

Oracle® Application Server

Adapter for CICS User's Guide

10g Release 3 (10.1.3.1.0)

B31001-01

January 2007

Copyright © 2006, 2007, Oracle. All rights reserved.

Primary Author: Jeanne Wiegelmann

Contributing Authors: Yishai Hadas, Dror Harari, Adeeb Massad, Meera Srinivasan, Bo Stern, Shashi Suravarapu, Costi Zaboura, Sheela Vasudevan, Marian Jones

The Programs (which include both the software and documentation) contain proprietary information; 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. This document is not warranted to be 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.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 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 we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface	vii
Audience.....	vii
Documentation Accessibility	vii
Related Documents	viii
Conventions	viii
 1 Introduction	
OracleAS Adapters for CICS Overview.....	1-1
OracleAS Adapter for CICS.....	1-1
OracleAS Adapter for CICS Queue	1-2
CICS Transient Data Queues (Intrapartition)	1-2
OracleAS Adapters for CICS Architecture.....	1-2
Integration Flow from Oracle Application Server to the Legacy Application	1-3
 2 Installing and Configuring OracleAS Adapters for CICS	
Preinstallation Tasks.....	2-1
IBM OS/390 or z/OS Hardware and Software Requirements.....	2-1
Hardware Requirements.....	2-1
Software Requirements	2-2
Windows Hardware and Software Requirements	2-2
Hardware Requirements	2-2
Software Requirements	2-3
Installing Oracle Connect on an IBM OS/390 or z/OS Series Platform.....	2-3
Installation Worksheet.....	2-3
Preinstallation Instructions.....	2-4
Installing the Kit.....	2-5
Installation Instructions.....	2-5
Postinstallation Instructions	2-8
Postinstallation Procedures	2-9
Starting the Daemon.....	2-10
Setting Up Oracle Connect for Reentrancy	2-11
Updating an Existing Oracle Connect Installation with CICS.....	2-11
Installing Oracle Studio.....	2-13
Installing Oracle Studio from the CD-ROM.....	2-13
Configuring Oracle Connect	2-13

Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio	2-14
Securing Access to Oracle Connect	2-14
Setting Password Access to Oracle Studio	2-15
Specifying Users with Administrative Rights	2-15
Setting Up Run-Time User Access to the IBM OS/390 or z/OS Platform	2-17
Modeling Interactions for OracleAS Adapter for CICS.....	2-18
Configuring an Oracle Connect Adapter	2-18
Generating Outbound Interactions	2-20
Generating Inbound Interactions	2-25
Modeling Interactions for OracleAS Adapter for CICS Queue.....	2-31
Preliminary Considerations.....	2-31
Configuring the CICS Queue Adapter	2-36
Generating Outbound Interactions	2-37
Viewing the XML Schema.....	2-41
Creating XML Schemas	2-41

3 Integrating OracleAS Adapters for CICS with OC4J

Overview of Integrating OracleAS Adapters for CICS with OC4J	3-1
Integrating the J2CA 1.5 CICS Adapter for Outbound.....	3-2
Configuring the J2CA 1.5 CICS Adapter for Outbound.....	3-2
Configuring Multiple Adapters	3-4
Updating Configuration Information	3-4
Using the CCI API to Develop Applications.....	3-4
Integrating the J2CA 1.5 CICS Queue Adapter for Inbound.....	3-5
Configuring the J2CA 1.5 CICS Queue Adapter for Inbound	3-5
Using the CCI API to Develop Message Endpoint Applications.....	3-9

4

Integrating OracleAS Adapters for CICS with Oracle BPEL Process Manager

Overview of Integrating OracleAS Adapters for CICS with Oracle BPEL Process Manager ...	4-1
Configuring Oracle BPEL Process Manager to interact with the OracleAS Adapter for CICS	4-2
Setting up the Connection to the Oracle Connect Server	4-2
Checking Metadata Availability Using Oracle JDeveloper	4-2
Configuring the WSDL for Outbound Applications	4-3
Configuring the WSDL for Inbound Applications.....	4-4

5 Advanced Features of OracleAS Adapters for CICS

Configuring the Daemon for High Availability	5-1
Adding a New Daemon Workspace Configuration.....	5-1
Editing the Workspace	5-2
Configuring the Server Mode.....	5-3
Configuring a Binding Environment	5-5
comm Category	5-6
debug Category	5-6
miscellaneous Category	5-7
odbc Category	5-8

oledb Category	5-8
optimizer Category	5-8
queryProcessor Category	5-8
transactions Category	5-9
tuning Category	5-9
Migration Considerations	5-9
Security Considerations	5-10
Setting Design Time Security	5-10
Setting Run-time Security	5-11
Transaction Support	5-11

6 Troubleshooting OracleAS Adapter for CICS

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

A Advanced Tuning of the Daemon

Daemon Control	A-1
Daemon Logging	A-3
Daemon Security	A-5
Workspaces	A-7
WS Info.	A-7
WS Server	A-9
WS Logging	A-13
WS Security	A-16

B Globalization Settings

Defining the Language and Codepage	B-1
--	-----

C Back-end Adapter Data Type Support

Data Type Mapping	C-1
-------------------------	-----

Index

Preface

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

This document describes the features of OracleAS Adapters for CICS that apply to the UNIX, Windows 2000, Windows XP, and Windows Server 2003 operating systems.

This preface covers the following topics:

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

Audience

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

- Installing and configuring OracleAS Adapters for CICS
- Diagnosing errors
- Using OracleAS to access CICS transactions

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

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. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

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

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

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Introduction

Oracle Application Server (AS) Adapters for CICS enable you to connect Oracle Application Server to a Customer Information Control System (CICS).

This section provides an overview of the features and architecture of OracleAS Adapters for CICS.

This section contains the following topics:

- [OracleAS Adapters for CICS Overview](#)
- [OracleAS Adapters for CICS Architecture](#)

OracleAS Adapters for CICS Overview

OracleAS Adapters for CICS include the following adapters:

- [OracleAS Adapter for CICS](#)
- [OracleAS Adapter for CICS Queue](#)

CICS is a family of application servers that provides industrial-strength, online transaction management and connectivity for mission-critical applications on IBM mainframe systems.

The application components in CICS are executable programs called transactions. These programs are typically written in COBOL but may possibly be written in other languages, such as PL/I, C, and Java. OracleAS Adapters for CICS interaction are mapped to specific programs.

OracleAS Adapter for CICS

OracleAS Adapters for CICS includes the following features:

- Uses COMMAREA transactions to communicate with clients, using a communication buffer described in the transaction program (such as a COBOL copybook). There is an input structure and an output structure (each with a limit of 32Kb). In many cases, transactions are written so that the input structure prefixes the output structure.
- Uses EXCI (EXternal Call Interface) to invoke CICS transactions from tasks running within the IBM OS/390 or z/OS system. EXCI is efficient because it uses cross-memory mechanisms.
- Enables storing valid CICS user names and passwords, which are passed to CICS for authentication as part of the client request.

- Captures and maintains a metadata schema for CICS by importing COBOL copybooks and transforming them into mapping definitions for Oracle Connect on the IBM OS/390 or z/OS Series platform and by associating the data structures with specific physical files.
- Uses an enterprise application integration (EAI) model. Users of a requesting application can model the interactions that they want implemented, specifying what each interaction does, when it occurs, and the inputs and outputs expected for each interaction.
- Maps data structures to facilitate access to CICS programs within Oracle Application Server.
- Supports two-phase commit and can fully participate in a distributed transaction.

Note: To use OracleAS Adapter for CICS with 2PC, you must install and configure RRS and also install CICS TS 1.3 or higher.

OracleAS Adapter for CICS Queue

OracleAS adapter for CICS Queue is used for pulling messages from a CICS transient data queue.

To pull message from the queue, the `getEvents` interaction is executed. This interaction is added to the adapter schema automatically. When executed, all messages in the queue are pulled.

CICS Transient Data Queues (Intrapartition)

Intrapartition refers to data on direct-access storage devices used with one or more programs running as separate tasks. Data directed to or from these internal queues must consist of variable-length records. All intrapartition transient data destinations are held as queues in the same VSAM data set, managed by CICS.

An intrapartition destination requires a resource definition containing information that locates the queue in the intrapartition data set. These queues can be associated with either a terminal or an output data set. When data is written to the queue by a user task, the queue can be used subsequently as input data by other tasks within the CICS region.

All access is sequential, governed by read and write pointers. Once a record has been read, it cannot be read by another task. Intrapartition data may ultimately be transmitted upon request to the terminal or retrieved sequentially from the output data set.

The intrapartition transient data queue is of the Logically Recoverable type. This data queue type is recovered on warm and emergency restarts. If a unit of work (UOW) updates a logically recoverable queue, and subsequently backs out the changes it has made, the changes made to the queue are also backed out. On a warm or emergency restart, the committed state of a logically recoverable intrapartition queue is recovered.

OracleAS Adapters for CICS Architecture

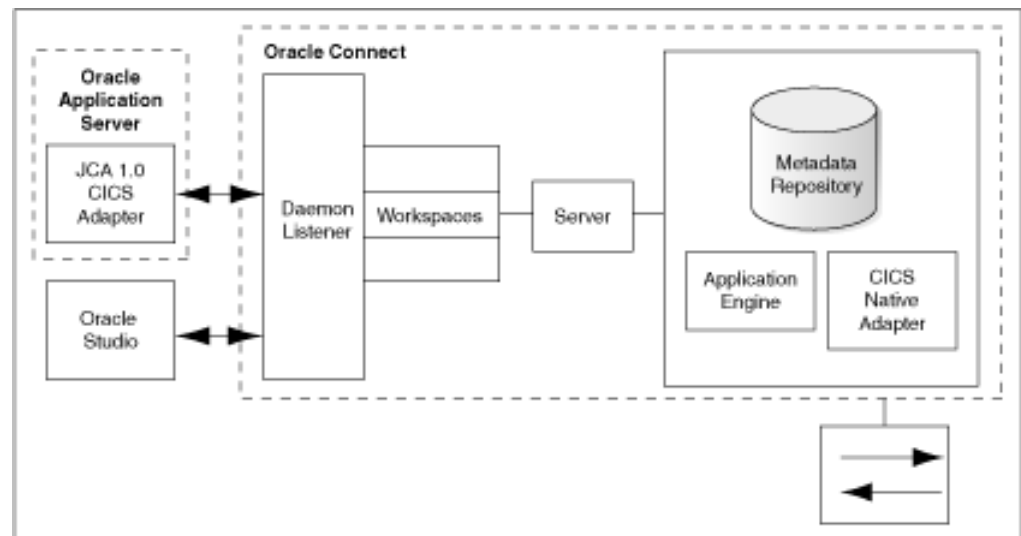
OracleAS adapters for CICS include the following components:

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

- Oracle Connect: Oracle Connect runs on the legacy system and handles requests from the J2CA 1.5 CICS adapter, that runs within Oracle Application Server Containers for J2EE (OC4J).
- Oracle Studio: Oracle Studio is the configuration tool for Oracle Connect. Configuration tasks using Oracle Studio are performed on a Windows platform. Oracle Studio uses perspectives that enables you to generate specific information necessary to model OracleAS Adapters for CICS.

Figure 1–1 illustrates the components of OracleAS Adapter for CICS:

Figure 1–1 OracleAS Adapter for CICS Architecture



See Also: *Oracle Application Server Adapter Concepts Guide*

Integration Flow from Oracle Application Server to the Legacy Application

The J2CA 1.5 CICS adapter converts the J2CA interaction invocation received from Oracle Application Server to XML format and passes the XML format to Oracle Connect on the legacy server. The daemon listens for the request coming from the J2CA 1.5 CICS adapter client and assigns a server process to handle the request.

The properties of the server process such as connection pooling requirements are defined by a workspace definition within the daemon. The server process includes an instance of the application engine, which converts the XML format into native structures understandable by CICS and passes the converted XML to the back-end adapter. The back-end adapter builds an interaction based on the metadata for the back-end adapter stored in the repository and the incoming converted XML, and passes it to the legacy application. The results of this execution are passed back to the application engine, using the back-end adapter, where these results are converted to XML and passed back to the client.

Installing and Configuring OracleAS Adapters for CICS

This section describes how to install Oracle Connect and Oracle Studio from the CD-ROM and how to configure Oracle Connect using Oracle Studio.

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

This section includes the following topics:

- [Preinstallation Tasks](#)
- [Installing Oracle Connect on an IBM OS/390 or z/OS Series Platform](#)
- [Updating an Existing Oracle Connect Installation with CICS](#)
- [Installing Oracle Studio](#)
- [Configuring Oracle Connect](#)

Preinstallation Tasks

Before installing OracleAS Adapters for CICS, ensure that your computer meets the following requirements:

- [IBM OS/390 or z/OS Hardware and Software Requirements](#)
- [Windows Hardware and Software Requirements](#)

IBM OS/390 or z/OS Hardware and Software Requirements

This section describes the following requirements for installing Oracle Connect on an IBM OS/390 or z/OS platform:

- [Hardware Requirements](#)
- [Software Requirements](#)

Hardware Requirements

The following table summarizes the hardware requirements for Oracle Connect.

Table 2–1 Oracle Connect Hardware Requirements

Hardware Component	Requirements
Processor	An IBM S/390 computer
Memory	The minimum requirement is 4 MB for each connection. A connection is defined as a connection to a server process or daemon. The actual memory requirement depends on such things as the size of the database and the number of databases accessed.
CD-ROM Drive	An internal or external CD-ROM drive
Disk Space (3380 and 3390 disks)	150 cylinders

Software Requirements

The following table summarizes the software requirements for Oracle Connect.

Table 2–2 Oracle Connect Software Requirements

Software Component	Requirements
Operating System	IBM OS/390 V2R5 or higher Or IBM z/OS Series V1R0 or higher
CICS TP Monitor	V4R1 or higher (recommended to use CICS V6R1 or higher) CICS EXCI support must be installed and IRCSTRT=YES must be specified in the CICS initialization parameters, so that the IRC (Inter Region Communication) starts. You can also set the IRC to open by issuing the following command: CEMT SET IRC OPEN. In addition, the IBM group DFH\$EXCI (or an equivalent user-defined group) must be installed in the CICS region using the CEDA RDO facility.
Oracle Application Server	Oracle Application Server 10g (10.1.3)

Windows Hardware and Software Requirements

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:

Table 2–3 Oracle Studio Hardware Requirements

Hardware Component	Requirements
Processor	An Intel or 100% compatible personal computer (PC), based on a Pentium processor
Memory	256 MB RAM
CD-ROM Drive	An internal or external CD-ROM drive
Disk Space for Oracle Studio	100 MB free disk space

Software Requirements

The following table summarizes the software requirements for Oracle Studio.

Table 2–4 Oracle Studio Software Requirements

Software Component	Requirements
Operating System	Microsoft Windows 2000 with service pack 2 or higher, or Microsoft Windows XP, or Microsoft Windows 2003
Microsoft	Network transport protocol software, TCP/IP, included with Microsoft Windows

Installing Oracle Connect on an IBM OS/390 or z/OS Series Platform

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

- [Installation Worksheet](#)
- [Preinstallation Instructions](#)
- [Installing the Kit](#)
- [Installation Instructions](#)
- [Postinstallation Instructions](#)

Note: If you have an Oracle Connect back-end adapter already installed on the IBM OS/390 or z/OS platform, then follow the instructions described in "[Updating an Existing Oracle Connect Installation with CICS](#)" on page 2-11.

The other back-end adapters that run on an IBM OS/390 or z/OS platform are:

- Oracle Application Server Adapter for IMS/DB
 - Oracle Application Server Adapter for IMS/TM
 - Oracle Application Server Adapter for VSAM
-

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 2–5 Preinstallation Information

Topic	Required Information	Default	Comment
General	Operating system	-	OS/390 V2R5 or higher or z/OS Series V1R0 or higher
-	Disk space	-	150 cylinders
-	Memory	-	The minimum requirement is 4MB for each connection. A connection is defined as a connection to a server process or daemon. The actual memory requirement depends on such things as the size of the database and the number of databases accessed.

Table 2–5 (Cont.) Preinstallation Information

Topic	Required Information	Default	Comment
-	Installation high-level qualifier	OCL1012	-
-	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	-

Table 2–6 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
Permission to update the security manager, such as RACF
Optionally, permission to specify an output class for Oracle Connect output

Table 2–7 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
Copy 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, ensure that the following information is available:

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

Before starting the installation, ensure 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.
- Permission to update the security manager, such as RACF

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

Oracle Connect for the IBM OS/390 or z/OS platform is contained in the following datasets:

- `OCL1012.TRANSMIT.KIT`
- `OCL1012.TRANSMIT.LOAD`

These datasets are provided on a CD-ROM in the following directory:

`Oracle_Connect\CICS_Legacy_Adapter`

Installing the Kit

Perform the following steps on the Mainframe:

1. Run the following command:

```
tso profile noprefix
```

The user name will not be used as part of the dataset name. On some systems this is the default.

2. Allocate datasets with the following space for each of these files:

- `OCL1012.TRANSMIT.KIT` = 130 tracks (3380 and 3390 disks)
- `OCL1012.TRANSMIT.LOAD` = 420 tracks (3380 and 3390 disks)

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

3. Using FTP, copy `OCL1012.TRANSMIT.KIT` and `OCL1012.TRANSMIT.LOAD` in the binary mode from the installation CD to the mainframe. You can replace the `OCL1012` high-level qualifier to any qualifier you want.

Installation Instructions

Perform the following steps to install Oracle Connect:

1. Run the following command at the TSO prompt:

```
RECEIVE INDSNAME('nnn.TRANSMIT.KIT')
```

Where `nnn` represents the high-level qualifier you want to assign for the Oracle Connect installation. Assign the high-level qualifier you specified in step 7 of the preinstallation procedure. The default value is `OCL1012`.

Note: You can use more than one high-level qualifier, such as `ACME.DEV.OCL1012`, 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 then the library is extracted to the current unit and volume.

3. Run the `PREPARE` member of the `nn.TRANSMIT.LIB` library:

```
ex PREPARE
```

Follow the instructions in the Response column in [Table 2–8](#) for each entry in the Screen column.

Table 2–8 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, then 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, then 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
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, then it must be entered as it will appear in the job. You can enter up to six lines. Enter a blank card to end input. If you do not enter a card, then the Oracle Connect default card is used.
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, before it is submitted, then 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, run the `NAVINST` member, as shown:

ex `NAVINST`

Follow the instructions in the Response column in [Table 2–9](#) for each entry in the Screen column.

Table 2–9 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, then answer Y, otherwise answer N.
THE SOFTWARE WILL BE INSTALLED UNDER THE HIGH LEVEL QUALIFIER THAT YOU WILL CHOOSE.	The high-level qualifier for the installation (referred to as <i>INSTROOT</i> throughout this guide)
ENTER THE HIGH LEVEL QUALIFIER ["QUALIFIER"]:	You can use more than one high-level qualifier (such as <code>ACME.DEV.VA10</code>). The total length must be less than or equal to twenty characters. The qualifiers can be the same as the ones used for the installation (step 1). The words <code>transmit</code> and <code>buildkit</code> 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 output. Assigning a device which is set on <code>HOLD</code> 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, then it must be entered as it will appear in the job. You can enter up to six lines. Enter a blank card to end input. If you do not enter a card, then the Oracle Connect default card is used.

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

Screen	Response
ADDING AND UPDATING ORACLE CONNECT FOR CICS 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.	In order to manage Oracle Connect on this computer from Oracle Studio, you need to enter a user account of a user who will have administrative authorization, or press Enter to enable any user to administer Oracle Connect on this computer. The administrative rights can be changed from within Oracle Studio after the installation.
ENTER A VALID USER NAME FOR AN ORACLE CONNECT ADMINISTRATOR [ALL]:	
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, before it is submitted, then 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, run the following command:

```
ex CICS
```

Follow the instructions in the Response column in [Table 2–10](#) for each entry in the Screen column.

Table 2–10 CICS Adapter-Specific Installation Prompts and Responses

Screen	Response
ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHEXCI]:	Enter the CICS EXCI load library name only if you do not want the default
PLEASE CONFIRM (YES/NO/QUIT) [YES]:	Confirm the entered details
ENTER THE ISPF LOAD LIBRARY NAME [ISP.SISPLD] :	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, then it must be entered as it will appear in the job. You can enter up to six lines. Enter a blank card to end input. If you do not enter a card, then the Oracle Connect default card is used.

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:

- [Postinstallation Procedures](#)
- [Starting the Daemon](#)

- [Setting Up Oracle Connect for Reentrancy](#)

Postinstallation Procedures

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

- 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 is installed.

- Using FTP, copy the BRANDBIN file, in the binary mode, from the Oracle Connect\CICS 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 following command:

```
"/setprog apf,add,dsn=INSTROOT.loadaut,volume=vol1002"
```

Where vol1002 is the volume where you installed Oracle Connect and *INSTROOT* is the high-level qualifier where Oracle Connect is installed.

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

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

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

- Move the *INSTROOT*.USERLIB(ATTTDAEMN) and *INSTROOT*.USERLIB(ATTSSVR) members to any active proclib, such as user.proclib. The ATTTDAEMN and ATTSSVR members are run as started tasks.

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

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

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

- Exit and save the changes.
- Change the following line in the ATTTDAEMN script to include the IP address and port of the IBM OS/390 or z/OS platform.

For example, before:

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

After:

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

Where *ip_address* is ip address of the computer, 2552 is the default port for starting the daemon and *IRPCDINI* is the default daemon configuration.

- The *ATTDAEMN* and *ATTSRVR* started tasks need permission to use an Open Edition TCP/IP stack. The owner must be a user with OMVS segment defined and OMVS UID=0000000000.
- In the security manager, such as RACF, define *ATTDAEMN* and *ATTSRVR* with a started task class and a general profile that enables the following:
 - Permission to issue master console commands.
 - START authority for the *ATTSRVR* job.
 - Access to an Open OS/390 segment, which defines access to TCP/IP OA sockets.
 - ALTER authority on datasets under *INSTROOT* to access to read, write, allocate, and delete datasets under *INSTROOT*.
- The installation includes a PS, *INSTROOT.DEF.GBLPARMS*, which contains global environment information. This PS is read at startup and the correct software version is used, based on the details provided in the startup task.

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

- *ATTSRVR*: located in an active proclib, such as *user.proclib*
- *ATTDAEMN*: located in an active proclib, such as *user.proclib*
- *NAVSQL*: located in *INSTROOT.USERLIB*
- The input during the installation procedure is written to *nnn.BUILDKIT.SRC(PARS)*. You can use this file to provide the same inputs if you rerun the installation, where *nnn* is the high-level qualifier you assign for the installation.
- For information about specifying Oracle Connect 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'
```

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

To submit the daemon as a job, uncomment the first two lines of the *ATTDAEMN JCL*, change the PARM line as described earlier, and run the job using the subcommand. 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='-KATTDAMN START 'INSTROOT.DEF.IRPCDINI''
//STEPLIB DD DSN=INSTROOT.LOADAUT,DISP=SHR
//SYSPRINT DD SYSOUT=A
//GBLPARMS DD DSN=INSTROOT.DEF.GBLPARMS,DISP=SHR
//SYSDUMP DD DUMMY
```

Setting Up Oracle Connect for Reentrancy

All Oracle Connect load modules are reentrant to enable subtasking. Therefore, move *INSTROOT.LOAD* to the Link Pack Area (LPA).

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

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

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

- Place the *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.

Updating an Existing Oracle Connect Installation with CICS

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

Table 2–11 Preinstallation Information

Topic	Required Information	Default	Comment
CICS	CICS EXCI load library name	CICS.CICS.SDFHEXCL	-

Table 2–12 Required Permissions

Permission
Permission to read the CICS EXCI library

In the *nnn.BUILDKIT.SRC* library, run the *CUSTOCL* member, as shown:

```
ex CUSTOCL
```

Follow the instructions in the Response column in [Table 2–13](#) for each entry in the Screen column.

Table 2–13 CICS 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, then 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, then enter the library where Program Specification Blocks (PSBs) are located.
ENTER YOUR PSB NAME [HOSPPSB] :	If you responded YES to working with IMS/DB, then enter the name of the PSB file to use.
PLEASE CONFIRM (YES/NO/QUIT) [YES] :	If you responded YES to working with IMS/DB, then 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 OracleAS Integration Adapter for IMS/DB, then respond YES.
ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHEXCI] :	If you responded YES to working with IMS/DB under CICS, then 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, then 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
ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHEXCI] :	Enter the CICS EXCI load library name only if you do not want the default.
PLEASE CONFIRM (YES/NO/QUIT) [YES] :	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 have Oracle Connect for VSAM already installed and you want to access VSAM data under CICS.
ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHEXCI] :	If you responded YES to working with VSAM under CICS, then enter the CICS EXCI load library name only if you do not want the default.
PLEASE CONFIRM (YES/NO/QUIT) [YES] :	If you responded YES to working with VSAM under CICS, then 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, then it must be entered as it will appear in the job. You can enter up to six lines. Enter a blank card to end input. If you do not enter a card, then the Oracle Connect default card is used.

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 2-8, as required.

Installing Oracle Studio

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

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

The other legacy adapters are:

- OracleAS Adapters for Tuxedo
 - OracleAS Adapter for IMS/DB
 - OracleAS Adapter for IMS/TM
 - OracleASAdapter for VSAM
-
-

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 Oracle Studio from the CD-ROM by running the self-extracting executable installation file, *OSL904-win32.exe*.

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

Configuring Oracle Connect

All modeling of Oracle Connect is performed using Oracle Studio. To use Oracle Studio, you first configure it to enable access to the IBM OS/390 or z/OS platform where CICS runs.

To configure Oracle Connect, refer to the following sections:

- [Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio](#)
- [Securing Access to Oracle Connect](#)
- [Modeling Interactions for OracleAS Adapter for CICS](#)
- [Modeling Interactions for OracleAS Adapter for CICS Queue](#)
- [Viewing the XML Schema](#)
- [Creating XML Schemas](#)

Note: The following tasks assume you have permission to access the IBM OS/390 or z/OS platform and that the Oracle Connect daemon is running on this computer.

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

Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio

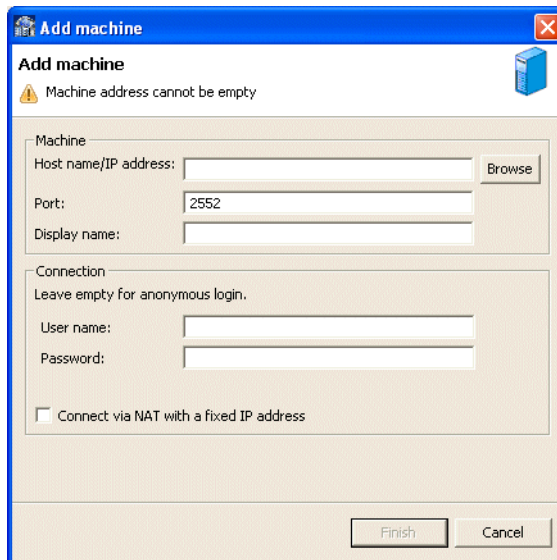
Perform the following steps to configure the IBM OS/390 or z/OS, using Oracle Studio:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. Right-click **Machines** in the Configuration Explorer and select **Add Machine**. The Add Machine screen is displayed, as shown in [Figure 2–1](#).
3. Enter the name of the computer you want to connect to, or click **Browse** to select the computer from the list of computers that is displayed and uses the default port, 2552.
4. Enter the username and password of the user who was specified as the administrator when Oracle Connect was installed.

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

The Add Machine screen is shown in the following figure:

Figure 2–1 The Add Machine screen



5. Click **Finish**.

The computer is displayed in the Configuration Explorer.

Securing Access to Oracle Connect

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

During modeling the following security mechanisms can be applied:

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

During runtime client access to Oracle Connect is provided by the user profile:

- [Setting Up Run-Time User Access to the IBM OS/390 or z/OS Platform](#)

Setting Password Access to Oracle Studio

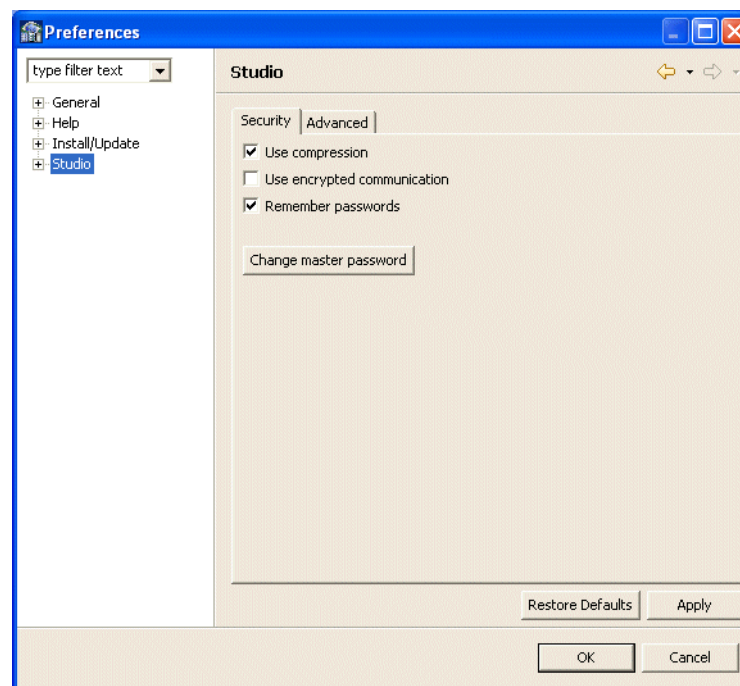
Initially, any operation performed using Oracle Studio does not require a password. You can set a password so that the first operation that involves accessing the server from Oracle Studio requires a password to be entered.

Perform the following steps:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**. Oracle Studio opens.
2. Select **Window** from the menu bar, and then select **Preferences**.

The Preferences screen is displayed as shown in the following figure:

Figure 2–2 The Preferences screen



3. Select the **Studio** node.
4. Click **Change master password**.
5. In the Change master password screen, 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 computer from Oracle Studio. This user can then authorize other users to make changes or to view the definitions for a selected computer. Adding a computer to Oracle Studio is described in ["Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio"](#) on page 2-14.

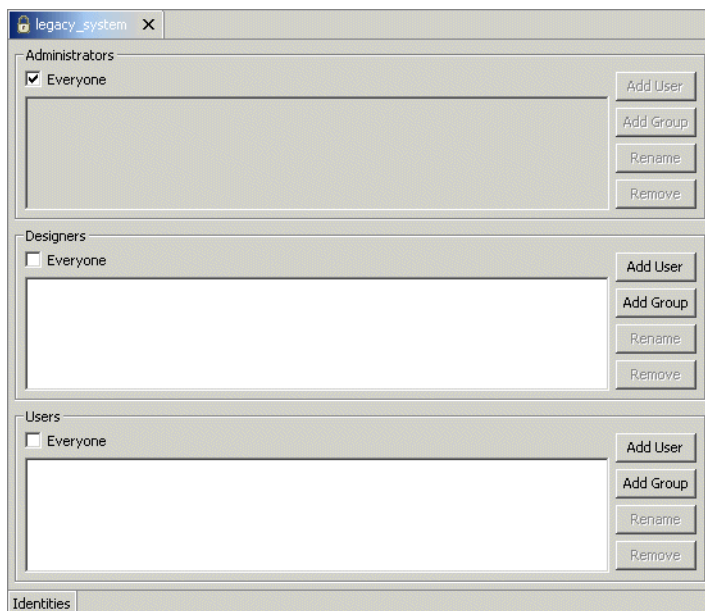
Note: The default during installation is to enable all users to be administrators.

Perform the following steps to specify a user with administrative rights:

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

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

Figure 2–3 The Identities screen



This screen has the following sections:

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

Designers: Designers can view all the definitions for the computer in Oracle Studio and can modify any of the definitions under the **Bindings** and **Users** nodes for the selected computer. For example, Oracle Studio database administrator can add new data sources and adapters and can change metadata definitions for a table in a data source.

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

3. Add users or groups of users by clicking **Add User** or **Add Group** for the relevant sections. The user or group that is added must be recognized as a valid user or group for the computer.

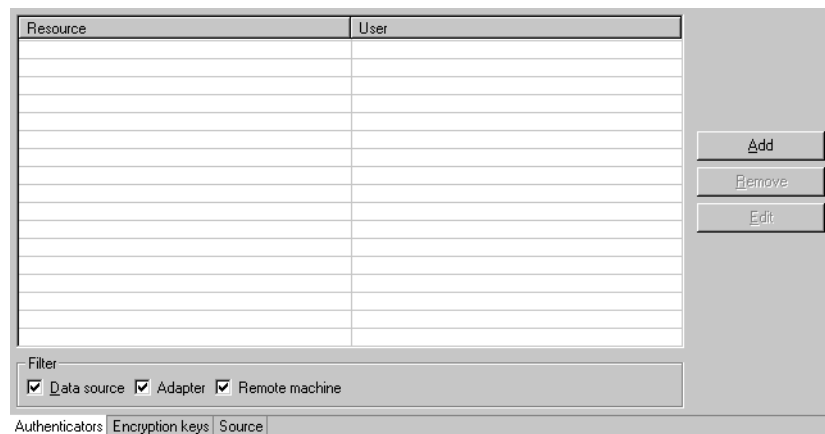
Once a name has been added to a section, only the user or group who logs on with that user name has the relevant authorization.

Setting Up Run-Time User Access to the IBM OS/390 or z/OS Platform

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

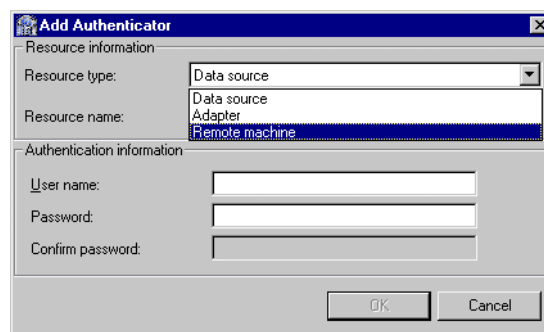
1. In the Configuration Explorer, expand the node of the computer for which you want to set the user name and password.
2. Expand the **Users** node.
3. Right-click the **NAV** user profile and select **Edit User**. The User editor is displayed, as shown in the following figure:

Figure 2–4 The User Editor pane



4. In the User editor, click **Add** to display the Add Authenticator screen.
5. Select **Remote Machine** from the **Resource type** list, as shown in the following figure:

Figure 2–5 The Add Authenticator screen



6. Enter the name of the IBM OS/390 or z/OS computer defined in Oracle Studio.
7. Enter the name and password used to access the computer and confirm the password.
8. Click **OK**.

Modeling Interactions for OracleAS Adapter for CICS

Modeling interactions for OracleAS Adapters for CICS involve defining an Oracle Connect back-end adapter using Oracle Studio. All the definitions specified in Oracle Studio are written to the IBM OS/390 or z/OS platform.

This section contains the following topics:

- [Configuring an Oracle Connect Adapter](#)
- [Generating Outbound Interactions](#)
- [Generating Inbound Interactions](#)

Configuring an Oracle Connect Adapter

To work with the Oracle Connect, you need to configure adapter definitions on the IBM OS/390 or z/OS platform to handle the interactions to and from a CICS program. You use Oracle Studio to specify these definitions.

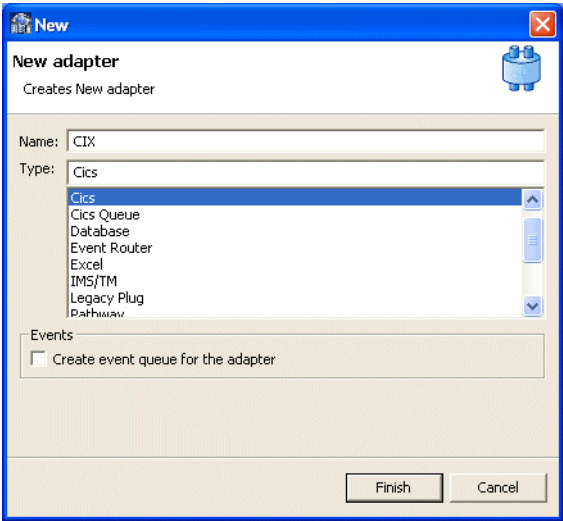
Perform the following steps:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. In the Configuration Explorer, expand the node of the computer defined in ["Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio"](#) on page 2-14.
3. Expand the **Bindings** node. The binding configurations available on this computer are listed.
4. Expand the **NAV** binding node. The NAV binding configuration includes branches for data sources and adapters that are located on the computer.
5. Right-click **Adapters** and select **New Adapter** to start the New Adapter wizard.
6. Enter a name for the back-end adapter.

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

7. Select **CICS** as the back-end adapter type from the **Type** list as shown in the following figure:

Figure 2–6 The New Adapter wizard

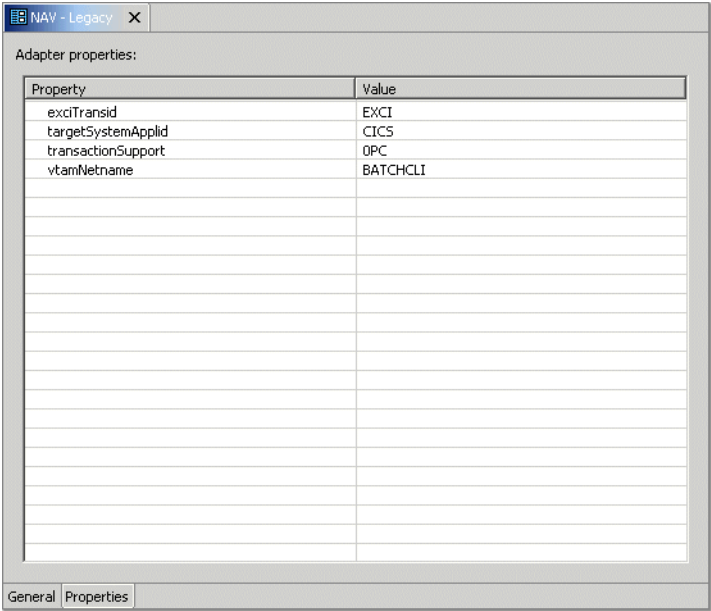


- 8. Select **Events**.
- 9. Click **Finish**. The back-end adapter is added to the list and the definition opens for editing.

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

- 10. Click the **Properties** tab and change any of the properties for the adapter, as required.
- The **Properties** tab is shown in the following figure:

Figure 2–7 The Adapter Properties tab



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

Table 2–14 CICS Adapter Properties

Property	Description
exciTransid	The CICS TRANSID. This value must be EXCI or a copy of this transaction
targetSystemApplid	The VTAM applid of the CICS target system. 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= <i>target_system</i> .
transactionSupport	For future use. The level of transaction support for this adapter that is reported to the transaction manager In the current version leave this value as 0PC
vtamNetname	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). If you plan to use the IBM defaults, then specify BATCHCLI as the VTAM_netname parameter, otherwise, define a specific connection (with EXCI protocol) and use the netname you provided there for this parameter.

Generating Outbound Interactions

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

If COBOL copybooks describing the adapter input and output structures are available, then you can import the adapter definition using the Metadata Import wizard in Oracle Studio Design perspective to generate interaction metadata. If the metadata is provided in a number of COBOL copybooks, with different filter settings (such as whether the first six columns are ignored or not), then you first import the metadata from copybooks with the same settings and later import the metadata from the other copybooks.

If COBOL copybooks describing the input and output structures are not available, then you need to manually define the metadata.

The following information is required during the import procedure:

The COBOL copybooks: These are copied to the computer running Oracle Studio as part of the import procedure.

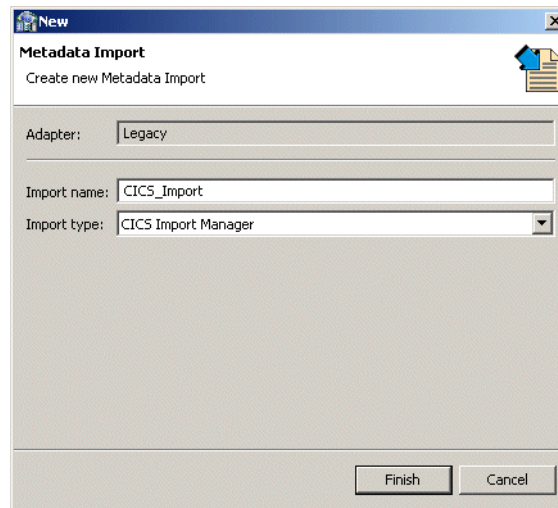
The names of the CICS programs run through the application adapter.

Use the Metadata Import procedure using Oracle Studio to generate interaction metadata, as follows:

1. In the Configuration Explorer, right-click the CICS back-end adapter defined.
2. Select **Edit Metadata** to display the Metadata tab, with the CICS back-end adapter displayed under the **Adapters** list.
3. Right-click the **Imports** node and select **New Import**. The New Metadata Import screen is displayed.
4. Enter a name for the import. The name can contain letters and numbers and the underscore character only.

5. Select **CICS Import Manager** from the **Import type** list, as shown in the following figure:

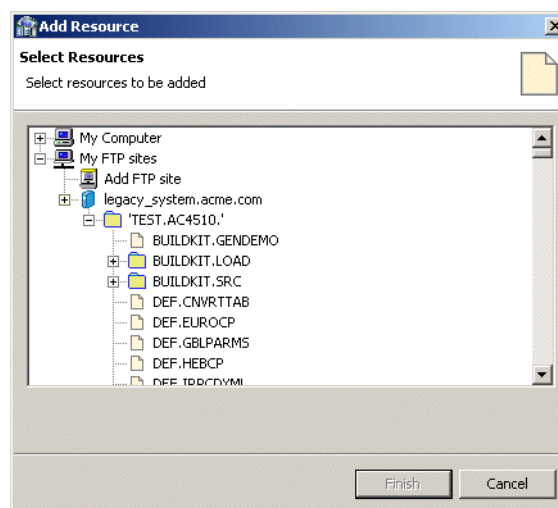
Figure 2–8 The Metadata Import wizard



1. Click **Finish**. The Metadata Import wizard is displayed.
2. Click **Add**. The Select Resources screen is displayed, which provides the option to select files from the local computer or copy the files from another computer.
3. If the files are on another computer, then right-click **My FTP Sites** and select **Add**. Optionally, double-click **Add FTP site**. The Add FTP Site screen is displayed.
4. Enter the server name or IP address where the COBOL copybooks reside and enter a valid username and password to access the computer (if anonymous access is used, click the Anonymous connection checkbox) then click **OK**.

The FTP site is added to the list of available sites, as shown in the following figure:

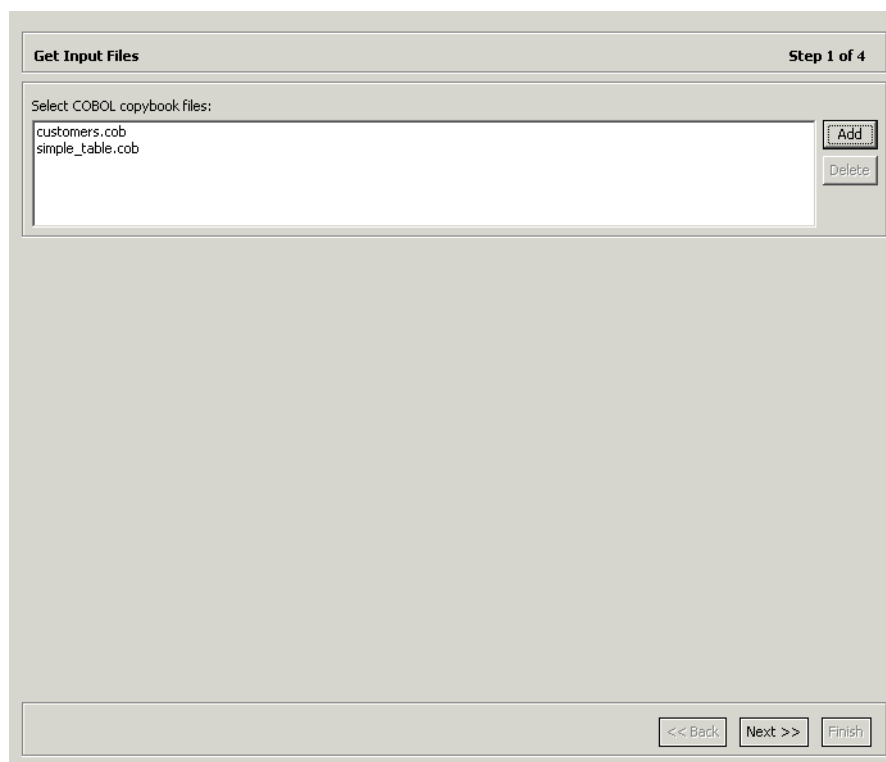
Figure 2–9 The Select Resources screen



Note: The selected server is accessed using the username as the high-level qualifier, which enables you to browse and transfer files.

5. Right-click the computer and select **Set Transfer Type**. Enter the transfer type (ASCII or BINARY) and click **OK**.
6. Expand the list of the added site until you find the necessary COBOL files. To change the high-level qualifier, right-click the computer and select **Change Root Directory**. Enter the new high-level qualifier enclosed in quotes, and click **OK**.
7. Select the required COBOL copybook file or files and click **Finish**. The selected file or files are displayed in the Metadata Import wizard as shown in the following figure:

Figure 2–10 The Get Input Files screen



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

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

8. Click **Next**. The Apply Filters screen is displayed as shown in the following figure.

Figure 2–11 The Apply Filters screen

Property	Value
<input type="checkbox"/> cobolFilter	
<input type="checkbox"/> compilerSourceSettings	
COMP_6 switch (for MICROFOCUS compiler source only)	COMP-6'2'
Compiler source	Default / Not known / Other
Storage mode (for MICROFOCUS compiler source only)	NOIBMCOMP
Ignore after column 72	true
Ignore first 6 columns	false
Prefix nested columns	true
Replace hyphens (-) in record and field names with underscores...	true
<input type="checkbox"/> template	
Case sensitive	false
Find	
Replace with	

9. Apply filters to the copybooks, as needed.

The following table lists the available filters:

Table 2–15 Metadata Filters

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

10. In the import wizard, click **Next. The Add Interactions screen is displayed, as shown in the following figure:**

Figure 2–12 The Add Interactions screen

Add Interactions Step 3 of 4

Add Interactions:

Name	Mode	Input	Output	Description
interaction1	sync-send-receive	CUSTOMERS	REQ1	

Add
Remove

Interaction-Specific Parameters

Property	Value
program	\$DFHCUST

<< Back Next >> Finish

11. Click **Add** to add an interaction for the CICS adapter. Provide the information as listed in the following table:

Table 2–16 Interaction Parameters

Parameter	Description
Name	The name of the interaction. You can change the default name specified.
Mode	<p>The interaction mode. You can select one of the following:</p> <ul style="list-style-type: none"> sync-send-receive: The interaction sends a request and expects to receive a response. This is the default mode. sync-receive: The interaction expects to receive a response sync-send: The interaction sends a request and does not expect to receive a response.
Input	<p>Identifies an input record. The input record is the data structure for the interaction. The records generated from the COBOL files specified at the beginning of the procedure are listed. Select the relevant record for the interaction.</p> <p>Note: You must specify an input record for each interaction before you can click Next. If the interaction does not require an input record, then the record specified here is ignored.</p>
Output	<p>Identifies an output record. The output record is the data structure for the results of the interaction. The records generated from the files specified at the beginning of the procedure are listed. Select the relevant record for the interaction.</p> <p>Note: You must specify an output record for the interaction if the mode is set to sync-send-receive or sync-receive, before you can click Next.</p>
Description	Free text describing the interaction.

Table 2–16 (Cont.) Interaction Parameters

Parameter	Description
Interaction-Specific Parameters	<p>CICS specific parameters:</p> <ul style="list-style-type: none"> ■ <code>commitMode</code>: The interaction commit mode. ■ <code>maxSegmentSize</code>: The maximum size of the segment. The default value is 32768 (bytes). ■ <code>transaction</code>: The name of the transaction that runs. You must specify a transaction name for each interaction. <p>The <code>maxSegmentSize</code> parameter enables dynamically splitting large messages into smaller segments. Although the default value can be changed, the logic of CICS transaction must correspond accordingly. The transaction must perform a GU call followed by a series of GN calls in order to compile the entire input message. A transaction output message, larger than 32K, is handled by the OTMS C/I interface which performs the task of assembling the output segments into a single buffer.</p>

12. Add as many interactions as necessary and click **Next** to generate the metadata definitions for the adapter.
13. Specify that you want to transfer the metadata from the Windows computer to the IBM OS/390 or z/OS platform and click **Finish**.

The metadata is imported based on the options specified and it is stored on the IBM OS/390 or z/OS platform. An XML representation of the metadata is also generated.

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

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

Generating Inbound Interactions

Inbound interactions are defined as events in Oracle Studio. When you defined the Oracle Connect for CICS back-end adapter with *Events* selected, as described in [Configuring an Oracle Connect Adapter](#) on page 2-18, an event adapter was defined automatically. The event adapter is defined with the same name as the back-end adapter with the word *event* appended to it. The back-end adapter and the event adapter are linked by Oracle Studio.

You can skip from the adapter definition to the event definition by right-clicking the adapter or event in the Configuration Explorer list and selecting the **Linked Event** or **Linked Adapter** option, respectively.

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

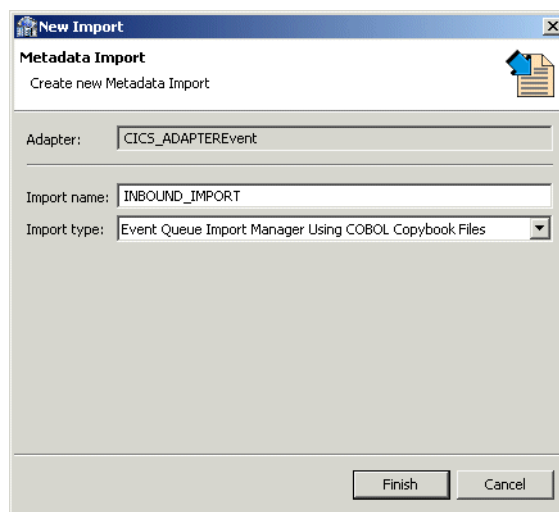
Note: The generation of inbound interactions involves similar steps to the steps described to generate outbound interactions. For details, see ["Generating Outbound Interactions"](#) on page 2-20.

Use the Metadata Import procedure in Oracle Studio to generate interaction metadata, as follows:

1. In the Configuration Explorer, right-click the CICS back-end adapter defined in ["Configuring an Oracle Connect Adapter"](#) on page 2-18.
2. Select **Linked Event**, to skip to the event adapter.
3. Right-click the event adapter and select **Edit Event**.
4. Select the **Properties** tab to add the names of Oracle Application Server users who can retrieve inbound interactions and OS/390 users who can send inbound interactions.
5. To add Oracle Application Server users, expand the **Routers** node and right-click the users property to add the user. The Properties tab is shown in the following figure:
6. Enter the name of Oracle Application Server user in the Value column for the item added.
7. To add OS/390 users, expand the **Senders** node and right click the users property to add the user.
8. Enter the name of the OS/390 user in the Value column for the item added.
9. Click **Save** to save the changes.
10. Right-click the event adapter in the Configuration Explorer and select **Edit metadata**, to display the Metadata tab, with the event adapter displayed under the Events list.
11. Right-click the Imports node and select **New Import**. The New Import screen is displayed.
12. Enter a name for the import. The name can contain letters and numbers and the underscore character only.
13. Select **Event Queue Import Manager Using Cobol COPYBOOK Input Files** as the import type, and click **Finish**.

The New Import screen is shown in the following figure:

Figure 2–13 The New Import screen



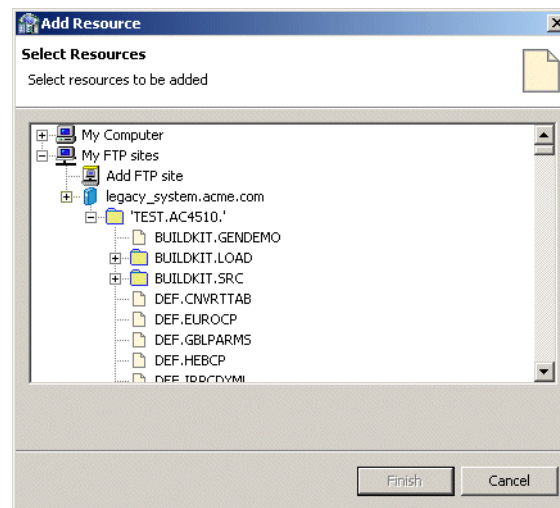
After defining an import type, the metadata import wizard opens in Oracle Studio. COBOL copybooks are used to create the metadata. The import wizard generates record structures, which are used for the record structures for inbound interactions.

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

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

The Add Resource screen is shown in the following figure:

Figure 2–14 The Add Resources screen



4. Right-click the computer and select **Set Transfer Type**. Enter the transfer type (ASCII or BINARY) and click **OK**.
5. Expand the node of the added site and locate the necessary COBOL files. To change the high-level qualifier, right-click the computer and select **Change Root Directory**. Enter a high-level qualifier enclosed in quotes, and click **OK**.
6. Select the file or files and click **Finish**. The selected file or files are displayed in the metadata import wizard, as shown in the following figure:

Figure 2–15 The Get Input Files screen

Get Input Files Step 1 of 5

Select COBOL copybook files:

orders.cob
customers.cob

Add
Delete

<< Back Next >> Finish

7. In the Import wizard, click **Next**. The Apply Filters screen is displayed as shown in the following figure:

Figure 2–16 The Apply Filters screen

Apply Filters Step 2 of 5

Click Next to analyze and convert the source files. You may also change various filter options.

Property	Value
<input checked="" type="checkbox"/> cobolFilter	
<input checked="" type="checkbox"/> compilerSourceSettings	
COMP_6 switch (for MICROFOCUS compiler ...	COMP-6/2'
Compiler source	Default / Not known / Other
Storage mode (for MICROFOCUS compiler s...	NOIBMCOMP
Ignore after column 72	true
Ignore first 6 columns	false
Prefix nested columns	true
Replace hyphens (-) in record and field names ...	true
<input checked="" type="checkbox"/> template	
Case sensitive	false
Find	
Replace with	

<< Back Next >> Finish

Main Source

8. Apply filters to the copybooks. The following table lists the available filters:

Table 2–17 Metadata Filters

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

9. Click **Next**. The Add Events screen is displayed.

10. Click **Add** to add an event for the CICS adapter and provide the following information:

Table 2–18 Event Parameters

Parameter	Description
Name	The name of the event. You can change the default name specified.
Mode	The event mode. You can select the async-send mode, which is the event 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 event. The records generated from the COBOL files specified at the beginning of the procedure are listed. Select the relevant record for the event. Note: You must specify an input record for each event before you can click Next . If the interaction does not require an input record, then the record specified here is ignored.
Description	Free text describing the event.

The Add Events screen is shown in the following figure:

Figure 2–17 The Add Events screen

Name	Mode	Input	Output	Description
event1	async-send	ORDERS		

11. Add the required events.
12. Click **Next** to generate the metadata definitions for the adapter.
13. Specify that you want to transfer the data from the Windows computer to the IBM OS/390 or z/OS platform, and click **Finish**, as shown in the following figure:

Figure 2–18 The Import Metadata screen

The metadata is imported based on the options specified and it is stored on the IBM OS/390 or z/OS platform. An XML representation of the metadata is also generated.

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

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

Modeling Interactions for OracleAS Adapter for CICS Queue

Modeling interactions for OracleAS Adapter for CICS Queue involves defining an Oracle Connect back-end adapter using Oracle Studio. All the definitions specified in Oracle Studio are written to the IBM OS/390 or z/OS platform.

This section contains the following topics:

- [Preliminary Considerations](#)
- [Configuring the CICS Queue Adapter](#)
- [Generating Outbound Interactions](#)

Preliminary Considerations

CICS TD queues can receive events with different structures. Therefore it is essential to appropriately pre-plan the mapping of these events, based on the various applications that write to these queues. If events with different structures are written to a queue, then variants should be used. These variants enable you to pre-define the fields which are used to identify the events that are read from the queue.

Queuing systems use headers which provide metadata for events written to its queues. When headers are used, each event is made of the header part and the data part. The header contains fields which represent the common fields of all the variants defined in the main record (AR_RECKEY in "[Adapter Definition](#)" on page 2-32), and variants which define multiple event types.

The adapter definition includes the `eventHeader` boolean parameter, which determines the structure of the event records. When set to `true`, it indicates that the payload of each event will have the following structure:

```
<event-name>
  <header... header-attributes/>
  <data ...>
  ...
</data>
</event-name>
```

Where *header-attributes* represent the common fields of all the variants defined.

If simple records are used, then the header field contains no fields and the data is simply the entire named record. When complex records are used (in this case, with a variant field), then the remaining header fields are the fields from the first level of the header's `eventsRecord` field, and the data part for a specific instance of an event is the Case of the variant field, as defined by the header's `eventsVariantField` parameter.

When the `eventHeader` parameter is set to `false`, CICS Queue handles the payload of each event as a single record structure. This record can be simple or complex (with variants). This is useful when the queue contains only one type of events.

The COBOL copybook used for the CICS Queue adapter metadata should be prepared in advance. This file should include the various event types that are sent to the CICS Queue. Next, you should prepare a file in `cob_adl` format (see "[COB_ADL Format](#)" on page 2-35), which includes the variant description: the redefined record, the selector field and the values for the variant cases in the order of appearance in the COBOL copybook (these files are used in the metadata import procedure for the OracleAS Adapter for CICS Queue).

The methodology for preparing a single, suitable COBOL copybook that will be imported as part of the Queue adapter metadata definition, is as follows:

- Copying a common root record which includes a SELECTOR field, and placing it as the root record in a new copybook.
- Copying the REDEFINED records, and placing them under the root record.
 - The first record is copied as is.
 - The subsequent records are also copied, but a code line, which sets the redefinition of the first record (REDEFINE), is added to each of them.

Example 2-1 Adapter Definition

Consider the following adapter definition sample. It is the metadata generated after importing the required files for the Queue adapter.

```
<?xml version="1.0" encoding="UTF-8"?>
<adapterDef type="CICSQ" transactionLevelSupport="2PC" name="sela_q1">
  <adapterSpec eventsHeader="false" eventsRecord="AR_RECORD"
    eventsVariantField="VAR_0" />
  <schema noAlignment="true">
    <record name="AR_RECORD">
      <field name="AR_SELECTOR" type="string" nativeType="numstr_u" size="3" />
      <field name="AR_RECKEY" type="AR_RECORD__AR_RECKEY" />
      <field name="VAR_0" type="AR_RECORD__VAR_0" selector="AR_SELECTOR" />
    </record>
    <record name="AR_RECORD__AR_RECKEY">
      <field name="AR_INST" type="string" nativeType="string" size="2" />
    </record>
    <variant implicit="true" name="AR_RECORD__VAR_0">
      <field name="AR_CNTL_RECORD" type="AR_RECORD__VAR_0__AR_CNTL_RECORD"
        case="1" />
      <field name="AR_TIR_RECORD" type="AR_RECORD__VAR_0__AR_TIR_RECORD"
        case="2" />
      <field name="AR_TCR_RECORD" type="AR_RECORD__VAR_0__AR_TCR_RECORD"
        case="3" />
    </variant>
    <record name="AR_RECORD__VAR_0__AR_CNTL_RECORD">
      <field name="AR_CNTL_ROOT" type="AR_RECORD__VAR_0__AR_CNTL_RECORD__AR
        _CNTL_ROOT" />
      <field name="AR_CNTL_SEGMENT" type="AR_RECORD__VAR_0__AR_CNTL_RECORD__AR
        _CNTL_SEGMENT" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_CNTL_RECORD__AR_CNTL_ROOT">
      <field name="AR_CNTL_CTR" type="string" nativeType="numstr_u" size="3" />
      <field name="FILLER" type="string" nativeType="string" size="6" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_CNTL_RECORD__AR_CNTL_SEGMENT">
      <field name="AR_CN_KEY" type="AR_RECORD__VAR_0__AR_CNTL_RECORD__AR_CNTL
        _SEGMENT__AR_CN_KEY" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_CNTL_RECORD__AR_CNTL_SEGMENT__AR_CN_KEY">
      <field name="AR_CN_TIR_CTR" type="string" nativeType="numstr_u" size="3" />
      <field name="AR_CN_TCR_CTR" type="string" nativeType="numstr_u" size="3" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_TIR_RECORD">
      <field name="AR_TIR_ROOT" type="AR_RECORD__VAR_0__AR_TIR_RECORD__AR_TIR
        _ROOT" />
      <field name="AR_TIR_SEGMENT" array="100" type="AR_RECORD__VAR_0__AR_TIR
        _RECORD__AR_TIR_SEGMENT" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_TIR_RECORD__AR_TIR_ROOT">
      <field name="AR_TIR_CTR" type="string" nativeType="numstr_u" size="3" />
    </record>
  </schema>
</adapterDef>
```

```

        <field name="FILLER2" type="string" nativeType="string" size="6" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_TIR_RECORD__AR_TIR_SEGMENT">
        <field name="AR_TI_KEY" type="AR_RECORD__VAR_0__AR_TIR_RECORD__AR_TIR_SEGMENT__AR_TI_KEY" />
        <field name="AR_TI_HRS_RANGE" type="AR_RECORD__VAR_0__AR_TIR_RECORD__AR_TIR_SEGMENT__AR_TI_HRS_RANGE" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_TIR_RECORD__AR_TIR_SEGMENT__AR_TI_KEY">
        <field name="AR_TI_CLASS" type="string" nativeType="string" size="3" />
        <field name="AR_TI_RESID" type="string" nativeType="string" size="1" />
        <field name="AR_TI_COLL" type="string" nativeType="string" size="2" />
        <field name="AR_TI_MAJOR" type="string" nativeType="string" size="4" />
        <field name="AR_TI_SPPGM" type="string" nativeType="string" size="3" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_TIR_RECORD__AR_TIR_SEGMENT__AR_TI_HRS_RANGE">
        <field name="AR_TI_LO_HR" type="string" nativeType="numstr_u" size="2" />
        <field name="AR_TI_HI_HR" type="string" nativeType="numstr_u" size="2" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_TCR_RECORD">
        <field name="AR_TCR_ROOT" type="AR_RECORD__VAR_0__AR_TCR_RECORD__AR_TCR_ROOT" />
        <field name="AR_TCR_SEGMENT" array="99" type="AR_RECORD__VAR_0__AR_TCR_RECORD__AR_TCR_SEGMENT" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_TCR_RECORD__AR_TCR_ROOT">
        <field name="AR_TCR_CTR" type="string" nativeType="numstr_u" size="3" />
        <field name="FILLER3" type="string" nativeType="string" size="6" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_TCR_RECORD__AR_TCR_SEGMENT">
        <field name="AR_TC_KEY" type="AR_RECORD__VAR_0__AR_TCR_RECORD__AR_TCR_SEGMENT__AR_TC_KEY" />
        <field name="AR_TC_HOURS" type="string" nativeType="numstr_u" size="2" />
        <field name="AR_TC_TUIT_TYPE" type="string" nativeType="string" size="1" />
    </record>
    <record name="AR_RECORD__VAR_0__AR_TCR_RECORD__AR_TCR_SEGMENT__AR_TC_KEY">
        <field name="AR_TC_COURS" type="string" nativeType="string" size="11" />
    </record>
</schema>
</adapterDef>

```

The following is a sample of three COBOL copybooks. Note the common AR-RECORD record (marked in bold) in each copybook. This field will be used as the root record of the final copybook.

```

01  AR-RECORD1.
03  AR-SELECTOR                PIC 9(03).
03  AR-RECKEY.
05  AR-INST                    PIC X(02).
**
**  CONTROL RECORD
**
03  AR-CNTL-RECORD.
05  AR-CNTL-ROOT.
07  AR-CNTL-CTR                PIC 9(03).
07  FILLER                    PIC X(06).
05  AR-CNTL-SEGMENT.
07  AR-CN-KEY.
09  AR-CN-TIR-CTR             PIC 9(03).

```

```
09  AR-CN-TCR-CTR  PIC 9(03).

01  AR-RECORD2.
    03  AR-SELECTOR          PIC 9(03).
    03  AR-RECKEY.
        05  AR-INST          PIC X(02).
**
**  TIR - INSTITUTIONAL TUITION RATES.
**
    03  AR-TIR-RECORD.
        05  AR-TIR-ROOT.
            07  AR-TIR-CTR      PIC 9(03).
            07  FILLER          PIC X(06).
        05  AR-TIR-SEGMENT  OCCURS 100 TIMES.
            07  AR-TI-KEY.
                09  AR-TI-CLASS PIC X(03).
                09  AR-TI-RESID PIC X(01).
                09  AR-TI-COLL  PIC X(02).
                09  AR-TI-MAJOR PIC X(04).
                09  AR-TI-SPPGM PIC X(03).
            07  AR-TI-HRS-RANGE.
                09  AR-TI-LO-HR  PIC 9(02).
                09  AR-TI-HI-HR  PIC 9(02).

01  AR-RECORD3.
    03  AR-SELECTOR          PIC 9(03).
    03  AR-RECKEY.
        05  AR-INST          PIC X(02).
**
**  TCR - COURSE TUITION AND FEES.
**
    03  AR-TCR-RECORD.
        05  AR-TCR-ROOT.
            07  AR-TCR-CTR      PIC 9(03).
            07  FILLER          PIC X(06).
        05  AR-TCR-SEGMENT  OCCURS 99 TIMES.
            07  AR-TC-KEY.
                09  AR-TC-COURS  PIC X(11).
            07  AR-TC-HOURS    PIC 9(02).
            07  AR-TC-TUIT-TYPE PIC X(01).
```

The following provides an example of the final COBOL copybook. Note the first record has been copied "as is", while the subsequent records have a REDEFINE appended to them.

```
01  AR-RECORD.
    03  AR-SELECTOR          PIC 9(03).
    03  AR-RECKEY.
        05  AR-INST          PIC X(02).
**
**  CONTROL RECORD
**
    03  AR-CNTL-RECORD.
        05  AR-CNTL-ROOT.
            07  AR-CNTL-CTR      PIC 9(03).
            07  FILLER          PIC X(06).
        05  AR-CNTL-SEGMENT.
```

```

07  AR-CN-KEY.
09  AR-CN-TIR-CTR  PIC 9(03).
09  AR-CN-TCR-CTR  PIC 9(03).
**
**  TIR - INSTITUTIONAL TUITION RATES.
**
03  AR-TIR-RECORD  REDEFINES  AR-CNTL-RECORD.
05  AR-TIR-ROOT.
07  AR-TIR-CTR      PIC 9(03).
07  FILLER          PIC X(06).
05  AR-TIR-SEGMENT  OCCURS 100 TIMES.
07  AR-TI-KEY.
09  AR-TI-CLASS PIC X(03).
09  AR-TI-RESID PIC X(01).
09  AR-TI-COLL  PIC X(02).
09  AR-TI-MAJOR PIC X(04).
09  AR-TI-SPPGM PIC X(03).
07  AR-TI-HRS-RANGE.
09  AR-TI-LO-HR  PIC 9(02).
09  AR-TI-HI-HR  PIC 9(02).
**
**  TCR - COURSE TUITION AND FEES.
**
03  AR-TCR-RECORD  REDEFINES  AR-CNTL-RECORD.
05  AR-TCR-ROOT.
07  AR-TCR-CTR      PIC 9(03).
07  FILLER          PIC X(06).
05  AR-TCR-SEGMENT  OCCURS 99 TIMES.
07  AR-TC-KEY.
09  AR-TC-COURS PIC X(11).
07  AR-TC-HOURS   PIC 9(02).
07  AR-TC-TUIT-TYPE PIC X(01).

```

COB_ADL Format

The following provides an example of the variant description file, which must comply to cob_adl format:

```
variant_record_name, selector_field_name, "case1", "case2", ..., "caseN"
```

For the above samples, the variant description file is as follows:

```
AR-CNTL-RECORD, AR-SELECTOR, "1", "2", "3"
```

CICS TD Queue Sample

The following is sample of the commands used to define a CICS TD Queue on the OS/390 or z/OS platform:

```

CEDA DEFINE TDQUEUE

TDqueue      ==> YTST
Group        ==> ATY
Description   ==>
TYPE         ==> INTra          Extra | INTra | INDirect
EXTRA PARTITION PARAMETERS
Databuffers  :                  1-255
DDname       :
DSname       :
Sysoutclass  :

```

```

Erroroption      :          Ignore | Skip
Opentime         :          Initial | Deferred
REWind           :          Leave | Reread
TYPEFile         :          Input | Output | Rdback
RECORDSize       :          0-32767
BLOCKSize        :          0-32767
RECORDFormat     :          Fixed | Variable
BLOCKFormat      :          Blocked | Unblocked

INTRA PARTITION PARAMETERS
Atifacility ==> File          Terminal | File | System
RECOVstatus ==> Logical      No | Physical | Logical
Facilityid ==>
TRAnsId ==>
TRiggerlevel ==> 00000      0-32767

Userid ==>
NDOUBT ATTRIBUTES
WAIT ==> Yes                No | Yes
WAITAction ==> Reject       Queue | Reject
NDIRECT PARAMETERS
Indirectname :
EMOTE PARAMETERS
REMOTENAME ==>
REMOTESYSTEM ==>

```

The ATYGTEVT CICS Program

The ATYGTEVT CICS program returns events from the CICS Transient Data Queue.

For the CICS programs to work transactionally with the TD queues, the TD queues must be set to `Logically Recoverable`, where only one task at a time can write to the TD queue and only when that task ends (or when it issues a sync-point) can other tasks write to that queue.

To use the ATYGTEVT program, perform the following steps:

1. Copy the program from `NAVROOT.HLQ` to a CICS DFHRPL library (such as `CICS.USER.LOAD`).
2. Define the ATYGTEVT program under CICS using any available group (such as `ATY` group) as follows:

```
CEDA DEF PROG(ATYGTEVT) G(ATY) LANG(C) DA(ANY) DE(ATTUNIT CICSQ GET EVENTS)
```

3. Install the ATYGTEVT program as follows:

```
CEDA IN G(ATY)
```

Configuring the CICS Queue Adapter

To work with the Oracle Connect, you need to configure adapter definitions on the IBM OS/390 or z/OS platform to handle the interactions from a CICS queue. You use Oracle Studio to specify these definitions.

Perform the following steps:

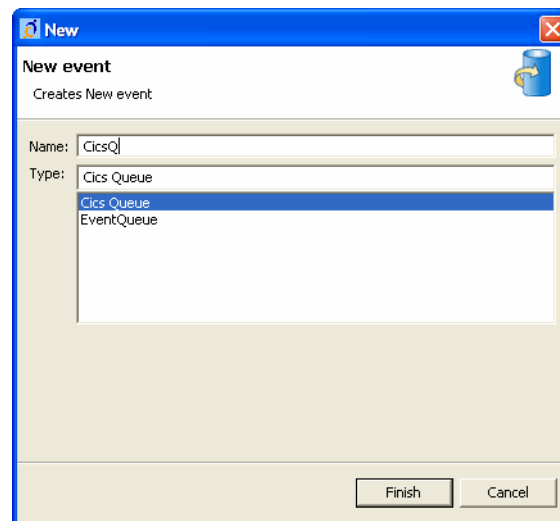
1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. In the Configuration Explorer, expand the node of the computer defined in ["Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio"](#) on page 2-14.

3. Expand the **Bindings** node. The binding configurations available on this computer are listed.
4. Expand the **NAV** binding node. The NAV binding configuration includes branches for data sources, adapters, and events that are located on the computer.
5. Right-click **Events** and select **New Event** to open the New Event wizard.
6. Enter a name for the queue event.

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

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

Figure 2–19 The New Adapter Screen



8. Click **Finish**.

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

Generating Outbound Interactions

OracleAS adapter for CICS Queue requires metadata describing the interaction, including the structure used to pass information.

During the import procedure, a COBOL copybook file and a variant description file are copied to the computer running Oracle Studio.

Perform the following steps to generate the interaction metadata:

1. In the Configuration Explorer, select the CICS Queue adapter defined in ["Configuring the CICS Queue Adapter"](#) on page 2-36.
2. Right-click the queue adapter and select **Edit metadata**.

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

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

The New Import screen is displayed.

4. Enter a name for the import. The name can contain letters, numbers and the underscore character only.
5. Select **CICS Queue Import Manager** from the **Import Type** list.
6. Click **Finish**.

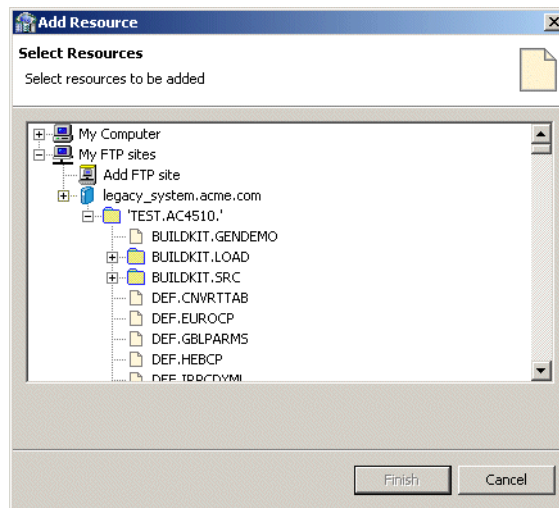
After defining an import type, the metadata import wizard opens in Oracle Studio. COBOL copybooks are used to create the metadata. The import wizard generates record structures, which are used to define the records for outbound interactions.

1. Click **Add** in the Metadata Import wizard. The Select Resources screen is displayed ([Figure 2–20](#)), which provides the option to select files from the local computer or copy the files from another computer.
2. If the files are on another computer, then right-click **My FTP Sites** and select **Add**. Optionally, double-click **Add FTP Site**. The Add FTP Site screen is displayed.
3. Enter the server name or IP address where the COBOL copybooks reside and enter a valid username and password to access the computer (if anonymous access is used, click **Anonymous connection**) then click **OK**. The FTP site is added to the list of available sites.

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

The Select Resource screen is shown in the following figure:

Figure 2–20 The Select Resources screen



4. Right-click the computer and select **Set Transfer Type**. Enter the transfer type (ASCII or BINARY) and click **OK**.
5. Expand the node of the added site and locate the necessary COBOL files. To change the high-level qualifier, right-click the computer and select **Change Root Directory**. Enter a high-level qualifier enclosed in quotes, and click **OK**.

6. Select the file or files and click **Finish**. The selected file or files are displayed in the metadata import wizard.

Figure 2–21 The Get Input Files screen

7. Click **Browse**, and navigate to the variant description COBOL file. If required, repeat steps 2-6, described above.
8. Click **Finish**.
9. In the Import wizard, click **Next**.

The Apply Filters screen is displayed as shown in the following figure:

Figure 2–22 The Apply Filters screen

Property	Value
<input checked="" type="checkbox"/> cobolFilter	
<input checked="" type="checkbox"/> compilerSourceSettings	
Ignore after column 72	true
Ignore first 6 columns	false
Prefix nested columns	true
Replace hyphens (-) in record and field names with underscores (_).	true
<input checked="" type="checkbox"/> template	
<input checked="" type="checkbox"/> VariantsDescriptionFilter	
Replace hyphens (-) in record and field names with underscores (_).	true

10. Apply filters to the copybooks. The available filters are listed in [Table 2–15](#). In addition, if required, apply the variant description filter.

Table 2–19 Variant Description Filter

Filter	Description
Replace hyphens (-) in record and field names with underscores (_).	When set to <code>true</code> , replaces all hyphens in either the record or field names in the metadata generated from the COBOL with underscore characters.

11. Click **Next**.

The Configure Adapter Specification screen is displayed, as shown in the following figure:

Figure 2–23 The Configure Adapter Specification screen

Configure Adapter Specification Step 3 of 4

Set the various adapter specification options and click Next to continue.

Property	Value
Events Record - The name of the COBOL Copybook record to be used.	
Events Variant Field - The variant (the redefined record) describing the events data. Leave empty for one event definition.	
Replace hyphens (-) with underscores (_) in 'Events Record' and 'Events Variant Field'. Should be consistent with the filter options.	true
Show the events header	false

12. In the Value column of the **Events Record** field, enter the name of the COBOL Copybook record that is used.
13. In the Value column of the **Events Variant Field** field, enter the name of the variant (the redefined record) which describes the relevant event's data. This property should be specified only ,if more than one event type is expected. Otherwise leave blank.
14. Specify whether to replace hyphens (`true/false`).
15. Specify whether to show the events header in the metadata (`true/false`).
16. Click **Next** to generate the metadata definitions for the adapter.
17. Specify if you want to transfer the data to the IBM OS/390 or z/OS platform, and click **Finish**.

Figure 2–24 The Import Metadata screen

Import Metadata Step 5 of 5

Click Finish to end the import process.

Deploy Metadata:
Do you want to transfer the metadata to the server?

☒ No
☐ Yes

<input type="button" value=" Next >> " /><input type="button" value=" Finish " />

Viewing the XML Schema

The XML describing the adapter interactions can be viewed in Oracle Studio Design perspective **Metadata** tab, under the adapter **Imports** node.

Creating XML Schemas

The XML schema describing the adapter interactions and the input and output records for these interactions are created automatically during the import procedure, as described in ["Generating Outbound Interactions"](#) on page 2-20 and in ["Generating Inbound Interactions"](#) on page 2-25.

Integrating OracleAS Adapters for CICS with OC4J

To deploy and integrate OracleAS Adapters for CICS with Oracle Application Server Containers for J2EE (OC4J), you need to configure the J2CA 1.5 CICS adapter.

This section includes the following topics:

- [Overview of Integrating OracleAS Adapters for CICS with OC4J](#)
- [Integrating the J2CA 1.5 CICS Adapter for Outbound](#)
- [Integrating the J2CA 1.5 CICS Queue Adapter for Inbound](#)

Overview of Integrating OracleAS Adapters for CICS with OC4J

Oracle Application Server provides a complete Java 2 Enterprise Edition (J2EE) environment that executes on the Java Virtual Machine (JVM) of the standard Java Development Kit (JDK). OC4J is J2EE certified and provides all the J2EE specific containers, APIs, and services. OC4J supports the J2CA 1.5 standard.

J2CA defines standard Java interfaces for simplifying the integration of applications with the EIS. OracleAS adapters are deployed as a Resource Adapter (RA) within the OC4J container.

The contract between the OC4J client application and the resource adapter is defined by the common client interface (CCI). The contract between the OC4J container and the resource adapter is defined by the service provider interface (SPI). The SPI API addresses the connection management, transaction management and the security management.

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

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

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

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

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

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

Message inflow contracts enable the resource adapter to send asynchronous messages to an application server message endpoints regardless of the messaging style, semantics, or delivering infrastructure. This enables a wide variety of message providers (like Java Message Service (JMS), or Java API for XML (JAXM)) to be plugged into an application server through the resource adapter.

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

Integrating the J2CA 1.5 CICS Adapter for Outbound

This section includes the following topics:

- [Configuring the J2CA 1.5 CICS Adapter for Outbound](#)
- [Using the CCI API to Develop Applications](#)

Configuring the J2CA 1.5 CICS Adapter for Outbound

To connect to the J2CA 1.5 CICS adapter under Oracle Application Server, you need to set the relevant parameters in the connection factory. You can perform this task by using Oracle Enterprise Manager, or by carrying out the following steps:

1. Open the following file in an editor:

```
root\j2ee\home\application-deployment\default\oracle\oc4j-ra.xml
```

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

2. In this file, set the following parameters for each connection.

```
<oc4j-connector-factories>
  <connector-factory location=" " connector-name="Oracle Legacy Adapter">
    <config-property name="userName" value=" " />
    <config-property name="password" value=" " />
    <config-property name="eisName" value=" " />
    <config-property name="serverName" value=" " />
    <config-property name="workspace" value=" " />
    <config-property name="portNumber" value=" " />
    <config-property name="persistentConnection" value=" " />
    <config-property name="keepAlive" value=" " />
    <config-property name="firewallProtocol" value=" " />
    <config-property name="connectTimeout" value=" " />
    <config-property name="encryptionProtocol" value=" " />
    <config-property name="encryptionKeyName" value=" " />
    <config-property name="encryptionKeyValue" value=" " />
    <config-property name="fakeXA" value=" " />
    <config-property name="useNamespace" value=" " />
    <config-property name="networkXMLProtocol" value=" " />
    <config-property name="exposeEventStreamMetadata" value=" " />
  </connector-factory>
</oc4j-connector-factories>
```

Note: `location` is the attribute that specifies the JNDI location where Oracle Application Server should bind the connection factory instance for application components.

[Table 3–1](#) provides a detailed description of these parameters.

Table 3–1 OC4J Connection Properties for Outbound Interactions

Property	Description
<code>eisName</code>	Required. Sets the name of the adapter to use. The adapter is defined in Oracle Connect server using Oracle Studio.
<code>serverName</code>	Required. Sets the TCP/IP address or host name where the Oracle Connect daemon is running.
<code>workspace</code>	Optional. Specifies the name of an Oracle Connect server workspace to use. The default workspace is <code>Navigator</code> .
<code>portNumber</code>	Optional. Specifies the TCP/IP port where the Oracle Connect daemon is running on the server. The default port is 2551.
<code>userName</code>	Optional. Specifies a user who can access the Oracle Connect server. The user is defined in the Oracle Connect daemon configuration. See Also: " Daemon Security " on page A-5 and " WS Security " on page A-16 for details about users allowed to access an Oracle Connect server.
<code>password</code>	Optional. Specifies a valid password for the user.
<code>persistentConnection</code>	Optional. Set to <code>true</code> or <code>false</code> . When set to <code>true</code> , connections can persist across multiple requests or connection context changes. It is recommended to set this property to <code>true</code> .
<code>keepAlive</code>	Optional. Set to <code>true</code> or <code>false</code> . When set to <code>true</code> , the socket used for the connection is always kept open. It is recommended to set this property to <code>true</code> .
<code>firewallProtocol</code>	Optional. Specifies the firewall protocol used: either <code>none</code> or <code>fixedNat</code> (the Nat protocol using a fixed address for the daemon). The default is set to <code>none</code> .
<code>connectTimeout</code>	Optional. Specifies the connection timeout in seconds. The default is 0, indicating that there is no connection timeout.
<code>encryptionProtocol</code>	Optional. Specifies the name of encryption protocol to use. The default is set to <code>null</code> . The RC4 protocol is supported.
<code>encryptionKeyName</code>	Optional. Specifies the name of the symmetric encryption key to use.
<code>encryptionKeyValue</code>	Optional. Specifies the value of the symmetric encryption key to use.
<code>fakeXa</code>	Optional. Set to <code>true</code> or <code>false</code> , according to the value of the <code>transaction support</code> property of the corresponding CICS adapter. When set to <code>true</code> , the XA APIs are internally converted to local transaction APIs.
<code>useNamespace</code>	Optional. Set to <code>true</code> or <code>false</code> . When set to <code>true</code> , the metadata record schema uses namespaces.
<code>networkXMLProtocol</code>	Optional. Specifies how the XML is passed over the network. <code>Binary</code> or <code>Text</code> can be selected.

Table 3–1 (Cont.) OC4J Connection Properties for Outbound Interactions

Property	Description
<code>exposeEventStreamMetadata</code>	Optional. When set to <code>true</code> , the standard XSD schema is used for the CICS/Q adapter inbound events metadata. When <code>false</code> , the native schema is used.

Configuring Multiple Adapters

Each J2CA 1.5 CICS adapter requires an entry in the `oc4j-ra.xml` file as described in ["Configuring the J2CA 1.5 CICS Adapter for Outbound"](#) on page 3-2.

See Also: *Oracle Application Server Adapter Concepts Guide*

Updating Configuration Information

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

Using the CCI API to Develop Applications

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

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

1. Select a `ConnectionFactory` object for the J2CA 1.5 CICS adapter.
2. Create a `Connection` object using the selected `ConnectionFactory`. A `Connection` is a handle to the underlying network connection to the EIS, which is identified in the `oc4j-ra.xml` file by the `serverName` property.
3. Create a `Connection` object using the selected `ConnectionFactory`. Specify the interaction properties using an `AttuInteractionSpec` object. The `AttuInteractionSpec` object has the following format:

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

The following table describes the properties that can be specified:

Table 3–2 Interaction Spec Properties

Property	Description
<code>name</code>	Specifies the interaction name to be executed.
<code>verb</code>	Specifies the mode for the interaction: <code>SYNC_SEND</code> , <code>SYNC_SEND_RECEIVE</code> , or <code>SYNC_RECEIVE</code> .
<code>timeOut</code>	Specifies the time (in milliseconds) to wait for an EIS to run the specified interaction.

The following is an `InteractionSpec` sample:

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

4. Invoke the `execute` method on the interaction to initiate a call to the EIS. Pass any data for the interaction as input and output records.
5. Once the interactions have been processed, close the `Interaction` and `Connection` objects.

Integrating the J2CA 1.5 CICS Queue Adapter for Inbound

This section includes the following topics:

- [Configuring the J2CA 1.5 CICS Queue Adapter for Inbound](#)
- [Using the CCI API to Develop Message Endpoint Applications](#)

Configuring the J2CA 1.5 CICS Queue Adapter for Inbound

The provider of the endpoint must supply the following information in the endpoint deployment descriptor file, called `ejb-jar.xml`:

- The properties of the `ActivationSpec` class of the CICS Queue adapter
- The type of message listener implemented in the endpoint

The `orion-ejb.xml` file must provide the name of the resource adapter.

OC4J searches the deployment descriptor of the resource adapter for the message listener type and sets the properties that are defined in the endpoint deployment descriptor file for the respective `ActivationSpec` class.

The resource adapter supports message listener types and `ActivationSpec` classes as follows:

Table 3–3 Message Listener Types and their ActivationSpec Classes

Message Listener Type	ActivationSpec Class
<code>javax.resource.cci.MessageListener</code>	<code>com.attunity.adapter.AttnActivationSpec</code>
<code>oracle.tip.adapter.api.OracleMessageListener</code>	<code>com.attunity.adapter.AttnOracleActivationSpec</code>

Example 3–1 Endpoint Deployment Descriptor (`ejb-jar.xml`)

```
<ejb-jar>
  <display-name>AttuMDB1</display-name>
  <enterprise-beans>
    <message-driven>
      <display-name>Attu Test Receiver Bean</display-name>
      <ejb-name>AttuTestReceiverBean</ejb-name>
      <ejb-class>attutestreceiverbean.AttnTestReceiverBean</ejb-class>
      <messaging-type>javax.resource.cci.MessageListener</messaging-type>
      <transaction-type>Container</transaction-type>
      <activation-config>
        <activation-config-property>
          <activation-config-property-name>userName</activation-config-property-name>
        </activation-config-property>
        <activation-config-property>
          <activation-config-property-name>password</activation-config-property-name>
        </activation-config-property>
      </activation-config>
    </message-driven>
  </enterprise-beans>
</ejb-jar>
```

```
<activation-config-property-name>firewallProtocol</activation-config-
  property-name>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>connectTimeout</activation-config-
    property-name>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>encryptionProtocol</activation-config-
    property-name>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>encryptionKeyName</activation-config-
    property-name>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>encryptionKeyValue</activation-config-
    property-name>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>workspace</activation-config-
    property-name>
  <activation-config-property-value>YTEST</activation-config-
    property-value>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>portNumber</activation-config-
    property-name>
  <activation-config-property-value>4820</activation-config-
    property-value>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>useNamespace</activation-config-
    property-name>
  <activation-config-property-value>true</activation-config-
    property-value>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>networkXMLProtocol</activation-config-
    property-name>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>idleTimeout</activation-config-
    property-name>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>messagesInBatch</activation-config-
    property-name>
  <activation-config-property-value>1</activation-config-
    property-value>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>support2PC</activation-config-
    property-name>
  <activation-config-property-value>true</activation-config-
    property-value>
</activation-config-property>
<activation-config-property>
  <activation-config-property-name>waitTime</activation-config-
    property-name>
```

```

        <activation-config-property-value>5</activation-config-
            property-value>
    </activation-config-property>
    <activation-config-property>
        <activation-config-property-name>retryInterval</activation-config-
            property-name>
        <activation-config-property-value>10</activation-config-
            property-value>
    </activation-config-property>
    <activation-config-property>
        <activation-config-property-name>eisName</activation-config-
            property-name>
        <activation-config-property-value>QVREAD</activation-config-
            property-value>
    </activation-config-property>
    <activation-config-property>
        <activation-config-property-name>serverName</activation-config-
            property-name>
        <activation-config-property-value>mvs5</activation-config-
            property-value>
    </activation-config-property>
    <activation-config-property>
        <activation-config-property-name>exposeEventStreamMetadata</activation-
            config-property-name>
        <activation-config-property-value>true</activation-config-
            property-value>
    </activation-config-property>
</activation-config>
</message-driven>
</enterprise-beans>
<assembly-descriptor>
<!-- NotSupported, Supports, Required, RequiresNew, Mandatory or Never -->
<container-transaction>
    <method>
        <ejb-name>
            AttuTestReceiverBean
        </ejb-name>
        <method-name>
            onMessage
        </method-name>
        <method-params>
            <method-param>javax.resource.cci.Record</method-param>
        </method-params>
    </method>
    <trans-attribute>
        Required
    </trans-attribute>
</container-transaction>
</assembly-descriptor>
</ejb-jar>

```

The following table lists the properties that are relevant for inbound interactions.

Table 3–4 OC4J Connection Properties for Inbound Interactions

Property	Description
eisName	Required. Sets the name of the adapter to use. The adapter is defined in Oracle Connect server using Oracle Studio.

Table 3–4 (Cont.) OC4J Connection Properties for Inbound Interactions

Property	Description
serverName	Required. Sets the TCP/IP address or host name where the Oracle Connect daemon is running.
messagesInBatch	Optional. Specifies the maximum number of messages that can be moved to an endpoint in batch. The default is set to 1.
support2PC	Optional. Enables 2PC transaction support. The default is set to <code>False</code> . Set to <code>true</code> or <code>false</code> , according to the value of the <code>transaction support</code> property of the corresponding event.
waitTime	Optional. Defines the maximum time (in seconds) for an empty transaction duration, and for the CICS adapter to return a "no messages" response (which will cause the current transaction to close). The default is set to 30 seconds.
retryInterval	Optional. Defines the sleep time after any detected problem before restarting inbound activity. The default is set to 15 seconds.
workspace	Optional. Specifies the name of an Oracle Connect server workspace to use. The default workspace is <code>Navigator</code> .
portNumber	Optional. Specifies the TCP/IP port where the Oracle Connect daemon is running on the server. The default port is 2551.
userName	Optional. Specifies a user who can access the Oracle Connect server. The user is defined in the Oracle Connect daemon configuration. See Also: " Daemon Security " on page A-5 and " WS Security " on page A-16 for details about users allowed to access an Oracle Connect server.
password	Optional. Specifies a valid password for the user.
firewallProtocol	Optional. Specifies the firewall protocol used: either <code>none</code> or <code>fixedNat</code> (the Nat protocol using a fixed address for the daemon). The default is set to <code>none</code> .
connectTimeout	Optional. Specifies the connection timeout in seconds. The default is 0, indicating that there is no connection timeout.
encryptionProtocol	Optional. Specifies the name of encryption protocol to use. The default is set to <code>null</code> . The RC4 protocol is supported.
encryptionKeyName	Optional. Specifies the name of the symmetric encryption key to use.
encryptionKeyValue	Optional. Specifies the value of the symmetric encryption key to use.
useNamespace	Optional. Set to <code>true</code> or <code>false</code> . When set to <code>true</code> , the metadata record schema uses namespaces.
networkXMLProtocol	Optional. Specifies how the XML is passed over the network. <code>Binary</code> or <code>Text</code> can be selected.
exposeEventStreamMetadata	Optional. When set to <code>true</code> (the default), the <code>EventStream</code> schema is sent to the endpoint.

Using the CCI API to Develop Message Endpoint Applications

The endpoint must implement the `onMessage` method. The record that the `onMessage` method receives is of the `CoreDomRecord` class, as shown in the following example.

Example 3–2 *onMessage method*

```
public Record onMessage(Record inMessage) throws javax.resource.ResourceException {

    ...
    CoreDOMWriter domW;
    domW = new CoreDOMWriter(false);
    Element outEl = ((CoreDomRecord)inMessage).getDom();
    String xml = domW.toXMLString(outEl);
    ...

    return null;

}
```

The adapter ignores the return values.

If the CICS Queue adapter describes two types of messages, `employee` and `department`, the XML data has the following input record structure:

- When the `exposeEventStreamMetadata` property is set to `true`:

```
<eventStream>
  < EMPLOYEE>
    ...
  </ EMPLOYEE>
  <DEPARTMENT>
    ...
  </DEPARTMENT>
</eventStream>
```

- When the `exposeEventStreamMetadata` property is set to `false`:

```
<getEventsResponse xmlns="noNamespace://QVREAD">
  <event eventName="EMPLOYEE" timestamp="2005-08-23T15:23:18">
    < EMPLOYEE>
      ...
    </ EMPLOYEE>
  </event>
  <event eventName=" DEPARTMENT " timestamp="2005-08-23T15:23:18">
    <DEPARTMENT>
      ...
    </DEPARTMENT>
  </event>
</getEventsResponse>
```

Integrating OracleAS Adapters for CICS with Oracle BPEL Process Manager

To deploy and integrate OracleAS Adapters for CICS with Oracle BPEL Process Manager, you need to configure BPEL Process Manager.

This section includes the following topics:

- [Overview of Integrating OracleAS Adapters for CICS with Oracle BPEL Process Manager](#)
- [Configuring Oracle BPEL Process Manager to interact with the OracleAS Adapter for CICS](#)

Overview of Integrating OracleAS Adapters for CICS with Oracle BPEL Process Manager

Oracle BPEL Process Manager provides a comprehensive solution for creating, deploying, and managing BPEL business processes. Oracle BPEL Process Manager is based on the Service Oriented Architecture (SOA) to provide enterprises with flexibility, interoperability, reusability, extensibility, and rapid implementation of Web services and business processes. It reduces the overall costs of management, modification, extension, and redeployment of existing business processes. Each business activity is a self-contained, self-describing, and modular application whose interface is defined by the WSDL, and the business process is modeled as a Web Service.

A Web Service is first published and then composed or orchestrated into business flows. Publishing a service is implemented by taking a function within an existing application or system and making it available in a standard way, while orchestration is implemented by composing multiple services into an end-to-end business process. The interactions that are defined as part of the configuration of the OracleAS Adapter for CICS are integrated into the orchestration as PartnerLinks. Every PartnerLink is linked to a WSDL that describes the Web service.

To integrating the OracleAS Adapter for CICS with Oracle BPEL Process Manager, you must perform the following tasks in the specified order:

1. [Installing and Configuring OracleAS Adapters for CICS](#)
2. [Integrating OracleAS Adapters for CICS with OC4J](#)
3. [Configuring Oracle BPEL Process Manager to interact with the OracleAS Adapter for CICS](#)

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

Configuring Oracle BPEL Process Manager to interact with the OracleAS Adapter for CICS

This section includes the following topics:

- [Setting up the Connection to the Oracle Connect Server](#)
- [Checking Metadata Availability Using Oracle JDeveloper](#)
- [Configuring the WSDL for Outbound Applications](#)
- [Configuring the WSDL for Inbound Applications](#)

Setting up the Connection to the Oracle Connect Server

Perform the following steps to set up the connection to the Oracle Connect server:

1. Open the Oracle BPEL Admin window.
2. On the Server tab, on the Configuration tab, specify the following:
 - **LegacyServer**: The IP address of the server where Oracle Connect is installed. For a single server, the default is `localhost`.
 - **LegacyPort**: The port number of the server where Oracle Connect is installed. For a single port, the default is `2551`.
3. Repeat the previous step for each Oracle Connect server to be used by Oracle BPEL Process Manager. Use a comma as a separator between the different servers and ports.
4. Click **Apply**.
5. Restart the server where Oracle BPEL Process Manager is installed.

Checking Metadata Availability Using Oracle JDeveloper

Perform the following steps to verify that the metadata of the Oracle Connect server is available in Oracle BPEL Process Manager:

1. Open Oracle JDeveloper.
2. On the **Connections** tab, expand the **Integration Server** node to view the list of OC4J servers.
3. Expand the node of the OC4J server on which you configured the JCA 1.5 CICS adapter (see [Integrating OracleAS Adapters for CICS with OC4J](#)).
4. Under the **Adapters** node, expand the **Legacy** node to view a list of the Oracle Connect servers that you defined by using the Oracle BPEL Admin window.
5. Under the node of the Oracle Connect server whose metadata you want to check, expand the node of the daemon (**IRPCD**) to view a list of workspaces.

A workspace includes adapters for either inbound or outbound applications.

6. Under the node of the workspace that contains the adapter that you want to work with, expand the node of the relevant adapter to view a list of interactions.

If you selected a workspace that includes adapters for inbound applications, the adapter contains a single interaction only, called `EventStream`.

7. Double-click an interaction to view the WSDL.

Configuring the WSDL for Outbound Applications

When you build an outbound application, Oracle BPEL Process Manager automatically creates the WSDL that corresponds to the interaction. The WSDL specifies the name of the adapter's connection factory as the value of the `adapterInstanceJndi` attribute of the `<jca:address>` element in the `<service>` section. This name is generated automatically. You need to verify that a connection factory with this name exists on the OC4J server. If it does not, you need to create it, or change the name of the connection factory to the name of a connection factory that exists.

The following is an example of a WSDL for outbound applications:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<definitions name="add" targetNamespace="http://xmlns.oracle.com/pcbpel/calc/add"
xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:legacyReq="noNamespace://calc"
xmlns:tns="http://xmlns.oracle.com/pcbpel/calc/add"
xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
xmlns:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
xmlns:pc="http://xmlns.oracle.com/pcbpel/" xmlns:legacyRes="noNamespace://calc">
  <types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="noNamespace://calc" targetNamespace="noNamespace://calc"
elementFormDefault="qualified" attributeFormDefault="unqualified">
      <xsd:element name="binput" type="binput"/>
      <xsd:complexType name="binput">
        <xsd:attribute name="p1" type="xsd:int"/>
        <xsd:attribute name="p2" type="xsd:int"/>
      </xsd:complexType>
    </xsd:schema>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="noNamespace://calc" targetNamespace="noNamespace://calc"
elementFormDefault="qualified" attributeFormDefault="unqualified">
      <xsd:element name="output" type="output"/>
      <xsd:complexType name="output">
        <xsd:attribute name="result" type="xsd:int"/>
      </xsd:complexType>
    </xsd:schema>
  </types>
  <message name="request">
    <part name="input_add" element="legacyReq:binput"/>
  </message>
  <message name="response">
    <part name="output_add" element="legacyReq:output"/>
  </message>
  <portType name="addPortType">
    <operation name="add">
      <input name="Input_add" message="tns:request"/>
      <output name="Output_add" message="tns:response"/>
    </operation>
  </portType>
  <binding name="addJCABinding" type="tns:addPortType">
    <jca:binding
XMLRecordConverterCallout="oracle.tip.adapter.fw.record.attunity.AttnXMLRecordConv
erterImpl"/>
    <operation name="add">
      <jca:operation FunctionName="add"
InteractionSpec="com.attunity.adapter.AttnInteractionSpec"
ExecutionTimeout="120"/>
    </operation>
  </binding>
</definitions>
```

```
        </operation>
    </binding>
    <service name="addService">
        <port name="addPort" binding="tns:addJCABinding">
            <jca:address adapterInstanceJndi="eis/legacy/calc"/>
        </port>
    </service>
    <plt:partnerLinkType name="addPartnerLinkType">
        <plt:role name="addRole">
            <plt:portType name="tns:addPortType"/>
        </plt:role>
    </plt:partnerLinkType>
</definitions>
```

Configuring the WSDL for Inbound Applications

When you build an outbound application, Oracle BPEL Process Manager automatically creates the WSDL that corresponds to the interaction, including the most important attributes of the `ActivationSpec` class, such as the name of the adapter, the server, the workspace, and the port number. The name of the workspace is the one that the `EventStream` interaction belongs to. All other properties have default values that you can modify.

The WSDL also specifies the name of the adapter's connection factory as the value of the `adapterInstanceJndi` attribute of the `<jca:address>` element in the `<service>` section. This name is generated automatically. If a connection factory with this name exists on the OC4J server, its values are taken. Otherwise, the properties specified by the `ActivationSpec` are used. If a value is specified by both the connection factory and the `ActivationSpec`, the `ActivationSpec` property overrides the value in the connection factory. If you want to use the value specified in the connection factory, you first need to delete the property from the `ActivationSpec`.

For a list of `ActivationSpec` properties, see the Oracle Application Server Containers for J2EE documentation.

The following is an example of a WSDL for outbound applications:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<definitions name="eventStream"
targetNamespace="http://xmlns.oracle.com/pcbpel/testCICSQ1/eventStream"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:legacyReq="noNamespace://testCICSQ1"
xmlns:tns="http://xmlns.oracle.com/pcbpel/testCICSQ1/eventStream"
xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
xmlns:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
xmlns:pc="http://xmlns.oracle.com/pcbpel/" xmlns:legacyRes="noNamespace://DEMO#">
    <types>
        <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="noNamespace://testCICSQ1" targetNamespace="noNamespace://testCICSQ1"
elementFormDefault="qualified" attributeFormDefault="unqualified">
            <xsd:element name="eventStream" type="eventStreamDescription"/>
            <xsd:complexType name="eventStreamDescription">
                <xsd:choice minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="EMPLOYEE_INFO__VAR_0__BIG_EMPLOYEE"
type="EMPLOYEE_INFO__VAR_0__BIG_EMPLOYEE"/>
                    <xsd:element name="EMPLOYEE_INFO__VAR_0__EMPLOYEE"
type="EMPLOYEE_INFO__VAR_0__EMPLOYEE"/>
                </xsd:choice>
            </xsd:complexType>
```

```

        <xsd:element name="EMPLOYEE_INFO__VAR_0__EMPLOYEE" type="EMPLOYEE_INFO__
VAR_0__EMPLOYEE" />
        <xsd:complexType name="EMPLOYEE_INFO__VAR_0__EMPLOYEE">
            <xsd:attribute name="LAST_NAME">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="15" />
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
            <xsd:attribute name="FIRST_NAME">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="10" />
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
            <xsd:attribute name="ADDRESS_1">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="15" />
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
            <xsd:attribute name="ADDRESS_2">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="15" />
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
            <xsd:attribute name="CARS" type="xsd:int" />
            <xsd:attribute name="SALARY" type="xsd:int" />
        </xsd:complexType>
        <xsd:element name="EMPLOYEE_INFO__VAR_0__BIG_EMPLOYEE" type="EMPLOYEE_
INFO__VAR_0__BIG_EMPLOYEE" />
        <xsd:complexType name="EMPLOYEE_INFO__VAR_0__BIG_EMPLOYEE">
            <xsd:attribute name="ID">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="10" />
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
            <xsd:attribute name="LAST_NAME2">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="100" />
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
            <xsd:attribute name="FIRST_NAME2">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="100" />
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
            <xsd:attribute name="ADDRESS_12">
                <xsd:simpleType>

```

```

        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="150"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="ADDRESS_22">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="150"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="CARS2" type="xsd:int"/>
<xsd:attribute name="SALARY2" type="xsd:int"/>
<xsd:attribute name="FILLER">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1500"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
</xsd:schema>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tns="noNamespace://DEMO#" attributeFormDefault="qualified"
elementFormDefault="qualified" targetNamespace="noNamespace://DEMO#">
    <element name="eventStream">
        <complexType/>
    </element>
</schema>
</types>
<message name="event">
    <part name="event_eventStream" element="legacyReq:eventStream"/>
</message>
<portType name="eventStreamPortType">
    <operation name="eventStream">
        <input name="Event_eventStream" message="tns:event"/>
    </operation>
</portType>
<binding name="eventStreamJCABinding" type="tns:eventStreamPortType">
    <pc:inbound_binding/>
    <operation name="eventStream">
        <jca:operation
ActivationSpec="com.attunity.adapter.AttnOracleActivationSpec"
EisName="testCICSQ1" ServerName="mvs5" UserName="" Password=""
Workspace="testCICSQ1" PortNumber="2551" FirewallProtocol="" ConnectTimeout="0"
EncryptionProtocol="" EncryptionKeyName="" EncryptionKeyValue=""
UseNamespace="true" NetworkXMLProtocol="" MessagesInBatch="50" Support2PC="true"
WaitTime="30" RetryInterval="15"/>
        <input/>
    </operation>
</binding>
<service name="eventStreamService">
    <port name="eventStreamPort" binding="tns:eventStreamJCABinding">
        <jca:address
ResourceAdapterClassName="com.attunity.adapter.AttnOracleResourceAdapter"
adapterInstanceJndi="eis/legacy/testCICSQ1"/>
    </port>
</service>
<plt:partnerLinkType name="eventStreamPartnerLinkType">

```

```
<plt:role name="eventStreamRole">
  <plt:portType name="tns:eventStreamPortType"/>
</plt:role>
</plt:partnerLinkType>
/definitions>
```

Advanced Features of OracleAS Adapters for CICS

Oracle Connect includes a number of tuning parameters that can improve performance. Specifically, the daemon can be configured to optimize communication between the IBM OS/390 or z/OS platform and a client. In addition, the binding environment can be tuned to optimize the request handling.

This section contains the following topics:

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

Configuring the Daemon for High Availability

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

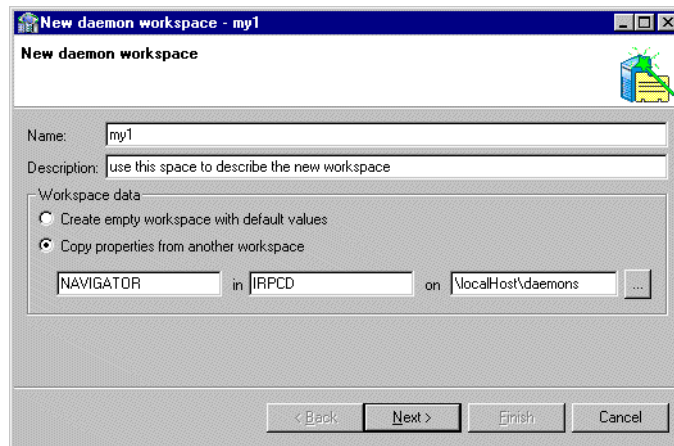
You can also have a number of daemon workspace configurations. Thus, you can create individual workspaces for use with different adapters.

Adding a New Daemon Workspace Configuration

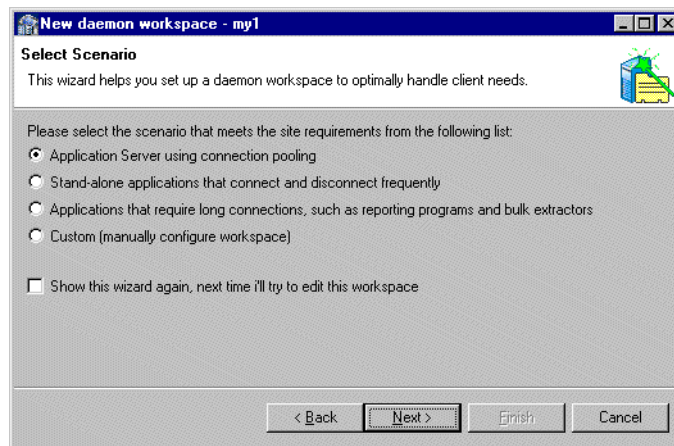
Use Oracle Studio to add a new daemon configuration. You can set up different daemon configurations for different situations.

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

1. From the **Start** menu, select **Programs, Oracle** and then select **Studio**.
2. In the Configuration Explorer, expand the node of the computer defined in ["Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio"](#) on page 2-14.
3. Expand the **Daemons** node. The available daemon configurations are listed.
4. Right-click **IRPCD** and select **New Workspace**. The New Daemon Workspace screen is displayed, as shown in the following figure:

Figure 5–1 The New Daemon Workspace screen

5. Specify a name for the new workspace and optionally, provide a description.
6. Specify whether you want default settings or copy the properties of an existing workspace.
To copy the properties of an existing workspace, click **Ellipsis** and select the workspace from which you want to copy the properties.
7. Click **Next**. The Select Scenario screen is displayed, as shown in the following figure:

Figure 5–2 The Select Scenario screen

8. Select the **Application Server using connection pooling** and click **Next**.
9. Continue through the wizard, specifying the required values for the workspace.
10. To complete the workspace definition, click **Finish**.

The new workspace is displayed under the **IRPCD** daemon node.

Editing the Workspace

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

Table 5–1 Workspace Definition Tabs

Tab	Description
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	This tab is not applicable for use with OracleAS Adapter for CICS.

To access these tabs, perform the following steps using Oracle Studio:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. In the Configuration Explorer, expand the computer defined in ["Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio"](#) on page 2-14.
3. Expand the **Daemons** node. The daemon configurations available on this computer are listed.
4. Expand the **IRPCD** node. The daemon workspaces are listed.
5. Right-click the required workspace and select **Edit Workspace**.
6. Click the required tab, which contains the information you want to edit. For full details of the tabs and the fields in these tabs, refer to ["Workspaces"](#) on page A-7.
7. After editing the workspace, click **Save**.

Configuring the Server Mode

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

- **singleClient**: Each client receives a dedicated server process. The account in which a server process runs is determined either by the client login information or by the specific server workspace.

This mode enables servers to run under a particular user account and isolates clients from each other (because each receives its own process). However, this server mode incurs a high overhead due to process startup times and may use a lot of server resources, as it requires as many server processes as concurrent clients).

- **multiClient**: Clients share a server process and are processed serially. This mode has low overhead because the server processes are 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 parameter (the maximum number of concurrent clients a server process for the current workspace accepts).

- **reusable:** This is an extension of the single client mode. Once the client processing finishes, the server process does not die and can be used by another client, reducing startup times and application startup overhead. This mode does not have the high overhead of single client mode because the servers are already initialized. However, this server mode may use a lot of server resources, as it requires as many server processes as concurrent clients).

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

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

Figure 5–3 The WS Server tab

The screenshot shows the 'WS Server' tab of a daemon workspace editor. The 'Workspace server mode' is set to 'multiClient'. The 'Reuse limit' section has 'Maximum' set to 50. The 'Clients per server limit' section has 'None' selected. The 'Server availability' section shows 'Initial number of servers' as 1 and 'Minimum number of available servers' as 0. The 'Resource limitations' section has 'Limit number of active servers' unchecked. The 'Server priority' section has 'Use default priority' selected. The bottom of the window shows a tab bar with 'WS Server' selected.

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 lower than the value specified in the Minimum number of available servers field, the daemon again starts server processes until the specified number of available servers is reached. The default for this parameter is 0, meaning that no servers are prestarted for this workspace.
- **Minimum number of available servers:** The minimum number of server processes in the prestarted server's pool before the Oracle Connect daemon resumes creating

new server processes (up to the number specified in the Initial number of servers field value, described earlier). If this parameter is set to a value greater than that of the Initial number of servers field value, the daemon considers the value to be the same as the value specified in the Initial number of servers field. In this case, a new server process is started and added to the pool each time a server process is removed from the pool and allocated to a client). The default for this parameter is 0, 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 earlier, 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

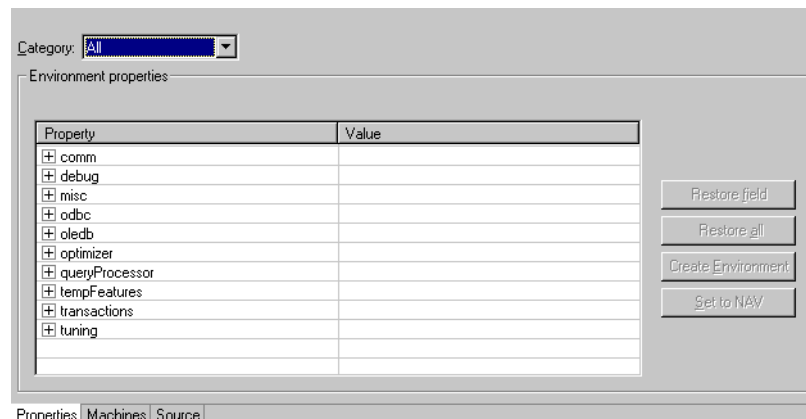
Each binding configuration includes the following:

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

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

1. From the **Start** menu, select **Programs, Oracle** and then select **Studio**.
2. In the Configuration Explorer, expand the node of the computer defined in ["Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio"](#) on page 2-14.
3. Expand the **Bindings** node. The available binding configurations are listed.
4. Right-click **NAV** and select **Edit Binding**.
5. In the Properties tab edit the required environment settings as needed. To edit an environment setting, click the property category node, and then click the required value to edit.

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

Figure 5–4 The Binding Properties tab

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 table lists the parameters that define the communication buffers:

Table 5–2 Comm Category Parameters

Parameter	Description
comCacheBufferSize	Specifies 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	Specifies the maximum bytes that can be written in one chunk on a socket. The default is -1 (no limitation).
comMaxXmlInMemory	Specifies the maximum size of an XML document held in memory. The default is 65535 bytes.
comMaxXmlSize	Specifies the maximum size of an XML document passed to another computer. The default is 65535 bytes.

debug Category

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

Table 5–3 *debug Category Parameters*

Parameter	Description
acxTrace	When set to <code>true</code> , the input xml sent to the back-end adapter and the output xml returned by the back-end adapter, are written to the log.
analyzerQueryPlan	This parameter is not applicable for use with OracleAS Adapters for CICS.
gdbTrace	This parameter is not applicable for use with OracleAS Adapters for CICS.
generalTrace	When set to <code>true</code> , 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 <code>generalTrace</code> is set to <code>true</code> .
oledbTrace	This parameter is not applicable for use with OracleAS Adapters for CICS.
optimizerTrace	This parameter is not applicable for use with OracleAS Adapters for CICS.
queryWarnings	This parameter is not applicable for use with OracleAS Adapters for CICS.
traceDir	This parameter is not applicable for use with OracleAS Adapters for CICS.

miscellaneous Category

The following parameters define miscellaneous operations, including globalization support and the directory where temporary files are written:

Table 5–4 *misc Category Parameters*

Parameter	Description
codepage	For use with globalization support to identify the codepage for the workspace. See also: Appendix A, "Advanced Tuning of the Metadata" .
cvtSeverityLevel	The data type conversion policy when a conversion error occurs: <ul style="list-style-type: none"> ■ 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	This parameter is not applicable for use with OracleAS Adapter for CICS.
language	Identifies the application language. A default codepage is selected based on the value specified for this parameter. See also: Appendix B, "Globalization Settings" .

Table 5–4 (Cont.) misc Category Parameters

Parameter	Description
<code>nlsString</code>	<p>Specifies the codepage used by a field whose data type is defined as "nlsString". Use this for a field whose codepage is other than that of the computer codepage. This parameter includes the following values:</p> <p>The name of the codepage.</p> <p>Whether the character set reads from right to left (as in middle eastern character sets). The default is set to <code>false</code>.</p>
<code>tempDir</code>	<p>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.</p>
<code>year2000Policy</code>	<p>Determines the way 2-digit years are converted into 4-digit years. When the parameter <code>year2000Policy</code> is not set, or when it is set to a value outside the range of values defined for the policy, as described in the following paragraphs, a default value of 5 and the Sliding Base Year policy is used. Two policies are provided:</p> <ul style="list-style-type: none"> Fixed Base Year: <code>year2000Policy</code> is set to a value greater than, or equal to 1900. In this case, the value of <code>year2000Policy</code> is the first 4-digit year after 1900 that can be represented by a 2-digit year. For example, if <code>year2000Policy</code> is set to 1905, the years 2000->2004 will be represented by 00->04. All other 2 digits will map to 19xx. This solution works data is present at the low end (close to the year 1900), which the user wants to keep with the current 2-digit format. <p>The user will probably change the base date only after ensuring that these old dates have been deleted from the data source.</p> <ul style="list-style-type: none"> Sliding Base Year: <code>year2000Policy</code> is set to a positive value less than 100. In this case, the value of <code>year2000Policy</code> represents the number of years ahead of the current year that can be represented by a 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 not applicable for use with OracleAS Adapters for CICS.

oledb Category

The `oledb` parameters are not applicable for use with OracleAS Adapters for CICS.

optimizer Category

The `optimizer` parameters are not applicable for use with OracleAS Adapters for CICS.

queryProcessor Category

The `queryProcessor` parameters are not applicable for use with OracleAS Adapters for CICS.

transactions Category

The following table lists the parameters that define the transaction support:

Table 5–5 Transactions Category Parameters

Parameter	Description
<code>commitOnDestroy</code>	TBD
<code>convertAllToDistributed</code>	When set to <code>true</code> , converts all simple transactions into distributed transactions.
<code>convertAllToSimple</code>	When set to <code>true</code> , converts all distributed transactions into simple transactions.
<code>disable2PC</code>	When set to <code>true</code> , disables two-phase commit capabilities, even in drivers that support two-phase commit.
<code>extendedLogging</code>	TBD
<code>logFile</code>	The high-level qualifier and name of the log file that logs activity when using transactions. The <code>logfile</code> parameter can also include the keyword <code>NORRS</code> after a comma (so that the format is <code>log,NORRS</code>) when RRS is not running on the OS/390 or z/OS platform.
<code>oleThreads</code>	TBD
<code>recoveryDelay</code>	The number of minutes from the start of a transaction before any recovery operation on that transaction can be attempted. The default is 15 minutes.
<code>timeLimit</code>	Specifies the amount of time to wait for a pending transaction to complete. This parameter is used when working under MS DTC where DTC is the transaction coordinator. A value of zero indicates a wait time of 1 msec.
<code>useCommitConfirmTable</code>	This parameter is not applicable for use with OracleAS Adapters for CICS.

tuning Category

The tuning parameters are not applicable for use with OracleAS Adapters for CICS.

Migration Considerations

You can migrate an adapter configuration from one platform to another. The configuration information is stored in the Oracle Connect repository on the source platform and is exported to an XML file which can then be imported to the target platform.

Note that when migrating a configuration, any file names and paths that are specific to the source platform must be changed to valid files on the target platform.

Perform the following steps to migrate an adapter configuration using Oracle Studio.

1. From the **Start** menu, select **Programs, Oracle** and then select **Studio**.
2. In the Configuration Explorer, right-click the computer defined in "[Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio](#)" on page 2-14 and select **Export XML definitions**.
3. Specify the path and name of the XML file, which stores the XML representation and complete configuration.

4. Edit any paths in the XML definition to the paths required on the target platform. For example, the setting for the `serverLogFile` parameter may need changing, depending on the platform.
5. Set up the target platform in Oracle Studio in the same way you set up the source platform, as described in ["Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio"](#) on page 2-14.
6. In the Configuration Explorer, right-click the target computer and select **Import XML definitions**.
7. Import the XML file to the target platform.

Security Considerations

Oracle Connect works within the confines of the legacy platform security system. For example, on an OS/390 computer with RACF installed, and with the workspace server mode set to `multi-tasking`, a `RACROUTE VERIFY` is performed for each task in the address space, according to the client connection.

In addition, Oracle Connect provides the following security components:

- A binary XML encryption mechanism, which is activated as follows:
 1. The client's first message to the server includes a pre-defined shared key, including the key name and value in the connection string. The server gets the key value for the key name passed from the client from the native object store (NOS).
 2. The server generates a random 128-bit RC4 session key which is returned encrypted to the client, using the shared key. If no predefined shared key is provided, then a predefined, hardcoded key is used (this key is hardcoded on the client and on the server).
 3. Passwords are always encrypted when passed over the wire, using an RC4, 128-bit session key, regardless of whether the entire session is encrypted or not.
 4. If a predefined shared key was provided, then the entire session is encrypted. Otherwise, only the password exchange is encrypted (using the hardcoded key).
- Credentials: Passwords and usernames exchanged over the network are encrypted using a pre-defined, hardcoded, 128-bit RC4 session key.
- Design Time: Security within Oracle Studio to grant access to Oracle Studio itself and to grant access to computers, user profiles and workspaces.
- Run-time: Security used to access resources from an application, including encryption over the network and controlling the daemon for the access.

Setting Design Time Security

Setting design time security is described in the following sections:

- Securing access to Oracle Studio is described in ["Setting Password Access to Oracle Studio"](#) on page 2-15.
- Securing rights to configure a computer in Oracle Studio is described in ["Specifying Users with Administrative Rights"](#) on page 2-15.

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

Setting Run-time Security

During run-time, security considerations are implemented as follows:

- When the client request accesses the legacy platform through the daemon, either anonymous access is allowed or a valid user name and password must be provided for the computer in the user profile. The `userName` and `password` properties in the J2CA 1.5 CICS adapter are used at this stage to access the daemon.

Note: The user name used to access the daemon must also be the name of a user profile used.

- Access by the client must be through a valid port, according to the list of ports specified in the Workspace Access section of the **WS Security** tab in Oracle Studio. See "[WS Security](#)" on page A-16.

Note: Access to the legacy platform through a firewall using the NAT protocol is specified when the computer is added to Oracle Studio.

- To be allocated a server process, the client must be granted anonymous access to the workspace or be listed in the Workspace Users section of the **WS Security** tab in Oracle Studio. See "[WS Security](#)" on page A-16.
- To access a server through a firewall, the daemon must be specified with a range of port numbers that are recognized by Oracle Connect. This is specified in the Enable Port Range section of the **WS Security** tab in Oracle Studio. See "[WS Security](#)" on page A-16.
- The ability to run commands on the daemon, such as starting or stopping a daemon or ending server processes is available only to administrators who have been registered in Oracle Connect as a daemon administrator. A client is registered as a valid daemon administrator in the **Daemon Security** tab in Oracle Studio, as described in "[Daemon Security](#)" on page A-5.

Note: You can also specify administrators who can run commands only at the level of the workspace. Specify these administrators in the **WS Security** tab, as described in "[WS Security](#)" on page A-16.

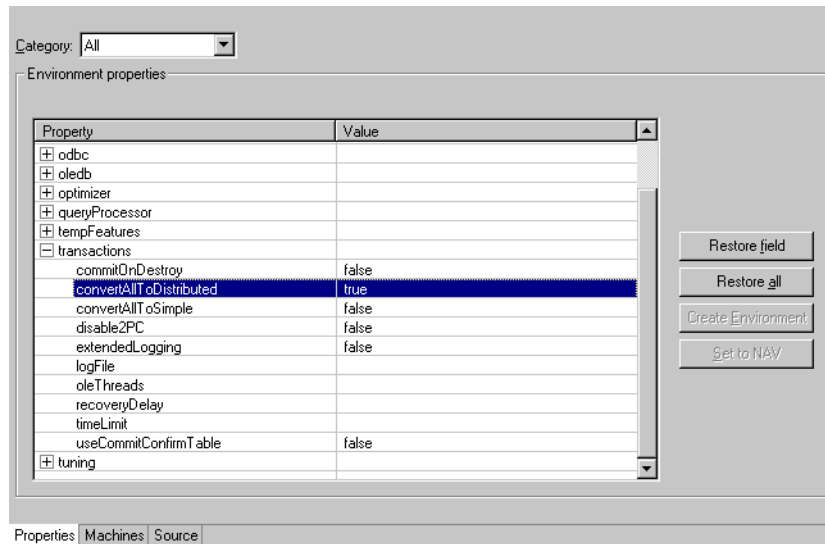
Transaction Support

OracleAS Adapters for CICS supports global transactions and can fully participate in a distributed transaction.

Note: To use OracleAS Adapters for CICS with 2PC, you must have RRS installed and configured.

To work with global transactions, set the `convertAllToDistributed` transaction environment property to `true`, as shown in the following figure:

Figure 5–5 The NAV binding Properties tab



To use the global transaction capability to access data on the OS/390 or z/OS computer, define every library in the ATTSRVR JCL as an APF-authorized library.

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

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

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

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

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

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

If RRS is not running, then OracleAS Adapters for CICS can participate in a distributed transaction, as the only one-phase commit resource, only if the `logFile` transaction environment property includes the keyword `NORRS`, as shown in the following figure:

Figure 5–6 The Properties tab

Property	Value
+ comm	
+ debug	
+ misc	
+ odbc	
+ oledb	
+ optimizer	
+ queryProcessor	
+ tempFeatures	
- transactions	
commitOnDestroy	false
convertAllToDistributed	true
convertAllToSimple	false
disable2PC	false
extendedLogging	false
logFile	logfile,NORRS
oleThreads	
recoveryDelay	
timeLimit	
useCommitConfirmTable	false
+ tuning	

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

Troubleshooting OracleAS Adapter for CICS

Troubleshooting Oracle AS Adapter for CICS involve checking various definitions and properties in Oracle Connect, including daemon status, workspace options, server parameters, and various system logs.

This section contains the following topics:

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

Troubleshooting the Daemon

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

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

This section contains the following topics:

- [Starting the Daemon](#)
- [Shutting Down the Daemon](#)
- [Monitoring the Daemon During Runtime](#)
- [Daemon Logs](#)

Starting the Daemon

The daemon is started when OracleAS Adapter for CICS is installed. In case you have shut down the daemon, as described in ["Shutting Down the Daemon"](#) on page 6-2, you can restart the daemon as described in the following task.

Note: The daemon is started on the IBM OS/390 or z/OS platform. It cannot be started remotely using Oracle Studio.

Task: Starting the Daemon

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

'/s ATTDAMN'

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

See Also: ["Starting the Daemon"](#) on page 2-10 for details about the ATTDAMN JCL.

Shutting Down the Daemon

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

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. Select the computer defined in ["Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio"](#).
3. Right-click the computer and select **Open Runtime Perspective**.
4. In the Runtime Explorer, right-click the computer and select **Shutdown Daemon**.

Monitoring the Daemon During Runtime

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

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. Right-click the computer defined in ["Setting Up the IBM OS/390 or z/OS Platform in Oracle Studio"](#) in the Configuration Explorer and select **Open Runtime Perspective**.

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

Daemon (Computer) Options

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

The following options are available at the daemon level:

Table 6–1 Daemon Options

Option	Description
Edit Daemon Configuration	Opens the daemon editor, enabling you to reconfigure the daemon. See Also: "Advanced Tuning of the Daemon" for details about the configuration settings.
Status	Checks the status of the daemon. The information about the daemon includes the daemon name, configuration used, the active client sessions, and logging information.
Reload Configuration	Reloads the configuration after any changes. Any servers currently started are not affected by the changed configuration. See Also: "Advanced Tuning of the Daemon" for details about the configuration settings.
View Log	Displays the daemon log. For details see "Daemon Logs" on page 6-4.
View Events	Displays the daemon events log.

Table 6–1 (Cont.) Daemon Options

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

Workspace Options

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

The following table lists the available options:

Table 6–2 Workspace Options

Option	Description
Edit Workspace Configuration	Opens the daemon editor to enable you to reconfigure the workspace. See Also: " Advanced Tuning of the Daemon " for details about the configuration settings.
Status	Checks the status of the workspace, whether it is available or not.
View Log	Displays the log for all servers for the workspace. For details see " Daemon Logs " on page 6-4.
View Events	Displays the workspace events log.
Recycle Servers	Closes all unused servers and prepares all active servers to close when the client disconnects. New connection requests are allocated with new servers.
Kill Servers	Immediately closes all active and unused servers. Note: It is recommended to use this option with caution, as it may lead to data loss.
Refresh	Refreshes the display.

Server Options

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

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

Table 6–3 Server Options

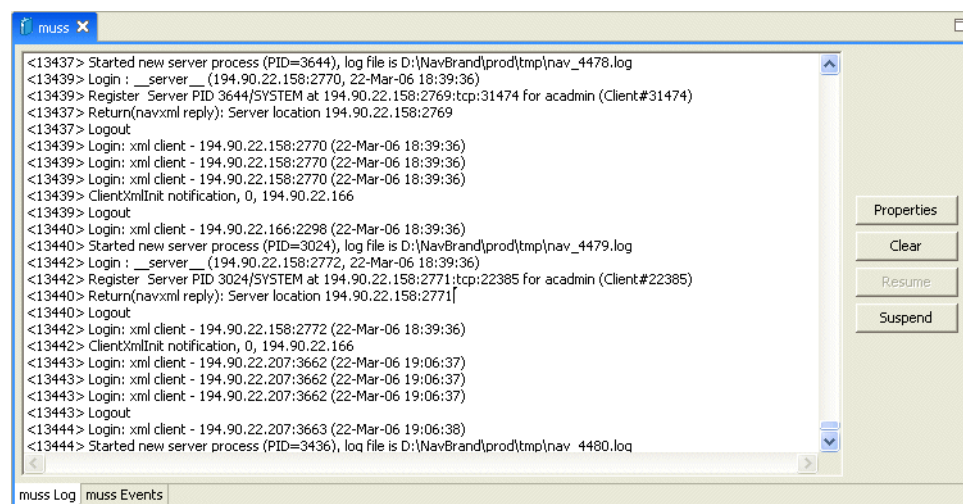
Option	Description
Status	Checks the status of the server. The information about the server includes the server mode and the number of active client sessions for the server.
View Log	Displays the server log. For details see "Daemon Logs" on page 6-4.
View Events	This option is for future use.
Kill server	Ends the server process, regardless of its activity status. Note: It is recommended to use this option with caution, as it may lead to data loss.
Refresh	Refreshes the display.

Daemon Logs

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

- Daemon log
- Workspace log
- Server process log

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



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

The Daemon Log Monitor

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

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

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

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

See Also: ["Daemon Logging"](#) on page A-3 for details.

The Workspace Log Monitor

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

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

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

See Also: ["WS Logging"](#) on page A-13.

The Server Log Monitor

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

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

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

See Also: ["WS Logging"](#) on page A-13.

Resolving Communication Errors

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

The following table describes the various scenarios that may exist when Oracle Application Server disconnects from the IBM OS/390 or z/OS computer.

Table 6–4 Scenarios When a Client Is Disconnected

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

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

- Daemon configuration settings.
- Oracle Connect security.
- TCP/IP subsystem. Oracle Application Server Adapter for CICS 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 computers involved and whether a portmapper is being used.

Resolving Specific Errors

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

C007: Server initialization failed.

Cause: The daemon failed to start its network service.

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

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

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

C008: Setting server event handler failed.

Cause: Internal error.

Action: Contact Oracle Support Services.

C009: IRPCD process has been terminated by user request.

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

Action: No action required.

C00A: Application %s not found.

Cause: The requested workspace does not exist.

Action: Check that the workspace defined in the oc4j-ra.xml file is also defined in the daemon configuration on the IBM OS/390 or z/OS platform. Use the **Status** option in the Runtime Manager perspective.

C00B: Invalid IRPCD client context.

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

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

C00C: Daemon request requires a server login.

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

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

C00D: Daemon request requires a client login.

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

Action: Reissue the command and specify a username and password.

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

See Also: ["Setting Up Run-Time User Access to the IBM OS/390 or z/OS Platform"](#) on page 2-17.

C00E: Daemon request requires an administrator login.

Cause: The requested daemon service requires an administrative login.

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

See Also: ["Daemon Security"](#) on page A-5.

C00F: Anonymous client logins are not allowed.

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

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

See Also: ["Daemon Security"](#) on page A-5.

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

See Also: ["Setting Up Run-Time User Access to the IBM OS/390 or z/OS Platform"](#) on page 2-17.

C010: Anonymous server logins are not allowed.

Cause: Internal error.

Action: Contact Oracle Support Services.

C011: Client has already timed out.

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

Action: Increase the Connect timeout value for the server workspace in the WS Info. tab of the daemon configuration.

See Also: ["WS Info."](#) on page A-7.

C012: Invalid username/password.

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

Action: See the daemon log file for the reason that the username/password were not accepted.

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

See Also: ["Setting Up Run-Time User Access to the IBM OS/390 or z/OS Platform"](#) on page 2-17.

Action: Make sure the daemon is started from an APF-authorized account that is allowed to check for system usernames and passwords.

C014: Client connection limit reached - try later.

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

Action: Increase the value of the Number of sub-tasks in the WS Server section of the daemon configuration.

See Also: ["WS Server"](#) on page A-9.

Action: Try running the command later.

C015: Failed to start server process.

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

Action: See the daemon and server logs for the reason the server did not start. For example, you might receive a 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 computer. For example:

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

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

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

C016: Unexpected server state.

Cause: Internal error.

Action: Contact Oracle Support Services.

C017: Active daemon clients exist. Shutdown canceled.

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

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

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

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

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

C01A: Lock %s not found.

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

Action: Contact Oracle Support Services.

C01B: Unexpected error in %s.

Cause: Internal error.

Action: Contact Oracle Support Services.

C01C: Cannot update configuration without _APPLICATIONS lock.

Cause: Internal error.

Action: Contact Oracle Support Services.

C01D: Need to lock the application first.

Cause: Internal error.

Action: Contact Oracle Support Services.

C01F: Cannot set configuration of a deleted application.

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: Cannot connect to the remote computer.

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

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

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

C021: Required variable %s not found

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

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

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

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

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

Action: Try to connect again.

Action: Increase the Connect timeout value for the server workspace in the WS Info. section of the daemon configuration.

See Also: ["WS Info."](#) on page A-7.

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

Action: Increase the value of the `Set maximum number of servers` and/or `Maximum` parameter for the `Clients per server limit` in the WS Server section of the daemon configuration.

See Also: ["WS Info."](#) on page A-7.

C023: Call made to unregistered module %d.

Cause: Internal error.

Action: Contact Oracle Support Services.

C024: Failed to create a socket.

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

Action: Check whether you have sufficient system privileges.

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

C025: Failed to set socket option %s

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

Action: Check whether you have sufficient system privileges.

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

C026: Failed to bind server to port %s

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

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

Action: Check whether you have sufficient system privileges.

C027: Cannot create TCP service for %s

Cause: An error occurred within the TCP/IP subsystem

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

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

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

Action: Enable the portmapper.

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

C029: Failed to create a server thread

Cause: Internal error.

Action: Contact Oracle Support Services.

C02A: Server thread failed to start

Cause: Internal error.

Action: Contact Oracle Support Services.

C02B: Stopping the %s server - no client

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

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

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

C02C: Unexpected event - a termination signal intercepted

Cause: Internal error.

Action: Contact Oracle Support Services.

C02D: Modified transport, context unknown/lost

Cause: Internal error.

Action: Contact Oracle Support Services.

C02F: Corrupted arguments passed to procedure

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

C031: Cannot register a non-module RPC %s

Cause: Internal error.

Action: Contact Oracle Support Services.

C032: An IRPCD program is required

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

C035: Out of memory

Cause: Not enough memory to service a client request.

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

C036: Failed to register RPC procedure module %s

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

C038: Cannot reregister RPC procedure number %x

Cause: Internal error.

Action: Contact Oracle Support Services.

C042: Remote call to %s failed; %s

Cause: Remote call to API failed.

Action: Check the daemon log file.

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

See Also: ["Daemon Logging"](#) on page A-3.

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

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

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

Action: Check that the daemon is up on the IBM OS/390 or z/OS platform. Use the Status option in the Runtime Manager perspective.

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

C045: Failed to create a service thread

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

Action: A system or process quota limit has been exceeded. Either increase the quota or lower the `Clients per server limit` field value in the WS Info. section of the daemon configuration.

See Also: ["WS Info."](#) on page A-7.

C047: %s out of memory

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

Action: Terminate unnecessary processes running on the server.

Action: Add more memory to the system.

Action: Allow the process to use more memory.

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

C066: Communication error with the server%s

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

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

Action: Check that the daemon is up on the IBM OS/390 or z/OS platform. Use the Status option in the Runtime Manager perspective.

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

C067: Unexpected error occurred in server function %s

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

Action: Contact Oracle Support Services.

C068: Fail to login daemon

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

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

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

C069: Fail to get server

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

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

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

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

Action: The server startup script failed.

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

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

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

C06A: Failed to connect to server

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

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

C06B: Disconnecting from server

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

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

C070: Server failed to send reply to the client

Cause: Server terminated unexpectedly.

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

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

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

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

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

C072: Reconnect to server %s

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

Action: No action is required.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: User/account has insufficient privileges.

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

See Also: ["Daemon Security"](#) on page A-5 or ["WS Security"](#) on page A-16.

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

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

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

E001: Failed in lock/release operation

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

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

J0006: Operation on already closed connection was requested

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

Action: Reopen the connection and try again.

J0028: Internal Error: Unknown XML tag %s

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

J0033: Internal Error: Attempt to work with closed socket

Cause: Internal error.

Action: Contact Oracle Support Services.

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

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

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

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

Cause: Internal error.

Action: Contact Oracle Support Services.

J0036: %s: %s

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

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

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

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: Internal error.

Action: Contact Oracle Support Services.

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

Cause: A request for metadata was not fulfilled.

Action: Check the validity of the request.

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

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

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

J0059: Value %s is invalid for attribute %s

Cause: A request for metadata was not fulfilled.

Action: Check the validity of the request.

J0068: Value must be of type string

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

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

J0069: Value must be of type MappedRecord

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

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

J0070: Value must be of type MappedRecord[]

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

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

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

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

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

J0072: Value must be of type Object[]

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

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

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

Cause: Encryption is not supported for nonpersistent connections.

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

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

Cause: The value pass.

Action: Change the argument used to a number.

Advanced Tuning of the Daemon

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

- [Daemon Control](#)
- [Daemon Logging](#)
- [Daemon Security](#)
- [Workspaces](#)

Daemon Control

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

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

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

Changes made to the daemon configuration are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager perspective.

The **Daemon Control** tab is shown in the following figure:

Figure 6–1 The Daemon Control tab

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

Table 6–5 Daemon Control tab Components

Field	Description
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	<p>The timeout period for <i>short</i> calls for all daemons. The definition of a short call is a call that should be completed in a few seconds. For example, most calls to a database such as DESCRIBE should be completed in a few seconds as opposed to call like a GETROWS call, which can take a long time. In heavily loaded or otherwise slow systems, even short calls such as calls to open a file, may take a significant amount of time. If a short call takes more than the specified time to complete, then the connection is stopped. The default value for this parameter is 60 seconds. Values of less than 60 seconds are considered to be 60 seconds.</p> <p>Specifying the timeout in a workspace overrides the value set in this field for that workspace.</p>

Table 6–5 (Cont.) Daemon Control tab Components

Field	Description
Connect timeout	<p>The time the client waits for a daemon server to start. If the daemon server does not start within this period, then the client is notified that the server did not respond. The value specified for this parameter serves as the default timeout for all the workspaces listed in the daemon configuration. The default value for this parameter is 60 seconds.</p> <p>Notes:</p> <ul style="list-style-type: none"> ■ Entering the timeout in a workspace overrides the value set in this field for that workspace. ■ Even if the XML source does not list this parameter in the workspace section, the workspace gets it using the default value. If you want to prevent a workspace from using the default value, you must enter a value of zero for this parameter in the workspace section.
Client idle timeout	<p>The maximum amount of time any daemon client may be idle before the connection with the server is closed.</p> <p>Specifying the timeout in a Workspace overrides this setting for that workspace.</p>

Daemon Logging

Using the **Daemon Logging** tab, you define the daemon log file settings, the log file structure and the location where the log is saved. In addition, use it to define the data that is logged and traced in the file.

The **Daemon Logging** tab is accessed as follows:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. Select the required computer from the Configuration Explorer.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Right-click the daemon in the Runtime Explorer and select **Edit Daemon Configuration**.
5. Click the **Daemon Logging** tab.
6. After making changes to the daemon, right-click the daemon and select **Reload Configuration**.

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

Changes made to the daemon configuration are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager perspective.

7. Right-click the daemon and select **End Unused Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **Daemon Logging** tab is shown in the following figure:

Figure 6–2 The Daemon Logging tab

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

Table 6–6 Daemon Logging tab Components

Field	Description
Daemon log file location	Specifies the daemon produces its log data. 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	Logs all daemon RPC function calls.
Daemon internal operations	Logs daemon internal operations.
Log trace information	Logs low-level RPC operations.
Display host and client domain name	Specifies whether the client host and domain name are logged rather than the client IP address. The default is false.
Trace options	Specifies the type of tracing being 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.

Table 6–6 (Cont.) Daemon Logging tab Components

Field	Description
System trace	Generates system-specific tracing of various operations.
Timing	Generates a timestamp for every entry to the server log file.
Binary XML log level	Sets the binary XML log level. Your options are: <ul style="list-style-type: none"> ■ debug ■ none (the default) ■ api ■ info
Server log filename format	Defines the name and location of the server log file. The field must specify the full path name. If no directory information is provided for the log file, then it will be located in the login directory of the account running Oracle Connect workstation.

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: server account login directory
- %P: server process ID
- %T: time (hhmmss)
- %U: server account name (username)

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

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

Daemon Security

The **Daemon Security** tab is used to:

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

The **Daemon Security** tab is accessed as follows:

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

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

Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.

7. Right-click the daemon and select **End Unused Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **Daemon Security** tab is shown in the following figure:

Figure 6–3 The Daemon Security tab

The screenshot shows the 'Daemon Security' tab in a software interface. At the top, under 'Administrator privileges', there are two radio buttons: 'All users' (selected) and 'Selected users only'. Below this is a large empty rectangular box. To the right of this box are four buttons: 'Add user', 'Add group', 'Rename', and 'Remove'. At the bottom of the tab, there is a 'Machine access' section with two checkboxes: 'Allow anonymous login' (checked) and 'Cached password' (unchecked). Below these are two text input fields: 'Encryption methods' (containing a single asterisk) and 'Domain name' (empty). At the very bottom of the window, there is a tab bar with four tabs: 'Daemon Control', 'Daemon Logging', 'Daemon Security' (which is the active tab), and 'Source'.

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

Table 6–7 Daemon Security tab Components

Field	Description
Administrators privileges	Identifies the users (accounts) allowed to perform administrative tasks (tasks that require administrative login).
All users	Enables all users to access the daemon and change the settings.

Table 6–7 (Cont.) Daemon Security tab Components

Field	Description
Selected users only	Identifies the names of users (accounts) and groups that can be administrators. ¹ If a user is not specified, the account from which the daemon was started is considered the administrator. Note that the daemon does not require the user to log in to the account on the system, but to log in to the daemon using the account name and password.
Machine access	Manages access to the computer.
Allow anonymous login	Whether workspaces allow anonymous logins (without user name/password entries). For the optimal level of security, keep this option unchecked and define a username for the Daemon Administrators parameter. If unchecked, then no workspace can have an anonymous client. If checked, then a particular workspace allows anonymous clients.
Cached password	Enables login passwords to be cached. This enhances performance by reducing login times for future connections from the same client in a session.
Encryption methods	Specifies the encryption method used to send information across the network. The default is an asterisk (*), meaning that all methods are acceptable. If an encryption method is specified, it must be used. The RC4 and DES3 protocols are currently supported.

¹ The name is prefixed with '@', to utilize the operating system GROUP feature.

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 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](#)
- WS Governing: This tab is not applicable for use with OracleAS Adapters for CICS

WS Info.

Using the **WS Info.** tab, you specify the features that control the operation of the workspace, such as the server type, the command procedure used to start the workspace and the binding configuration associated with this workspace.

The **WS Info.** tab is accessed as follows:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. Select the required computer from the Configuration Explorer.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Expand the Daemons node to display the workspaces in the Runtime Explorer.

5. Right-click the workspace and select **Edit Workspace Configuration**. The WS Info. tab opens.
6. After making changes to the workspace, right-click the daemon and select **Reload Configuration**.

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

Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.

7. Right-click the daemon and select **End Unused Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **WS Info.** tab is shown in the following figure:

Figure 6–4 The WS Info tab

The screenshot displays the 'WS Info' tab within a configuration window. The fields are as follows:

- Workspace name:** Navigator
- Description:** Default Workspace
- Startup script:** (empty text box)
- Server type:** native
- Workspace binding name:** (empty text box)
- Workspace database name:** (empty text box)
- Timeout parameters:**
 - Client idle timeout:** 0
 - Connect timeout:** 60

The bottom of the window shows a series of tabs: Daemon Control, Daemon Logging, Daemon Security, **WS Info**, WS Server, WS Logging, WS Security, WS Governing, and Source.

The **WS Info.** tab comprises fields, as listed in the following table:

Table 6–8 WS Info tab Components

Field	Description
Workspace name	The name used to identify the workspace. Note: The default configuration includes the default Navigator workspace. This workspace is automatically used if a workspace is not specified as part of the connection settings.
Description	A description of the workspace.
Startup script	The full path name of the script that starts the workspace server processes. The script specified here must always activate the <code>nav_login</code> procedure and then run the server program (<code>svc</code>). If you do not specify the directory, the startup procedure is taken from the directory where the daemon resides. Oracle Connect includes a default startup script, which it is recommended to use.
Server type	This field is not applicable for use with OracleAS Adapters for CICS.

Table 6–8 (Cont.) WS Info tab Components

Field	Description
Workspace binding name	This field is not applicable for use with OracleAS Adapters for CICS.
Timeout parameters	<p>The time the client waits for the workspace server to start. If the workspace server does not start within this period, then the client is notified that the server did not respond. Specifying the timeout here overrides the default setting, specified in the Control section.</p> <p>See Also: "Daemon Control" on page A-1 for details about the Daemon Control section.</p>
Client idle timeout	The maximum amount of time a workspace client can be idle before the connection with the server is closed.
Connect timeout	The time the client waits for a workspace server to start. If the workspace server does not start within this period, then the client is notified that the server did not respond.

WS Server

Using the **WS Server** tab, 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** tab is accessed as follows:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. Select the required computer from the Configuration Explorer.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Expand the Daemons node to display the workspaces in the Runtime Explorer.
5. Right-click the workspace and select **Edit Workspace Configuration**.
6. Click the **WS Server** tab.
7. After making changes to the workspace, right-click the daemon and select **Reload Configuration**.

Notes:

- You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Edit Daemon**.
 - Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.
-

8. Right-click the daemon and select **End Unused Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **WS Server** tab is shown in the following figure:

Figure 6–5 The WS Server tab

The screenshot shows the 'WS Server' tab of a configuration window. At the top, 'Workspace server mode' is set to 'multiClient'. Below this are two sections: 'Reuse limit' and 'Clients per server limit'. 'Reuse limit' has radio buttons for 'None' and 'Maximum: 50' (selected). 'Clients per server limit' has radio buttons for 'None' (selected) and 'Maximum: 0'. The 'Server availability' section contains 'Initial number of servers: 1' and 'Minimum number of available servers: 0', with checkboxes for 'Keep when daemon ends' and 'Set maximum number of servers: 0'. The 'Resource limitations' section has a checkbox for 'Limit number of active servers: 0'. The 'Server priority' section has radio buttons for 'Use default priority' (selected) and 'Use priority specified: 0'. At the bottom is a tab bar with the following tabs: Daemon Control, Daemon Logging, Daemon Security, WS Info, WS Server (selected), WS Logging, WS Security, WS Governing, and Source.

Workspace server mode
multiClient

Reuse limit	Clients per server limit
<input type="radio"/> None	<input checked="" type="radio"/> None
<input checked="" type="radio"/> Maximum: 50	<input type="radio"/> Maximum: 0

Server availability	
Initial number of servers: 1	Minimum number of available servers: 0
<input type="checkbox"/> Keep when daemon ends	<input type="checkbox"/> Set maximum number of servers: 0

Resource limitations
<input type="checkbox"/> Limit number of active servers: 0

Server priority
<input checked="" type="radio"/> Use default priority
<input type="radio"/> Use priority specified: 0

Tab Bar
Daemon Control Daemon Logging Daemon Security WS Info WS Server WS Logging WS Security WS Governing Source

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

Table 6–9 WS Server tab Components

Field	Description
Workspace server mode	<p>Specifies the type of new server processes that the daemon starts up. The daemon supports the following server modes:</p> <ul style="list-style-type: none"> ■ singleClient: Each client receives a dedicated server process. The account in which a server process runs is determined either by the client login information or by the specific server workspace. This mode enables servers to run under a particular user account and isolates clients from each other, as each receives its own process. However, this server mode incurs a high overhead due to process startup times and can use a lot of server resources as it requires as many server processes as concurrent clients. ■ multiClient: Clients share a server process and are processed serially. This mode has low overhead because the server processes are 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. ■ multiThreaded: This mode is not applicable for use with OracleAS Adapter for CICS. ■ reusable: An extension of single-client mode. Once the client processing finishes, the server process does not die and can be used by another client, reducing startup times and application startup overhead. This mode does not have the high overhead of single-client mode because the servers are already initialized. However, this server mode can use a lot of server resources as it requires as many server processes as concurrent clients. 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	<p>Sets the maximum number of times a particular server can be reused. A one-client server can be reused after its (single) client has disconnected. Reuse of servers enhances startup performance because it avoids the need to repeat initialization. The default for this field is none (0), indicating that server reuse is unlimited. This parameter is disabled only if the server mode value is singleClient.</p>
Clients per server limit	<p>Sets the maximum number of clients a server process for the current workspace accepts. The default for this field is none (0), indicating that the number of clients for each server is unlimited. This field is enabled only if the server mode value is multiClient or multiThreaded.</p>

Table 6–9 (Cont.) WS Server tab Components

Field	Description
Server availability	<p data-bbox="613 268 1154 323">Specifies the number of servers in a pool of servers, available to be assigned to a client.</p> <p data-bbox="613 336 992 363">The following options are available:</p> <ul style="list-style-type: none"> <li data-bbox="613 375 1227 583">■ Initial number of servers: The number of server processes that are prestarted for this workspace when the daemon starts up. When the number of available server processes drops lower than the value specified in the Minimum number field, the daemon again starts server processes until this number of available server processes is reached. The default for this field is 0. <li data-bbox="613 602 1227 810">■ Minimum number: The minimum number of server processes in the prestarted pool before the daemon resumes creating new server processes (to the value specified in the Initial number of servers field). If this field is set to a value higher than the Initial number of servers field, the daemon uses the value specified in the Initial number of servers field. The default for this field is 0. <li data-bbox="613 829 1227 1062">■ Keep when daemon ends: When a daemon is shutdown, all the servers started by that daemon are also killed, even if they are active. Set this field to <code>true</code> if you want the servers for the workspace to remain active, even after the daemon has been shut down. If this field is set to <code>true</code>, it is the responsibility of the system operator or manager to ensure that the servers are eventually killed. This must be done at the system level. <li data-bbox="613 1081 1227 1337">■ Set maximum number of servers: The maximum number of available server processes. Once this number is reached, no new nonactive server processes are created for the particular workspace. For example, if a number of server processes are released at the same time, so that there are more available server processes than specified by this field, the additional server processes higher than this value are terminated. The default for this field is zero, meaning that there is no maximum.

Table 6–9 (Cont.) WS Server tab Components

Field	Description
resource limitations	<p>Specifies the number of servers that can be in use at any one time. The more servers used, the greater the system resources that are used.</p> <p>The following options are available:</p> <ul style="list-style-type: none"> ■ 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	<p>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.</p> <p>The following priority options are available:</p> <ul style="list-style-type: none"> ■ 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 the **WS Logging** tab, you specify parameters to log, that occur with the workspace server process.

The **WS Logging** tab is accessed as follows:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. Select the required computer from the Configuration Explorer.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Expand the Daemons node to display the workspaces in the Runtime Explorer.
5. Right-click the workspace and select **Edit Workspace Configuration**.
6. Click the WS Logging tab.
7. After making changes to the workspace, right-click the daemon and select **Reload Configuration**.

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

Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.

8. Right-click the daemon and select **End Unused Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **WS Logging** tab is shown in the following figure:

Figure 6–6 The WS Logging tab

The screenshot shows the 'WS Logging' configuration window. At the top, there is a checkbox labeled 'Specific log file format:' which is checked, followed by a text field containing 'Nav_%.i.log'. Below this is a section titled 'Trace options' containing several checkboxes: 'No timeout', 'Call trace', 'RPC trace', 'Sockets', 'Extended RPC trace', 'System trace' (which is checked), and 'Timing'. The next section is 'Event Information:', which contains three sub-sections: 'Logging' with radio buttons for 'none' (selected), 'error', and 'debug'; 'Server' with checkboxes for 'Connect' and 'Disconnect'; and 'Client' with checkboxes for 'Connect' and 'Disconnect'. The final section is 'Audit' with radio buttons for 'None' (selected), 'Statistics', 'Headers', and 'Detailed'.

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

Table 6–10 WS Logging tab Components

Field	Description
Specific log file format	<p>Defines the name and location of the server log file if you want the data written to a file instead of SYSOUT for the server process. The parameter must specify the name and the high level qualifier.</p> <p>The following tokens can appear in the log file template and will be replaced accordingly:</p> <ul style="list-style-type: none"> ■ %A: workspace name ■ %D: date (yymmdd) ■ %I: instance number of the given workspace server ■ %L: server account's login directory ■ %P: server's process ID ■ %T: time (hhmmss) ■ %U: server's account name (username)
Trace options	<p>Specifies the type of tracing to be performed. The following tracing options are available:</p> <ul style="list-style-type: none"> ■ No timeout: 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.
Logging	<p>Specifies the level of events that are logged for the workspace. The following event levels are available:</p> <ul style="list-style-type: none"> ■ none: The event log only displays the IP addresses of client that have logged in and out from the workspace. ■ error: The event log displays the IP addresses of client that have logged in and out from the workspace as well as any errors that have been generated. ■ debug: The event log displays the IP addresses of client that have logged in and out from the workspace as well as any errors that have been generated and all trace results that were specified in the Daemon Logging tab.

Table 6–10 (Cont.) WS Logging tab Components

Field	Description
Server	Specifies the server connection events to log. The following server events are available: <ul style="list-style-type: none"> ■ Connect: The event log displays the server connection events. ■ Disconnect: The event log displays the server disconnect events.
Client	Specifies the type of tracing performed. The following client events are available: <ul style="list-style-type: none"> ■ Connect: The event log displays the client connection events. ■ Disconnect: The event log displays the client disconnect events.
Audit	This group is not applicable for use with OracleAS Adapter for CICS.

WS Security

Using the **WS Security** tab, you specify the level of security at the workspace level, as opposed to the daemon level, which is set in the **Daemon Security** tab.

See Also: ["Daemon Security"](#) on page A-5 for details about security.

The **WS Security** tab is used to:

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

The **WS Security** tab is accessed as follows:

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

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

Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.

8. Right-click the daemon and select **End Unused Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **WS Security** tab is shown in the following figure:

Figure 6–7 The WS Security tab

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

Table 6–11 WS Security tab Components

Field	Description
Administration	Defines the users (accounts) allowed to perform administrative tasks (tasks that require administrative login) on this workspace.
Administrator privileges	<p>Identifies the users (accounts) with administrator privileges. The following options are available:</p> <ul style="list-style-type: none"> ■ All users: Indicates that anyone can access the workspace and change the settings. ■ Selected users only: The names of users (accounts) and groups that can be administrators. <p>Note: If a user is not specified here, the user specified in the Workspace users area will have administrator rights for this workspace. In this case, if all users are selected for the Workspace users area, then all users have administrator rights for this workspace.</p> <ul style="list-style-type: none"> ■ Allow Listing: Determines whether this workspace appears in the list of workspaces.
Workspace account	Defines the users (accounts) allowed to access the workspace, firewall access ports, workspace account, and anonymous login permissions.

Table 6–11 (Cont.) WS Security tab Components

Field	Description
Workspace users	<p>Lists the users who are allowed to use the workspace.</p> <ul style="list-style-type: none">■ All users: Indicates that any user who has logged on to the daemon can use the workspace.■ Selected users only: Specifies users (accounts) and groups that can use the workspace. Note: If a user is not specified, any user who has logged on to the daemon can use the workspace.■ Enable ports range: Defines 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 can be accessed through a firewall.■ Use specific workspace account: Defines 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: Defines whether this workspace can be accessed without authentication (user name/password). If anonymous login is allowed, specify the server account name to use. If this field is not specified, then the value in the Workspace account field is used.

Globalization Settings

OracleAS Adapters for CICS provides the globalization support for the following languages:

- Arabic
- English (the default)
- French
- German
- Greek
- Hebrew
- Italian
- Japanese
- Korean
- Portuguses
- Simple Chinese
- Spanish
- Traditional Chinese
- Turkish

This appendix describes how to define the language support.

Defining the Language and Codepage

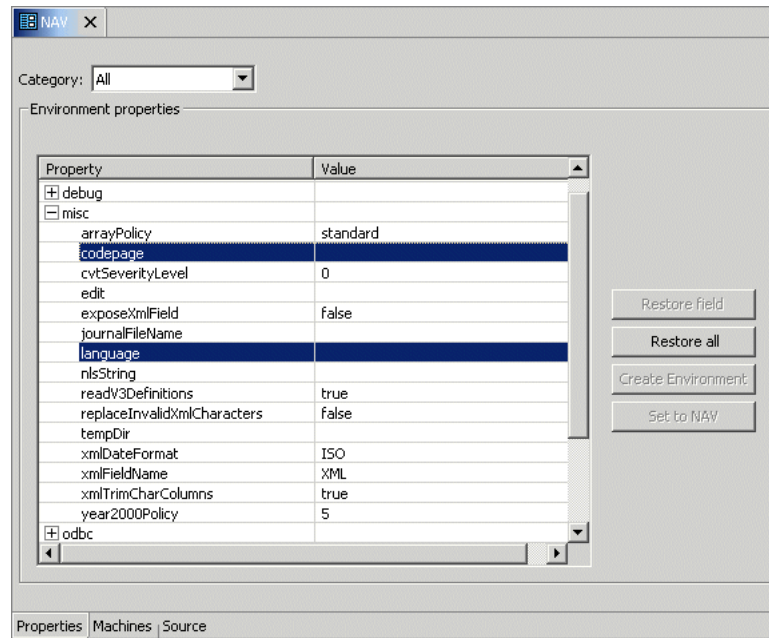
The language and codepage parameters are accessed from the computer where Oracle Studio is installed.

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

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. Select and expand the required computer node from the Configuration Explorer.
3. Expand the Binding node.
4. Right-click **NAV** and select **Edit Binding**.
5. In the binding **Properties** tab, expand the **misc** node.

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

Figure 6–8 The Properties tab



6. Enter a valid value for the language. See [NLS Language Codes](#)
7. Optionally, in the **codepage** field, specify the codepage required.

You can skip this step, and specify a language (see the previous step). In this case, a default codepage is used.

The following table lists the codepages:

Table 6–12 NLS Language Codes

Language Name	Language Code	ASCII Platforms (Default)	EBCDIC Platforms (Default)	Alternative Codepages (EBCDIC based unless noted otherwise)
Arabic	ARA	ISO-8859-6	IBM420	Windows-1256 (ASCII based)
Chinese - Simplified	SCHI	GB2312	IBM935	-
Chinese - Traditional	TCHI	BIG5	IBM937	-
English UK	ENUK	ISO-8859-15	IBM1146	IBM037, IBM500, IBM1140, IBM1148 ISO-8859-1 (ASCII based)
English US	ENUS	ISO-8859-15	IBM1140	IBM500, IBM1148 ISO-8859-1 (ASCII based)
French	FRE	ISO-8859-15	IBM1147	IBM037, IBM500, IBM1140, IBM1148 ISO-8859-1 (ASCII based)

Table 6–12 (Cont.) NLS Language Codes

Language Name	Language Code	ASCII Platforms (Default)	EBCDIC Platforms (Default)	Alternative Codepages (EBCDIC based unless noted otherwise)
German	GER	ISO-8859-15	IBM1141	IBM037, IBM500, IBM1140, IBM1148 ISO-8859-1 (ASCII based)
Greek	GRK	ISO-8859-7	IBM875	-
Hebrew	HEB	ISO-8859-8	IBM424	-
Italian	ITL	ISO-8859-15	IBM1144	IBM037, IBM500, IBM1140, IBM1148 ISO-8859-1 (ASCII based)
Japanese	JPN	SJIS EUC (Solaris) VMS-JP (VMS)	IBM939	-
Korean	KOR	KSC5601 MS949 (Win)	IBM933	MS949, EUC-KR (both ASCII based)
Latin International	LAT	ISO-8859-15	IBM1148	IBM037, IBM1140 ISO-8859-1 (ASCII based)
Portuguese	POR	ISO-8859-15	IBM1140	IBM500, IBM1148 ISO-8859-1 (ASCII based)
Russian	RUS	ISO-8859-5	IBM1154	-
Spanish	SPA	ISO-8859-15	IBM1145	IBM037, IBM500, IBM1140, IBM1148 ISO-8859-1 (ASCII based)
Turkish	TUR	ISO-8859-9	IBM1155	IBM857

Back-end Adapter Data Type Support

OracleAS Adapters for CICS supports a number of data types that are used when defining metadata in Oracle Studio. The data types are mapped from the COBOL data types during the import procedure.

Note: The mapping of data types between OracleAS Adapters for CICS and Oracle Application Server is performed internally by Oracle Connect.

Data Type Mapping

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

Table 6–13 Data Type Mapping: COBOL and Oracle Connect Back-end Adapter

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

Table 6–13 (Cont.) Data Type Mapping: COBOL and Oracle Connect Back-end Adapter

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

Index

A

acxTrace parameter, 5-7
analyzerQueryPlan parameter, 5-7
application language parameter, 5-7

B

buffer size, comCacheBufferSize parameter, 5-6

C

CD-ROM drive requirements
 IBM S/390, 2-2
 PC, 2-2
codepage parameter, 5-7
comCacheBufferSize parameter, 5-6
comm parameters
 comCacheBufferSize, 5-6
 comMaxSocketSize, 5-6
 comMaxXmlSize, 5-6
comMaxSocketSize parameter, 5-6
comMaxXmlSize parameter, 5-6
cvtSeverityLevel parameter, 5-7

D

daemon
 logging, A-3
 security, A-5
 server modes, 5-3
 shutting down, 6-2
 starting, 6-1
 timeout, 6-6
data types
 atomic metadata, C-1
 cvtSeverityLevel, 5-7
 cvtSeverityLevel parameter, 5-7
 nlsString parameter, 5-8
debug parameters
 acxTrace, 5-7
 analyzerqueryPlan, 5-7
 environment, 5-6
 generalTrace, 5-7
 logFile, 5-7
 oledbTrace, 5-7
 optimizerTrace, 5-7

 queryWarnings, 5-7
 traceDir, 5-7
disk space requirements
 IBM S/390, 2-2
 PC, 2-2

E

environment parameters
 acxTrace, 5-7
 analyzerQueryPlan, 5-7
 codepage, 5-7
 comCacheBufferSize, 5-6
 comMaxSocketSize, 5-6
 comMaxXmlSize, 5-6
 cvtSeverityLevel, 5-7
 debug, 5-6
 generalTrace, 5-7
 language, 5-7
 logFile, 5-7
 miscellaneous, 5-7
 nlsString, 5-8
 odbc, 5-8
 oledb, 5-8
 oledbTrace, 5-7
 optimizer, 5-8
 optimizerTrace, 5-7
 queryProcessor, 5-8
 queryWarnings, 5-7
 tempDir, 5-8
 traceDir, 5-7
 transactions, 5-9
 tuning, 5-9
 year2000Policy, 5-8
error log, logFile parameter, 5-7

G

generalTrace parameter, 5-7
generating interactions
 inbound, 2-25
 outbound, 2-37

H

hardware requirements

- IBM S/390 CD-ROM drive, 2-2
- IBM S/390 disk space, 2-2
- IBM S/390 memory, 2-2
- PC CD-ROM drive, 2-2
- PC disk space, 2-2
- PC memory, 2-2
- PC processor, 2-2
- UNIX, 2-2

I

- IBM S/390
 - installing, 2-5
 - postinstallation, 2-8
 - preinstallation, 2-4
- inbound interactions, generating, 2-25
- installing
 - IBM S/390, 2-5
 - Windows, 2-13
- interactions
 - inbound, 2-25
 - outbound, 2-37

L

- language parameter, 5-7
- log files
 - daemon options, A-3
 - logFile parameter, 5-7
- logFile parameter, 5-7
- logging
 - daemon configuration, A-3
 - optimizer strategy, 5-7
 - trace information, 5-7

M

- maximum size, XML documents, 5-6
- memory requirements
 - IBM S/390, 2-2
 - PC, 2-2
- metadata, atomic data types, C-1
- Microsoft software requirements, 2-3
- miscellaneous parameters
 - codepage, 5-7
 - cvtSeverityLevel, 5-7
 - environment, 5-7
 - language, 5-7
 - nlsString, 5-8
 - tempDir, 5-8
 - year2000Policy, 5-8

N

- nlsString parameter, 5-8

O

- odbc environment parameters, 5-8
- oledb environment parameters, 5-8
- oledbTrace parameter, 5-7

- operating system requirements
 - PC, 2-3
 - UNIX, 2-2
- optimizer
 - environment parameters, 5-8
 - optimizerTrace parameter, 5-7
 - traceDir parameter, 5-7
 - writing plan to file, 5-7
- optimizerTrace parameter, 5-7
- Oracle Application Server requirements, 2-2
- outbound interactions, generating, 2-37

P

- postinstallation, IBM S/390, 2-8
- preinstallation, IBM S/390, 2-4
- processor requirements
 - PC, 2-2
 - UNIX, 2-2

Q

- query optimizer
 - logging strategy, 5-7
 - traceDir parameter, 5-7
 - writing plan to file, 5-7
- queryProcessor environment parameters, 5-8
- queryWarnings parameter, 5-7

R

- requirements
 - PC hardware, 2-2
 - PC software, 2-3
 - UNIX hardware, 2-1
 - UNIX software, 2-2

S

- security configuration, daemon, A-5
- servers
 - configuring modes, 5-3
 - reusable, 5-4
 - Reuse limit daemon parameter, 5-4, A-11
- SMS
 - See system managed storage
- sockets, comMaxSocketSize parameter, 5-6
- software requirements
 - Microsoft, 2-3
 - Oracle Application Server, 2-2
 - PC operating system, 2-3
 - UNIX operating system, 2-2
- system managed storage, 2-4

T

- tempDir parameter, 5-8
- temporary files, 5-8
- timeout
 - client idle, 6-6
 - daemon, 6-6

trace information, logging, 5-7
traceDir parameter, 5-7
transactions environment parameters, 5-9
tuning environment parameters, 5-9

U

UNIX

hardware requirements, 2-1
software requirements, 2-2

W

Windows, installing, 2-13
Workspace server mode, A-11

X

XML documents, maximum size, 5-6

Y

Y2K

See year2000Policy parameter
year2000Policy parameter, 5-8

