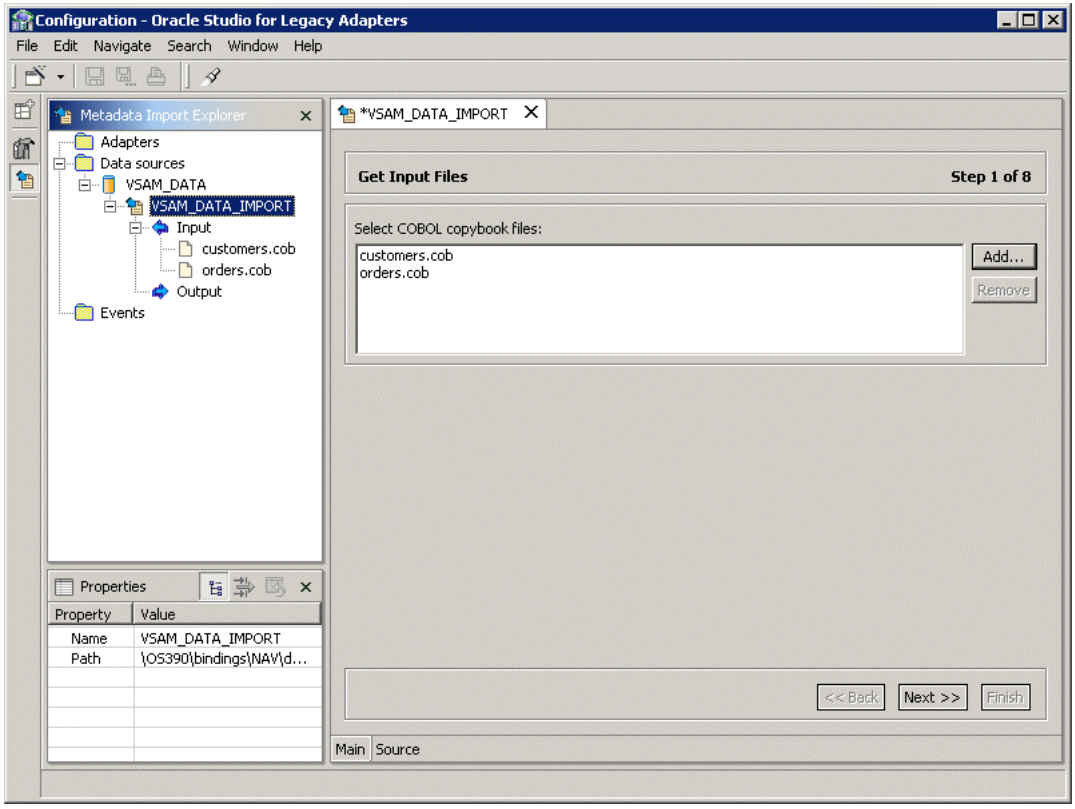
9. Click **Finish** to start the transfer. The **Select** window is redisplayed with the list of transferred files.



10. Select the files you want to use in the import and click **OK**.

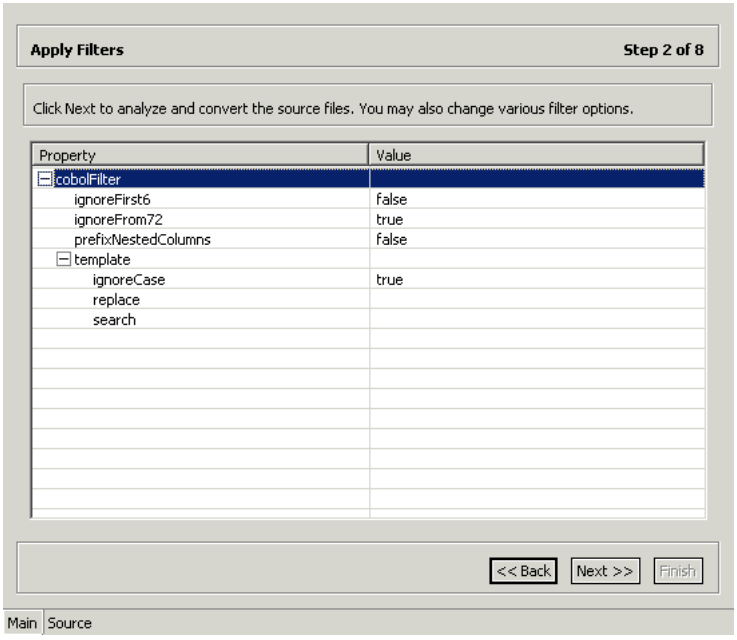
Note: You can import the metadata from one COBOL copybook and later add to this metadata by redoing the import using different COBOL copybooks.

The format of the COBOL copybooks must be the same. 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 case, you must do separate imports.



The files selected are displayed in the wizard. In the Metadata Import Explorer the files used for the input and created for the output can be seen under the Input and Output nodes respectively.

11. When you have completed adding the list of copybooks, click **Next**.



12. Apply filters to the copybooks.

    The following filters are available:

    ignoreFirst6 - Ignore the first six columns in the COBOL copybook.

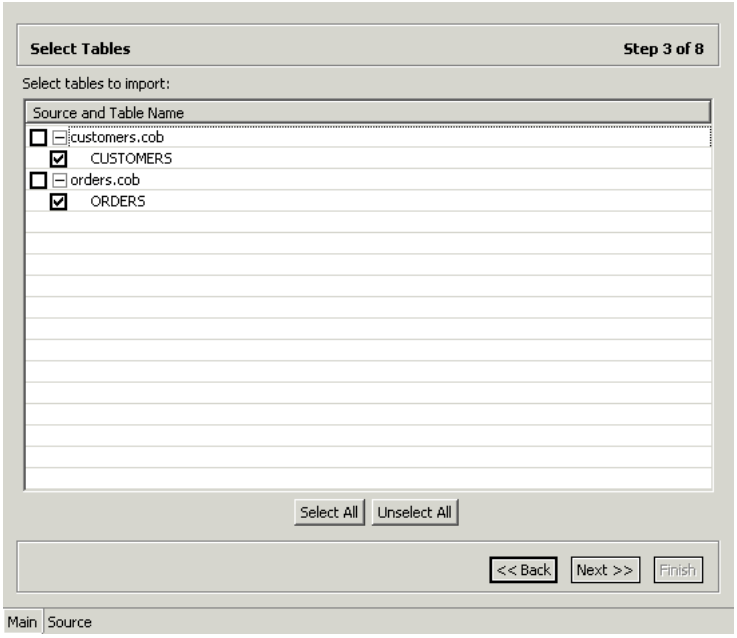
    ignoreFrom72 - Ignore columns 73 to 80 in the COBOL copybook.

    prefixNestedColumns - Prefix all nested columns with the previous level heading.

    In addition, you can specify a search string and the string that will replace this search string in the generated metadata, and whether the replacement is dependent on the case of the found string.

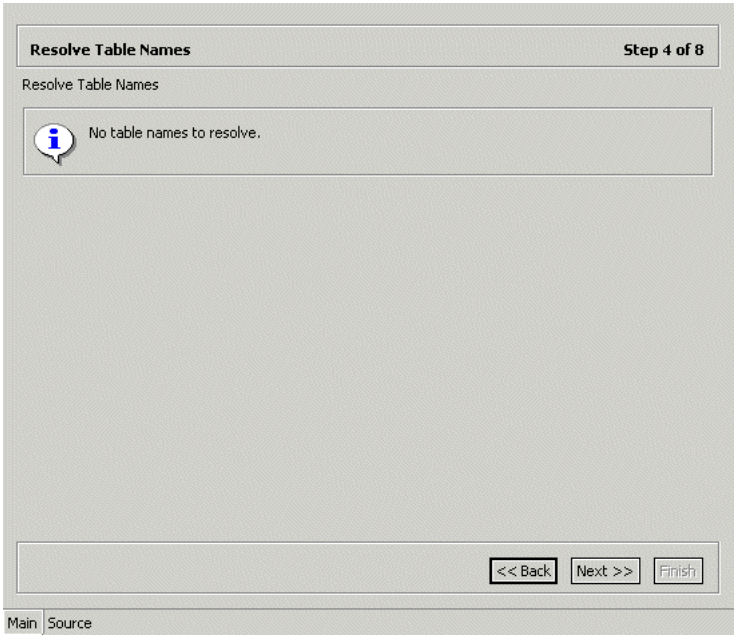
13. Click **Next** to apply the filter settings.

14. The records that are identified in the COBOL copybooks are displayed. You can select the records you want to import.

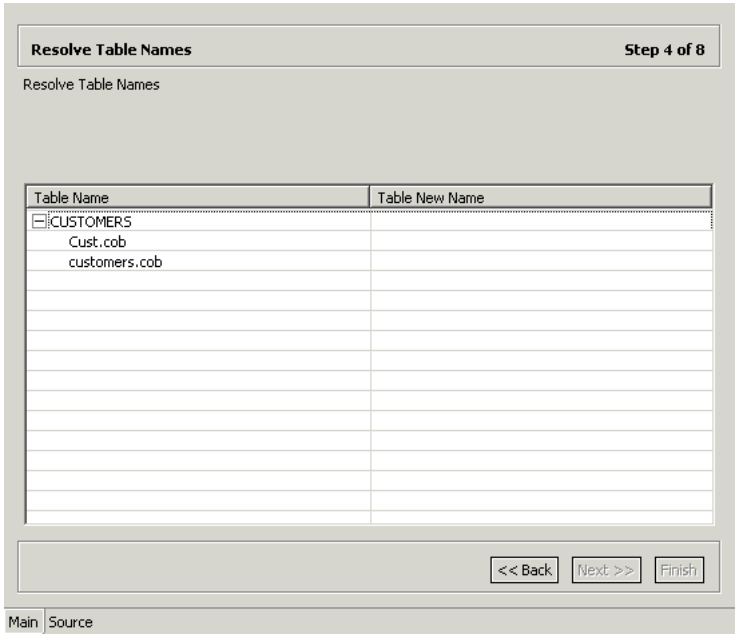


15. Select the records to import and click **Next**.

The import manager identifies the names of the records in the COBOL copybooks that will be imported as tables. If the names are unique, the following window is displayed:



However, if there are multiple records with the same name, the following window is displayed:

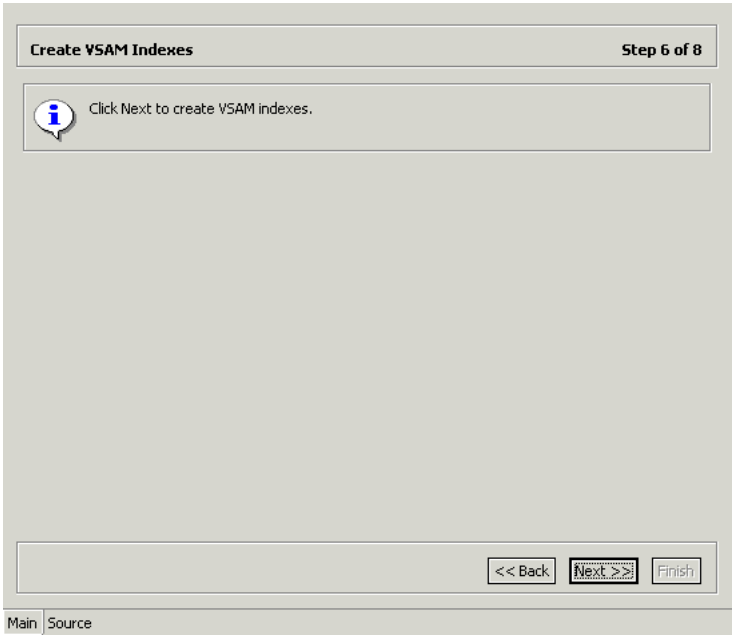


In this case you have to specify a table name you want to use for each of the names displayed, according to the COBOL copybook where the record is defined.

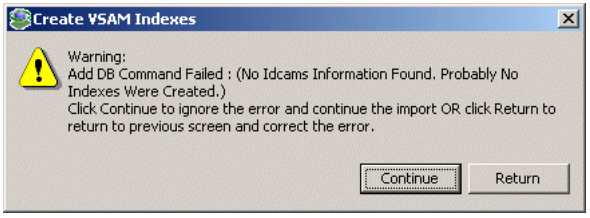
16. Click **Next**.
17. The next step is different, depending on whether the VSAMCICS Import Manager or the VSAM Import Manager is used.

    If the **VSAMCICS** Import Manager is used, specify both physical file name and the logical file name for each record listed.

    If the **VSAM** Import Manager is used, specify the physical file name for each record listed, including the high-level qualifiers.
18. Click **Next**.
19. The OS/390 machine must be accessed to retrieve information to build metadata about the indexes.



If you try to build the indexes and the step fails for any reason (such as the OS/390 machine is temporarily not accessible), a warning message similar to the following message is issued and you continue to the next step.

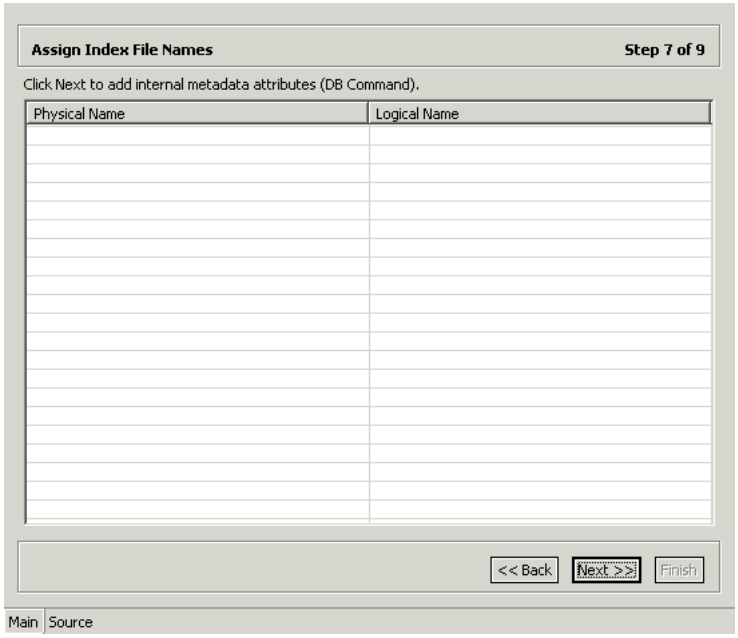


Click **Continue** to go to the next step.

Note: If the indexes are not generated, you can define them manually in the Metadata perspective of Oracle Studio. For details, refer to "Metadata for the VSAM Data Source" on page A-2.

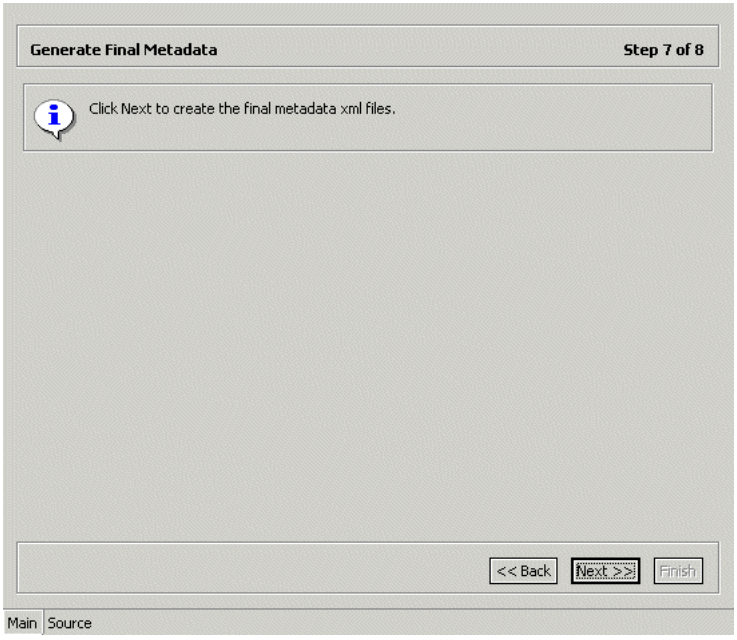
---

20. The next step (Assign Index File Names) applies only if the VSAM under CICS Import Manager is used.

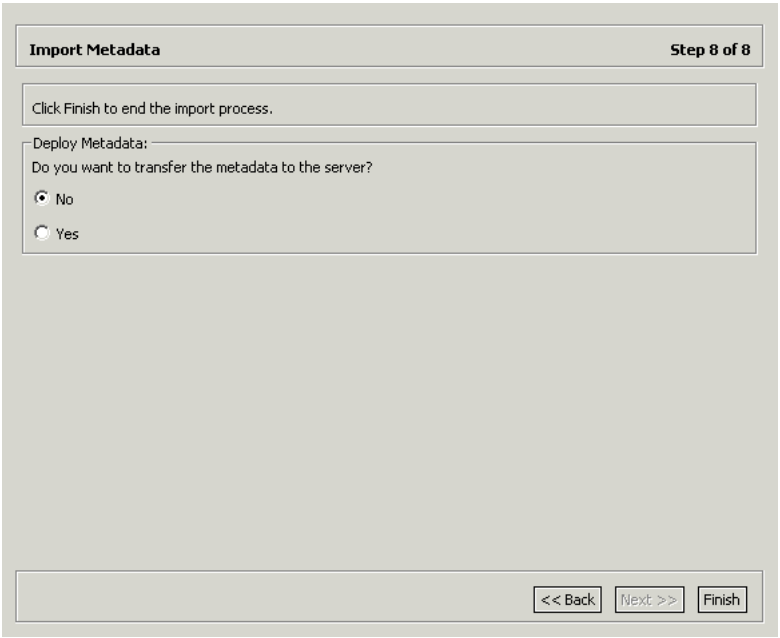


For each table listed, specify next to the alternate indexes for the table the logical file name for the index.

21. Click **Next**.

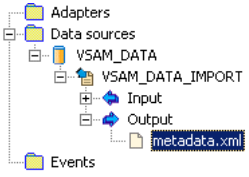


22. Click **Next** to generate the metadata.



23. Specify that you want to transfer the metadata from the Windows machine to the OS/390 machine and click **Finish**.

The metadata is imported based on the options specified and it is stored on the OS/390 machine with the data source. 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 perspective in Oracle Studio. You can also make any fine adjustments to the metadata and maintain it, as necessary.

> **See Also:** "Metadata for the VSAM Data Source" on page A-2 for details about the data source metadata

Setting Up an Oracle Connect for the VSAM Backend Adapter

To work with Oracle Connect for VSAM, you need to set up an adapter definition on the OS/390 machine to handle the interactions to and from VSAM. You use Oracle Studio to specify these definitions.

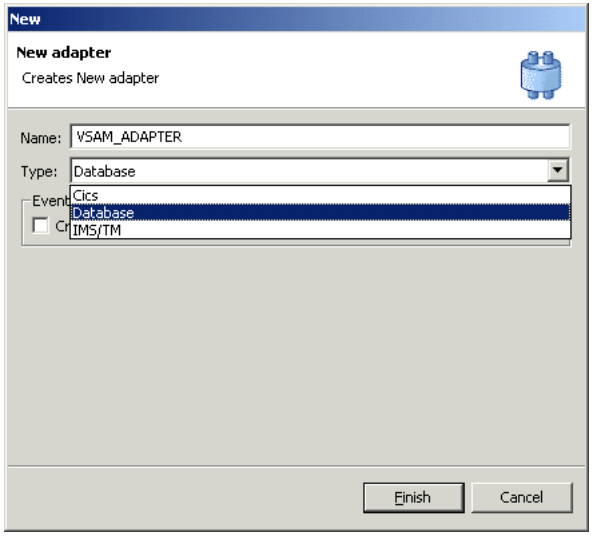
Task: Setting Up the Adapter

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. In the Configuration Explorer, click the + next to the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Click the + next to the **Bindings** node. The binding configurations available on this machine are listed.
4. Click the + next to the **NAV** binding node. The NAV binding configuration includes branches for data sources and adapters that are located on the machine.
5. Right-click **Adapters** and choose **New Adapter** from the popup menu, to open the New Adapter wizard.
6. In the **New** window, enter a name for the adapter.

   Note: The name specified for the adapter is also the name specified setting up the delivery channel in Oracle Application Server ProcessConnect. For details see, "Setting Up the Delivery Channel" on page 4-2.

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

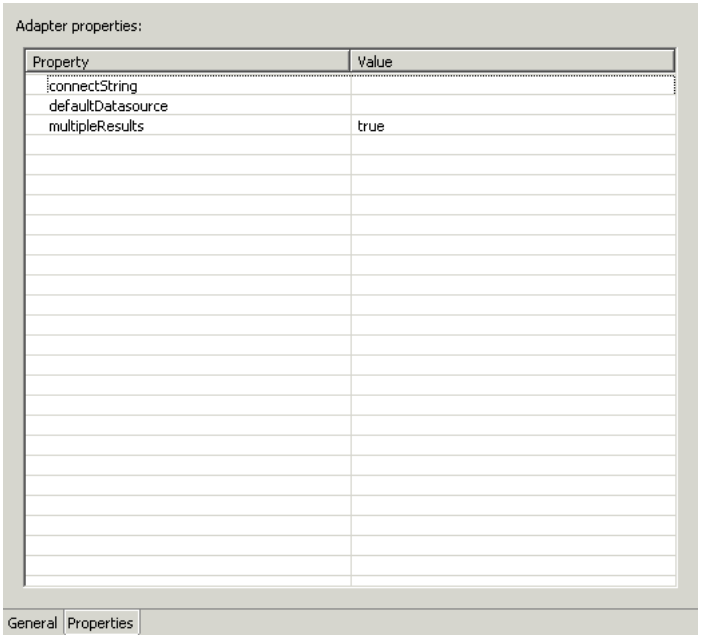
7. Select the `Database` adapter type from the `Type` drop-down list:



8. Check the **Create event queue for the adapter** checkbox.

> **Note:** Checking this box enables you to generate inbound interactions, as described in Generating Inbound Interactions on page 3-33.

9. Click **Finish**. The Adapter is added to the tree and the definition opened for editing.
10. Select the **Properties** tab.



11. Set the following properties for the adapter:

Property	Value	Description
connectString	-	For internal use only
defaultDatasource	*ds_name* For example, VSAM_DATA	The name of a data source in the binding configuration defined in "Setting Up the VSAM Data Source" on page 3-2
multipleResults	true	For internal use only

Note: You must specify the VSAM data source name for the defaultDatasoure property.

Generating Outbound Interactions

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

Use the Metadata perspective in Oracle Studio to generate interaction metadata for the adapter.

- Task 1: Setting Up the Metadata Perspective
- Task 2: Generating Outbound Interactions
- Task 3: Modifying the Generated Outbound Interactions

Task 1: Setting Up the Metadata Perspective

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. In the Configuration Explorer, click the + next to the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Click the + next to the **Bindings** node. The binding configurations available on this machine are listed.
4. Click the + next to the **NAV** binding node.
5. Click the + next to the **Adapters** node.
6. Right-click the Oracle Connect for VSAM backend adapter defined in "Setting Up an Oracle Connect for the VSAM Backend Adapter" on page 3-22.
7. Choose **Edit metadata** from the popup menu, to open the Metadata perspective, with the VSAM adapter displayed under the adapters list.

Task 2: Generating Outbound Interactions

1. In the Metadata perspective, right-click the Interactions node and choose **New** from the popup menu, to open the New Interaction wizard.

   The New Interaction wizard opens with two options displayed, to create interactions:

   - Automatic - Four interactions are generated for each VSAM table enabling the following SQL to be executed:

     SELECT

     INSERT

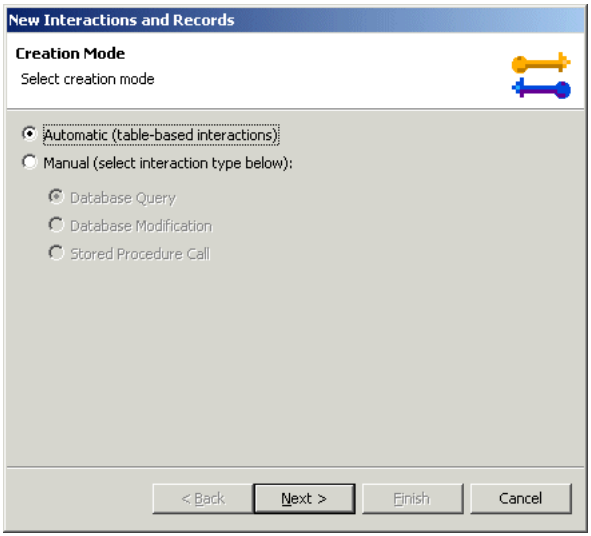
     UPDATE

     DELETE

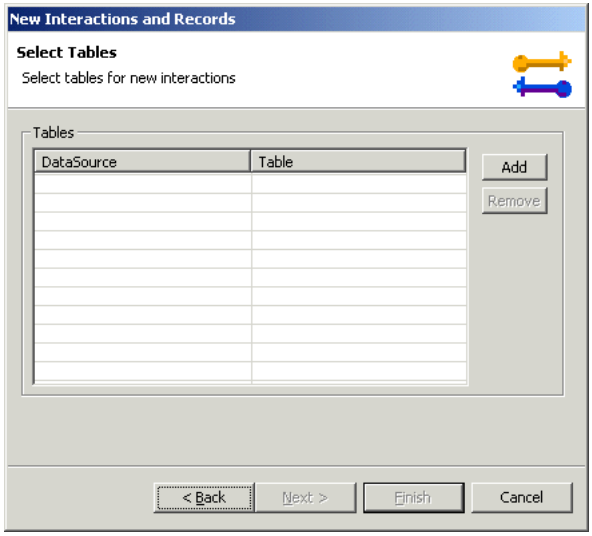
- Manual - One interaction is generated, based on the type of SQL selected: Database Query (a SELECT statement) or Database Modification (an INSERT, UPDATE or DELETE statement).

Note: VSAM does not support the stored procedure option.

It is recommended that you choose the automatic mode to create interactions.



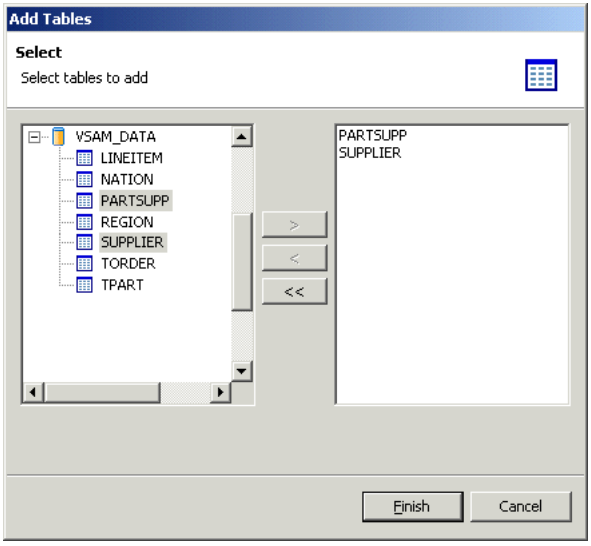
2. Click **Next**. The Select Tables window opens, enabling you to add tables from the VSAM data source that you want to access with the interaction.



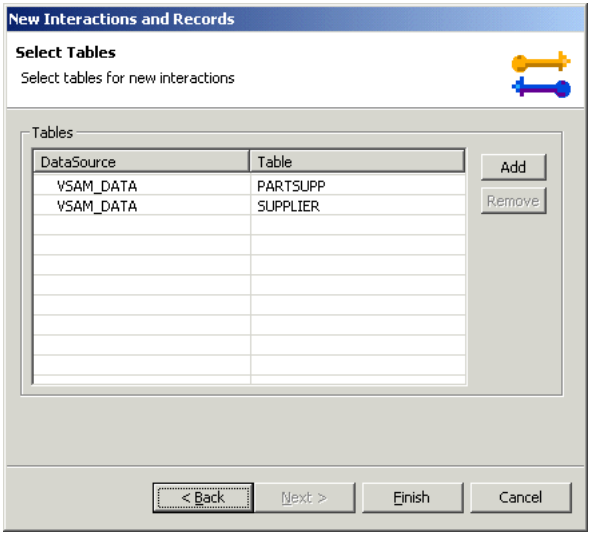
3. Click **Add** to add tables.

   The data sources that have been defined and all the tables, for each data source, that have had metadata defined for them are displayed.

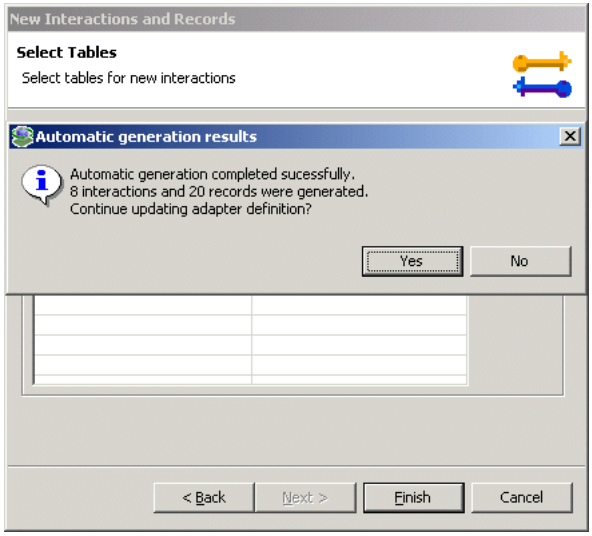
   Select the tables that you want to access with the interaction and click the right arrow to move these tables into the right pane.



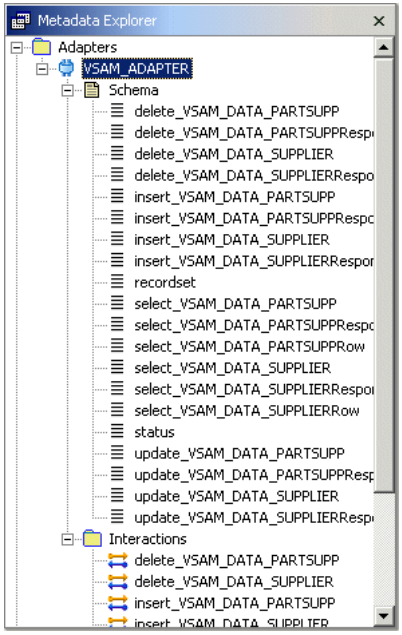
4. Click **Finish**. The selected tables are displayed.



5. 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.



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

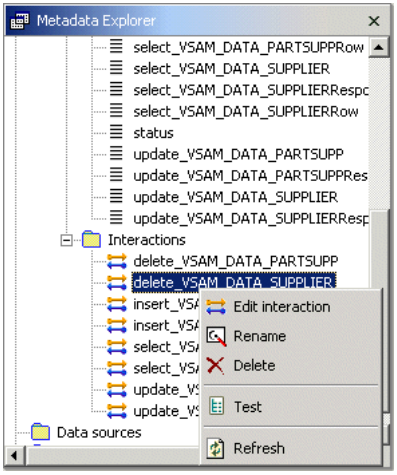


The SQL used in the interactions can be modified to the exact application requirements using the Metadata perspective, as described in "Task 3: Modifying the Generated Outbound Interactions" on page 3-30.

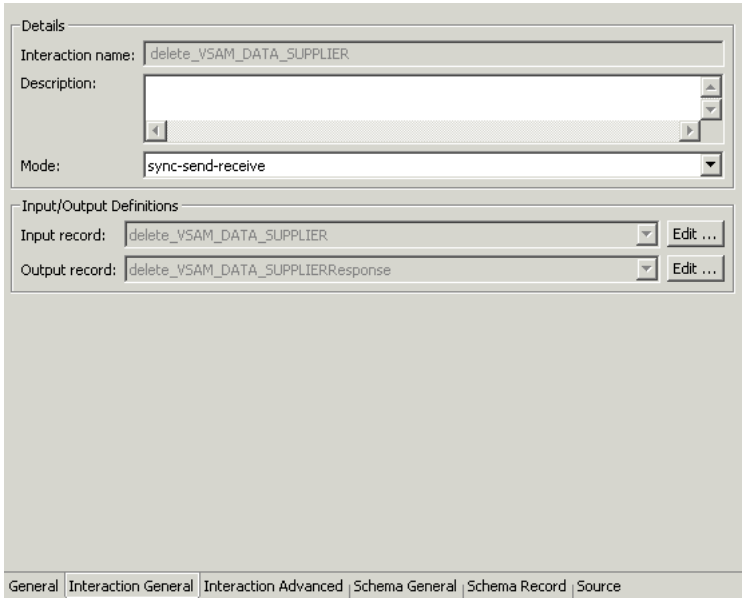
Task 3: Modifying the Generated Outbound Interactions

You can modify the interaction definitions to the exact requirements of the application in the Metadata perspective. The following example uses the DELETE interaction generated in the previous task to describe how the interactions can be modified.

1. In the Metadata perspective, right-click the interaction to modify and choose **Edit Metadata** from the popup menu.



The adapter metadata editor opens, displaying the **Interaction General** tab.

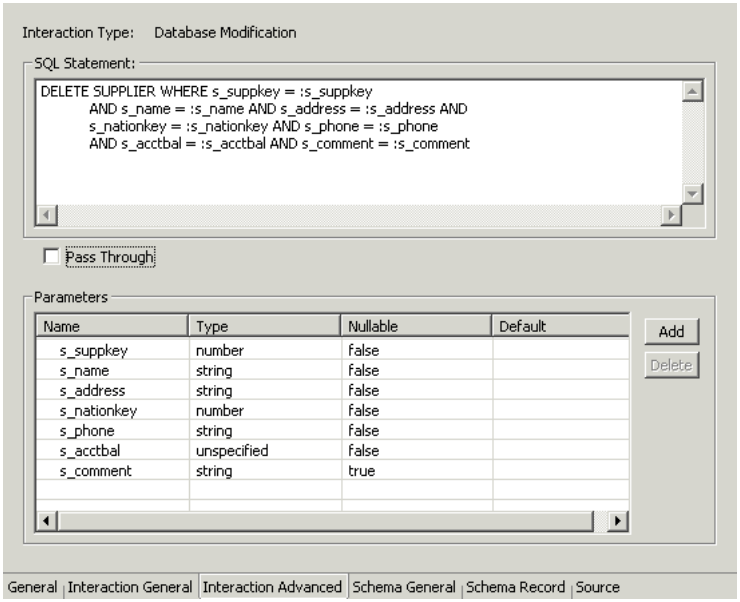


The **Interaction General** tab displays general information about the way the interaction is executed. You can add a description of the interaction and define the mode of operation for the interaction. The following modes are available:

- **sync-send-receive** - The interaction sends a request and expects to receive a response.
- **sync-send** - The interaction sends a request and does not expect to receive a response.
- **sync-receive** - The interaction expects to receive a response.

The information for a request is passed in the input record. The information for the response from the VSAM data source is passed in the output record.

2. Click the **Interaction Advanced** tab to display specific information about the interaction.

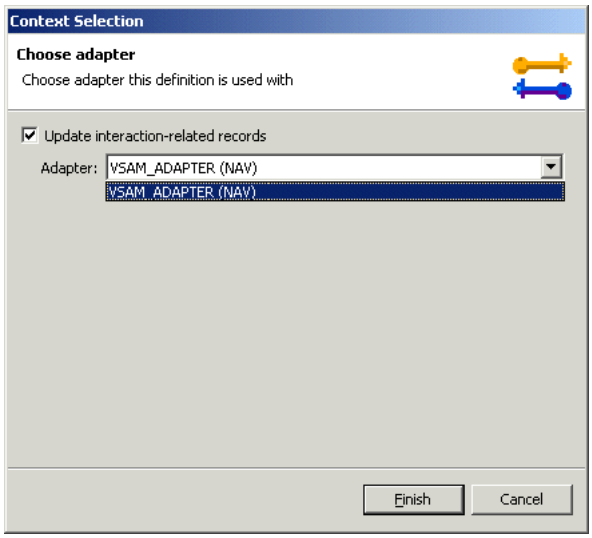


As, required, change the SQL and the parameters associated with the SQL.

Parameters are specified in a SET clause or in a WHERE clause with the following format:

`:parameter_name`

3. Dependent on the changes made to the SQL, when you close the editor, or click the **Save** button, the Context Selection window opens.



Specify the adapter from the list and check the **Update interaction-related records** checkbox. Any changes that need making to the record structures in the schema part of the metadata are done automatically.

Note: The interaction records are built based on all the fields in the table and cannot be changed manually, even if you change the SQL so that less fields are involved.

Note: Modifying interactions is the same as described above, which uses a DELETE SQL statement, for all types of SQL statements (INSERT, UPDATE, etc.).

See Also: "Interaction Metadata for the Adapter" on page A-12 for details about the adapter metadata

Generating Inbound Interactions

Inbound interactions are defined as events in Oracle Studio. When you defined the Oracle Connect for VSAM backend adapter with the **Create event queue for the adapter** box checked, as described in "Task: Setting Up the Adapter" on page 3-22,

an event adapter was defined automatically. The event adapter is defined with the same name as the backend adapter with the word *event* appended to it. The backend adapter and the event adapter are linked by Oracle Studio. You can jump from one adapter definition to the other by right-clicking the adapter or event in the **Configuration Explorer** list and choosing the Linked Event or Linked Adapter option respectively.

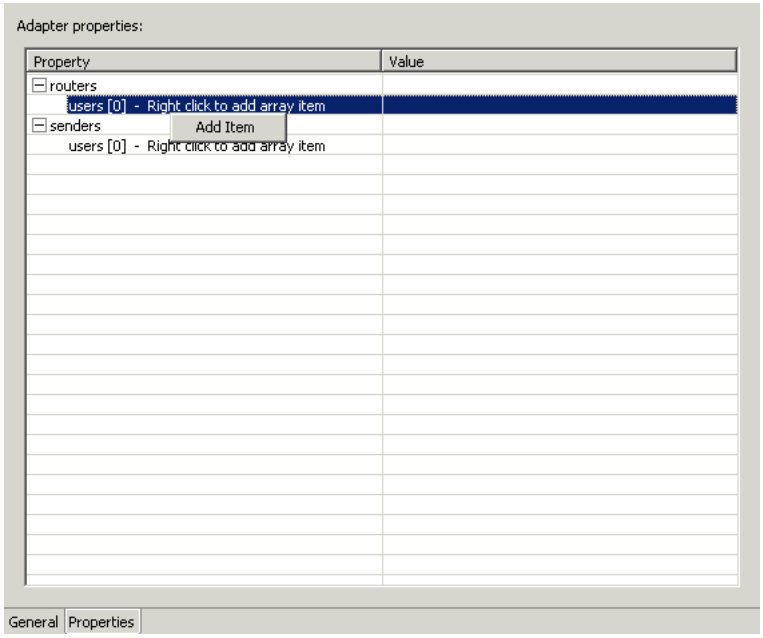
The event adapter requires metadata describing the inbound interactions, including the structures used to pass information to and from the adapter.

Use the Import Metadata perspective in Oracle Studio to generate interaction metadata for the adapter.

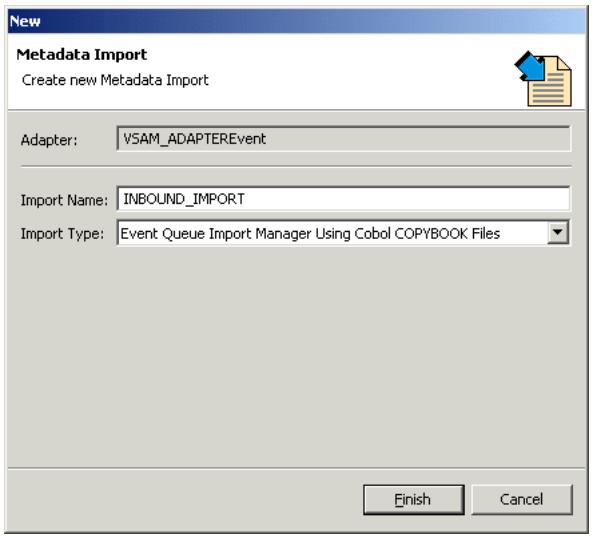
- Task 1: Defining an Import Procedure
- Task 2: Generating Inbound Interactions

Task 1: Defining an Import Procedure

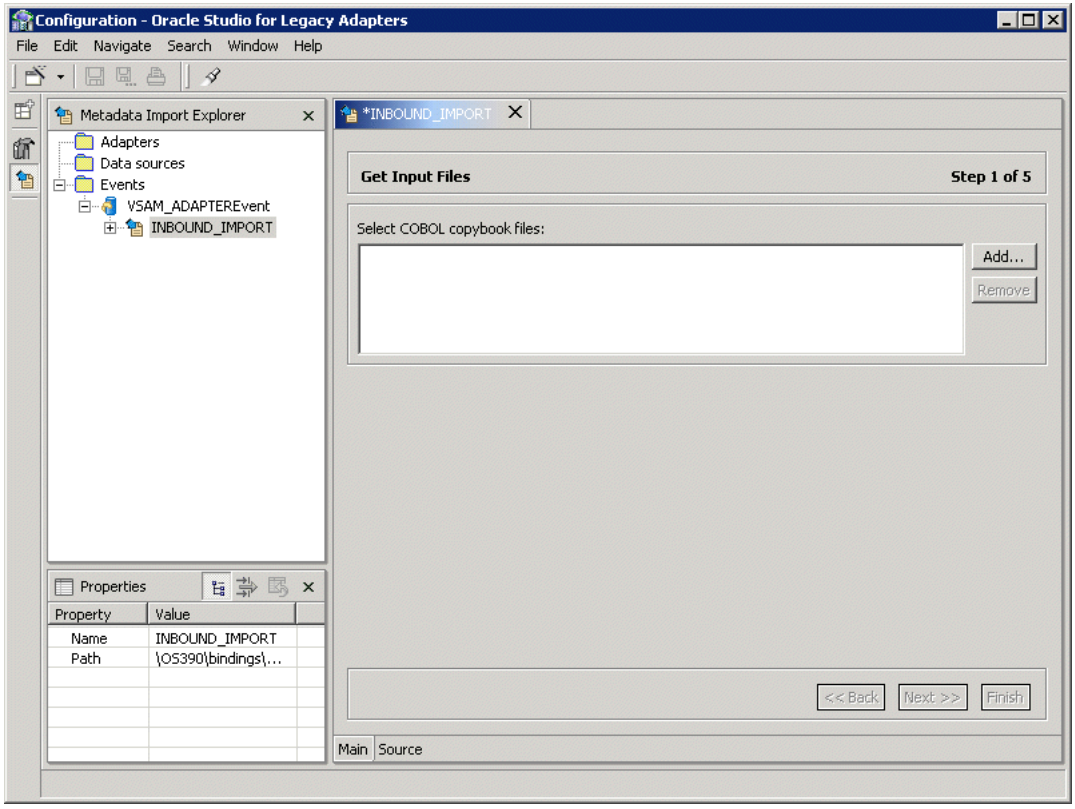
1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. In the Configuration Explorer, click the + next to the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Click the + next to the **Bindings** node. The binding configurations available on this machine are listed.
4. Click the + next to the **NAV** binding node.
5. Click the + next to the **Adapters** node.
6. Right-click the VSAM adapter defined in "Setting Up an Oracle Connect for the VSAM Backend Adapter" on page 3-22.
7. Choose **Linked Event** from the popup menu, to jump to the event adapter.
8. Right-click the event adapter and choose **Edit Event** from the popup menu.
9. Select the **Properties** tab to add the names of Oracle Application Server ProcessConnect users who can retrieve inbound interactions and OS/390 users who can send inbound interactions.
10. To add Oracle Application Server ProcessConnect users, click the + next to the **routers** node and right click the users property to add the user.



11. Enter the name of the Oracle Application Server ProcessConnect user in the Value column for the item added.
12. To add OS/390 users, click the + next to the **senders** node and right click the users property to add the user.
13. Enter the name of the OS/390 user in the Value column for the item added.
14. Click the **Save Editor Contents** button to save the changes.
15. Right-click the event adapter in the **Configuration Explorer** and choose **Import** from the popup menu, to open the Metadata Import perspective, with the event adapter displayed under the Events list.
16. Right-click the event adapter and choose **New Import** from the popup menu, to open the New Metadata Import wizard.
17. Enter a name for the import. The name can contain letters and numbers and the underscore character only.
18. Select **Event Queue Import Manager Using Cobol COPYBOOK Input Files** for the `Import Type`.



19. Click **Finish**. The metadata import wizard opens in Oracle Studio.



Task 2: Generating Inbound Interactions

After defining an import as described in the preceding task, the metadata import wizard opens in Oracle Studio. COBOL copybooks are used to create the metadata. The import wizard generates record structures, which are used for the record structures for inbound interactions.

> **Note:** The generation of inbound interactions involves similar steps to the steps described in "Task 2: Generating Metadata for the VSAM Data" on page 3-7.

1. Click **Add** to add COBOL copybooks.

   A Select window opens. This window lists any files selected.

2. Click **Add**.

   The Add Resource window opens, which provides the option to select files from the local machine or FTP the files from another machine.

3. If the files are on another machine, select the **Import resource from remote machine** option to start a file transfer process from the machine.

4. Click **Next** to open the File Transfer wizard.

5. Enter the server name where the COBOL copybooks reside and, if not using anonymous access, enter a username and password to access the machine.

6. Click **Next**. The machine is accessed and the directory structure displayed, enabling you to browse to the files to be transferred.

   You access the OS/390 machine at the high-level qualifier for the user name. To change the high-level qualifier, enter the name of the high-level qualifier in the Directory name field and click the arrow button.

7. Navigate to the location where the files reside and select the files to be transferred and the transfer method (**Text** or **Binary**)

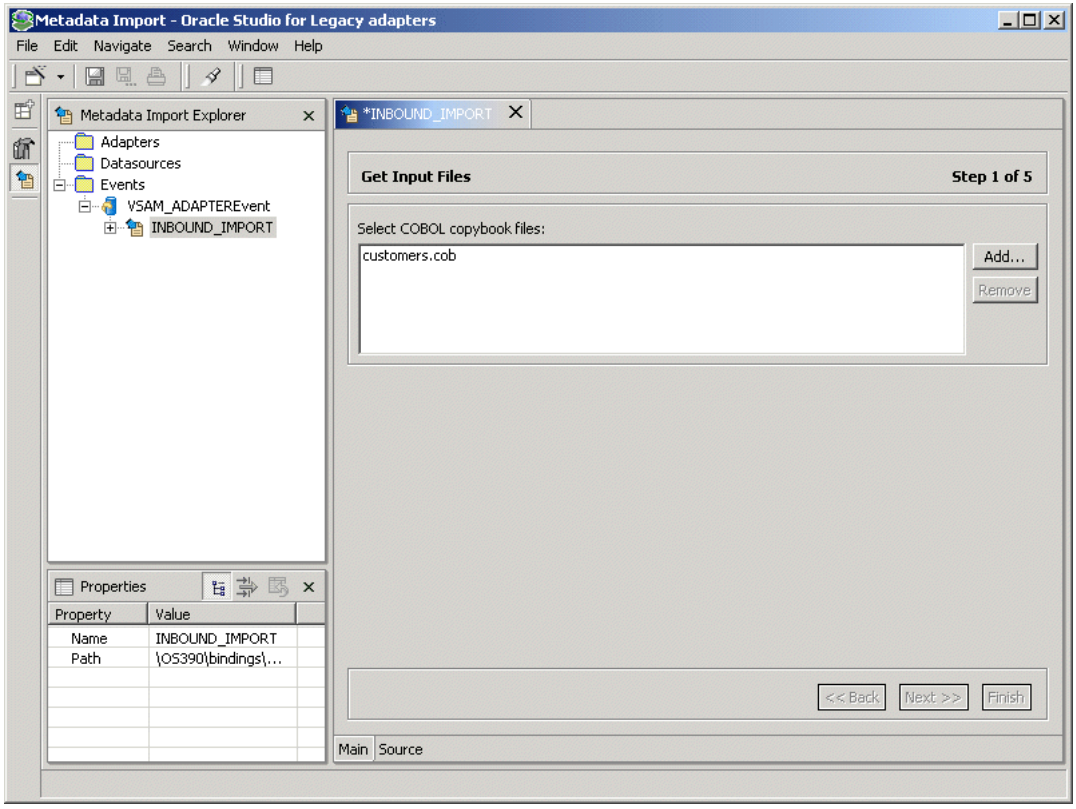
8. Click **Next**. A summary screen is displayed with details of what is going to be transferred.

9. Click **Finish** to start the transfer. The **Select** window is redisplayed with the list of transferred files.

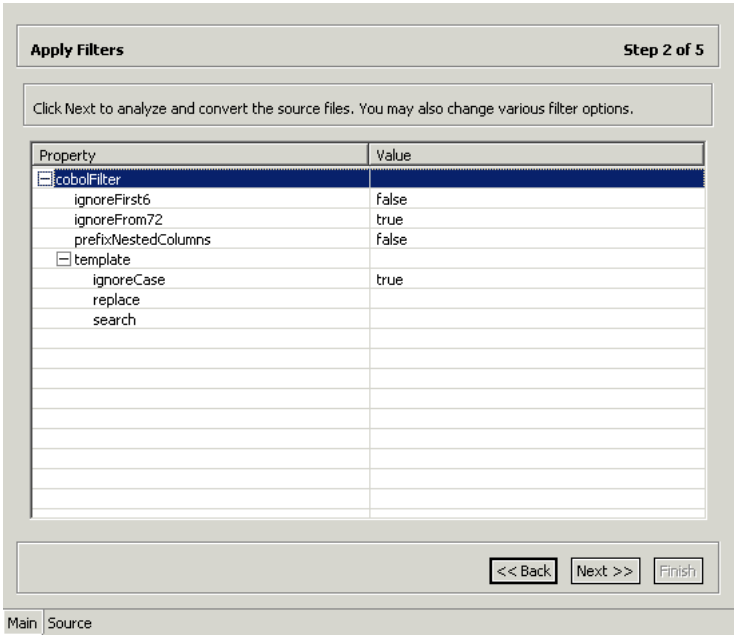
10. Select the files you want to use in the import and click **OK**.

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

The format of the COBOL copybooks must be the same. 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 case, you must do separate imports.



11. In the metadata import wizard, click **Next** after you have completed adding the list of copybooks.



12. Apply filters to the copybooks.

    The following filters are available:

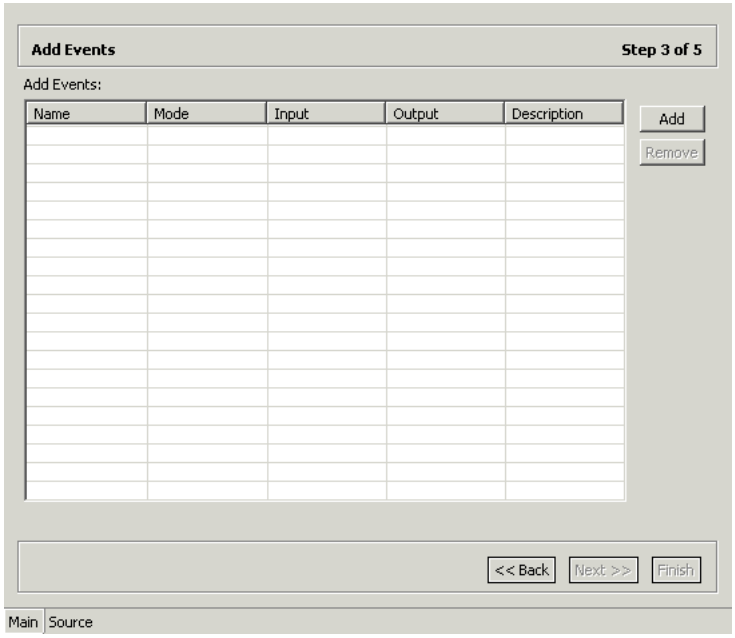
    ignoreFirst6 - Ignore the first six columns in the COBOL copybook.

    ignoreFrom72 - Ignore columns 73 to 80 in the COBOL copybook.

    prefixNestedColumns - Prefix all nested columns with the previous level heading.

    In addition, you can specify a search string and the string that will replace this search string in the generated metadata, and whether the replacement is dependent on the case of the found string.

13. Click **Next** to apply the filter settings. The Add Events window opens.



14. Click **Add** to add an interaction for the VSAM adapter.

    Interaction name - The name of the interaction.

    Mode - The interaction mode:

    async-send - The inbound interaction sends a request and does not expect to receive a response.

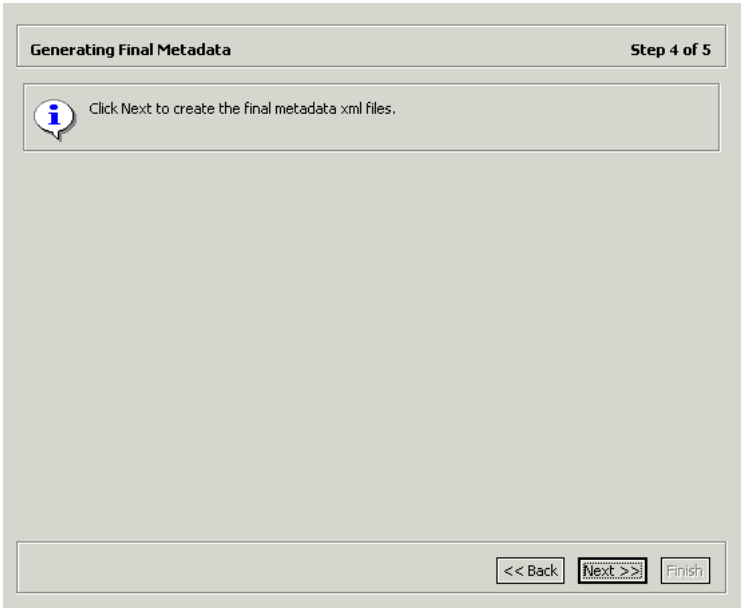
    Input - Identifies an input record. The input record is the data structure for the inbound interaction. The records generated from the COBOL programs specified at the beginning of the procedure are listed in a drop-down list. Select the relevant record for the interaction.

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

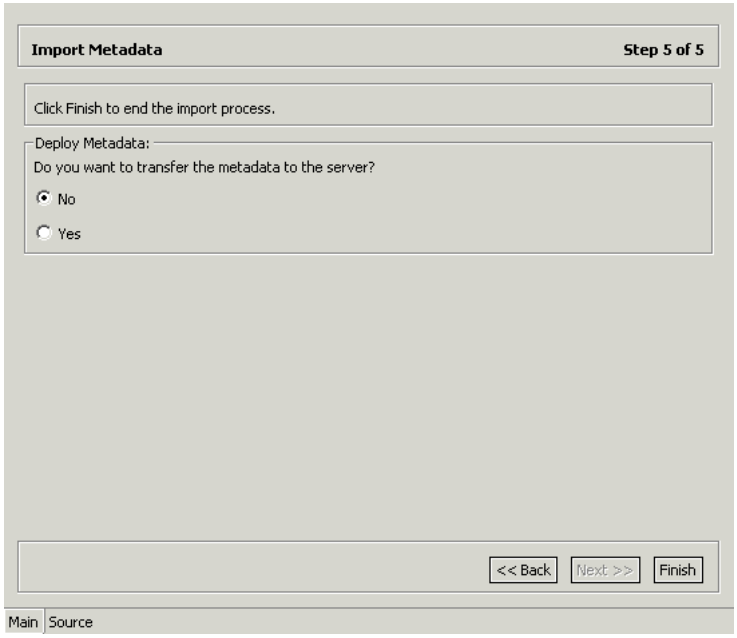
    Output - For internal use. You cannot enter a value in this field.

    Description - Free text describing the interaction.

15. Add as many interactions as necessary and then click **Next**.



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



17. Specify that you want to transfer the metadata from the Windows machine to the OS/390 machine and click **Finish**.

The metadata is imported based on the options specified and it is stored on the OS/390 machine. 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 perspective in Oracle Studio. You can also make any fine adjustments to the metadata and maintain it, as necessary.

> **See Also:** "Interaction Metadata for the Adapter" on page A-12 for details about the adapter metadata

4

Using Interactions in a Business Process

After the VSAM data source and the Oracle Connect for VSAM backend adapter have been modeled as described in Chapter 3, "Modeling Interactions for Oracle Application Server Integration Adapter for VSAM", the interactions can be used in a business process in Oracle Application Server ProcessConnect. You define a delivery channel for the VSAM adapter part of the Oracle Application Server Integration Adapter for VSAM. This delivery channel is used during the modeling in Oracle Application Server ProcessConnect with the interactions defined for the adapter in Oracle Studio in a business process.

This chapter contains the following sections:

- Adding the VSAM Adapter in Oracle Application Server ProcessConnect
- Setting Up the Delivery Channel
- Modifying an Adapter in Oracle Studio
- Comparing a ProcessConnect XSD to an Oracle Connect Record Schema

Adding the VSAM Adapter in Oracle Application Server ProcessConnect

When adding the adapter type during the application definition, specify `VSAM Adapter`.

> **See Also:** *Oracle Application Server ProcessConnect User's Guide* for details of setting up applications in Oracle Application Server ProcessConnect

Setting Up the Delivery Channel

Part of the application definition includes adding a delivery channel for the Oracle Application Server Integration Adapter for VSAM.

The delivery channel identifies where the OS/390 database adapter is located and the necessary security and encryption requirements to access the adapter.

Task: Setting Up the Delivery Channel

1. In the Create Delivery Channel page for the adapter, add the following information to create a delivery channel.

Field	Description
Name	A name for this delivery channel You must provide a value for this field
The Adapter Name	The name of the Oracle Connect for VSAM backend adapter to use, as defined in Oracle Studio. You can check the name specified for the adapter in Oracle Studio You must provide a value for this field
The Workspace to use	The name of a server workspace to use. Leave blank to use the default workspace. If you used another workspace when modeling the adapter in Oracle Studio, you can check the workspace name in the Configuration perspective in Oracle Studio
The Remote machine Name	The server machine TCP/IP host name. The host name is the name of the machine where the Oracle Connect for VSAM backend adapter was defined using Oracle Studio You must provide a value for this field

Field	Description
The Port to connect	The TCP/IP port where the daemon is running on the server. The default port is 2552. If you specified another port when modeling the adapter in Oracle Studio, you can check the port specified by right-clicking on the machine in Oracle Studio and checking the port number displayed
Protocol Name	The firewall protocol used: either none or fixedNat (the Nat protocol using a fixed address for the daemon). The default is none
Timeout via Connect	Connection timeout in seconds. The default is 0, meaning that there is no connection timeout
Encryption protocol name	Name of encryption protocol to use. The default is null. The RC4 protocol is supported
Encryption key name	Name of the symmetric encryption key to use
Encryption key value	Value of the symmetric encryption key to use
The User Name	The back-end user name for authentication. Leave blank if anonymous access is allowed.
The User Password	The back-end user password for authentication. Leave blank if anonymous access is allowed
Directory to store rejected inbound interactions	The directory where files with information about rejected inbound interactions are stored. Each rejected inbound interaction is stored in a file whose name consists of a timestamp and the interaction name.

2. Click **Apply**.

   The delivery channel is defined for the application and the Delivery Channels Details page for the added delivery channel is displayed.

You can now select any of the interactions defined for the adapter.

> **Note:** You cannot add, modify or delete interactions for the adapter from Oracle Application Server ProcessConnect. Any changes to the interactions must be done in Oracle Studio.

Modifying an Adapter in Oracle Studio

The procedure for making changes in Oracle Studio to an interaction or to the interaction input and output records and having the change reflected in Oracle Application Server ProcessConnect depends on whether the configuration was deployed or not.

Before the configuration has been deployed, delete the interaction from the application (and all relevant items that reference the interaction) and after making the change in Oracle Studio, add the changed interaction to the application.

After the configuration has been deployed, you cannot change the interaction until you have performed the following procedure:

1. Stop both the adapter framework and integration manager processes.
2. Delete the interaction from the application (and all relevant items that reference the interaction).
3. Make the changes to the interaction in Oracle Studio.
4. In the same application, add the changed interaction.
5. Redeploy the configuration.
6. Start both the adapter framework and integration manager processes.

Comparing a ProcessConnect XSD to an Oracle Connect Record Schema

The interaction that is modeled in Oracle Studio is automatically translated to an xsd presentation in Oracle Application Server ProcessConnect. The xsd can be viewed by displaying the record in the Native Dataypes window (under Modeling > Datatypes).

The following example shows the Oracle Application Server ProcessConnect xsd for an input record.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="noNamespace://OS390/Navigator/CustAccount#findCustAccountInput"
targetNamespace="noNamespace://OS390/Navigator/CustAccount#findCustAccountInput"
elementFormDefault="qualified">
  <xsd:element name="findCustAccountInput" type="findCustAccountInput"/>
  <xsd:complexType name="findCustAccountInput">
    <xsd:attribute default="ATCLIENT" name="EYE-CATCH" type="xsd:string"/>
    <xsd:attribute name="ACCT-NUM" type="xsd:string"/>
```

```
  </xsd:complexType>
</xsd:schema>
```

The Oracle Connect metadata for the input record, which was the source for the above xsd, is part of the metadata definition:

```
<record name="findCustAccountInput">
  <field name="EYE-CATCH" type="string" default="ATCLIENT"
         size="8" nativeType="string"/>
  <field name="ACCT-NUM" type="string" size="9" nativeType="numstr_u"/>
</record>
```

The Oracle Connect metadata definition can be viewed in Oracle Studio as follows:

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Click the + next to the **Bindings** node.
4. Click the + next to the **NAV** binding node.
5. Click the + next to the **Adapters** node to display the adapters.
6. Right-click the CICS adapter in the **Configuration Explorer** and choose **Edit Metadata** to open the Metadata perspective.
7. Right-click the adapter in the Metadata Explorer and choose **Edit definition**.
8. Choose the Source tab to review the metadata definition.

The metadata definition includes the following parts:

- General information about the adapter
- Information about each interaction, including the names of the input and output record
- The record schemas used to support the interactions, including input and output (response) record schemas

The following example shows the Oracle Application Server ProcessConnect xsd for an output record.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="noNamespace://OS390/Navigator/CustAccount#findCustAccountResponse"
targetNamespace="noNamespace://OS390/Navigator/CustAccount#findCustAccountRespon
se" elementFormDefault="qualified">
```

```
  <xsd:element name="findCustAccountResponse" type="findCustAccountResponse"/>
  <xsd:complexType name="findCustAccountResponse">
    <xsd:attribute name="RET-CODE" type="xsd:string"/>
    <xsd:attribute name="ACCT-ID" type="xsd:string"/>
    <xsd:attribute name="ACCT-FIRST-NAME" type="xsd:string"/>
    <xsd:attribute name="ACCT-LAST-NAME" type="xsd:string"/>
    <xsd:attribute name="ACCT-BIRTH-DATE" type="xsd:string"/>
    <xsd:attribute name="ACCT-MARRIED" type="xsd:int"/>
    <xsd:attribute name="ACCT-CHILDREN" type="xsd:int"/>
    <xsd:attribute name="ACCT-TOTAL-ASSETS" type="xsd:int"/>
    <xsd:attribute name="ACCT-EMPLOYED" type="xsd:int"/>
    <xsd:attribute name="ACCT-OVERTIME-HOURS" type="xsd:int"/>
    <xsd:attribute name="ACCT-COMMISSION" type="xsd:float"/>
    <xsd:attribute name="ACCT-SALARY" type="xsd:double"/>
    <xsd:attribute name="ACCT-BALANCE" type="xsd:decimal"/>
    <xsd:attribute name="ACCT-VACATION-DAYS" type="xsd:string"/>
  </xsd:complexType>
</xsd:schema>
```

The Oracle Connect metadata for the output record has the following structure:

```
<record name="findCustAccountResponse">
  <field name="RET-CODE" type="string" nativeType="int4"/>
  <field name="ACCT-ID" type="string" size="9" nativeType="numstr_u"/>
  <field name="ACCT-FIRST-NAME" type="string" size="20" nativeType="string"/>
  <field name="ACCT-LAST-NAME" type="string" size="20" nativeType="string"/>
  <field name="ACCT-BIRTH-DATE" type="string" size="8" nativeType="numstr_u"/>
  <field name="ACCT-MARRIED" type="int" nativeType="int2"/>
  <field name="ACCT-CHILDREN" type="int" nativeType="int4"/>
  <field name="ACCT-TOTAL-ASSETS" type="int" nativeType="int8"/>
  <field name="ACCT-EMPLOYED" type="int" nativeType="int2"/>
  <field name="ACCT-OVERTIME-HOURS" type="int" nativeType="int4"/>
  <field name="ACCT-COMMISSION" type="float" nativeType="single"/>
  <field name="ACCT-SALARY" type="double" nativeType="dfloat"/>
  <field name="ACCT-BALANCE" type="numeric" size="9"
         scale="-2" nativeType="decimal"/>
  <field name="ACCT-VACATION-DAYS" type="string" size="3"
         nativeType="numstr_s"/>
</record>
```

5

Oracle Application Server Integration Adapter for VSAM Features and Restrictions

After the Oracle Application Server Integration Adapter for VSAM is installed and the modeling described in Chapter 3, "Modeling Interactions for Oracle Application Server Integration Adapter for VSAM" and Chapter 4, "Using Interactions in a Business Process" completed, you can use the Oracle Application Server Integration Adapter for VSAM to invoke interactions that access VSAM data.

This chapter contains the following sections:

- Accessing VSAM Using SQL
- Sending Inbound Interactions Using a CICS Transaction

> **Note:** When accessing VSAM directly, you model inbound interactions using API functions in COBOL, as described in Appendix E, "COBOL APIs to Applications".

- High Availability with Oracle Connect
- Known Restrictions

Accessing VSAM Using SQL

VSAM does not include a native SQL-based interface. The Oracle Application Server Integration Adapter for VSAM provides support for standard ANSI '92 SQL to select, insert, update and delete VSAM records within the adapter interactions. Additionally, you can incorporate enhancements into the SQL specified in an interaction to handle hierarchical data in VSAM.

A hierarchical query is a query whose result is a hierarchy of rowsets reflecting parent-child relationships.

Rowsets with arrays of structures as columns are modeled such that the rows of an array constitute the children of a column in the containing parent row.

Hierarchical queries enable you to do the following:

- Arrange rowsets resulting from a query in a hierarchy, reflecting a parent-child relationship
- Manipulate data that is stored hierarchically

You can handle this type of data in the following ways:

- By including the hierarchical data as a parent-child relationship
- By flattening the hierarchical data
- By using virtual tables to represent the data

See Also: Appendix C, "Supported SQL Syntax and SQL Enhancements" on page C-1 for details

Generating Hierarchical Results

A hierarchical query nests a `SELECT` statement as one of the columns of the rowset retrieved by a nested `SELECT` statement. You use braces ({}) to delimit the nesting. This type of query generates a child rowset, which enables you to incorporate drill-down operations in the application.

Example

The following hierarchical query, when specified in an interaction, produces a child rowset:

```
SELECT C_name,
  {SELECT O_orderkey,
    {SELECT L_partkey, L_linenumber
      FROM lineitem
```

```
        WHERE L_orderkey = O_orderkey} AS items
    FROM torder WHERE O_custkey=C_custkey} AS orders
FROM customer
```

The result has a three-tier hierarchical structure. The main ("root") rowset has two columns. The second column includes another ("child") rowset. This child rowset includes `items`, a column that includes another child rowset.

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.

Example

The following query, when specified in an interaction, accesses a child rowset:

```
select name from employees->hchild
```

This example assumes hierarchical data with a parent rowset called `employees`. This rowset has ordinary columns plus a column called `hchild` (one row for each of the children of this employee). The query lists all of the children of all of the employees.

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. This is equivalent to specifying a left outer join between the parent rowset and a child rowset, and the resulting rowset reproduces each row of the parent, combining it with one row for each of its children.

The nested `SELECT` statement can reference a child rowset (using the parent->child syntax) only in its `FROM` clause.

The examples that follow, when specified in interactions, assume hierarchical data with a parent rowset called `employees`. This rowset has ordinary columns plus a column called `hchild` (one row for each of the children of this employee).

Compare the following queries:

```
SELECT emp_id,(select name
  from employees->hchild)from employees
```

And

```
SELECT emp_id,(select name
  from e->hchild)from employees e
```

The first query, without an alias, lists for each employee all of the children of all of the employees. The second query uses an alias and produces a list containing the children stored in an array called hchild belonging to each employee.

The following query retrieves the number of children that study in each city for every city where the company has employees.

```
SELECT city, (SELECT COUNT(*)
                FROM employees->hchild->hschool A
                  WHERE A.city = B.city)
  FROM cities B
```

Sending Inbound Interactions Using a CICS Transaction

The Oracle Application Server Integration Adapter for VSAM includes a set of APIs that enable sending a native event as an inbound interaction from a COBOL program, using a CICS transaction.

> **See Also:** Appendix E, "COBOL APIs to Applications" for details of APIs that can be used when not working under CICS

To send an inbound interaction, perform the following tasks:

- Task 1: Setting Up the IBM OS/390 Machine
- Task 2: Setting Up a Call to the CICS Transaction

Task 1: Setting Up the IBM OS/390 Machine

Before you can use the CICS transaction, set it up using the following procedure:

1. Copy *INSTROOT*.LOAD(ATTCICSD) to a CICS DFHRPL library.
2. Copy *INSTROOT*.LOAD(TRANS3GL) to a CICS DFHRPL library.

Note: Make sure that the CICS Socket Interface is enabled. You can enable this interface by issuing the following CICS command:

```
EZAO START CICS
```

Refer to the *TCP/IP V3R2 For MVS: CICS Sockets Interface Guide*. If you are not sure if the system is configured with the Socket Interface, try running the EZAC transaction. If the transaction produces a screen, you should be able to run the EZAO startup transaction. If not, see if the transaction has been defined in a group that has not been installed, for example: `CEDC V TRANS(EZAC) G(*).`

If it is defined in a group, install that group and try running EZAO again. If not, you have to configure CICS as outlined in the *TCP/IP V3R2 For MVS: CICS Sockets Interface Guide*.

3. Set up CICS resource definitions for the COBOL program from which you want to send a native event as an inbound interaction.

   The following JCL can be used as a template:

```
//ATTCSD  JOB  'ORACLE','CSD',MSGLEVEL=1,NOTIFY=&SYSUID
//STEP1   EXEC PGM=DFHCSDUP,REGION=512K,
//          PARM='CSD(READWRITE),PAGESIZE(60),NOCOMPAT'
//STEPLIB  DD  DSN=<HLQ1>.SDFHLOAD,DISP=SHR
//DFHCSD   DD  UNIT=SYSDA,DISP=SHR,DSN=<HLQ2>.CSD
//OUTDD    DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
*/*****************************************************************/
*/*  ORACLE CICS Definitions                                      */
*/*                                                               */
*/*****************************************************************/
*---------------------------------------------------------------------*
* Note: Install GROUP(ORA) - CEDA IN G(ORA)                    *
*     If you are rerunning this, uncomment the DELETE command. *
*---------------------------------------------------------------------*
*
* Start ORACLE RESOURCES:
*
* DELETE                   GROUP(ORA)
  DEFINE PROGRAM(ATTCICSD) GROUP(ORA) LANGUAGE(C) DATALOCATION(ANY)
         DE(Oracle DLL)
```

```
  DEFINE PROGRAM(TRANS3GL) GROUP(ORA) LANGUAGE(C) DATALOCATION(ANY)
         DE(Oracle DLL)
  DEFINE PROGRAM(<PROG>) GROUP(ORA) LANGUAGE(<LANG>) DATALOCATION(ANY)
         DE(Oracle)
  DEFINE TRANSACTION(<ORATRAN>) GROUP(ORA) PROGRAM(<PROG>)
         TASKDATAL(ANY)
         DE(Oracle TRAN ID)
  LIST                    GROUP(ORA)
*
* End   ORACLE RESOURCES
*
/*
//
```

Make the following changes before running the JCL:

1. Modify the JCL, as follows:
   - Change the JOB card to suit the site.
   - Change <HLQ1> to point to the CICS SDFHLOAD library.
   - Change <HLQ2> to point to the CICS CSD dataset.
   - Change <LANG> to the LE370 for COBOL.
   - Change <PROG> to the COBOL program name.
   - If you are calling the COBOL program from a CICS transaction, change <ORATRAN> to the CICS transaction name that calls the COBOL program.
2. From CICS, install the ORA group, by issuing the following command:

   ```
   CEDA IN G(ORA)
   ```

Task 2: Setting Up a Call to the CICS Transaction

The COBOL program sets up a buffer in the CICS COMMAREA that contains the information needed for the inbound interaction and then calls the `TRANS3GL` transaction to send the interaction.

The buffer is formatted as follows:

Parameter	Size	Description
Version	4	The version of the APIs used. The expected value is 1

Parameter	Size	Description
ServersUrl	256	The URL of the OS/390 machine and the port number where Oracle Connect for VSAM runs. For example, IP1:2552, where IP1 is the URL and 2552 is the port
Username	64	A valid username to access the OS/390 machine
Password	64	A valid password for the user name
Workspace	64	A daemon workspace. The default is Navigator
AdapterName	64	The name of the adapter defined in Oracle Studio
SchemaFileName	256	For future use. Leave blank
EncKeyName	64	For future use. Leave blank
EncKeyValue	256	For future use. Leave blank
InteractionName	64	The name of the interaction as defined in Oracle Studio
Flags	4	The following flags are available: ■ 1 - A trace of the XML is performed ■ 2 - A trace of the communication calls is performed ■ 3 - Both the XML and communication calls are traced ■ 4 - The NAT firewall protocol is used, with a fixed address for the daemon ■ 5 - A trace of the XML is performed and the NAT firewall protocol is used, with a fixed address for the daemon ■ 6 - A trace of the communication calls is performed and the NAT firewall protocol is used, with a fixed address for the daemon ■ 7 - Both the XML and communication calls are traced and the NAT firewall protocol is used, with a fixed address for the daemon
Input format	4	The following formats are available: ■ 0 - Input is provided as XML ■ 1 - Input is provided using parameters

Parameter	Size	Description
Input	-	The size of the input depends on the value specified in the `Input size` parameter If the Input format is set to `0` (XML), the input is formatted as follows: ■ The first four bytes specify the size of the input XML string ■ The next 64 bytes specifies the name of the record used for the output (the inbound interaction) ■ The next bytes (to the exact length specified in the first four bytes) specify the input XML string. For example: `<findorder ORDER_NO='17' />` where `findorder` is the inbound interaction name If the Input format is set to `1` (the input is done using parameters), the input is formatted as follows: ■ The first four bytes specify the number of parameters ■ The next 4 bytes specify the maximum size of any parameter value ■ The next 64 bytes specify the name of the record used for the output (the inbound interaction) ■ The next 32 bytes specify the name of the parameter ■ The next bytes (to the exact length specified in the first four bytes) specify the input parameter ■ The following bytes repeat the last two entries until all the parameters are specified

The buffer is defined as follows:

```
*
*  COBOL COPY OF DATA BUFFER
*
01  COMM-DATA-BUFF  PIC X(5000).

01  COMM-DATA-BUFF-ERROR REDEFINES COMM-DATA-BUFF.
    05  COMM-ERROR-STATUS  PIC S9(8) COMP SYNC.
    05  COMM-ERROR-MSG     PIC X(256).

01  COMM-DATA-BUFF-INPUT REDEFINES COMM-DATA-BUFF.

    05  INPUT-COMMAREA-3GL.
        10   INCOM-VERSION           PIC S9(8) COMP SYNC.
        10   INCOM-SERVERS-URLS      PIC X(256).
```

```
*                     /* IP1:PORT[,IP2:PORT] [,...] */
        10   INCOM-USER-NAME         PIC X(64).
        10   INCOM-PASSWORD          PIC X(64).
        10   INCOM-WORKSPACE         PIC X(64).
        10   INCOM-ADAPTER-NAME      PIC X(64).
        10   INCOM-SCHEMA-FILE-NAME  PIC X(256).
        10   INCOM-ENC-KEY-NAME      PIC X(64).
        10   INCOM-ENC-KEY-VALUE     PIC X(256).
        10   INCOM-INTERACTION-NAME  PIC X(64).
        10   INCOM-DW-FLAGS          PIC S9(8) COMP SYNC.
        10   INCOM-INP-FORMAT        PIC S9(8) COMP SYNC.
        10   INCOM-EXEC-INPUT.
             15  INCOM-XML-BUFF.
                 20   INCOM-XML-ILEN  PIC S9(8) COMP SYNC.
                 20   INCOM-XML-INTER-OUTREC-NAME
                                          PIC X(64).
*     ====>>> CHANGE ??? TO LEN SPECIFIED IN INCOM-XML-ILEN
                 20   INCOM-XML-INPUT PIC X(???).
             15  INCOM-PARAM-BUFF REDEFINES INCOM-XML-BUFF.
                 20  INCOM-PARAM-COUNT     PIC S9(8) COMP SYNC.
                 20  INCOM-PARAM-VALUE-LEN PIC S9(8) COMP SYNC.
                 20  INCOM-PARAM-INT-OUTREC-NAME     PIC X(64).
*     ====>>> CHANGE ??  TO COUNT SPECIFIED IN INCOM-PARAM-COUNT
                 20  INCOM-PARAM-NAME-VALUE OCCURS ?? TIMES.
                     25  INCOM-PARAM-NAME    PIC X(32).
*     ====>>> CHANGE ?? TO LEN SPECIFIED IN INCOM-PARAM-VALUE-LEN
                     25  INCOM-PARAM-VALUE   PIC X(??).

01  COMM-DATA-BUFF-OUTPUT REDEFINES COMM-DATA-BUFF.

    05  COMM-OUT-STATUS   PIC S9(8) COMP SYNC.
    05  COMM-OUT-LENGTH   PIC S9(8) COMP SYNC.

    05  COMM-OUT-DATA     PIC X(4992)
```

The TRAN3GL transaction is called as follows:

```
EXEC CICS LINK PROGRAM("TRANS3GL")
            COMMAREA(commDataBuff)
            LENGTH(iCommSize);
```

where:

commDataBuff - The buffer with the interaction details, used in the COMMAREA.

iCommSize - The size of the buffer. This value is also used to determine the size of the output string. Thus make sure the value is big enough for the expected output.

After defining the COMMAREA and calling the TRAN3GL transaction in the COBOL program, compile and move the COBOL program to a CICS DFHRPL (LOAD) library.

The output includes a 4 byte success flag: Zero for success, otherwise failure. The output overrides the input. If the result is failure, an error message with a length of 256 bytes is returned.

If XML was specified for the input and the result is success, the output is formatted as XML, as follows:

- The first four bytes specify the size of the output
- The following bytes make up the XML output

If parameters were specified for the input and the result is success, the output is formatted as follows:

- The first four bytes specify the size of the output
- The next 32 bytes specify the name of the output attribute
- The next bytes (to the exact length specified for the input string in the) specify the output value
- The following bytes repeat the last two entries until all the output is specified

High Availability with Oracle Connect

You can configure a pool of server processes so that a server process is always available for a client request.

> **See Also:** "Configuring the Daemon for High Availability" on page 7-2 for details about high availability

Known Restrictions

If you encounter incompatibility problems not listed in this section, please contact Oracle Support Services. The following section describes the known restrictions and includes suggestions for dealing with them when possible:

- VSAM Data Source Restrictions
- Array Updates

> **Note:** If you have any questions or concerns about the restrictions, contact Oracle Support Services.

VSAM Data Source Restrictions

When accessing VSAM data, the following restrictions apply:

- Transactions are not supported when accessing VSAM directly. When accessing VSAM under CICS, one-phase commit transactions are supported.
- SQL `DELETE` operations are not supported for ESDS files.

VSAM Under CICS

When accessing VSAM under CICS, the following additional restrictions apply:

- Using an alternate index to access an ESDS file is not supported.
- A non-unique alternate index for a KSDS file is not supported.

VSAM Not Under CICS

When accessing VSAM directly (and not under CICS), the following restrictions apply:

- Locking is not supported

- An RRDS file cannot have an alternate index.
- The primary key of a KSDS file must be one segment only (however, it can be several consecutive fields). You cannot modify the primary key value of a KSDS file.

Array Updates

Arrays (generated from a COBOL Copybook `OCCUR`) cannot be updated.

6

Troubleshooting the Oracle Application Server Integration Adapter for VSAM

Troubleshooting Oracle Application Server Integration Adapter for VSAM can involve checking both Oracle Application Server ProcessConnect and Oracle Connect for VSAM.

This chapter contains the following sections:

- Design-Time Troubleshooting
- Troubleshooting the Daemon

Design-Time Troubleshooting

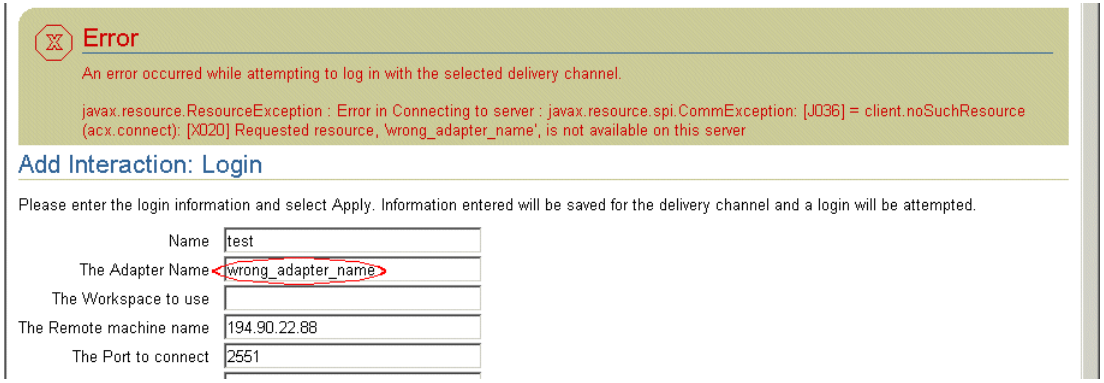
The most common cause of errors when defining the Oracle Application Server Integration Adapter for VSAM is incorrect definitions in the delivery channel. If you encounter problems during the modeling phase, check the delivery channel definitions.

> **See Also:** Task: Setting Up the Delivery Channel on page 4-2 for details of setting up the delivery channel

The following errors occur during the modeling of interactions, when using a delivery channel that was defined incorrectly.

Specifying the Wrong Adapter Name

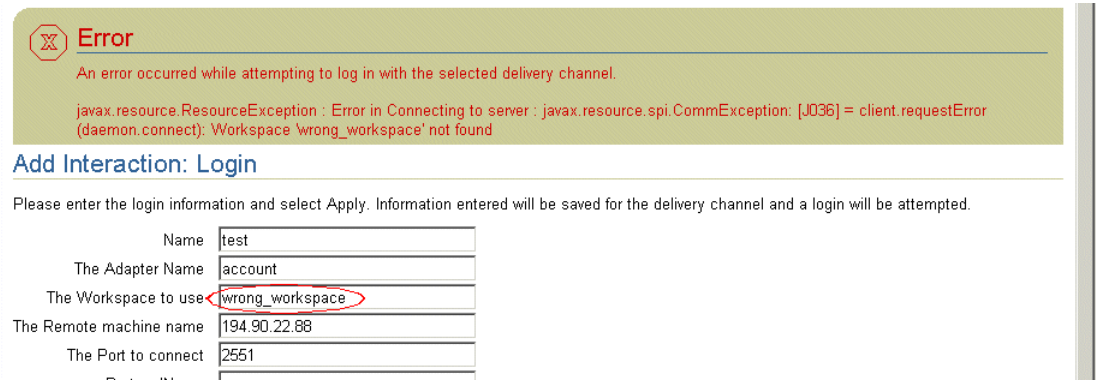
The following error occurs when the wrong adapter name is specified in the delivery channel:



Check that the name specified for the adapter is the same as the name specified in the delivery channel.

Specifying the Wrong Workspace Name

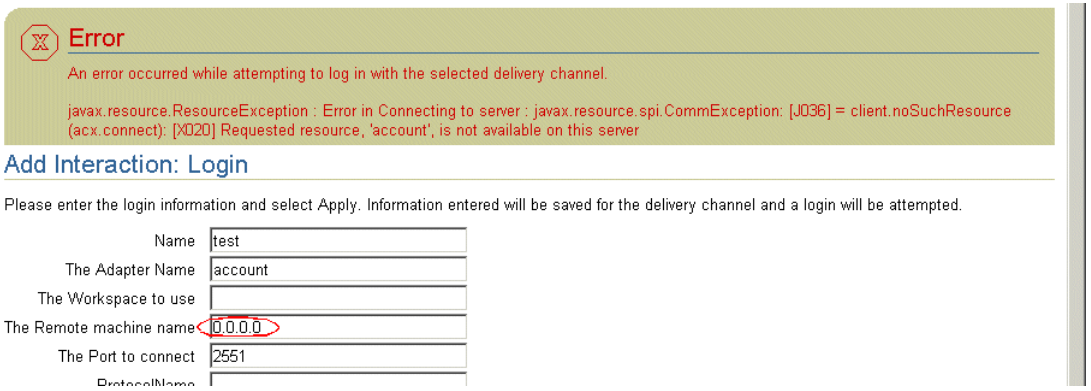
The following error occurs when the wrong workspace name is specified in the delivery channel:



Check that the name specified for the adapter is the same as the name specified in the delivery channel.

Specifying the Wrong Machine

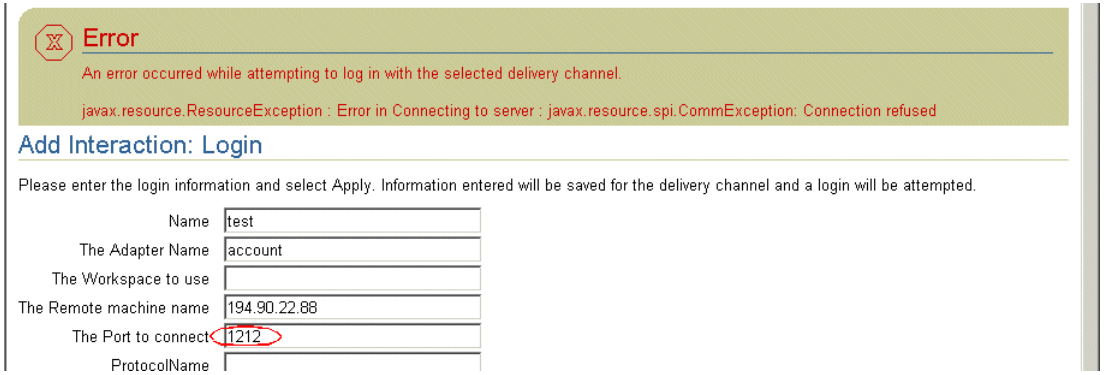
The following error occurs when the remote machine is specified incorrectly:



Check the definition of the remote machine in the delivery channel definition.

Specifying the Wrong Port Number

The following error occurs when the port number to connect to is specified incorrectly:



Check the port number defined in the delivery channel. The default port number is 2552.

When the Connection with the OS/390 Machine Is Down

If the delivery channel definitions are correct, the problem is probably with the connection with the OS/390 machine itself. In this case, check that you have a connection to the OS/390 machine in Oracle Studio.

For example you receive a connection timeout error when the daemon is not running on the mainframe:

Troubleshooting the Daemon

A daemon runs on the OS/390 machine. The daemon is responsible for allocating a server process for a client. The daemon resides in a single address space and is executed as a started task.

The daemon authenticates clients, authorizes requests for a server process within a certain server workspace and provides the clients with the required servers. When a client requests a connection, the daemon allocates a server process to handle this connection, and refers the client to the allocated process. This may be a new process (dedicated to this client) or an already-started process. Further communication between the client session and the server process is direct and does not involve the

daemon. The daemon is notified when the connection ends and the process is either killed or remains for use by another client

The daemon supports multiple server configurations called workspaces. Each workspace defines accessible data sources, applications, environment settings, security requirements and server allocation rules. The allocation of servers by the daemon is based on the workspace that the client uses to access the data source. Thus, a client can access a data source using one workspace, where a server process is allocated from an existing pool of servers, or the client can access a data source using a different workspace, where a new server process is allocated for each client request.

The daemon loads configuration settings such as various operational parameters as well as the list of workspaces accessible through the daemon.

> **See Also:** Appendix D, "Daemon Configuration Settings" for details about the configuration settings

Troubleshooting the daemon and the communication between Oracle Application Server ProcessConnect and Oracle Connect for VSAM is performed using Oracle Studio to monitor the daemon and server activity and control what happens to the daemon and server processes.

This section contains the following sections:

- Starting the Daemon
- Shutting Down the Daemon
- Monitoring the Daemon During Runtime
- Daemon Logs
- Resolving Communication Errors

Starting the Daemon

If for any reason you shut down the daemon, as described in "Shutting Down the Daemon" on page 6-6, you can restart the daemon as described in the following task.

Note: The daemon is started on the OS/390 machine only. 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:

```
'/s ATTDAEMN'
```

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

> **See Also:** "Starting the Daemon" on page 2-14 for details about the ATTDAEMN JCL

Shutting Down the Daemon

If you need to shut down the daemon for any reason (for example, you want to re-IPL the system), use Oracle Studio.

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. Right-click the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19 in the **Configuration Explorer** and choose **Open Runtime Perspective** from the popup menu.
3. In the Runtime Explorer, right-click the machine and choose **Shutdown Daemon** from the popup menu.

Monitoring the Daemon During Runtime

Use Oracle Studio to monitor the daemon during runtime.

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. Right-click the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19 in the **Configuration Explorer** and choose **Open Runtime Perspective** from the popup menu.

You can manage the daemon by right-clicking the relevant node: daemon, workspace or server process.

Daemon (Machine) Options

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

The following options are available at the machine level:

Status - Checks the status of the daemon. The information about the daemon includes the name of the daemon configuration used, the active client sessions and logging information.

Reload Configuration - Reloads the configuration after any changes. Any servers currently started are not effected by the changed configuration.

> **See Also:** Appendix D, "Daemon Configuration Settings" for details about the configuration settings

View Log - View the daemon log. For details see Daemon Logs on page 6-8.

View Events - For future use.

Daemon Properties - Displays information about the machine where the daemon is running, such as the physical address and any username and password needed to access the machine.

Shutdown Daemon - Shuts down the daemon on the machine.

Rename - Enables changing the name of the daemon displayed in the tree.

Remove - Removes the machine from the perspective tree.

Edit Daemon Configuration - Open the daemon editor to enable you to reconfigure the daemon.

> **See Also:** Appendix D, "Daemon Configuration Settings" for details about the configuration settings

Refresh - Refreshes the display.

Workspace Options

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

The following options are available at the workspace level:

Status - Checks the status of the workspace, whether it is available or not.

Reload Configuration - Reloads the configuration of the workspace after any changes. Any servers currently started are not effected by the changed configuration.

View Log - View the log for all servers for the workspace. For details see Daemon Logs on page 6-8.

View Events - For future use.

End Unused Servers - Ends all server processes for the workspace that are not active. All available and unconnected servers are terminated and any connected servers are marked and terminated on release.

Use this option when changes to the daemon configuration were made after servers were started up. On the next operation, servers are restarted, based on the new settings.

End All Servers - Ends all the active server processes for the specified workspace, regardless of whether the server has an active client.

Edit Workspace Configuration - Open the daemon editor to enable you to reconfigure the workspace.

> **See Also:** Appendix D, "Daemon Configuration Settings" for details about the configuration settings

Refresh - Refreshes the display.

Server Options

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

The following options are available at the server level:

Status - Checks the status of the server. The information about the server includes the server mode and the number of active client sessions for the server.

View Log - View the server log. For details see Daemon Logs on page 6-8.

View Events - For future use.

End - Ends the server process.

Refresh - Refreshes the display.

Daemon Logs

Oracle Connect for VSAM produces a number of logs that you can use to troubleshoot problems. The daemon manages the following logs:

- The daemon log
- The workspace log

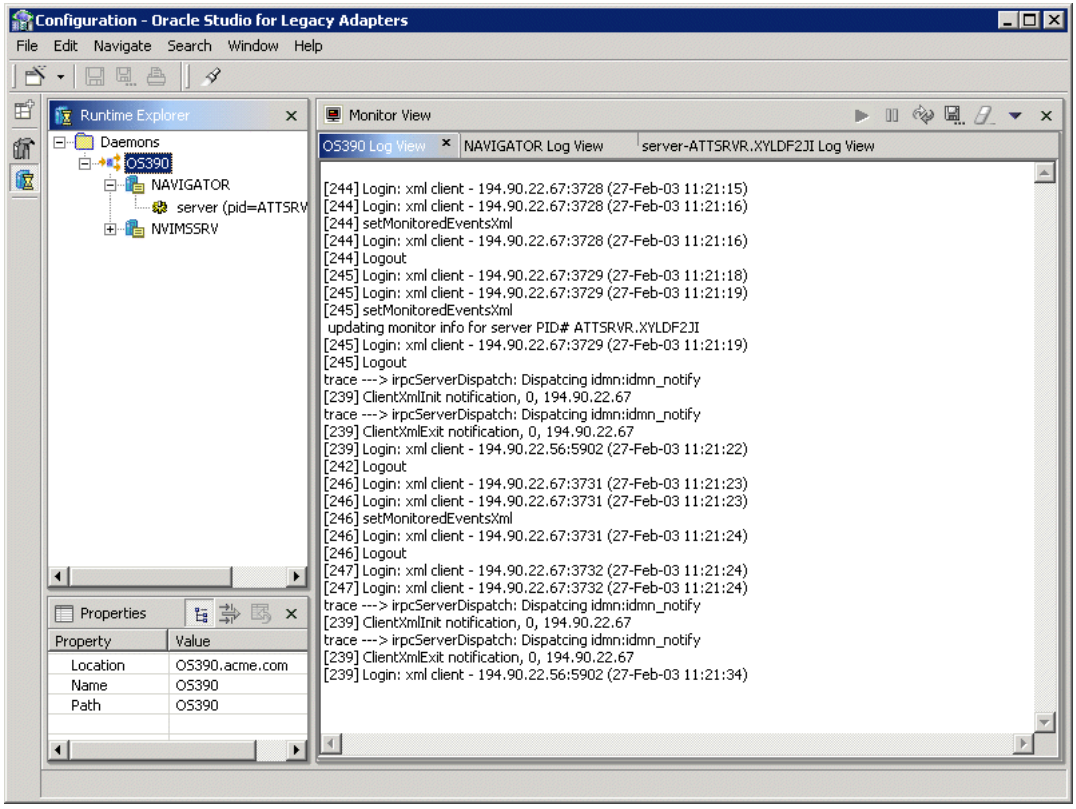
- The server process log

The Runtime Manager perspective of Oracle Studio provides a monitor for these logs.

Display the required log by right-clicking the level you want (daemon, workspace or server) and choosing the View Log option.

The Daemon Log Monitor

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



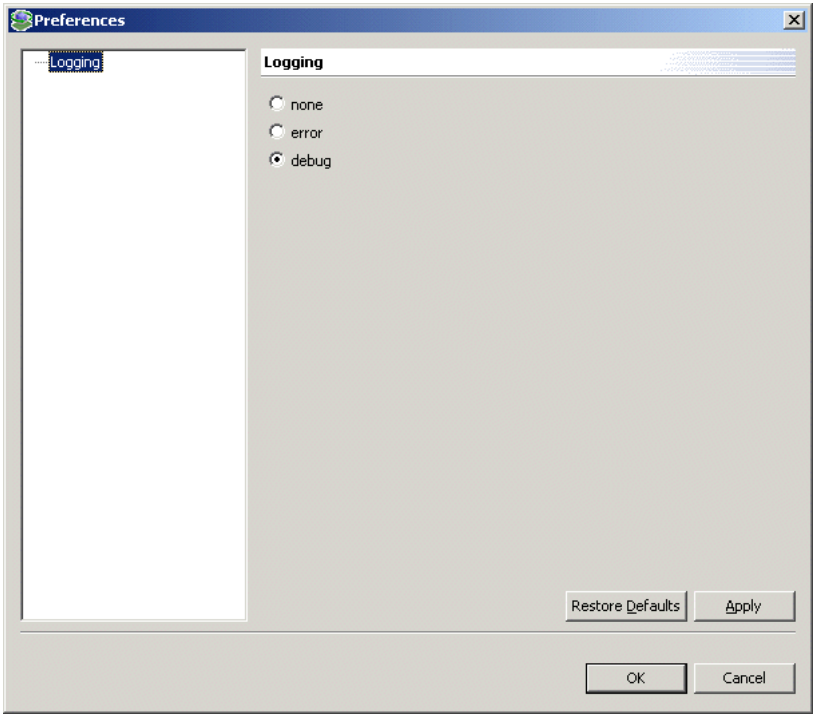
You can change the level of logging by clicking the **Update Monitor** button on the toolbar. The following levels of logging are available:

none - The log displays who has logged in and out from the daemon.

error - The log displays who has logged in and out from the daemon and any errors that have been generated.

debug - The log displays who has logged in and out from the daemon, any errors that have been generated and any tracing that has been specified in the daemon configuration.

See Also: "Daemon Logging" on page D-4



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 the **Update Monitor button** on the toolbar. The following levels of logging are available:

none - The log displays who has connected and disconnected from the server process.

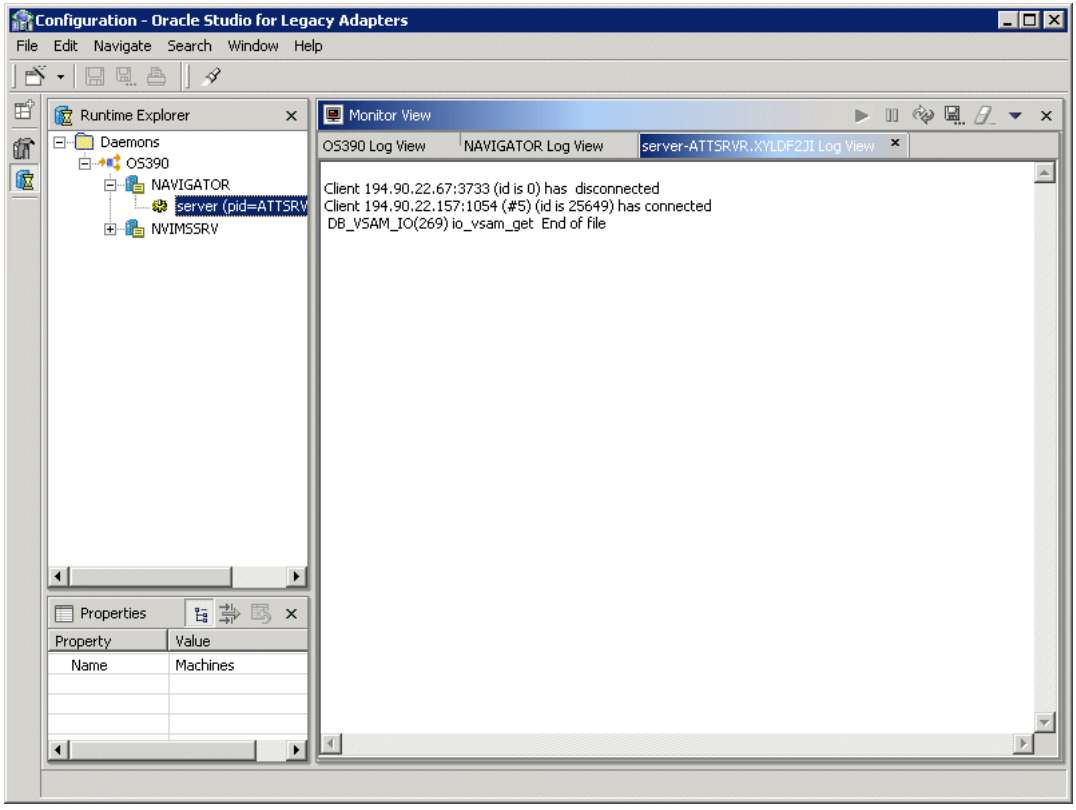
error - The log displays who has connected and disconnected from the server process and any errors that have been generated.

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: "WS Logging" on page D-16 for details

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 the **Update Monitor** button on the toolbar. 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: "WS Logging" on page D-16 for details

The following table describes the log monitor toolbar buttons and their functionality.

Table 6–1 Log Monitor Toolbar Buttons

Button	Name	Function
	Resume	Resumes monitoring the machine after the monitor has been suspended. This button toggles with the suspend button.
	Suspend	Suspends the monitor. This button toggles with the resume button.
	Update Monitor	Updates the information being monitored.
	Save as	Enables specifying a name and location for saving the log.
	Clear view	Clears the monitor view.
	Menu	Displays a menu with the options available using the buttons.

Resolving Communication Errors

When Oracle Studio disconnects from the OS/390 machine, the machine is displayed in Oracle Studio with an X in a red circle. If this situation occurs, try to access the machine later.

The following table describes the various scenarios that may exist when Oracle Application Server ProcessConnect disconnects from the OS/390 machine.

Table 6–2 Scenarios When a Client Is Disconnected

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

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

- The daemon configuration settings
- Oracle Connect for VSAM security
- The TCP/IP subsystem. Oracle Application Server Integration Adapter for VSAM uses TPC/IP for internal intermachine communications.
- System details, such as the account name and password of the administrator account, the IP address of the machines 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 delivery channel for the adapter.

Action: Check the TCP/IP subsystem on the current machine 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 machine with the port specified in the delivery channel for the adapter.

C008: Setting server event handler failed.

Cause: Internal error.

Action: Contact Oracle Support Services.

C009: IRPCD process has been terminated by user request.

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

Action: No action required.

C00A: Application %s not found.

Cause: The requested workspace does not exist.

Action: Check that the workspace defined in the delivery channel is also defined in the daemon configuration on the OS/390 machine. Use the Status option in the Runtime Manager perspective.

C00B: Invalid IRPCD client context.

Cause: A non-Oracle Connect legacy adapter 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 legacy adapter server or program was trying to use a daemon service reserved for Oracle Connect for VSAM 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/password.

Action: Edit the user profile in Oracle Studio to specify a valid username and password for the OS/390 machine.

> **See Also:** "Setting Up Runtime User Access to the IBM OS/390 Machine" on page 2-26 for details

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:** "Daemon Security" on page D-7 for details

C00F: Anonymous client logins are not allowed.

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

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

> **See Also:** "Daemon Security" on page D-7 for details

Action: Edit the user profile in Oracle Studio to specify a valid username and password for the OS/390 machine.

> **See Also:** "Setting Up Runtime User Access to the IBM OS/390 Machine" on page 2-26 for details

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. section of the daemon configuration.

See Also: "WS Info." on page D-10 for details

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 OS/390 machine.

See Also: "Setting Up Runtime User Access to the IBM OS/390 Machine" on page 2-26 for details

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 number of sub-tasks per address space in the WS Server section of the daemon configuration.

See Also: "WS Server" on page D-12 for details

Action: Try running the command later.

C015: Failed to start server process.

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

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

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

To set impersonation: APF authorize all the steplibs in the server script on OS/390 machine. For example:

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

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

In the WS Security tab of the Navigator workspace under the daemon node in the Configuration perspective, check `Use specific workspace account` the and clear the `Workspace account` field of all values.

C016: Unexpected server state.

Cause: Internal error.

Action: Contact Oracle Support Services.

C017: Active daemon clients exist. Shutdown canceled.

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

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

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

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

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

C01A: Lock %s not found.

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

Action: Contact Oracle Support Services.

C01B: Unexpected error in %s.

Cause: Internal error.

Action: Contact Oracle Support Services.

C01C: Cannot update configuration without _APPLICATIONS lock.

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 machine.

Action: Check that the name specified for the machine in the delivery channel 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 machine 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 for VSAM 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 for VSAM.

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

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

Cause: An Oracle Connect for VSAM 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: "WS Info." on page D-10 for details

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

Action: Increase the number of sub-tasks per address space in the WS Server section of the daemon configuration.

See Also: "WS Server" on page D-12 for details

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 machine 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 machine 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 for VSAM 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 port specified in the delivery channel 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 machine 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 for VSAM 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 for VSAM 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.

C02E: Call made to non-existing procedure %d

Cause: This error typically is caused by a client of a newer version calling an old server.

Action: Verify that the client and server are using the same version of Oracle Connect for VSAM.

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 and/or add memory to the system.

C036: Failed to register RPC procedure module %s

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

C038: Cannot re-register RPC procedure number %x

Cause: Internal error.

Action: Contact Oracle Support Services.

C042: Remote call to %s failed; %s

Cause: Remote call to API failed.

Action: Check the daemon log file.

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

See Also: "Daemon Logging" on page D-4 for details

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

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

Action: Check the remote machine definition in the delivery channel for the adapter.

Action: Check the daemon is up on the OS/390 machine. Use the Status option in the Runtime Manager perspective.

Action: Check the network connection by trying to ping the host machine 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 in the WS Info. section of the daemon configuration.

See Also: "WS Info." on page D-10 for details

C047: %s out of memory

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

Action: Kill 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 for VSAM daemon or server failed, or an established session with a server has failed.

Action: Check the remote machine definition in the delivery channel.

Action: Check the daemon is up on the OS/390 machine. 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 machine 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 machine.

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 re-install Oracle Connect for VSAM on the server.

C069: Fail to get server

Cause: The Oracle Connect for VSAM daemon on the server machine 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 for VSAM daemon log file on the server

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

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

Action: The server startup script failed.

Action: The username given is not allowed to use the requested server. Use an authorized 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, above.

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.

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

C06B: Disconnecting from server

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

Action: Oracle Connect for VSAM automatically tries to re-establish a connection with a server upon the next SQL command issued against the server. Once the network or machine failure is corrected, the connection to the daemon is re-established 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 machine failure or a server program failure caused the connection to abort. The currently active transaction is aborted as well.

Action: Normally, Oracle Connect for VSAM 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 and/or server machine should be fixed before connection can be resumed.

Action: In case of a server crash not related to callable user code, contact Oracle support.

C072: Reconnect to server %s

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

Action: No action is required.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: User/account has insufficient privileges.

Action: Grant administrative privileges to the user/account via the Administrator parameter of the Daemon Security or WS Security sections in the daemon configuration.

> **See Also:** "Daemon Security" on page D-7 or "WS Security" on page D-18 for details

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.

7

Configuring the OS/390 Working Environment

Oracle Connect for VSAM includes a number of tuning parameters that can improve performance. Specifically, the daemon can be configured to optimize communication between the OS/390 machine and the client machine and the binding environment can be tuned to optimize the way actual requests are handled.

This chapter contains the following sections:

- Configuring the Daemon for High Availability
- Configuring a Binding Environment

Configuring the Daemon for High Availability

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

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

The workspace that will be used is specified in the delivery channel, as described in Setting Up the Delivery Channel on page 4-2.

Task: 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.

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. In the Configuration Explorer, click the + next to the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Click the + next to the **Daemons** node. The daemon configurations available on this machine are listed.
4. Right-click **IRPCD** and select New Workspace from the popup menu.
5. In the New daemon workspace window, specify a name the new workspace and optionally a description.
6. 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 the ellipsis button and select the workspace from which you want to copy the properties.

Once the workspace is defined the daemon editor is displayed, showing the **WS Info** tab.

Task: Editing the Workspace

You edit a workspace using the following tabs:

- **WS Info** - Specifies general information including the server type, the command procedure used to start the workspace, the binding configuration associated with this workspace and the timeout parameters.
- **WS Server Mode** - Specifies workspace server information including features that control the operation of the servers started up by the workspace and allocated to clients.
- **WS Logging** - Specifies parameters for logging and the format to use for a log file.
- **WS Security** - Specifies administration privileges, user access, ports available to access the workspace and workspace account specifications.
- **WS Governing** - Specifies information used to govern how a VSAM query is executed.

Task: Configuring the Server Mode

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

Single Client - 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 (since 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 (since it requires as many server processes as concurrent clients).

Multi-Client - Clients share a server process and are processed serially.

This mode has low overhead since the server processes are already initialized. However, because clients share the same process, they may impact one another, especially if they issue lengthy queries.

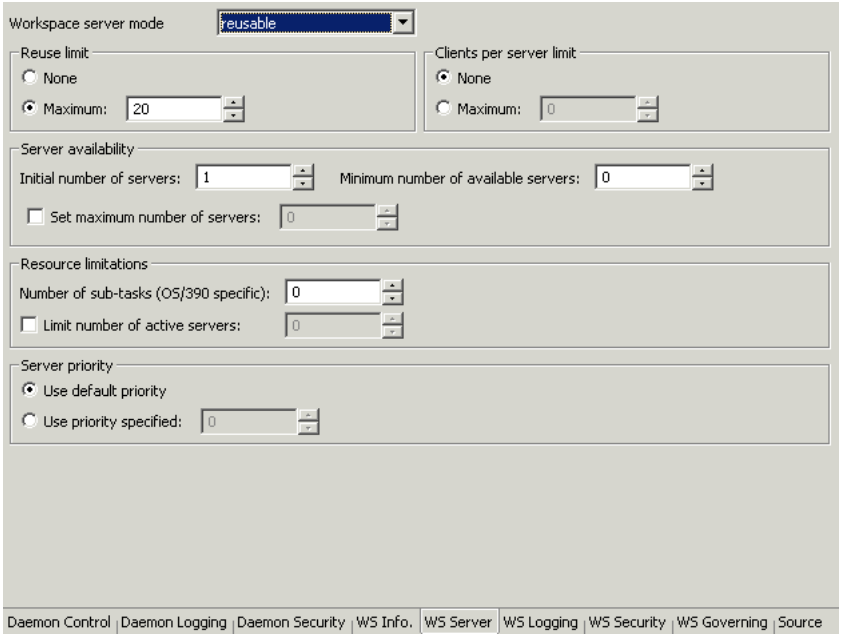
The number of clients that share a process is determined by the `Clients per server limit` (the maximum number of concurrent clients a server process for the current workspace accepts).

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 since the servers are already initialized. However, this server mode may use a lot of server resources (since 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 since it eliminates the need to repeat initializations. However, reuse runs a risk of higher memory leakage over time. The default for the `Reuse limit` field value is **None**, indicating that no reuse limit is enforced.

Set the server mode in the WS Server tab of the daemon editor:



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

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

Initial number of servers - The number of server processes that are prestarted for this workspace when the daemon starts 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 below the value specified in the `Minimum number of available servers` field, the daemon again starts server processes until the specified number of available servers is reached. The default for this parameter is 0, meaning that no servers are prestarted for this workspace.

Minimum number of available servers - The minimum number of server processes in the prestarted server's pool before the Oracle Connect daemon resumes creating new server processes (up to the number specified in the `Initial number of servers field` value, described above). If this parameter is set to a value greater than that of the `Initial number of servers` field value, the daemon considers the value to be the same as the value specified in the `Initial number of servers` field. In this case, a new server process is started and added to the pool each time a server process is removed from the pool and allocated to a client). The default for this parameter is 0, meaning that new servers are created only when there are no other available servers.

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

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 above, you can set additional server processes as sub-tasks by specifying this parameter. Thus, setting 10 servers and 10 prestarted sub-tasks results in 100 tasks started (10 sub-tasks for each process).

Configuring a Binding Environment

Each binding configuration includes the following:

- Environment settings, which are used to configure the environment used by any of the VSAM data sources and database adapters defined in the binding
- VSAM data sources on the current machine

- Application adapters on the current machine
- Inbound queues for adapters on the OS/390 workstation machine. (Inbound queues are referred to as events in Oracle Studio)

Configuring data sources, adapters and inbound queues are described in Chapter 3, "Modeling Interactions for Oracle Application Server Integration Adapter for VSAM". The following section describes the environment settings.

Task: Setting a Binding Environment

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. In the Configuration Explorer, click the + next to the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Click the + next to the **Bindings** node. The binding configurations available on this machine 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 click the value to edit. The binding environment is divided into the following categories:

- comm Category
- debug Category
- miscellaneous Category
- odbc Category
- oledb Category
- optimizer Category
- queryProcessor Category
- transactions Category
- tuning Category

comm Category

The following parameters define communication buffers:

comCacheBufferSize - The size of a memory buffer on a client, which is used by the Oracle Connect client/server to store read-ahead data. The default is 200000 bytes.

comMaxSocketSize - The maximum bytes that can be written in one chunk on a socket. The default is -1 (no limitation).

comMaxXmlInMemory - The maximum size of an XML document held in memory. The default is 65535 bytes.

comMaxXmlSize - The maximum size of an XML document passed to another machine. The default is 65535 bytes.

debug Category

The following parameters define debugging and logging operations.

acxTrace - When set to "true", the input xml sent to the backend adapter and the output xml returned by the backend adapter, are written to the log.

analyzerQueryPlan - For internal use.

gdbTrace - For internal use.

generalTrace - When set to "true", logs general trace information. The default writes only error messages to the log.

logFile - The high-level qualifier of the log file for messages. The following types of message are written to the log:

- Error messages
- Trace information and information about the query optimization strategy if generalTrace is set to "true"

oledbTrace - For internal use.

optimizerTrace - When set to "true", trace information and information about the Query Optimizer strategy is written to the log.

queryWarnings - When set to "true", generates a log of Query Processor warnings.

traceDir - For internal use.

miscellaneous Category

The following parameters define miscellaneous operations, including national language support (NLS) and the directory where temporary files are written.

codepage - For use with National Language Support (NLS) to identify the codepage for the workspace.

> **See Also:** Appendix F, "National Language Settings"

cvtSeverityLevel - The data type conversion policy when a conversion error occurs:

- 0 (Default) - The data in the output column will be a null or empty value.
- 1 - The data in the output column will be a null or empty value and the error is reported to the log.
- 2 - An error is reported and processing stops.

edit - For internal use.

language - Identifies the application language. A default codepage is selected based om the value specified for this parameter.

> **See Also:** Appendix F, "National Language Settings"

nlsString - Specifies the codepage used by a field whose data type is defined as "nlsString". You use this for a field whose codepage is other than that of the machine's codepage. This parameter includes the following values:

- The name of the codepage.
- Whether the character set reads from right to left (as in middle eastern character sets). The default is "false".

tempDir - 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.

year2000Policy - Determines the way 2-digit years are converted into 4-digit years. When the parameter year2000Policy is not set, or when it is set to a value outside the range of values defined for the policy, as described below, 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 4-digit year after 1900 that can be represented by a 2-digit year. For example, if year2000Policy is set to 1905, the years 2000->2004 will be represented by 00->04. All other 2 digits will map to 19xx.
>
> This solution is most appropriate if there is live data at the low end (close to the year 1900), which the user wants to keep with the current 2-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 2-digit number. With each passing year the earliest year that can be represented by a 2-digit number changes to a year later.

odbc Category

The odbc parameters are for internal use.

oledb Category

The oledb parameters are for internal use.

optimizer Category

The following parameters enable you to customize the performance.

goal - The optimization policy, which can be specified as one of the following:

- **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.

noHashJoin - When set to "true", disables hash join optimization. When hash joins are enabled, a significant amount of disk space is required. Therefore, if the system does not have much available disk space, disable hash join optimization.

noSemiJoin - For internal use.

preferredSite - For internal use.

semiJoinInValuesFactor - For internal use.

queryProcessor Category

The following parameters enable you to fine tune how queries are processed.

compileAfterLoad - For internal use.

dsoThreadingBoth - For internal use.

emptyStringIsNull - When set to "true", all empty strings are reported as "NULL".

firstTreeExtensions - The maximum size allowed for an SQL query after compilation. The default value is 150 KB.

forceQualifiedNames - For internal use.

ignoreSegmentBindFailure - For internal use.

maxColumnsInParsing - The maximum number of columns that a query references. The default is 500 columns.

maxSqlCache - The maximum number of SQL queries stored in cache memory. The default value is 3.

noCommandReuse - When set to "true", sets Query Processor not to cache for reuse the executed state of a query.

noCompilationCache - When set to "true", sets Query Processor not to save a successfully compiled statement in the cache.

noDSPropertyCache - For internal use.

noInsertParameterization - When set to "true", sets Query Processor not to parameterize constants in INSERT statements.

noMarkRowFailedFetch - When set to "true", enables Query Processor to continue fetching data after a failure in retrieving a row.

noMetadataCaching - When set to "true", Query Processor does not take object metadata from the cache but directly from VSAM.

noParallelExecution - For internal use.

noQueryParametrization - When set to "true", sets Query Processor not to convert constants into parameters when accessing VSAM.

noQueryReadAhead - For internal use.

noSessionPooling - For internal use.

noSQSCache - For internal use.

noThreadedReadAhead - For internal use.

noThreads - For internal use.

noTreeAutoExtend - For internal use.

optimisticForUpdate - When set to "true", optimistic locking becomes the default for locking behavior.

parserDepth - The maximum depth of the expression tree.

proceduresCacheSize - For internal use.

promptDbUserPassword - For internal use.

tokenSize – The maximum length of a string in an SQL query. The minimum value is 64. The default value is 350.

useAlternateQualifier - For internal use.

useTableFilterExpressions – When set to "true", enables the use of tables that have filter expressions specified in their metadata.

transactions Category

The transactions parameters are for internal use.

tuning Category

The following parameters define tuning.

dsmMaxBufferSize - The maximum size of a cache memory. This cache is used when memory is required on a temporary basis (as when Oracle Connect sorts data for a query output, for a subquery, or for aggregate queries). This cache size is not used for hash joins and lookup joins (see the hashBufferSize parameter, below). The default is 1000000 bytes.

dsmMaxSortBufferSize - The maximum size of the sort buffers. Use this parameter instead of dsmMaxBufferSize for sorts only. The default is 1000000 bytes.

dsmMidBufferSize - The maximum size of the index cache. This cache is not used for hash joins and lookup joins. The default is 100000 bytes.

hashBufferSize - The amount of cache memory that is available for each hash join or lookup join. The default is 1000000 bytes.

hashEnableParallelism - Both sides of a hash join are read concurrently. The default is "true".

hashMaxDiskSpace - The maximum amount of disk space (in MBs) that a query can use for hash joins. The default is unlimited (all the free space on the allocated

disk). If a query requires more space than allocated via this parameter, the query execution will stop. The minimum value for this parameter is 20 MB.

Temporary files are written per query. Therefore, if several users can execute queries at the same time, adjust the amount of space available, so that the total that can be allocated at any one time does not exceed the available space.

hashMaxOpenFiles - The maximum number of files that a query can open at one time for use when performing hash joins. The number assigned to this parameter must not exceed the system maximum. The default is 90 files.

A

The Oracle Connect VSAM Metadata Definitions

Oracle Studio includes an import utility to generate metadata for the VSAM data source, using COBOL copybooks. Once imported, the metadata is maintained in the Metadata perspective of Oracle Studio.

Oracle Studio also includes a wizard to help define the adapter inbound and outbound interactions and input and output structures used for these interactions. The interactions and input and output structures are maintained as metadata in the Metadata perspective of Oracle Studio.

This appendix contains the following sections:

- Metadata for the VSAM Data Source

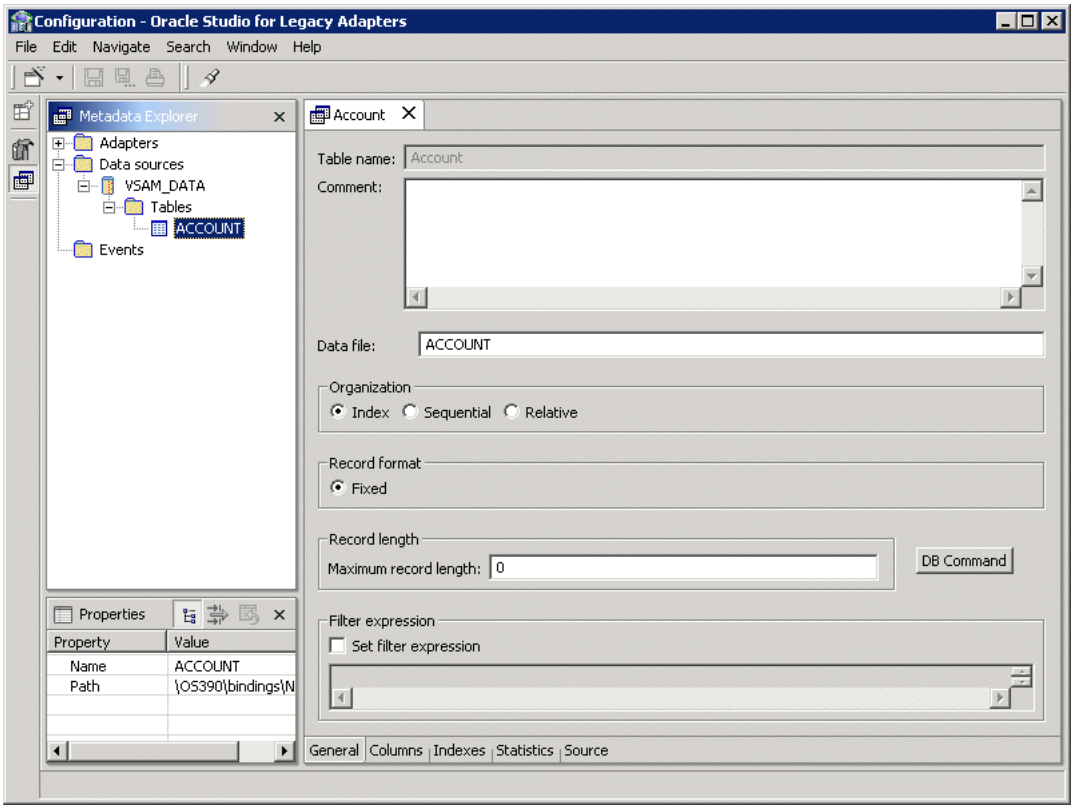
> **Note:** The Oracle Application Server Integration Adapter for VSAM provides a relational model for the VSAM data. Thus, relational terminology is used. For example, the metadata for a VSAM record is referred to as table metadata.

- Interaction Metadata for the Adapter

Metadata for the VSAM Data Source

Use Oracle Studio to maintain the metadata for the VSAM data source.

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle** > **Studio** > **Studio**.
2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Click the + next to the **Bindings** node.
4. Click the + next to the **NAV** binding node.
5. Click the + next to the **Data sources** node to display the data sources.
6. Right-click the VSAM data source in the **Configuration Explorer** and choose **Edit Metadata** to open the Metadata perspective.



7. Right-click the table in the Metadata Explorer you want to view and choose **Edit**.

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

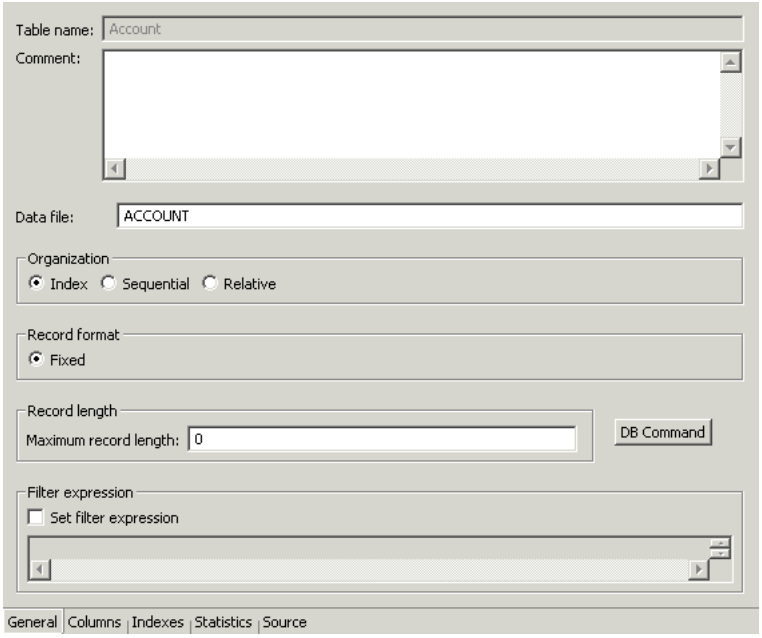
- General Tab
- Columns Tab
- Indexes Tab
- Statistics Tab

The following tab provides an XML representation of the data source metadata:

- Source 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 has the following fields:

Table name - The name of the table.

Comment - An optional description of the table.

Data File - The location of the file containing the table.

Organization - How the record represented by the table is organized. The options that are displayed depend on the record. The following options are available:

- Index
- Sequential
- Relative - used with RRDS files. Access to a specific record number of a relative file is performed by using a pseudo column to specify the record position. The hash symbol (#) is used to specify a pseudo column. For example:

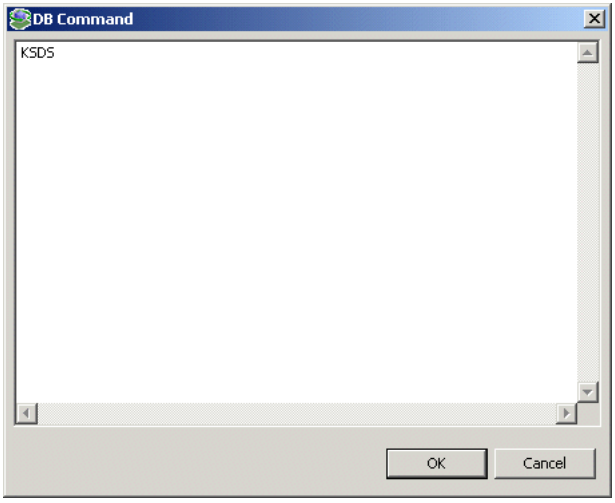
```
SELECT * FROM colleges WHERE # = 6
```

Record format - 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 - How the record represented by the table is formatted:

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.

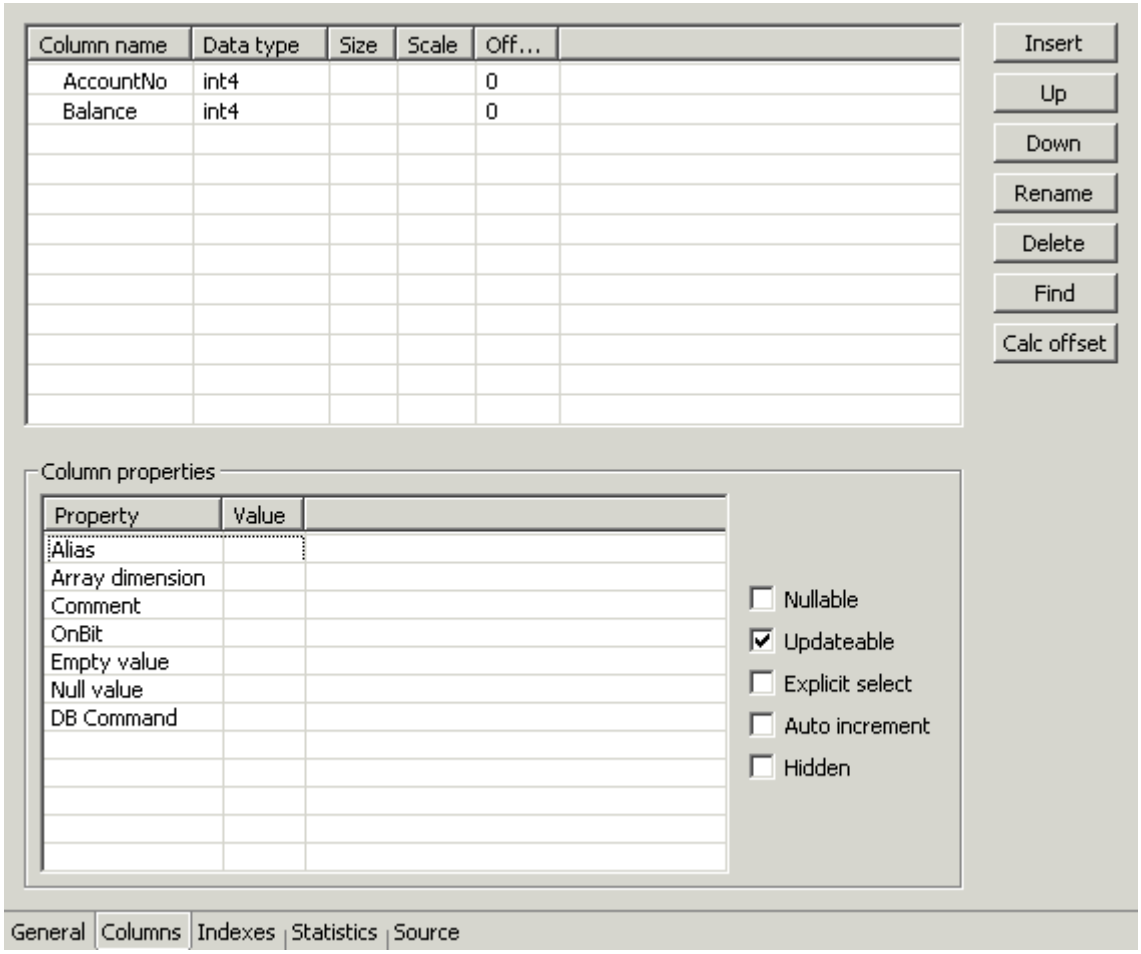
VSAM specific commands are displayed in the DB Command window, which you open by clicking **DB Command**.



The information displayed is generated automatically when the metadata is generated and should not be modified.

Columns Tab

Use the **Columns** tab to specify metadata describing the columns of the table.



The **Columns** tab has the following fields:

Column Name - The name of the column.

Data Type - The datatype of the data the column contains. Selecting this field displays a list of the possible datatypes.

> **See Also:** "VSAM Data Source Metadata Datatypes" on page B-6 for details about the available datatypes

Size - The size allocated for the column.

Scale - The definition depends on the datatype. For decimal datatypes this value 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 datatypes this value is the total number of digits. The number must be negative.

Offset - An absolute offset for the field in a record.

The **Column Properties** relate to the selected column.

Alias - A name used to replace the default virtual table name for an array. Virtual table names are generated by appending the array name to the record name. Thus, 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 to be usable, specify an Alias to replace the long name.

Array dimension - The maximum number of occurrences of the group of columns that make up the array.

Comment - A comment about the specific column.

OnBit - The position of the bit in a BIT field and the starting bit in a BITS field.

Empty value - The value for the field during an insert operation, when a value is not specified

Null value - The null value for the field during an insert operation, when a value is not specified.

DBcommand - VSAM-specific commands for the column. The information displayed is generated automatically when the metadata is generated and should not be modified.

Nullable - The current field can contain NULL values.

Updateable - The current field can be updated.

Explicit select - The current field is not returned when you execute a `SELECT * FROM...` statement. To return this field, you must explicitly ask for it (in a query such as `SELECT NATION_ID, SYSKEY FROM NATION` where `SYSKEY` is a field defined with Explicit Select).

Note: You cannot include the asterisk (*) in a query where you want to retrieve a field defined with Explicit Select. That is, a statement such as `SELECT *, SYSKEY FROM NATION` will not return `SYSKEY`.

You can disable this attribute by specifying the disableExplicitSelect attribute in Oracle Studio, in the properties for the data source.

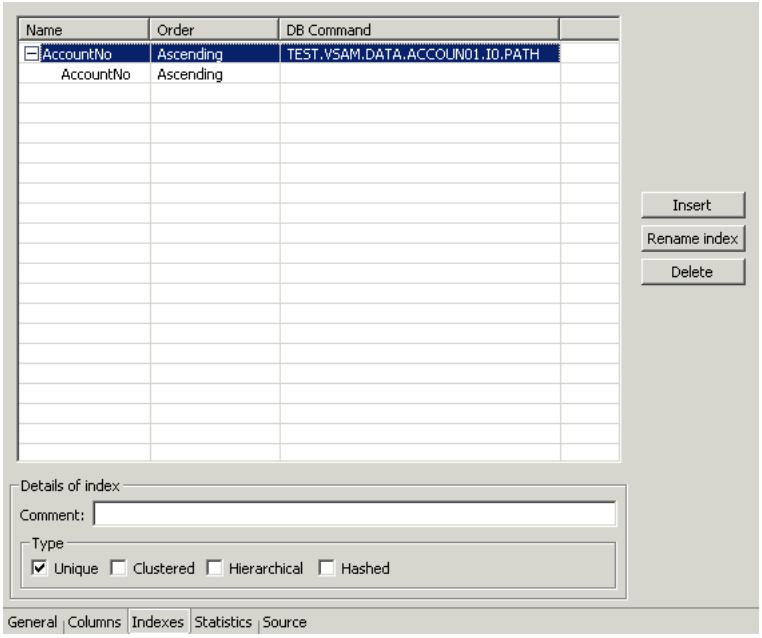
Auto increment - The current field is updated automatically by the data source during an INSERT statement and should not be explicitly specified in the INSERT statement. The INSERT statement should include an explicit list of values. This attribute is used for fields such as an order number field whose value is incremented each time a new order is entered to the data source.

Hidden - The current field is hidden from users.

Indexes Tab

Use the **Indexes** tab to specify metadata describing the indices 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 has the following fields:

Name - The names of existing indexes for the current table.

Order - The ordering of the rows retrieved by the index.

DB Command - VSAM-specific commands for the index. The information displayed is generated automatically when the metadata is generated and should not be modified.

The following index details can be specified:

Comment - An optional comment on the selected index.

Type - Indicates the type of the index:

Unique - Indicates that the current index is unique.

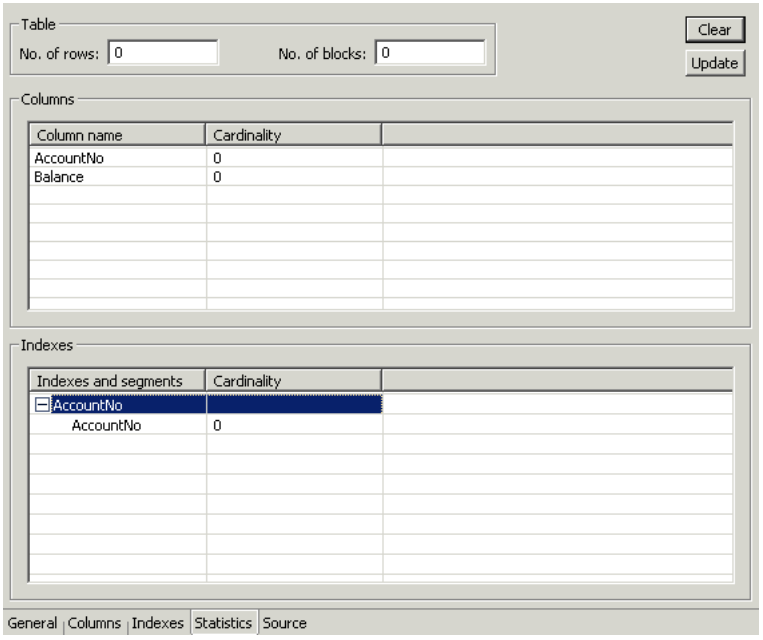
Clustered - Indicates that the current index is clustered.

Hierarchical - Indicates that the current index is hierarchical.

Hashed - Indicates that the current index is hashed.

Statistics Tab

Use the **Statistics** tab to update statistics for a table.



The **Statistics** tab has the following fields:

No. of rows - The approximate number of rows in the table. If the value is -1, the number of rows in the table is unknown (a value was not supplied and the update statistics utility was not run to update the value). A value of 0 indicates that this table is empty.

No. of blocks - The approximate number of blocks in the table.

> **Warning: If neither the number of rows nor the number of blocks is specified for a table, queries over the table might be executed in a nonoptimal manner.**

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

Column name - The columns in the table.

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

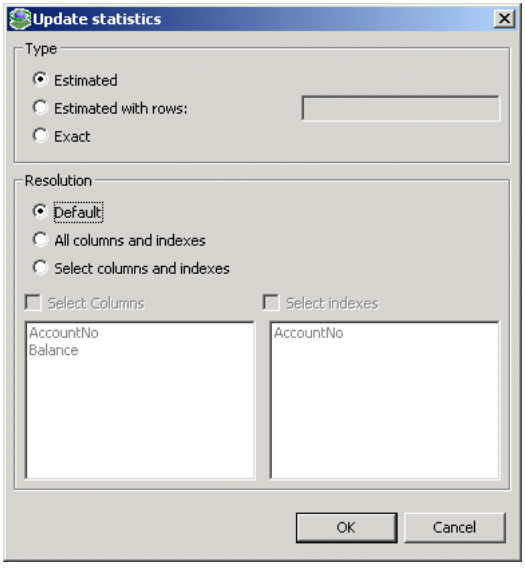
Use the Indexes group box to specify cardinality for the columns in each of the indexes in the table:

Indexes and segments - The indexes and segments in the table.

Cardinality - The number of distinct key values in the index. If the value is -1, 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.

Generating Statistics

Click the **Update** button in the **Statistics** tab to generate updated statistics for the table.



The following fields are used to update statistics:

Use the **Type** group box to specify the following:

Estimated - An estimation of the amount of statistical information returned.

Estimated with rows - An estimation of the amount of statistical information returned. The estimate includes an estimation of the number of rows in the table. Specify the number in the text box. This number is used to shorten the time to produce the statistics, assuming that the value specified here is the correct value, or close to the correct value.

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

Exact - The exact statistical information returned. Note that this can be a lengthy task and can lead to disk space problems with large tables.

Default - Only information about the table and indexes is collected. Information for partial indexes and columns is not collected.

All columns and indexes - Information about the table, indexes, partial indexes and columns is collected.

Select columns and indexes - Enables you to select the columns and indexes for which you want to collect statistics. In the enabled list of columns or indices left click those columns you want included (you can use shift-click and control-click to select a number of columns or indices).

Source Tab

The **Source** tab displays the XML representation of the metadata.

Interaction Metadata for the Adapter

Use Oracle Studio to maintain the metadata for the VSAM adapter.

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Click the + next to the **Bindings** node. The binding configurations available on this machine are listed.
4. Click the + next to the **NAV** binding node to display the adapters.
5. Right-click the VSAM adapter in the **Configuration Explorer** and choose **Edit Metadata** to open the Metadata perspective.
6. Right-click the adapter in the Metadata Explorer and choose **Edit definition**.

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

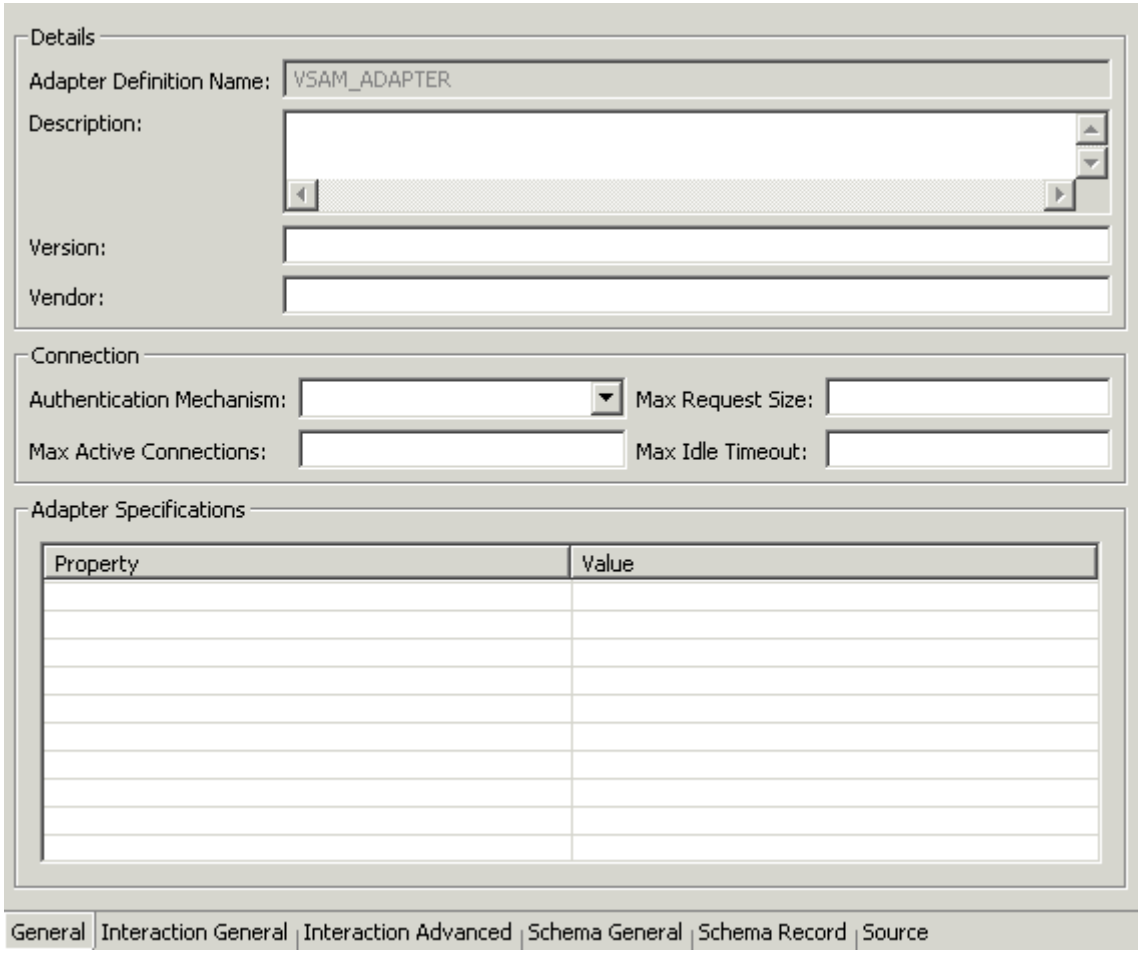
- General Tab
- Interaction General Tab
- Interaction Advanced Tab
- Schema General Tab
- Schema Record Tab

The following tab provides an XML representation of the metadata:

- Source Tab

General Tab

Use the **General** tab to maintain information generally describing the adapter and the connection to it.



The **General** tab has the following fields:

Adapter definition name - The name of the adapter definition.

Description - An identifying description of the adapter.

Version - The adapter version.

Vendor - The vendor which the adapter came from.

Authentication mechanism - The authentication to access the adapter. The available mechanisms are:

- kerbv5
- none
- basic-password

Max request size - The maximum size in bytes for an XML request or reply. Larger messages are rejected with an error.

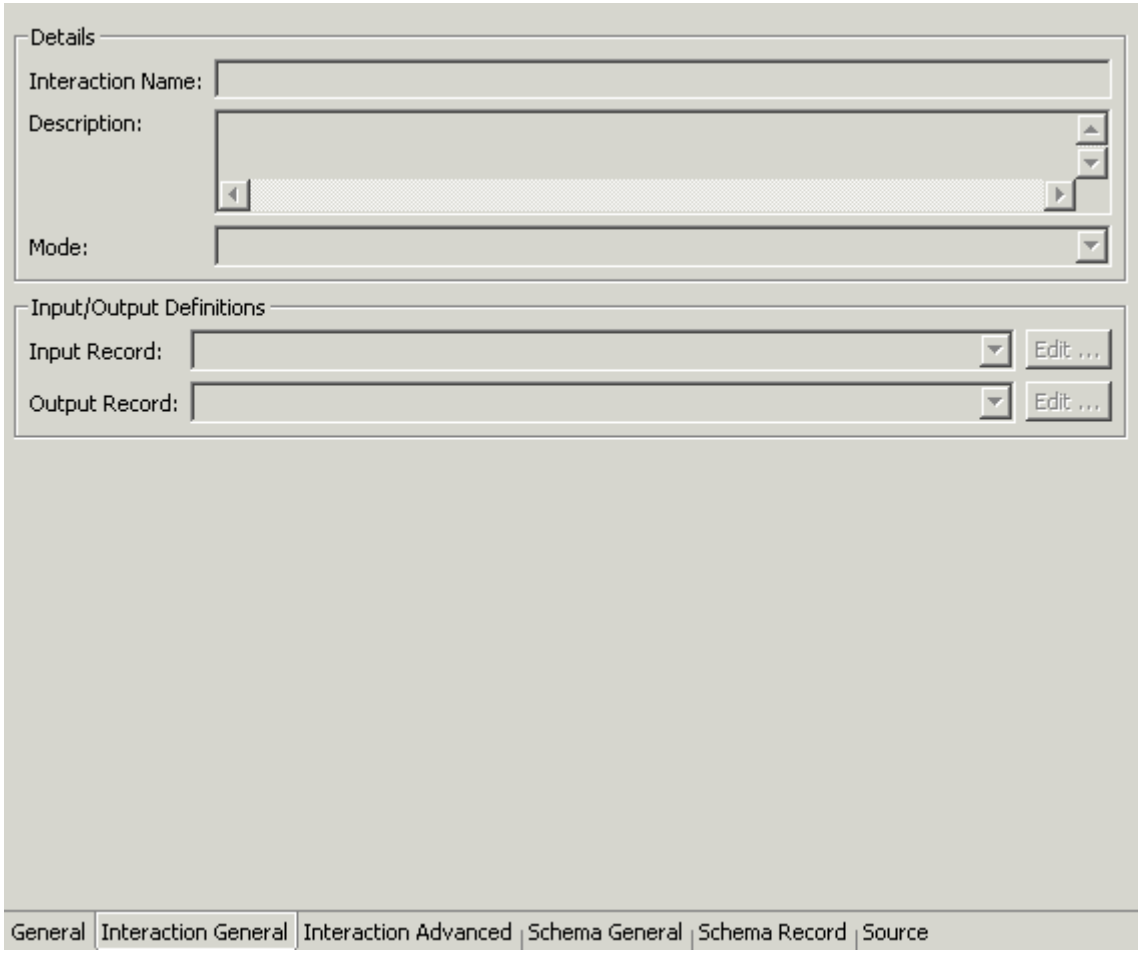
Max active connections - The maximum number of simultaneous connections for an adapter (per process).

Max idle timeout - The maximum time, in seconds, that an active connection can stay idle. After that time, the connection is closed.

Adapter Specifications - Adapter-specific properties.

Interaction General Tab

Use the **Interaction General** tab to define the general details of the interaction in addition to its input and output definitions.



The **Interaction General** tab has the following fields:

Interaction name - The name of the interaction.

Description - A descriptive identifier for the interaction.

Mode - The interaction mode:

- **sync-send-receive** - The outbound interaction sends a request and expects to receive a response.
- **sync-send** - The outbound interaction sends a request and does not expect to receive a response.

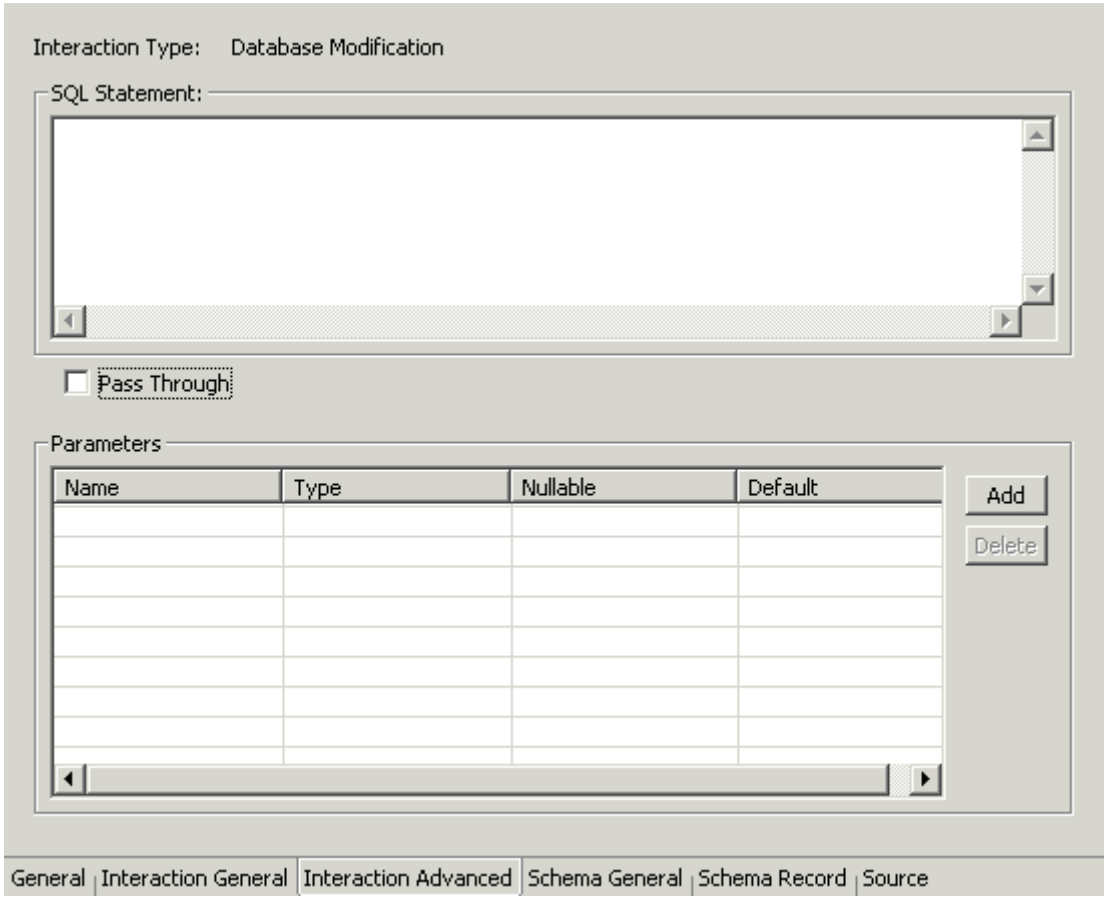
- **sync-receive** - The outbound interaction expects to receive a response.
- **async-send** - The inbound interaction sends a request and does not expect to receive a response.

Input record - Identifies an input record.

Output record - Identifies an output record for the results of an outbound interaction.

Interaction Advanced Tab

Use the **Interaction Advanced** tab to define specific interaction properties.



The **Interaction Advanced** tab has the following fields:

SQL Statement - The SQL statement that is executed on the VSAM data source by the interaction. Parameters are specified in the SQL, in a SET clause or in a WHERE clause, with the following format:

```
:parameter_name
```

Pass Through - For internal use.

Parameters - Defines the property and value of parameters specific to the interaction.

Name - The name of the parameter.

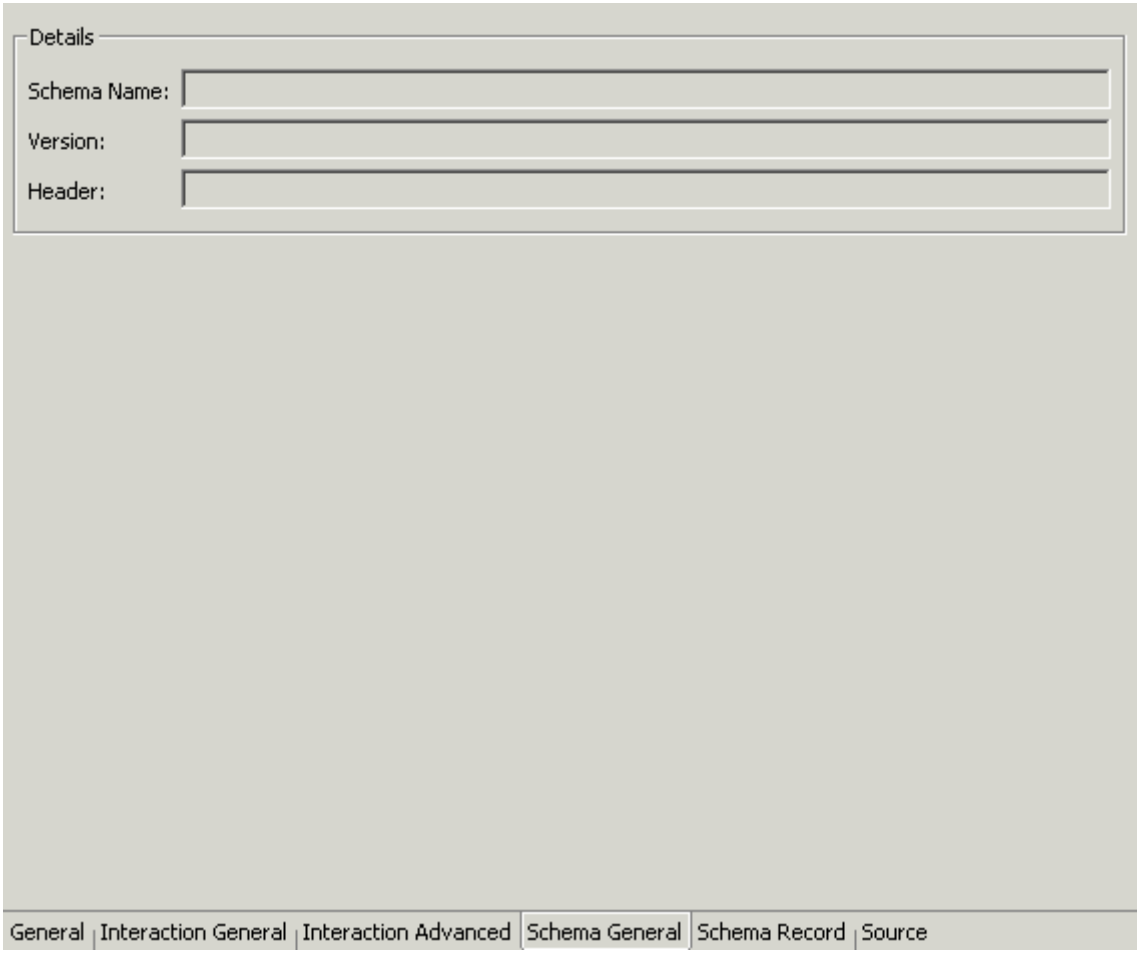
Type - The type of the parameter (such as number or string).

Nullable - Whether the value can be null or not.

Default - A default value for the parameter.

Schema General Tab

Use the **Schema General** tab to define the general details of the input and output record structures for the interaction.



The **Schema General** tab has the following fields:

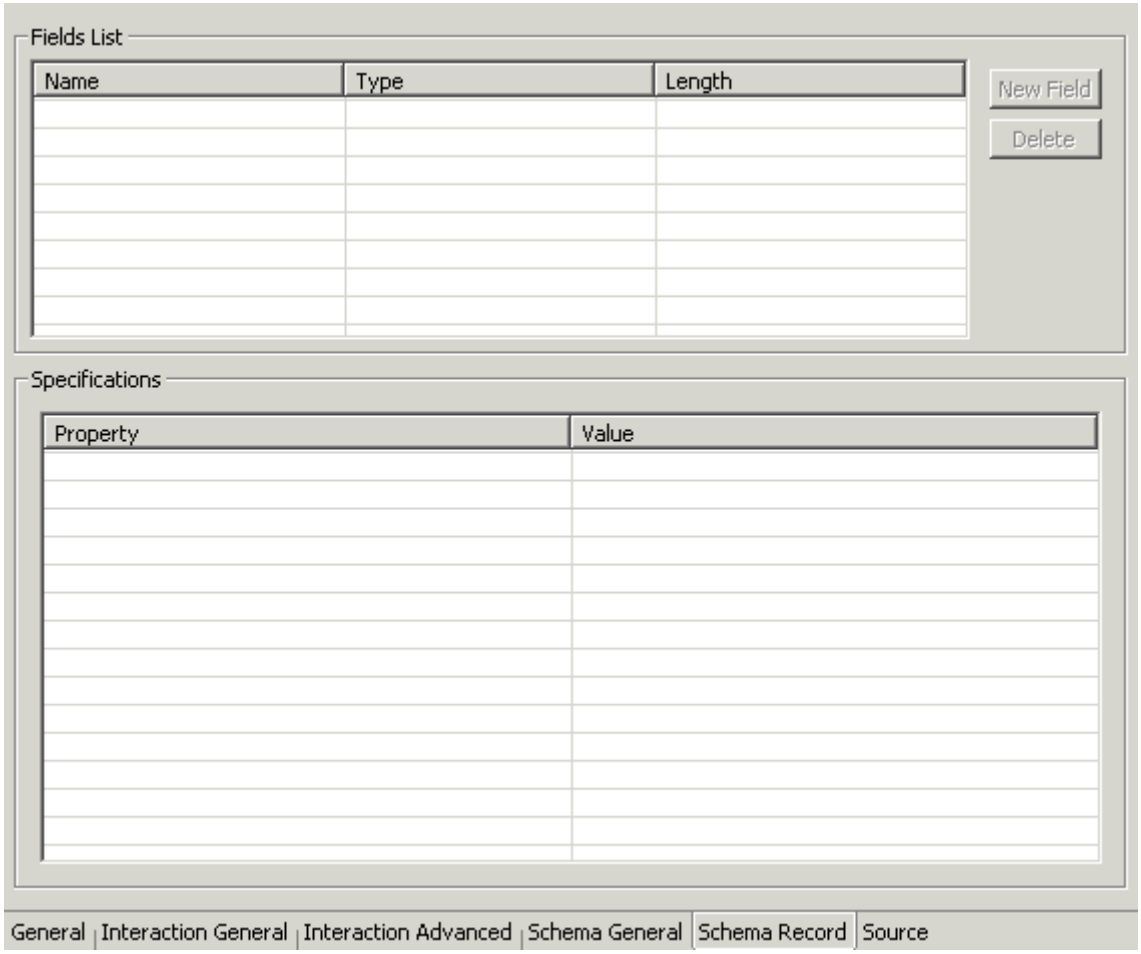
Schema name - The name of the adapter.

Version - The schema version.

Header - A C header, file to map between the data structure and the adapter.

Schema Record Tab

Use the **Schema Record** tab to define the input and output record structures for the interaction.



The **Schema Record** tab has the following fields:

The **Fields List** defines single data items within a record.

Name - The name of the field.

Type - The datatype of the field. The following are valid datatypes:

Binary

Boolean

Byte

Date

Double

Enum

Float

Int

Long

Numeric[(p[,s])]

Short

String

Time

Timestamp

Length - The size of the field including a null terminator, when the datatype supports null termination (such as the cstring data type).

Use the **Specifications** box to specify field properties.

Source Tab

The **Source** tab displays the XML representation of the adapter metadata.

B

Datatype Support

The Oracle Application Server Integration Adapter for VSAM supports a number of datatypes that are used when defining metadata for both the VSAM data source and VSAM backend adapter in Oracle Studio. The datatypes are mapped from the COBOL datatypes during the import procedure.

> **Note:** The mapping of datatypes between Oracle Application Server Integration Adapter for VSAM and Oracle Application Server ProcessConnect is performed internally by Oracle Connect for VSAM.

This appendix contains the following sections:

- Datatype Mapping for the VSAM Data Source
- VSAM Data Source Metadata Datatypes
- Datatype Mapping for Oracle Connect for VSAM

Datatype Mapping for the VSAM Data Source

The following table shows the mapping performed to generate metadata for a VSAM data source from COBOL copybooks.

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

Table B–1 Datatype Mapping: COBOL and VSAM Data Source

COBOL Datatype	Oracle Connect Datatype	Details
BINARY (3 <= *size* < 5 with fractional data)	scaled_int2	Signed word integer Size = 2 bytes Range: -32768 to +32767
BINARY (3 <= *size* < 5 without fractional data)	int2	Signed word integer Size = 2 bytes
BINARY (5 <= *size* < 10 with fractional data)	scaled_int4	Signed longword Size = 4 bytes Range: -2147483648 to +2147483647
BINARY (5 <= *size* < 10 without fractional data)	int4	Signed longword Size = 4 bytes
BINARY (*size* < 3 with fractional data)	scaled_int1	Signed byte integer Size = 1 byte Range: -128 to +127
BINARY (*size* < 3 without fractional data)	int1	Signed byte integer Size = 1 byte
BINARY (*size* >= 10 with fractional data)	scaled_int8	Signed quadword Size = 8 bytes Range: -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807
BINARY (*size* >= 10 without fractional data)	int8	Signed quadword Size = 8 bytes
COMP (3 <= *size* < 5 with fractional data)	scaled_int2	Signed word integer Size = 2 bytes Range: -32768 to +32767

Table B–1 (Cont.) Datatype Mapping: COBOL and VSAM Data Source

COBOL Datatype	Oracle Connect Datatype	Details
COMP (3 <= *size* < 5 without fractional data)	int2	Signed word integer Size = 2 bytes
COMP (5 <= *size* < 10 with fractional data)	scaled_int4	Signed longword Size = 4 bytes Range: -2147483648 to +2147483647
COMP (5 <= *size* < 10 without fractional data)	int4	Signed longword Size = 4 bytes
COMP (*size* < 3 with fractional data)	scaled_int1	Signed byte integer Size = 1 byte Range: -128 to +127
COMP (*size* < 3 without fractional data)	int1	Signed byte integer Size = 1 byte
COMP-2	dfloat	Double floating point number Size = 8 bytes Range: 0.29E-38 to 1.7E38 Precision: 16 digits
COMP-3	decimal	Packed decimal Maximum number of digits = 31 Maximum fractions = 11 Length = int (number of digits/2) + 1 bytes
COMP-4 (5 <= *size* < 10 with fractional data)	scaled_int4	Signed longword Size = 4 bytes Range: -2147483648 to +2147483647
COMP-4 (5 <= *size* < 10 without fractional data)	int4	Signed longword Size = 4 bytes
COMP-4 (*size* < 5 with fractional data)	scaled_int2	Signed word integer Size = 2 bytes Range: -32768 to +32767
COMP-4 (*size* < 5 without fractional data)	int2	Signed word integer Size = 2 bytes

Table B–1 (Cont.) Datatype Mapping: COBOL and VSAM Data Source

COBOL Datatype	Oracle Connect Datatype	Details
COMP-4 (*size* >= 10 with fractional data)	scaled_int8	Signed quadword Size = 8 bytes Range: -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807
COMP-4 (*size* >= 10 without fractional data)	int8	Signed quadword Size = 8 bytes
COMP-5 (5 <= *size* < 10 with fractional data)	scaled_int4	Signed long integer Size = 4 bytes Range: -2147483648 to +2147483647
COMP-5 (5 <= *size* < 10 without fractional data)	int4	Signed long integer Size = 4 bytes
COMP-5 (*size* < 5 with fractional data)	scaled_int2	Signed word integer Size = 2 bytes Range: -32768 to +32767
COMP-5 (*size* < 5 without fractional data)	int2	Signed word integer Size = 2 bytes
COMP-5 (*size* >= 10 with fractional data)	scaled_int8	Signed quadword Size = 8 bytes Range: -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807
COMP-5 (*size* >= 10 without fractional data)	int8	Signed quadword Size = 8 bytes
COMP-X (3 <= *size* < 5 with fractional data)	scaled_int2	Signed word integer Size = 2 bytes Range: -32768 to +32767
COMP-X (3 <= *size* < 5 without fractional data)	int2	Signed word integer Size = 2 bytes
COMP-X (5 <= *size* < 10 with fractional data)	scaled_int4	Signed long integer Size = 4 bytes Range: -2147483648 to +2147483647

Table B–1 (Cont.) Datatype Mapping: COBOL and VSAM Data Source

COBOL Datatype	Oracle Connect Datatype	Details
COMP-X (5 <= *size* < 10 without fractional data)	int4	Signed long integer Size = 4 bytes
COMP-X (*size* < 3 with fractional data)	scaled_int1	Signed byte integer Size = 1 byte Range: -128 to +127
COMP-X (*size* < 3 without fractional data)	int1	Signed byte integer Size = 1 byte
COMP-X (*size* >= 10 with fractional data)	scaled_int8	Signed quadword Size = 8 bytes Range: -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807
COMP-X (*size* >= 10 without fractional data)	int8	Signed quadword Size = 8 bytes
INDEX	int4	Signed long integer Size = 4 bytes
[SIGN [IS]] LEADING	numstr_nl	Signed numeric string. Sign is the first character of the string Maximum number of digits = 31 Maximum fractions = 11
[SIGN [IS]] LEADING SEPARATE [CHARACTER]	numstr_nl	Signed numeric string. Sign is the first character of the string Maximum number of digits = 31 Maximum fractions = 11
NATIVE-2	int2	Signed word integer Size = 2 bytes
NATIVE-4	int4	Signed long integer Size = 4 bytes
NATIVE-8	scaled_int8	Signed quadword Size = 8 bytes Range: -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807

Table B–1 (Cont.) Datatype Mapping: COBOL and VSAM Data Source

COBOL Datatype	Oracle Connect Datatype	Details
PACKED-DECIMAL	decimal	Packed decimal Maximum number of digits = 31 Maximum fractions = 11 Length = int (number of digits/2) + 1 bytes
POINTER	int4	Signed long integer Size = 4 bytes
[SIGN [IS]] TRAILING	numstr_nr	Signed numeric string. Sign is the last character of the string. Maximum number of digits = 31 Maximum fractions = 11
[SIGN [IS]] TRAILING SEPARATE [CHARACTER]	numstr_nr	Signed numeric string. Sign is the last character of the string Maximum number of digits = 31 Maximum fractions = 11

VSAM Data Source Metadata Datatypes

The following table shows all the valid datatypes that can be used to describe the metadata for a VSAM data source.

Note: This section is for reference only. The only datatypes that should be used are those generated for the COBOL copybook, as described in "Datatype Mapping for the VSAM Data Source" on page B-2.

Table B–2 VSAM Metadata Datatypes

Oracle Connect Datatype	Details
bit	Single bit within a byte Size = 1 byte Format: `<field name="name" datatype="bits" onBit="n" ... />`, where *n* specifies the bit (within a byte) the field uses

Table B–2 (Cont.) VSAM Metadata Datatypes

Oracle Connect Datatype	Details
bits	A signed number of bits within a byte Size = 1 bit to 1 byte Format: `<field name="name" datatype="bits" onBit="n" size="m"... />`, where *n* specifies the bit (within a byte) to start from and *m* is the number of bits. If *n* is not specified then n defaults to 1 for the first occurrence of the field and is contiguous thereafter. The maximum number of bits you can map is 32
cstring	Null-terminated string of alphanumeric characters; maximum length must be specified. An extra byte is required for the null flag
date6	Date in a string having the form YYMMDD
date8	Date in a string having the form YYYYMMDD
decimal	Packed decimal Maximum number of digits = 31 Maximum fractions = 11 Length = int (number of digits/2) + 1 bytes
dfloat	Double floating point number Size = 8 bytes Range: 0.29E-38 to 1.7E38 Precision: 16 digits
double	Double floating number Size = 8 bytes Range: 0.56E-308 to 0.90E308 Precision: 15 digits
filler	Allocation for future use, string type; length must be specified
image	Image data (Binary large object)
int_date	Date in a four byte integer having the form YYMMDD or YYYYMMDD Example, 23-Jul-1998 has the form: 980723 or 19980723
int1	Signed byte integer Size = 1 byte
int2	Signed word integer Size = 2 bytes

Table B–2 (Cont.) VSAM Metadata Datatypes

Oracle Connect Datatype	Details
int3	Signed integer Size = 3 bytes
int4	Signed longword Size = 4 bytes
int6	Signed integer Size = 6 bytes
int8	Signed quadword Size = 8 bytes
jdate	Julian date Size = 2 bytes Bits 0-6: (non-century) year Bits 7-15: day of the year
logical	Signed long integer Values: 1 for true, 0 for false.
numstr_bdn	Signed numeric string. Sign is the first character of the string. Maximum number of digits = 31 Maximum fractions = 11 **Note**: the number of fractions includes the decimal point
numstr_nl	Signed numeric string. Sign is the first character of the string Maximum number of digits = 31 Maximum fractions = 11
numstr_nlo	Signed numeric string. A left overpunched sign is implemented. Maximum number of digits = 31 Maximum fractions = 11
numstr_nr	Signed numeric string. Sign is the last character of the string. Maximum number of digits = 31 Maximum fractions = 11
numstr_s	Signed numeric string. A right overpunched sign is implemented Maximum number of digits = 31 Maximum fractions = 11

Table B–2 (Cont.) VSAM Metadata Datatypes

Oracle Connect Datatype	Details
numstr_u	Unsigned numeric string Maximum number of digits = 31 Maximum fractions = 11
numstr_zoned	Signed numeric string Maximum number of digits = 31 Maximum fractions = 11
phdate	Size = 2 bytes Bits 0-6: (non-century) year Bits 7-10: number of month Bits 11-15: day of month
scaled_int1	Signed byte integer Size = 1 byte Range: -128 to +127
scaled_int2	Signed word integer Size = 2 bytes Range: -32768 to +32767
scaled_int3	Signed integer Size = 3 bytes
scaled_int4	Signed long integer Size = 4 bytes Range: -2147483648 to +2147483647
scaled_int6	Signed integer Size = 6 bytes
scaled_int8	Signed quadword Size = 8 bytes Range: -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807
scaled_uint1	Unsigned byte integer Size = 1 byte Range: 0 to 254
scaled_uint2	Unsigned word integer Size = 2 bytes Range: 0 to 65534

Table B–2 (Cont.) VSAM Metadata Datatypes

Oracle Connect Datatype	Details
scaled_uint4	Unsigned long integer Size = 4 bytes Range: 0 to +4,294,967,294.
scaled_uint4	Unsigned long integer Size = 4 bytes Range: 0 to 4294967294
str_date	Atomic date string Size = 10 characters Format = YYYY-MM-DD
str_time	Atomic date string Size = 8 characters Format = HH:MM:SS
string	String of alphanumeric characters; length must be specified
text	Text data (Binary large object)
ubits	An unsigned number of bits within a byte Size = 1 bit to 1 byte Format: `<field name="name" datatype="bits" onBit="n" size="m" ... />`, where *n* specifies the bit (within a byte) to start from and *m* is the number of bits. If *n* is not specified then n defaults to 1 for the first occurrence of the field and is contiguous thereafter The maximum number of bits you can map is 31
uint2	Unsigned word integer Size = 2 bytes Range: 0 to +65534.
uint6	Unsigned integer Size = 6 bytes
unicode	Null-terminated alphanumeric unicode string; maximum length must be specified
unspecified	Unknown datatype, string type, length must be specified
varstring	16 bit count, followed by a string
varstring4	32 bit count, followed by a string

Datatype Mapping for Oracle Connect for VSAM

The following table shows the mapping performed to generate metadata for the VSAM backend adapter from COBOL copybooks.

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

Table B–3 Datatype Mapping: COBOL and Oracle Connect for VSAM

COBOL Datatype	Xsd Datatype
BINARY (with fractional data)	string
BINARY (without fractional data)	int
COMP (with fractional data)	string
COMP (without fractional data)	int
COMP-2	double
COMP-3	string
COMP-4 (with fractional data)	string
COMP-4 (without fractional data)	int
COMP-5 (with fractional data)	string
COMP-5 (without fractional data)	int
COMP-X (with fractional data)	string
COMP-X (without fractional data)	int
INDEX	int
[SIGN [IS]] LEADING	string
[SIGN [IS]] LEADING SEPARATE [CHARACTER]	string
NATIVE-2	int
NATIVE-4	int
NATIVE-8	string
PACKED-DECIMAL	string
POINTER	int
[SIGN [IS]] TRAILING	string

Table B–3 (Cont.) Datatype Mapping: COBOL and Oracle Connect for VSAM

COBOL Datatype	Xsd Datatype
[SIGN [IS]] TRAILING SEPARATE [CHARACTER]	string

C

Supported SQL Syntax and SQL Enhancements

The Oracle Application Server Integration 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

The Oracle Application Server Integration Adapter for VSAM provides full support for `SELECT`, `DELETE`, `INSERT`, and `UPDATE` statements, when specified in interactions.

Supported Operators

The Oracle Application Server Integration Adapter for VSAM enables the use of the following operators in SQL statements:

- Arithmetic Operators
- Comparison Operators

Table C–1 **Arithmetic Operators**

Operator
+
-
*
/

Table C–2 **Comparison Operators**

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

Supported Functions

The Oracle Application Server Integration Adapter for VSAM enables the use of the following functions in SQL statements:

- String Functions
- Group Functions
- Mathematical Functions

Table C–3 String Functions

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

Table C–4 Group Functions

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

Table C–4 (Cont.) Group Functions

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

Table C–5 Mathematical Functions

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

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:

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.

D

Daemon Configuration Settings

The daemon runs on the OS/390 machine and is responsible for allocating a server process for a client. The daemon resides in a single address space and is executed as a started task. When started, the daemon loads configuration settings such as various operational parameters as well as the list of workspaces accessible through the daemon.

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

This appendix contains the following sections:

- Daemon Control
- Daemon Logging
- Daemon Security
- Workspaces

Daemon Control

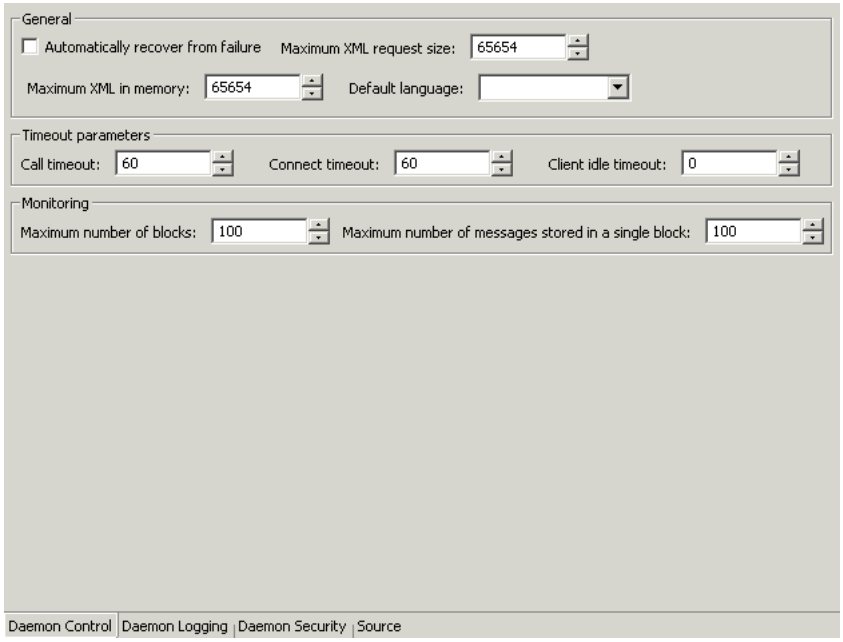
The **Daemon Control** section specifies various control options.

The **Daemon Control** section is accessed as follows:

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Right-click the machine and choose **Open Runtime Perspective**.
4. Right-click the daemon in the **Runtime Explorer** and choose **Edit Daemon Configuration**. The **Daemon Control** tab opens.
5. After making changes to the daemon, right-click the daemon and choose **Reload Configuration**.

Note: You can also change daemon settings in the using the **Configuration Explorer**, by selecting a machine and drilling-down the list until the daemon is displayed. Right-click the daemon and choose **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.



The following fields are displayed:

Automatically recover from failure - The daemon restarts automatically if it fails for any reason (any error that causes the daemon process to terminate, such as network process lost or the CPU running the daemon crashes and the backup daemon is defined on another CPU). All available and unconnected servers are terminated and any connected servers are marked and terminated on release. Also the backup starts a backup for itself.

The backup appends a new log file to the log of the original daemon, adding a line indicating that a backup daemon was started.

Maximum XML request size - The maximum number of bytes that the daemon handles for an XML document.

Maximum XML in memory - The maximum amount of space reserved for the XML in memory.

Default language - The language that the daemon supports. This setting is used when working with a client with a code page different from the server code page.

Call timeout - The timeout period for *short* calls for all daemons. The definition of a short call is a call that should be completed in a few seconds. For example, most

calls to a database such as DESCRIBE should be completed in a few seconds as opposed to call like a GETROWS call, which can take a long time. In heavily loaded or otherwise slow systems, even short calls such as calls to open a file, may take a significant amount of time. If a short call takes more than the specified time to complete, the connection is aborted. The default value for this parameter is 60 seconds. Values of less than 60 seconds are considered to be 60 seconds.

Specifying the timeout in a workspace overrides the value set in this field for that workspace.

Connect timeout - The time the client waits for a daemon server to start. If the daemon server does not start within this period, 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.

Specifying the timeout in a workspace overrides the value set in this field for that workspace.

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.

Daemon Logging

The **Daemon Logging** section defines the daemon log file settings, the log file structure and the location where the log is saved. In addition it defines the data that is logged and traced in the file.

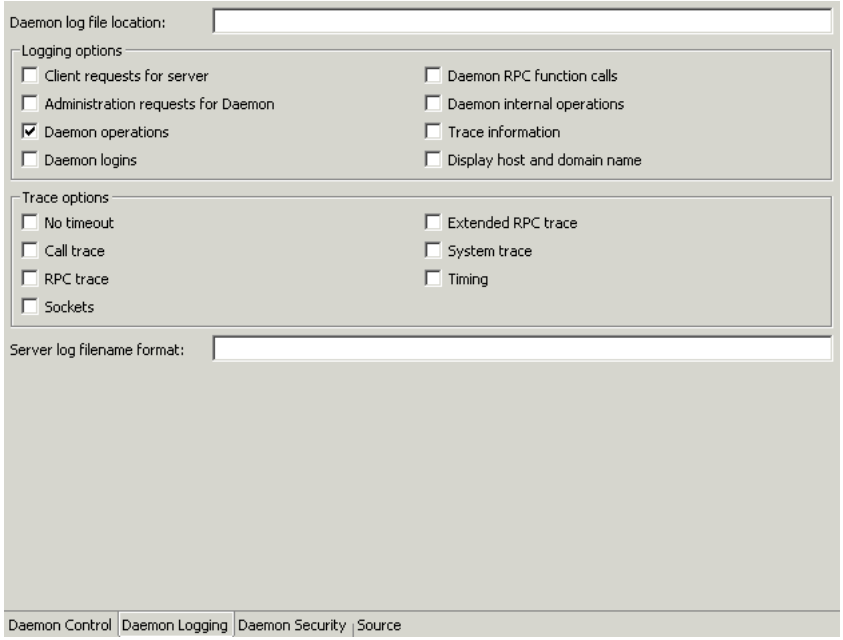
Note: By default the log entries are written to SYSOUT for the daemon process.

The **Daemon Logging** section is accessed as follows:

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Right-click the machine and choose **Open Runtime Perspective**.

4. Right-click the daemon in the **Runtime Explorer** and choose **Edit Daemon Configuration**.
5. Choose the **Daemon Logging** tab.
6. After making changes to the daemon, right-click the daemon and choose **Reload Configuration**.

> **Note:** You can also change daemon settings in the using the **Configuration Explorer**, by selecting a machine and drilling-down the list until the daemon is displayed. Right-click the daemon and choose **Edit Daemon**.
>
> Changes made to the daemon configuration in the way are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.



The following fields are displayed:

Daemon log file location - Where the daemon produces its log data if you want the data written to a file instead of SYSOUT for the daemon process. The full path must be specified.

Logging options - Specifies what tracing is performed.

Client requests for server - Logs client requests for server activations; this provides logging of the process IDs of the started servers along with the location of the log files.

Administration requests for daemon - Logs all of the administration requests for the daemon.

Daemon operations - Logs all of the daemon operations.

Daemon logins - Logs daemon logins.

Daemon RPC function calls - Log all daemon RPC function calls.

Daemon internal operations - Log daemon internal operations.

Log trace information - Logs low-level RPC operations.

Display host and client domain name - Whether the client host and domain name are logged rather than the client IP address. The default is false.

Trace options - Specifies what tracing is performed.

No timeout - Disables the standard RPC timeouts, setting them to a long duration (approximately an hour) to facilitate debugging.

Call trace - Generates a message in the server log file for each RPC function called. This is useful for troubleshooting the server.

RPC trace - Enables debugging messages on the server.

Sockets - Generates a message in the server log file for each socket operation.

Extended RPC trace - Generates a verbose message in the server log file for each low-level RPC function called. This is useful for troubleshooting the server.

System trace - Generates system-specific tracing of various operations.

Timing - Generates a timestamp for every entry to the server log file.

Server log filename format - The name of the server log file if you want the data written to a file instead of SYSOUT for the server process.

The following tokens can appear in the log file template and will be replaced accordingly:

%A - workspace name

%D - date (yymmdd)

%I - instance number of the given workspace server

%L - the path to *INSTROOT*.TMP. If you specify a file name without this path, the file is created under *INSTROOT*.TMP.*filename*, where *INSTROOT* is the high-level qualifier where Oracle Connect for VSAM is installed.

%P - server's process ID

%T - time (hhmmss)

%U - server's account name (username)

For example, a log file template %L.ATTSRVR%I can produce a log file such as: *INSTROOT*.TMP.ATTSRVR5, where *INSTROOT* is the high-level qualifier where Oracle Connect for VSAM is installed.

Daemon Security

The **Daemon Security** section is used for the following:

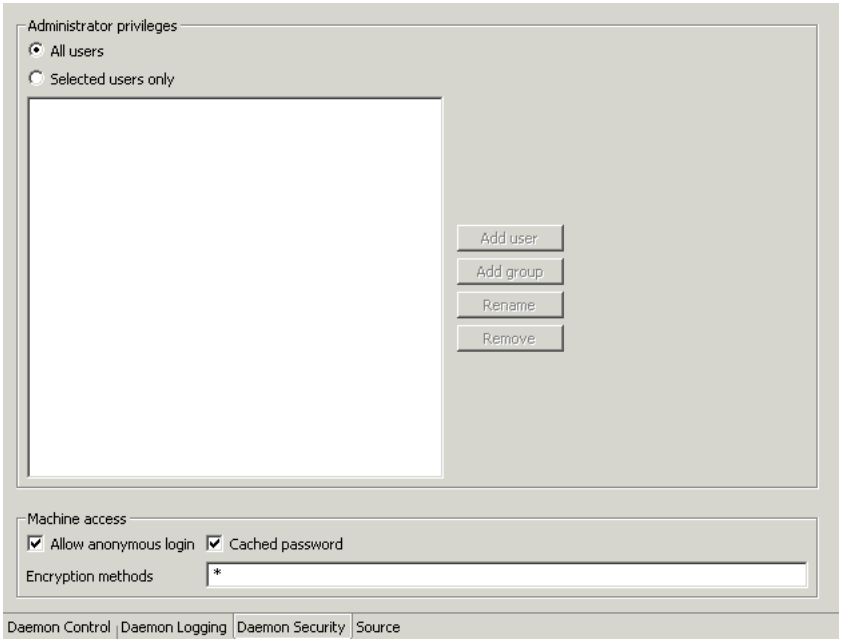
- To grant administration rights for the daemon
- To determine access to the machine

The **Daemon Security** section is accessed as follows:

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Right-click the machine and choose **Open Runtime Perspective**.
4. Right-click the daemon in the **Runtime Explorer** and choose **Edit Daemon Configuration**.
5. Choose the **Daemon Security** tab.
6. After making changes to the daemon, right-click the daemon and choose **Reload Configuration**.

Note: You can also change daemon settings in the using the **Configuration Explorer**, by selecting a machine and drilling-down the list until the daemon is displayed. Right-click the daemon and choose **Edit Daemon**.

Changes made to the daemon configuration in the way are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.



The following fields are displayed:

Administrators privileges - Identifies the users (accounts) allowed to perform administrative tasks (tasks that require administrative login).

All users - Anyone can access the daemon and change the settings.

Selected users only - The names of users (accounts) and groups that can be administrators.[1]

Note: If a user is not specified, the account from which the daemon was started is considered the administrator. Note that the daemon does not require the user to log in to the account on the system, but to log in to the daemon using the account name and password.

Machine access - Manages access to the machine.

Anonymous login allowed - Whether workspaces allow anonymous logins (without user name/password entries). For the optimal level of security, keep this option unchecked and define a username for the Daemon Administrators parameter.

If unchecked, no workspace can have an anonymous client. If checked, a particular workspace allows anonymous clients.

Cached passwords - Enables login passwords to be cached. This enhances performance by reducing login times for future connections from the same client in a session.

Encryption methods - The encryption method being used to send information across the network. The default is an asterisk (*), meaning that all methods are acceptable. If an encryption method is specified, it must be used. The RC4 and DES3 protocols are currently supported.

Workspaces

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 machine for the duration of the client request. Each workspace has its own definition. The workspace definition is divided into the following groups:

- WS Info.
- WS Server
- WS Logging
- WS Security

[1] The name is prefixed with '@', to utilize the operating system GROUP feature.

- WS Governing

Warning: **You must not delete the Navigator workspace.**

WS Info.

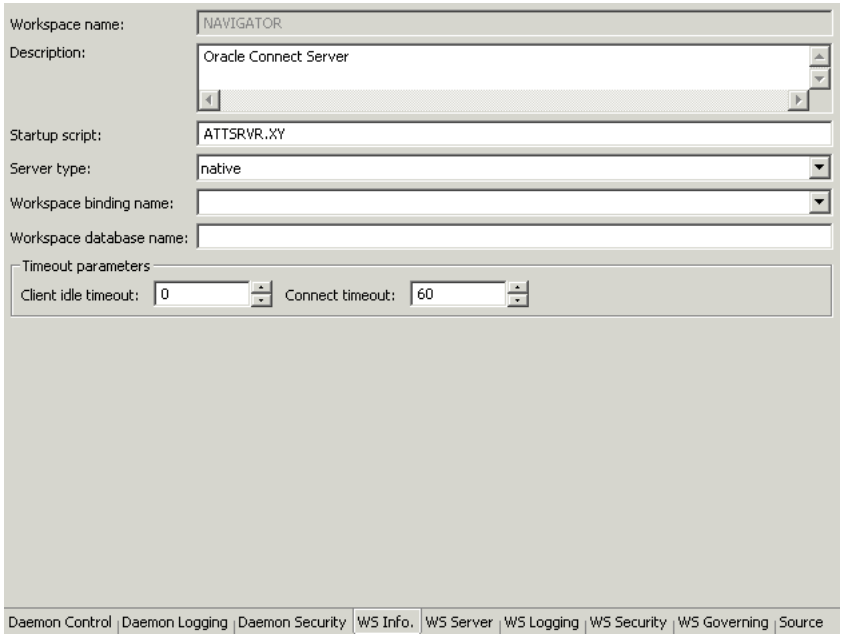
Using **WS Info.** you specify the features that control the operation of the workspace: the server type, the command procedure used to start the workspace and the binding configuration associated with this workspace.

The **WS Info.** section is accessed as follows:

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Right-click the machine and choose **Open Runtime Perspective**.
4. Drill-down the daemon to display the workspaces in the **Runtime Explorer**.
5. Right-click the workspace and choose **Edit Workspace Configuration**. The **WS Info.** tab opens.
6. After making changes to the daemon, right-click the daemon and choose **Reload Configuration**.

Note: You can also change workspace settings in the using the **Configuration Explorer**, by selecting a machine and drilling-down the list until the daemon is displayed. Drill-down the daemon to list the workspaces. Right-click the workspace and choose **Edit Workspace**.

Changes made to the daemon configuration in the way are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.



The following fields are displayed:

Workspace name - The name used to identify the workspace.

> **Note:** The default configuration includes the default Navigator Workspace. This workspace is automatically used if a workspace is not specified as part of the connection settings.

Description - A description of the workspace.

Startup script - The full pathname 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 for VSAM includes a default startup script, which it is recommended to use.

Specify only the script name, since the server is activated as a started task.

Note: The installation includes a default startup script: `ATTSRVR`.

Server type - For internal use only.

Workspace binding name - For internal use only.

Timeout parameters - The time the client waits for the workspace server to start. If the workspace server does not start within this period, the client is notified that the server did not respond. Specifying the timeout here overrides the default setting, specified in the Control section.

> **See Also:** "Daemon Control" on page D-2 for details about the Control section

Client idle timeout - The maximum amount of time a workspace client can be idle before the connection with the server is closed.

Connect timeout - The time the client waits for a workspace server to start. If the workspace server does not start within this period, 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.

WS Server

Using **WS Server**, you specify 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.

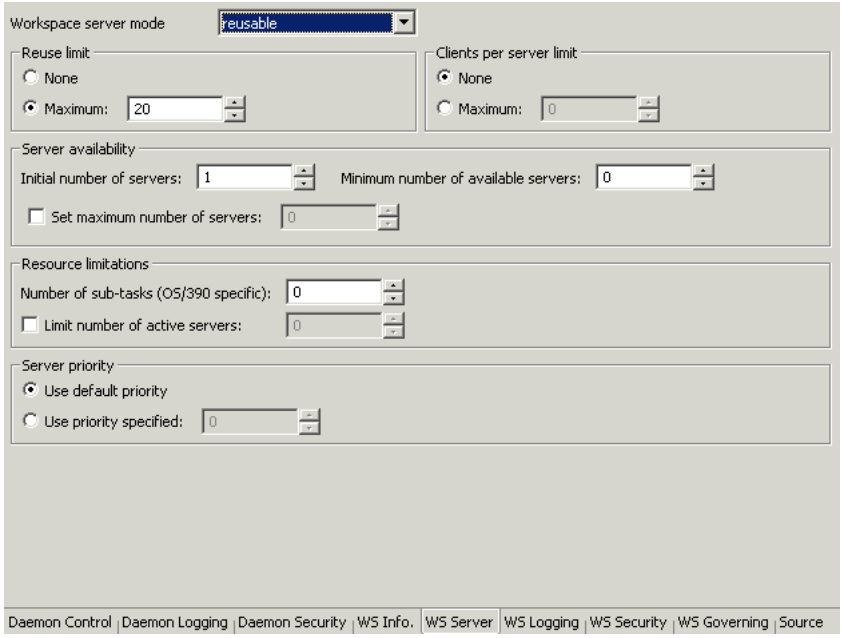
The **WS Server** section is accessed as follows:

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Right-click the machine and choose **Open Runtime Perspective**.
4. Drill-down the daemon to display the workspaces in the **Runtime Explorer**.
5. Right-click the workspace and choose **Edit Workspace Configuration**.

6. Choose the **WS Server** tab.
7. After making changes to the daemon, right-click the daemon and choose **Reload Configuration**.

Note: You can also change workspace settings in the using the **Configuration Explorer**, by selecting a machine and drilling-down the list until the daemon is displayed. Drill-down the daemon to list the workspaces. Right-click the workspace and choose **Edit Workspace**.

Changes made to the daemon configuration in the way are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.



The following fields are displayed:

Workspace server mode - The type of new server processes that the daemon starts up. The daemon supports the following server modes:

Single client - 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 (since 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 (since it requires as many server processes as concurrent clients).

Multi-client - Clients share a server process and are processed serially. This mode has low overhead since 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.

Multi-threaded - For internal use only.

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 since the servers are already initialized. However, this server mode can use a lot of server resources (since it requires as many server processes as concurrent clients).

Note: 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.

Reuse limit - 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 enabled only if the server mode value is `Multi-client` or `reusable`.

Clients per server limit - 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 per server is unlimited. This field is enabled only if the server mode value is `Multi-client`.

Server availability

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 below the value specified in the `Minimum number` field (see below), 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.

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 above this value are terminated. The default for this field is 0, meaning that there is no maximum.

Resource limitations

Number of subtasks - The number of subtasks for a server that are prestarted for this workspace when the daemon starts up. Thus, setting 10 prestarted servers and 10 subtasks results in 100 tasks started (10 subtasks for each process).

Limit number of active servers - The maximum number of active server processes (either available or in use). Once reached, no new server processes will be created for the particular workspace and client connections would be rejected if there is no available server to accept them. Once the number of active servers drops below the maximum (for example, a client disconnects from a server and the server terminates), new servers can again be started. If the value of this field is set to a nonzero value lower than the value for the **Initial number of servers** field, the daemon assumes it is set to the same value as the **Initial number of servers** field. The default for this field is 0, meaning that no maximum is enforced.

Server Priority - 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.

Use default priority - Sets the priority as 0. There is no specific priority for this workspace.

Use priority - Enables setting the priority.

WS Logging

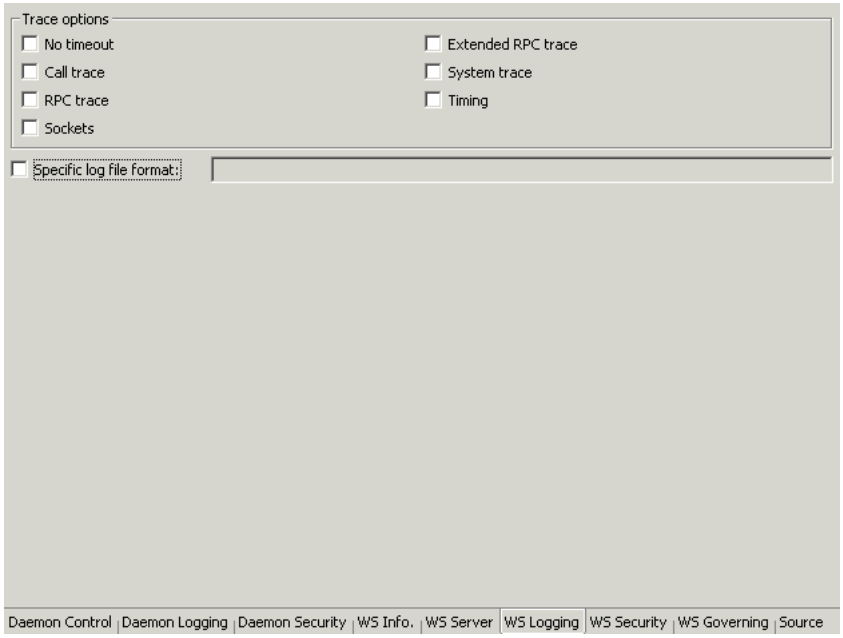
Using **WS Logging** you specify parameters to log that occur with the workspace server process.

The **WS Logging** section is accessed as follows:

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Right-click the machine and choose **Open Runtime Perspective**.
4. Drill-down the daemon to display the workspaces in the **Runtime Explorer**.
5. Right-click the workspace and choose **Edit Workspace Configuration**.
6. Choose the **WS Logging** tab.
7. After making changes to the daemon, right-click the daemon and choose **Reload Configuration**.

Note: You can also change workspace settings in the using the **Configuration Explorer**, by selecting a machine and drilling-down the list until the daemon is displayed. Drill-down the daemon to list the workspaces. Right-click the workspace and choose **Edit Workspace**.

Changes made to the daemon configuration in the way are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.



The following fields are displayed:

Trace options - Specifies what tracing is performed.

No timeout - Disables the standard RPC timeouts, setting them to a long duration (approximately an hour) to facilitate debugging.

Call trace - Generates a message in the server log file for each RPC function called. This is useful for troubleshooting the server.

RPC trace - Enables debugging messages on the server.

Sockets - Generates 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 - Generates a verbose message in the server log file for each low-level RPC function called. This is useful for troubleshooting the server.

System trace - Generates operating system-specific tracing.

Timing - Generates a timestamp for every entry to the server log file.

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 full pathname. If no directory information is provided for the log file, it will be located in the login directory of the account running the server.

The following tokens can appear in the log file template and will be replaced accordingly:

%A - workspace name

%D - date (yymmdd)

%I - instance number of the given workspace server

%L - the path to *INSTROOT*.TMP. If you specify a file name without this path, the file is created under *INSTROOT*.TMP.filename, where *INSTROOT* is the high-level qualifier where Oracle Connect for VSAM is installed.

%P - server's process ID

%T - time (hhmmss)

%U - server's account name (username)

Note: Unless this parameter is specified, the log entries are written to SYSOUT for the job specified in the **Workspace startup** parameter.

WS Security

Using **WS Security** you specify the level of security at the workspace level, as opposed to the daemon level, which is set in the Security section of the daemon.

> **See Also:** For details about the Security section, see "Daemon Security" on page D-7

The **WS Security** section is used for the following:

- To grant administration rights for the workspace
- To determine access to the workspace by a client

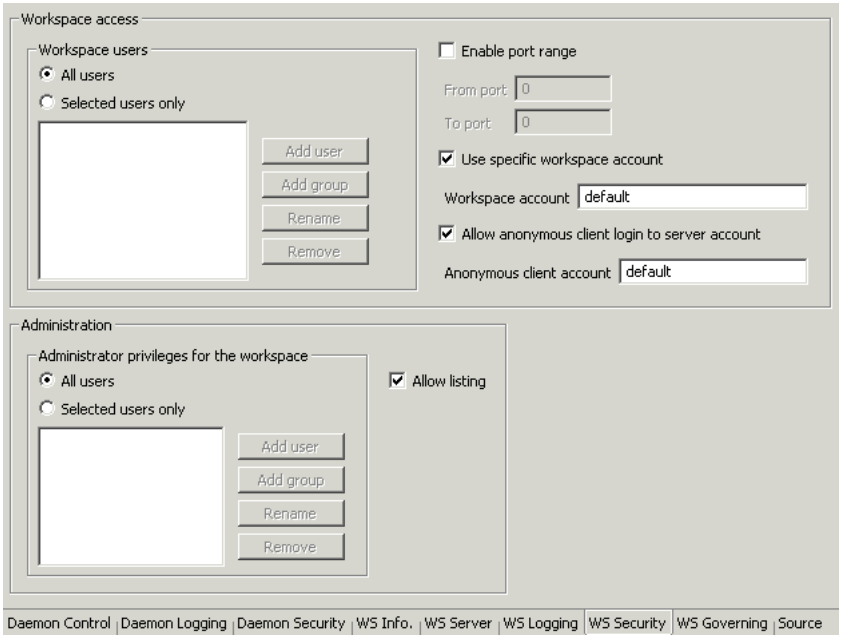
The **WS Security** section is accessed as follows:

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.

2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Right-click the machine and choose **Open Runtime Perspective**.
4. Drill-down the daemon to display the workspaces in the **Runtime Explorer**.
5. Right-click the workspace and choose **Edit Workspace Configuration**.
6. Choose the **WS Security** tab.
7. After making changes to the daemon, right-click the daemon and choose **Reload Configuration**.

Note: You can also change workspace settings in the using the **Configuration Explorer**, by selecting a machine and drilling-down the list until the daemon is displayed. Drill-down the daemon to list the workspaces. Right-click the workspace and choose **Edit Workspace**.

Changes made to the daemon configuration in the way are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.



The following fields are displayed:

Administration - Identifies the users (accounts) allowed to perform administrative tasks (tasks that require administrative login) on this workspace. For example, a user with administrative rights to a workspace can refresh the specific workspace servers using the `IRPCD` command with the `Refresh` *`workspace`* option.

Administrator privileges - Identifies the users (accounts) allowed to perform administrative tasks to the workspace (tasks that require administrative login).

All users - Anyone can access the workspace and change the settings.

Selected users only - The names of users (accounts) and groups that can be administrators.[2]

[2] The name is prefixed with '@', to utilize the operating system GROUP feature.

Note: If a user is not specified, the users specified in the `Workspace users` field (see below) have administrator rights for this workspace. In this case, if all users are allowed for the `Workspace users` field, all users have administrator rights for this workspace.

Allow Listing - Determines whether this workspace appears in the list of workspaces.

Workspace access - Defines the users (accounts) allowed to access this workspace.

Workspace users - Lists the users who are allowed to use the workspace (after logging on). If All users is specified, any user who has logged on to the daemon can use the workspace.

All users - Any user who has logged on to the daemon can use the workspace.

Selected users only - The names of users (accounts) and groups that can use the workspace.[3]

Note: If a user is not specified, any user who has logged on to the daemon can use the workspace.

Enable ports range - The firewall ports through which you access the workspace. Specifies 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 for VSAM can be accessed through a firewall.

Use specific workspace account - The operating system account used for the workspace. If not specified, the account name that was provided by the client is used.

Allow anonymous client login to server account - Whether this workspace can be invoked without authentication (user name/password). If anonymous login is allowed, specify the server account name to use. If this field is not specified, the value in the `Workspace account` field is used.

[3] The name is prefixed with '@', to utilize the operating system GROUP feature.

WS Governing

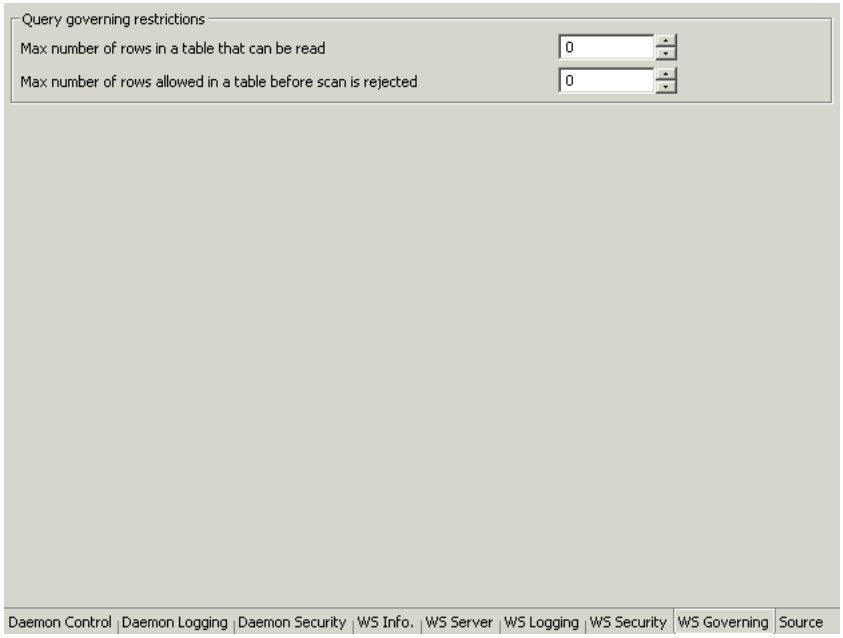
Using **WS Governing** you manage the way queries are executed. Since query governing parameters are defined at the workspace level, the restrictions apply to all queries.

The **WS Governing** section is accessed as follows:

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle > Studio > Studio**.
2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Right-click the machine and choose **Open Runtime Perspective**.
4. Drill-down the daemon to display the workspaces in the **Runtime Explorer**.
5. Right-click the workspace and choose **Edit Workspace Configuration**.
6. Choose the **WS Governing** tab.
7. After making changes to the daemon, right-click the daemon and choose **Reload Configuration**.

Note: You can also change workspace settings in the using the **Configuration Explorer**, by selecting a machine and drilling-down the list until the daemon is displayed. Drill-down the daemon to list the workspaces. Right-click the workspace and choose **Edit Workspace**.

Changes made to the daemon configuration in the way are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.



The following fields are displayed:

Max Number of Row in a Table That Can Be Read Parameter - Restricts the 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 Parameter - Restricts the number of table rows that can be scanned. This parameter impacts on the query booth during query optimization and execution.

During Query Optimization the value set is compared to the table cardinality. If the table cardinality is greater than the restriction the scan strategy is ignored as a possible strategy, unless it is the only available strategy.

During Query Execution a scan is limited to the value set. When the number of rows scanned exceeds the number stated, the query returns an error.

Note: You must refresh the daemon as well as reloading the configuration after changing the values in the WS Governing tab. The daemon configuration is reloaded by right-clicking the daemon in the Oracle Studio Runtime perspective and choosing Reload Configuration. Servers not in use are refreshed by right-clicking the daemon workspace in the Oracle Studio Runtime perspective and choosing End Unused Servers.

E

COBOL APIs to Applications

The Oracle Application Server Integration Adapter for VSAM includes a set of APIs that enable sending a native event as an inbound interaction from a COBOL program.

To use the APIs, you must include the following in the COBOL program:

- The `ACXAINIT` function - This function loads the API functions. The function does not have any parameters.
- Under CICS, link the COBOL program with the stub *`INSTROOT`*`.FIXLIB(ACX3GL)`, where *INSTROOT* is the high-level qualifier where Oracle Connect for VSAM is installed

This appendix describes the APIs and functions:

- Connection APIs
- Transaction APIs
- The Execute Function
- The Get Event Function
- The Ping Function
- The Get Error Function

> **See Also:** "Sending Inbound Interactions Using a CICS Transaction" on page 5-4 for details of how to use a CICS transaction that calls the APIs instead of using the APIs

Connection APIs

The following functions handle the connection and connection context for an inbound interaction:

- The Connect Function
- The Disconnect Function

There are two kinds of connections:

Transient Connection - Transient connections are created for use within a single inbound interaction. A transient connection is disconnected when an inbound interaction ends, or when the connection context changes (that is, with the `connect`, `setConnection` or `disconnect` functions).

Persistent Connection - Persistent connections can persist across multiple inbound interactions or connection context changes. Persistent connections are disconnected upon an explicit `disconnect` function or when a connection idle timeout expires.

The Connect Function

The `Connect` function establishes a new connection context. All the interactions defined take place within a connection context.

Upon a successful connect, a connection context is established and an implicit setConnection is performed with the newly created connection ID. A failed `connect` function leaves the inbound interaction with no connection context (that is, if a connection context was established prior to invoking the `connect` function, that connection context will no longer be in effect).

The function returns an integer of 1 (TRUE) or 0 (FALSE), to determine the success of the function.

Syntax

ACXACNCT using ServersUrls Username Password Workspace AdapterName Persistent IdleTimeout ConnectMode DefinitionFileName KeyName KeyValue ConnectHandle

Table E–1 ACXACNCT Parameters

Parameter	Usage	Description
ServersUrls	Input	The address of the server being connected to in the format: *server_name*:*port*. Or: *TCP/IP_address*:*port*. A series of servers, separated by commas, can be specified. The connection is made to the first server, in the list, that is up
Username	Input	The username required by the adapter
Password	Input	The user password required by the adapter
Workspace	Input	Optional. The name of the workspace where the adapter associated to the connection runs. The default workspace is Navigator
AdapterName	Input	The name of the adapter to connect to To supply the adapter schema as a parameter for the ACXACNCT call, build an ESDS VSAM file, which holds the adapter definition. Insert the schema into this VSAM file You can insert the schema into the VSAM file by creating a table with one column with a datatype of char X and use an INSERT INTO SQL statement to enter the data
Persistent	Input	Persistent connection indication. If *true*, a persistent connection is created. Otherwise a transient connection is created. The default is *false*.
IdleTimeout	Input	A per-connection client idle timeout setting (in seconds). If the client does not use the connection for the specified amount of time, the connection is disconnected by the server and its associated resources released. This setting is limited by the server side maximum idle connection timeout setting
ConnectMode	Input	The mode of the connection: **0** - Immediate connection **1** - Deferred connection - the connection is made when the execute function is performed
DefinitionFileName	Input	The name and path of the local definition used for the adapter. If omitted, the API queries the underlying adapter for the metadata This parameter should be specified, when the adapter is remote

Table E–1 (Cont.) ACXACNCT Parameters

Parameter	Usage	Description
KeyName	Input	The name of the encryption key
KeyValue	Input	The value associated with the encryption key
ConnectHandle	Output	A pointer to the connection. A pointer is always returned when the connection fails. This enables calling the `getError` function to determine what caused the error The `Disconnect` function must always be called to clear the connection handle

The Disconnect Function

The `Disconnect` function destroys the current connection context. All the resources associated with the current connection (persistent or transient) are released.

Syntax

ACXADSCO using ConnectHandle

Table E–2 ACXADSCO Parameters

Parameter	Usage	Description
ConnectHandle	Input	A pointer to the connection

Transaction APIs

Transaction APIs are used in the following scenarios:

Non-transacted operation - The adapter works in auto-commit mode. Work is committed immediately and automatically upon execution. This operation mode is the default operation mode when no transaction APIs are used, or when the `setAutoCommit` function sets auto-commit to true.

Local transaction operation - When auto-commit is set to false, the first interaction starts a transaction that lasts until an explicit commit (using the `transactionCommit` function) or an explicit rollback (using the `transactionRollback` function) occurs. All interactions performed in between are part of that transaction.

Note: 'local' is used here to indicate the scope of the transaction, rather than its location: the local transaction may be running on a remote machine.

The following functions handle transaction operations:

- The Set Autocommit Function
- The Transaction Commit Function
- The Transaction Rollback Function

The Set Autocommit Function

The `Set Autocommit` function sets the auto-commit mode of the connection.

Syntax

ACXASCMT using ConnectHandle AutoCommit

Table E–3 ACXASCMT Parameters

Parameter	Usage	Description
ConnectHandle	Input	A pointer to the connection.
AutoCommit	Input	New auto-commit mode of the connection. If set to *true*, each interaction immediately commits once executed. The auto-commit mode must be turned off if multiple interactions need to be grouped into a single transaction and committed or rolled back as a unit When auto-commit is reset and no global transaction is in progress, any interaction starts a local transaction. The client is required to use transactionCommit or transactionRollback at the appropriate time to commit or rollback the transaction The auto-commit mode is *true* by default and is reset if a distributed (global) transaction is started

The Transaction Commit Function

The `Transaction Commit` function commits the work done under the global or local transaction.

Syntax

ACXACMIT using ConnectHandle Xid

Table E–4 ACXACMIT Parameters

Parameter	Usage	Description
ConnectHandle	Input	A pointer to the connection.
Xid	Input	For internal use only. Supply an empty value

The Transaction Rollback Function

The `Transaction Rollback` function rolls back the work done under the (global) transaction.

Syntax

ACXARBCK using ConnectHandle Xid

Table E–5 ACXARBCK Parameters

Parameter	Usage	Description
ConnectHandle	Input	A pointer to the connection
Xid	Input	For internal use only. Supply an empty value

The Execute Function

The `Execute` function executes a given interaction against the application. The function supplies the input record that is required by the application.

Syntax

ACXAEXEC using ConnectHandle InterationName BufferIn BufferOut BufferOutLen

Table E–6 ACXAEXEC Parameters

Parameter	Usage	Description
ConnectHandle	Input	A pointer to the connection
InteractionName	Input	Name of interaction to execute
BufferIn	Input	A pointer to the input record
BufferOut	Output	For internal use only. Supply an empty value

Table E–6 (Cont.) ACXAEXEC Parameters

Parameter	Usage	Description
BufferOutLen	Output	For internal use only. Supply an empty value

The Get Event Function

The `Get Event` function determines the event to wait for and how long to wait. When an event is received, the function returns the results of performing the event.

Syntax

ACXAGTEV using ConnectHandle EventName Wait Keep BufferOut BufferOutLen

Table E–7 ACXAGTEV Parameters

Parameter	Usage	Description
ConnectHandle	Input	A pointer to the connection
EventName	Input	Name of event to wait for
Wait	Input	Length of time to wait to receive the event, in seconds
Keep	Input	Whether the event should be stored in the repository or deleted once finished. The default is false (to delete the event)
BufferOut	Output	A pointer to the output record
BufferOutLen	Output	The length of the output record

The Ping Function

The `Ping` function returns, in a pingResponse response, information about an active adapter.

Syntax

ACXAPING using ConnectHandle OutputStruct

Table E–8 ACXAPING Parameters

Parameter	Usage	Description
ConnectHandle	Input	A pointer to the connection
OutputStruct	Output	The return information describing the structure of the adapter

The Get Error Function

The `Get Error` function returns error information.

Syntax

ACXAGTER using ConnectHandle Error Status

Table E–9 ACXAGTER Parameters

Parameter	Usage	Description
ConnectHandle	Input	A pointer to the connection
Error	Output	The error message returned by the function
Status	Output	The status of the returned error

F

National Language Settings

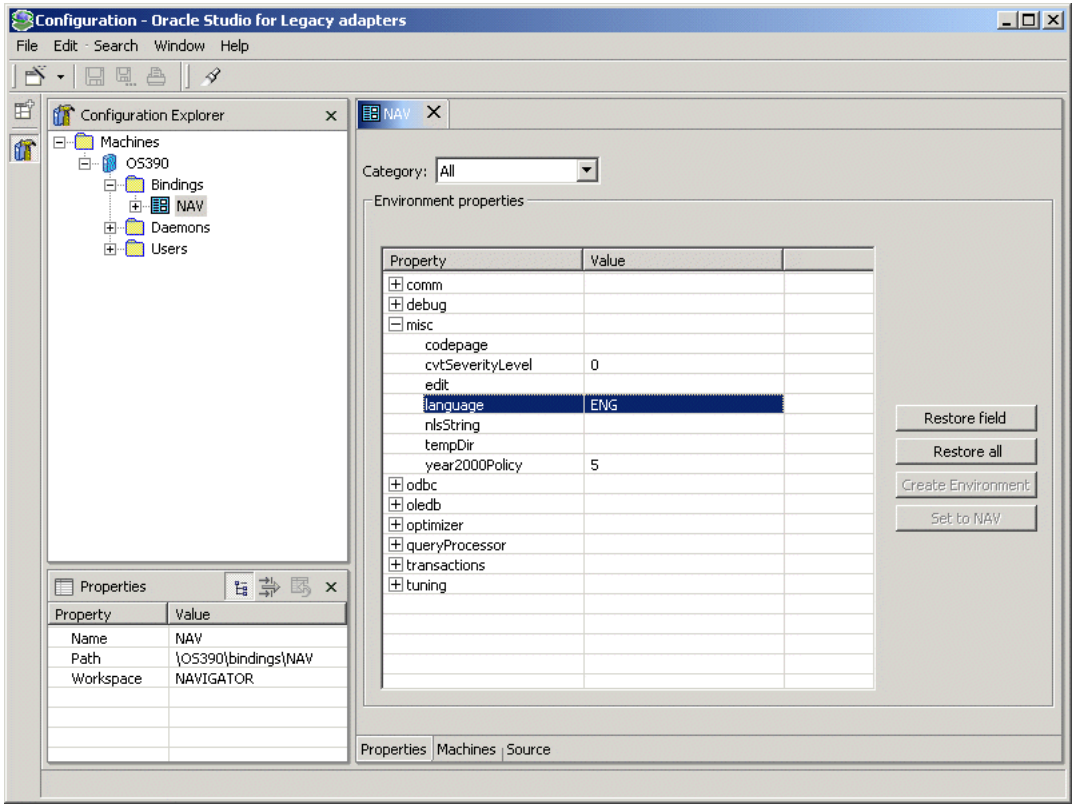
National Language Support (NLS) is provided by the Oracle Application Server Integration Adapter for VSAM for the following languages:

- Arabic
- English (the default)
- Hebrew
- Japanese
- Korean
- Simple Chinese
- Spanish
- Traditional Chinese

This appendix describes how to set up language support.

Defining the Language and Codepage

1. On the machine where Oracle Studio is installed, from the **Start** menu, choose **Start** > **Programs** > **Oracle** > **Studio** > **Studio**.
2. Select the machine defined in "Configuring Oracle Connect for VSAM" on page 2-19.
3. Right-click the **NAV** binding node and Choose **Edit Binding**.
4. In the **Binding Properties** tab, click the + next to the **misc** node.



5. Enter a valid value for the language. The following language codes are valid:

   ARA - Arabic

   ENG - English

   HEB - Hebrew

JPN - Japanese

KOR - Korean

SCHI - Simple Chinese

SPA - Spanish

TCHI - Traditional Chinese

6. Optionally, in the codepage field, specify the codepage required.

   You can skip this step, and just specify a language (see the previous step). In this case, a default codepage is used. The following shows the default codepages:

   ARA - AR8ISO8859P6

   ENG - EBCDIC

   HEB - IW8EBCDIC424

   JPN - JA1616DBCS

   KOR - KO16DBCS

   SCHI - ZHS16DBCS

   SPA - WE8EBCDICLATIN

   TCHI - ZHT16DBCS

   The following table lists the supported codepages according to language:

Table F–1 Supported Codepages

Language	Supported Codepage Values	Description
Arabic	AR8ISO8859P6	ISO 8859-6 Arabic (ASCII)
English	EBCDIC	-
English	D8EBCDIC273	EBCDIC codepage 273 8-bit Austrian German
English	DK8EBCDIC277	EBCDIC codepage 277 8-bit Danish
English	S8EBCDIC278	EBCDIC codepage 278 8-bit Swedish
English	I8EBCDIC280	EBCDIC codepage 280 8-bit Italian
English	WE8EBCDIC285	EBCDIC codepage 285 West European
English	F8EBCDIC297	EBCDIC codepage 297 8-bit French

Table F–1 (Cont.) Supported Codepages

Language	Supported Codepage Values	Description
English	WE8EBCDIC500	EBCDIC codepage 500 West European
Hebrew	IW8EBCDIC424	EBCDIC codepage 424 8-bit Latin/Hebrew (new EBCDIC)
Hebrew	IW8EBCDIC806	EBCDIC codepage 806 8-bit Latin/Hebrew (old EBCDIC)
Japanese	JA16DBCS	IBM EBCDIC 16-bit with Latin characters
Japanese	JA16EBCDIC930	IBM DBCS codepage 390 16-bit This codepage has no encoding for lowercase Latin letters. All resources should be defined in uppercase.
Korean	KO16DBCS	IBM EBCDIC 16-bit
Simple Chinese	ZHS16DBCS	IBM EBCDIC 16-bit Simple Chinese
Spanish	WE8EBCDICLATIN	EBCDIC codepage with Spanish extensions
Spanish	WE8EBCDIC284	EBCDIC codepage 284 8-bit Latin American/Spanish
Traditional Chinese	ZHT16DBCS	IBM EBCDIC 16-bit Traditional Chinese

Note: The Codepages for the European languages are based on English being chosen as the language in the Language field.

Index

A

acxTrace parameter, 7-7
analyzerQueryPlan parameter, 7-7
application language
  parameter, 7-8
arithmetic operators, C-2
array updates, 5-12
arrays
  flattening hierarchical data, 5-3, C-5
autocommit
  setting using setAutoCommit verb, E-5

B

buffer size
  comCacheBufferSize parameter, 7-7
Business Objects
  qualifying catalog and tables names, 7-10

C

cache
  max queries in, 7-10
cache memory
  dsmMaxBufferSize parameter, 7-11
CD-ROM drive requirements
  IBM S/390, 2-2
  PC, 2-3
CICS
  requirements, 2-3
COBOL, E-6
  execute function, E-6
COBOL API
  connect verb, E-2
  disconnect verb, E-4
  setAutoCommit verb, E-5
  transaction verbs, E-4
  transactionCommit verb, E-5
  transactionRollback, E-6
codepage parameter, 7-8
comCacheBufferSize parameter, 7-7
comm parameters
  comCacheBufferSize, 7-7
  comMaxSocketSize, 7-7
  comMaxXmlSize, 7-7
comMaxSocketSize parameter, 7-7
comMaxXmlSize parameter, 7-7
committing a transaction, E-5
comparison operators, C-2
compileAfterLoad parameter, 7-10
compiling procedures and views, 7-10
connect verb, E-2
connecting
  connect verb, E-2
  using COBOL, E-2
connection
  destroying with disconnect, E-4
CREATE PROCEDURE statement
  maximum procedures in cache memory, 7-11
cvtSeverityLevel parameter, 7-8

D

daemon
  logging, D-4
  security, D-7
  server modes, 7-3

shutting down, 6-6
starting, 6-5
timeout, 6-13
workspaces, 6-5
daemon configuration
logging section, D-4
security section, D-7
data source properties
disabling caching, 7-10
data type conversions
cvtSeverityLevel parameter, 7-8
data types
cvtSeverityLevel, 7-8
nlsString, 7-8
datatypes
metadata atomic datatypes, B-2, B-11
debug environment parameters, 7-7
debug parameters
acxTrace, 7-7
analyzerqueryPlan, 7-7
generalTrace, 7-7
logFile, 7-7
oledbTrace, 7-7
optimizerTrace, 7-7
queryWarnings, 7-7
traceDir, 7-7
disconnect verb, E-4
disk space requirements
IBM S/390, 2-2
PC, 2-3
distributed transactions
VSAM, 5-11
dsmMaxBufferSize parameter, 7-11
dsmMaxSortBufferSize parameter, 7-11
dsmMidBufferSize parameter, 7-11
dsoThreadingBoth parameter, 7-10

E

edit parameter, 7-8
emptyStringIsNull parameter
queryProcessor parameters
emptyStringIsNull, 7-10
environment parameters
acxTrace, 7-7
analyzerQueryPlan, 7-7
codepage, 7-8
comCacheBufferSize parameter, 7-7
comMaxSocketSize, 7-7
comMaxXmlSize, 7-7
cvtSeverityLevel, 7-8
debug, 7-7
dsmMaxBufferSize, 7-11
dsmMaxSortBufferSize, 7-11
dsmMidBufferSize, 7-11
dsoThreadingBoth, 7-10
edit, 7-8
emptyStringIsNull, 7-10
firstTreeExtensions, 7-10
forceQualifiedNames, 7-10
generalTrace, 7-7
goal, 7-9
hashBufferSize, 7-11
hashEnableParallelism, 7-11
hashMaxDiskSpace, 7-11
hashMaxOpenFiles, 7-12
ignoreSegmentBindFailure, 7-10
language, 7-8
logFile, 7-7
maxColumnsInParsing, 7-10
maxSqlCache, 7-10
miscellaneous, 7-7
nlsString, 7-8
noCommandReuse, 7-10
noCompilationCache, 7-10
noDSPropertyCache, 7-10
noHashJoin, 7-9
noInsertParameterization, 7-10
noMarkRowFailedFetch, 7-10
noMetadataCaching, 7-10
noParallelExecution, 7-10
noQueryParametrization, 7-10
noQueryReadAhead, 7-10
noSemiJoin, 7-9
noSessionPooling, 7-10
noSQSCache, 7-10
noThreadedReadAhead, 7-10
noThreads, 7-11
noTreeAutoExtend, 7-11
odbc, 7-9

oledb, 7-9
oledbTrace, 7-7
optimisticForUpdate, 7-11
optimizer, 7-9
optimizerTrace, 7-7
parserDepth, 7-11
preferredSite, 7-9
proceduresCacheSize, 7-11
promptDbUserPassword, 7-11
queryProcessor, 7-10
queryWarnings, 7-7
semiJoinInValuesFactor, 7-9
tempDir, 7-8
tokenSize, 7-11
traceDir, 7-7
transactions, 7-11
tuning, 7-11
useAlternatequalifier, 7-11
useTableFilterExpressions, 7-11
year2000Policy, 7-8
error log
logFile parameter, 7-7
execute function, E-6

F

first row optimization
goal parameter, 7-9
firstTreeExtensions parameter, 7-10
forceQualifiedNames parameter, 7-10

G

generalTrace parameter, 7-7
goal parameter, 7-9
Group functions, C-3

H

hardware requirements
IBM S/390 CD-ROM drive, 2-2
IBM S/390 disk space, 2-2
IBM S/390 memory, 2-2
IBM S/390 processor, 2-2
PC CD-ROM drive, 2-3
PC disk space, 2-3
PC memory, 2-3
PC processor, 2-3
hash disk space
hashMaxDiskSpace parameter, 7-11
hash joins
disabling, 7-9
hashMaxOpenFiles parameter, 7-12
hash memory
hashBufferSize parameter, 7-11
hashBufferSize parameter, 7-11
hashEnableParallelism parameter, 7-11
hashing
hashEnableParallelism parameter, 7-11
hashMaxDiskSpace parameter, 7-11
hashMaxOpenFiles parameter, 7-12
hierarchical queries
flattening hierarchical data, 5-3, C-5

I

IBM S/390
installation, 2-7
postinstallation, 2-12
preinstallation, 2-6
ignoreSegmentBindFailure parameter, 7-10
index cache
dsmMidBufferSize parameter, 7-11
INSERT statement
disabling parameterization of constants, 7-10
installation
IBM S/390, 2-7
installing
Windows, 2-19

J

joins
disabling hash joins, 7-9
hashBufferSize parameter, 7-11
hashEnableParallelism parameter, 7-11
hashMaxDiskSpace parameter, 7-11
hashMaxOpenFiles parameter, 7-12

K

known restrictions, 5-11

L

language parameter, 7-8
locking
  optimistic as default, 7-11
log files
  daemon options, D-4
  logFile parameter, 7-7
logFile parameter, 7-7
logging
  optimizer strategy, 7-7
  trace information, 7-7

M

maxColumnsInParsing parameter, 7-10
maximum size, 7-7
maxSqlCache parameter, 7-10
memory
  dsmMaxBufferSize parameter, 7-11
  dsmMaxSortBufferSize parameter, 7-11
memory for index cache
  dsmMidBufferSize parameter, 7-11
memory requirements
  IBM S/390, 2-2
  PC, 2-3
metadata
  atomic datatypes, B-2, B-11
  disabling retrieval from cache, 7-10
Microsoft software requirements, 2-4
miscellaneous environment parameters, 7-7
miscellaneous parameters
  codepage, 7-8
  cvtSeverityLevel, 7-8
  edit, 7-8
  language, 7-8
  nlsString, 7-8
  tempDir, 7-8
  year2000Policy, 7-8
multithreading
  disabling, 7-11

N

NAV_UTIL
  text editor, 7-8
nlsString parameter, 7-8
noCommandReuse parameter, 7-10
noCompilationCache parameter, 7-10
noDSPropertyCache parameter, 7-10
noHashJoin parameter, 7-9
noInsertParameterization parameter, 7-10
noMarkRowFailedFetch parameter, 7-10
noMetadataCaching parameter, 7-10
noParallelExecution parameter, 7-10
noQueryParametrization parameter, 7-10
noQueryReadAhead parameter, 7-10
noSemiJoin parameter, 7-9
noSessionPooling parameter, 7-10
noSQSCache parameter, 7-10
noThreadedReadAhead parameter, 7-10
noThreads parameter, 7-11
noTreeAutoExtend parameter, 7-11

O

odbc environment parameters, 7-9
oledb environment parameters, 7-9
oledbTrace parameter, 7-7
operating system requirements
  IBM S/390, 2-3
  PC, 2-4
optimisticForUpdate parameter, 7-11
optimizer environment parameters, 7-9
optimizer files
  traceDir parameter, 7-7
optimizer parameters
  goal, 7-9
  noHashJoin, 7-9
  noSemiJoin, 7-9
  preferredSite, 7-9
  semiJoinInValuesFactor, 7-9
optimizer plan
  writing to file, 7-7
optimizerTrace parameter, 7-7
Oracle Application Server ProcessConnect
    requirements, 2-3

P

parallel processing
  disabling, 7-10
parserDepth parameter, 7-11
password
  setting automatic prompt for, 7-11
pooling
  disabling, 7-10
postinstallation
  IBM S/390, 2-12
preferredSite parameter, 7-9
preinstallation
  IBM S/390, 2-6
procedures
  columns
    maximum referenced in query, 7-10
  maximum in cache memory, 7-11
  reading from disk, 7-10
  setting to compile, 7-10
proceduresCacheSize parameter, 7-11
processor requirements
  IBM S/390, 2-2
  PC, 2-3
promptDbUserPassword parameter, 7-11

Q

queries
  disabling parallel processing, 7-10
  disabling parameterization of constants, 7-10
  disabling read-ahead functionality, 7-10
  disabling reuse, 7-10
  disabling saved compilation, 7-10
  enabling retrieval after failure, 7-10
  maximum cached in memory, 7-10
  maximum length of string, 7-11
  maximum size after compilation, 7-10
query optimization
  goal parameter, 7-9
query optimizer
  logging strategy, 7-7
  writing plan to file, 7-7
query optimizer files
  traceDir parameter, 7-7
queryProcessor environment parameters, 7-10
queryProcessor parameters
  compileAfterLoad
    environment parameters
      compileAfterLoad, 7-10
  dsoThreadingBoth, 7-10
  firstTreeExtensions, 7-10
  forceQualifiedNames, 7-10
  ignoreSegmentBindFailure, 7-10
  maxColumnsInParsing, 7-10
  maxSqlCache, 7-10
  noCommandReuse, 7-10
  noCompilationCache, 7-10
  noDSPropertyCache, 7-10
  noInsertParameterization, 7-10
  noMarkRowFailedFetch, 7-10
  noMetadataCaching, 7-10
  noParallelExecution, 7-10
  noQueryParametrization, 7-10
  noQueryReadAhead, 7-10
  noSessionPooling, 7-10
  noSQSCache, 7-10
  noThreadedReadAhead, 7-10
  noThreads, 7-11
  noTreeAutoExtend, 7-11
  optimisticForUpdate, 7-11
  parserDepth, 7-11
  proceduresCacheSize, 7-11
  promptDbUserPassword, 7-11
  tokenSize, 7-11
  useAlternatequalifier, 7-11
  useTableFilterExpressions, 7-11
queryWarnings parameter, 7-7

R

read-ahead functionality
  disabling, 7-10
reading from disk, 7-10
requirements
  IBM S/390 hardware requirements, 2-2
  IBM S/390 software requirements, 2-2
  PC hardware requirements, 2-3
  PC software requirements, 2-4
rolling back transactions, E-6

row optimization
goal parameter, 7-9

S

security information
setting automatic prompt for, 7-11
segmented data sources
response upon failure, 7-10
semiJoinInValuesFactor parameter, 7-9
server modes
configuring, 7-3
servers
reusable, 7-4
Reuse limit daemon parameter, D-14
ReuseLimit daemon parameter, 7-4
sessions
disabling pooling, 7-10
setAutoCommit verb, E-5
SMS
system managed storage, 2-6
sockets
comMaxSocketSize parameter, 7-7
software
Microsoft products, 2-4
software requirements
CICS, 2-3
IBM S/390 operating system, 2-3
Microsoft, 2-4
Oracle Application Server ProcessConnect, 2-3
PC operating system, 2-4
sort buffer
dsmMaxSortBufferSize parameter, 7-11
SQL
flattening hierarchical data, 5-3, C-5
stored queries
maximum in cache memory, 7-11
string functions, C-3
system managed storage
SMS, 2-6

T

tempDir parameter, 7-8
temporary files
tempDir parameter, 7-8
threading model, 7-10
timeout
client idle, 6-13
tokenSize parameter, 7-11
TP Monitor requirements
CICS, 2-3
trace information
logging, 7-7
traceDir parameter, 7-7
transaction verbs, E-4
transactionCommit verb, E-5
transactionRollback verb, E-6
transactions
committing with transactionCommit, E-5
rolling back with transactionRollback, E-6
transactions environment parameters, 7-11
tuning
preferredSite parameter, 7-9
tuning environment parameters, 7-11
tuning parameters
dsmMaxBufferSize, 7-11
dsmMaxSortBufferSize, 7-11
dsmMidBufferSize, 7-11
hashBufferSize, 7-11
hashenableParallelism, 7-11
hashMaxDiskSpace, 7-11
hashMaxOpenFiles, 7-12

U

useAlternatequalifier parameter, 7-11
useTableFilterExpressions parameter, 7-11

V

verbs
connection, E-2
views, 7-10
setting to compile, 7-10
VSAM
setting up, 3-2
VSAM driver, 5-11
distributed transactions, 5-11

W

Windows
  installing, 2-19

X

XML documents, 7-7
  maximum size, 7-7

Y

Y2K
  year2000Policy parameter, 7-8