# Contents

## Preface

- Audience .......................................................... ix
- Documentation Accessibility ........................................... ix
- Related Documents ....................................................... x
- Conventions ................................................................. x

## 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

- IBM z/OS Hardware and Software Requirements .................. 2-1
  - Hardware Requirements ........................................... 2-1
  - Software Requirements .............................................. 2-2

### Installing Oracle Connect on an IBM z/OS Series Platform

- Installation Worksheet ................................................. 2-3
- Preinstallation Instructions ............................................ 2-4
- Installing the Kit ......................................................... 2-4
- Installation Instructions ................................................... 2-5
- Post-Installation Instructions ............................................ 2-8
  - Post-Installation Procedures ...................................... 2-8
  - Starting the Daemon .................................................. 2-9
  - Setting Up Oracle Connect for Reentrancy ................. 2-10

### Updating an Existing Oracle Connect Installation with CICS

### Installing Oracle Studio

- Oracle Studio Requirements ........................................... 2-11
- Installing Oracle Studio on Windows .............................. 2-12
- Installing Oracle Studio on Linux ................................. 2-12
  - Installing Oracle Studio with the Wizard (SH installation) ...................... 2-12
  - Installing with a Silent Installation (RPM) .................................. 2-12
Configuring Oracle Connect ................................................................. 2-12
  Setting Up the IBM z/OS Platform in Oracle Studio .......................... 2-13
  Securing Access to Oracle Connect .................................................. 2-14
  Setting Password Access to Oracle Studio ...................................... 2-14
  Specifying Users with Administrative Rights .................................... 2-15
  Setting Up Run-Time User Access to the IBM z/OS Platform .......... 2-16
Modeling Interactions for OracleAS Adapter for CICS ....................... 2-17
  Configuring an Oracle Connect Adapter ......................................... 2-17
  Generating Outbound Interactions .................................................. 2-19
  Generating Inbound Interactions ..................................................... 2-25
Modeling Interactions for OracleAS Adapter for CICS Queue .......... 2-31
  Preliminary Considerations ............................................................ 2-31
  Configuring the CICS Queue Adapter ............................................. 2-37
  Generating Outbound Interactions .................................................. 2-38
  Viewing the XML Schema ............................................................... 2-42
  Creating XML Schemas ................................................................. 2-42

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 Oracle Net Services Administrator’s Guide BPEL Process Manager Examples

  Getting Started .................................................................................. 5-1
  Deploy a BPEL Outbound Process ..................................................... 5-2
    Beginning Tasks ............................................................................... 5-2
    Design-Time Configuration ............................................................. 5-3
    Create a BPEL Project for a BPEL Outbound Process ..................... 5-3
    Create a Database Adapter to Read the Doctor Data ...................... 5-4
    Design a BPEL Outbound Process to Read the String ..................... 5-5
    Create a Partner Link to the Database Adapter ............................... 5-6
6 Configuring an OracleAS Adapter for CICS for Outbound Interactions

Creating Outbound Interactions with the OracleAS Adapter for CICS ............................................. 6-1
  Requirements ......................................................................................................................................... 6-1
Create the findDoctor and findPatient Interactions ........................................................................... 6-1
  Prepare the System .............................................................................................................................. 6-2
Set up Machine Access to Oracle Connect ............................................................................................. 6-2
Add a CICS Adapter ............................................................................................................................. 6-2
Import the Metadata ............................................................................................................................... 6-3

7 Troubleshooting OracleAS Adapter for CICS

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

8 Advanced Features of OracleAS Adapters for CICS

Configuring the Daemon for High Availability ...................................................................................... 8-1
  Adding a New Daemon Workspace Configuration .................................................................................. 8-1
  Editing the Workspace .......................................................................................................................... 8-2
  Configuring the Server Mode ................................................................................................................ 8-2
Configuring a Binding Environment ....................................................................................................... 8-6
  Debug .................................................................................................................................................. 8-7
  General .................................................................................................................................................. 8-8
  Language .............................................................................................................................................. 8-8
  Modeling .............................................................................................................................................. 8-9
  ODBC ............................................................................................................................................... 8-9
  OLEDB .............................................................................................................................................. 8-9
  Optimizer ............................................................................................................................................ 8-9
  Query Processor ................................................................................................................................. 8-9
  Parallel Processing .............................................................................................................................. 8-9
  Transactions ....................................................................................................................................... 8-9
  Tuning ................................................................................................................................................ 8-10
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.

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 to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.
Accessibility of Code Examples in Documentation
Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

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

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

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

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>boldface</td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td>italic</td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Oracle Application Server (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 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 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 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 back 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 or Linux computer. 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:

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

**Note:** In addition to the installation procedures described in this section, the J2CA 1.5 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 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 z/OS Hardware and Software Requirements

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

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

- Hardware Requirements
- Software Requirements

**Hardware Requirements**

The following table summarizes the hardware requirements for Oracle Connect.
Installing Oracle Connect on an IBM z/OS Series Platform

Software Requirements
The following table summarizes the software requirements for Oracle Connect.

Table 2–2 Oracle Connect Software Requirements

<table>
<thead>
<tr>
<th>Software Component</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>IBM z/OS</td>
</tr>
<tr>
<td>CICS TP Monitor</td>
<td>V4R1 or higher (recommended to use CICS V6R1 or higher)</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>The operating system must be able to use the TCP/IP protocol for using the Internet.</td>
</tr>
<tr>
<td>C Runtime Library</td>
<td>The C runtime library has all the standard C runtime programs.</td>
</tr>
<tr>
<td>Oracle Application Server</td>
<td>Oracle Application Server 10g (10.1.3.4)</td>
</tr>
</tbody>
</table>

Installing Oracle Connect on an IBM z/OS Series Platform

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

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

**Note:** If you have an Oracle Connect back-end adapter or CDC adapter already installed on the IBM z/OS platform, then follow the instructions described in "Updating an Existing Oracle Connect Installation with CICS" on page 2-10.
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 2–3 Preinstallation Information

<table>
<thead>
<tr>
<th>Topic</th>
<th>Required Information</th>
<th>Default</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Operating system</td>
<td>-</td>
<td>z/OS</td>
</tr>
<tr>
<td></td>
<td>Disk space</td>
<td>-</td>
<td>150 cylinders</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
<td>-</td>
<td>The minimum requirement is 4MB for each connection. A connection is defined as a connection to a server process or daemon. The actual memory requirement depends on such things as the size of the database and the number of databases accessed.</td>
</tr>
<tr>
<td></td>
<td>Installation high-level qualifier</td>
<td>OCL10134</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Volume</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Unit</td>
<td>3390</td>
<td>SMS only: unit where SMS resides.</td>
</tr>
<tr>
<td></td>
<td>Output class</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>JCL job card</td>
<td>-</td>
<td>An optional card (up to 6 lines) to replace the prefix job (entered as it will appear in the job)</td>
</tr>
<tr>
<td>CICS</td>
<td>ISPF load library name</td>
<td>ISP.SISPLOAD</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>CICS EXCI load library name</td>
<td>CICS.CICS.SDFHEXC1</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 2–4 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 2–5 Installation Checklist

<table>
<thead>
<tr>
<th>Step</th>
<th>Comment/Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>tso profile noprefix</td>
<td>Ensures that the user name is not used as part of the dataset name allocated in the next steps</td>
</tr>
<tr>
<td>Allocate dataset: {HLQ}.TRANSMIT.KIT</td>
<td>130 tracks (3390), format=FB, record length=80, block size=3120</td>
</tr>
<tr>
<td>Allocate dataset: {HLQ}.TRANSMIT.LOAD</td>
<td>500 tracks (3390), format=FB, record length=80, block size=3120</td>
</tr>
<tr>
<td>FTP files to z/OS</td>
<td>FTP using binary mode</td>
</tr>
<tr>
<td>RECEIVE INDSNAME('{HLQ}.TRANSMIT.KIT')</td>
<td>-</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Comment/Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>da('([HLQ].TRANSMIT.LIB) UNIT(unit) VOLUME(volume)') -</td>
<td></td>
</tr>
<tr>
<td>EX {HLQ}.TRANSMIT.LIB(PREPARE)</td>
<td>Successful MAXCC is 0, 4 or 8</td>
</tr>
<tr>
<td></td>
<td>BUILDKIT.SRC, BUILDKIT.LOAD, and</td>
</tr>
<tr>
<td></td>
<td>BUILDKIT.GENDEMO created</td>
</tr>
<tr>
<td>EX {HLQ}.BUILDKIT.SRC(NAVINST)</td>
<td>Successful MAXCC is 0 or 4</td>
</tr>
</tbody>
</table>

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

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

These datasets are provided 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:

   - OCL10134.TRANSMIT.KIT = 130 tracks (3380 and 3390 disks)
   - OCL10134.TRANSMIT.LOAD = 500 tracks (3380 and 3390 disks)

   For each dataset: RECFM=FB and LRECL=80. The block size is 3120.
3. Using FTP, copy OCL10134.TRANSMIT.KIT and OCL10134.TRANSMIT_LOAD in the binary mode to the mainframe. You can replace the OCL10134 high-level qualifier to any qualifier you want.

**Installation Instructions**

Perform the following steps to install Oracle Connect:

1. Run the following command at the TSO prompt:

   \[ \text{RECEIVE INDSTYPE('nnn.TRANSMIT.KIT')} \]

   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 OCL10134.

   **Note:** You can use more than one high-level qualifier, such as ACME.DEV.OCL10134, with the following conditions:
   - The total length must be less than or equal to twenty characters.
   - The words transmit and buildkit cannot be used as high-level qualifiers.

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

   \[ \text{da('nnn.TRANSMIT.LIB') [UNIT(unit) VOLUME(volume)]} \]

   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:

   \[ \text{ex PREPARE} \]

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

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

Table 2–7  Installation Prompts and Responses

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

```
   ex CICS
```

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

<table>
<thead>
<tr>
<th>Table 2–8 CICS Adapter-Specific Installation Prompts and Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Screen</strong></td>
</tr>
<tr>
<td>ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHEXCI]:</td>
</tr>
<tr>
<td>PLEASE CONFIRM (YES/NO/QUIT) [YES]:</td>
</tr>
<tr>
<td>ENTER THE ISPF LOAD LIBRARY NAME [ISPISIPLOAD]:</td>
</tr>
<tr>
<td>PLEASE CONFIRM (YES/NO/QUIT) [YES]:</td>
</tr>
<tr>
<td>ENTER THE OUTPUT CLASS FOR INSTALLATION OUTPUT [A]:</td>
</tr>
<tr>
<td>DO YOU WANT TO USE THE DEFAULT JOB CARD Y/N [Y]</td>
</tr>
</tbody>
</table>
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.

**Post-Installation Instructions**

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

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

**Post-Installation Procedures**

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

- Allocate a dataset for `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 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=vol002
```

Where `vol002` 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(ATTDAEMN)` and `INSTROOT.USERLIB(ATTSRVR)` members to any active proclib, such as user.proclib. The `ATTDAEMN` and `ATTSRVR` members are run as started tasks.

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

  - Run `INSTROOT.USERLIB(NAVCMD)` and enter `EDIT DAEMON IRPCDINI` at the prompt.

  - Change the `startupScript` parameter from `ATTSRVR` to the new name for the server:

    ```
    <Workspace name="Navigator"
    ```
Installing Oracle Connect on an IBM z/OS Series Platform

startupScript="NEW_NAME"
serverMode="reusable"

- Exit and save the changes.

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

  For example, before:

  // PARM='^B START IRPCDINI'

  After:

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

  Where \textit{ip_address} is ip address of the computer, 2551 is the default port for starting the daemon and \textit{IRPCDINI} is the default daemon configuration.

- The ATTDAEMN and ATTSSRVR 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 ATTSSRVR with a started task class and a general profile that enables the following:
  
  - Permission to issue master console commands.
  - \textsc{START} authority for the ATTSSRVR job.
  - Access to an Open z/OS segment, which defines access to TCP/IP OA sockets.
  - \textsc{ALTER} authority on datasets under \textit{INSTROOT} to access to read, write, allocate, and delete datasets under \textit{INSTROOT}.

- The installation includes a PS, \textit{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:

  - ATTSSRVR: located in an active proclib, such as user.proclib
  - ATTDAEMN: located in an active proclib, such as user.proclib
  - NAVSQL: located in \textit{INSTROOT.USERLIB}

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

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

Starting the Daemon

Activate \textit{INSTROOT.USERLIB(ATTDAEMN)} as a started task to invoke the daemon.

For example, in the SDSF screen enter the following:

'/s ATTDAEMN'

Where \textit{INSTROOT} is the high-level qualifier where Oracle Connect is installed.
To submit the daemon as a job, uncomment the first two lines of the `ATTDAEMN` JCL, change the PARM line as described earlier, and run the job using the subcommand. The `ATTDAEMN` JCL is similar to the following:

```jcl
//*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'
//STELIB DD DSN=INSTROOT.LOADAUT,DISP=SHR
//SYSPRINT DD SYSOUT=A
//GBLPARMS DD DSN=INSTROOT.DEF.GBLPARMS,DISP=SHR
// EXEC PGM=IRPCD,COND=((1,EQ,STEP1),(2,EQ,STEP1)),
// PARM='-KATTDAEMN START ''INSTROOT.DEF.IRPCDINI''
//STELIB 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 `STELIB` list:

```jcl
// 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 2–9 Preinstallation Information

<table>
<thead>
<tr>
<th>Topic</th>
<th>Required Information</th>
<th>Default</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS</td>
<td>CICS EXCI load library name</td>
<td>CICS.CICS.SDFHEXCI</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2–10 Required Permissions

<table>
<thead>
<tr>
<th>Permission</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Permission to read the CICS EXCI library</td>
<td></td>
</tr>
</tbody>
</table>
In the nnn.BUILDKIT.SRC library, run the CICS member, as shown:

```
ex CICS
```

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

<table>
<thead>
<tr>
<th>Screen</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTER THE CICS EXCI LOAD LIBRARY NAME [CICSTS13.CICS.SDFHXEXCI] :</td>
<td>Enter the CICS EXCI load library name only if you do not want the default.</td>
</tr>
<tr>
<td>PLEASE CONFIRM (YES/NO/QUIT) [YES] :</td>
<td>Confirm the entered details</td>
</tr>
<tr>
<td>ENTER THE ISPF LOAD LIBRARY NAME [ISPSISLOAD] :</td>
<td>Enter the ISPF load library name only if you do not want the default.</td>
</tr>
<tr>
<td>PLEASE CONFIRM (YES/NO/QUIT) [YES] :</td>
<td>Confirm the entered details</td>
</tr>
<tr>
<td>ENTER THE OUTPUT CLASS FOR INSTALLATION OUTPUT [A] :</td>
<td>Enter the output class for Oracle Connect output. Assigning a device which is set on HOLD prevents the loss of log information when the Oracle Connect started tasks finish (the default is A).</td>
</tr>
<tr>
<td>DO YOU WANT TO USE THE DEFAULT JOB CARD Y/N [Y]</td>
<td>A job card is displayed. If you want to use a replacement card, then it must be entered as it will appear in the job. You can enter up to six lines. Enter a blank card to end input. If you do not enter a card, then the Oracle Connect default card is used.</td>
</tr>
</tbody>
</table>

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

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

### Installing Oracle Studio

This following sections explain how to install Oracle Studio.

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

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

### Oracle Studio Requirements

The following are the hardware requirements for Oracle Studio:

- Processor: Intel or 100% compatible computer, based on a Pentium processor
- Memory: 256 MB
- Disk space: 120 MB of free disk space
You can install Oracle Studio on the following operating systems:

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

**Installing Oracle Studio on Windows**

Oracle Studio is installed with a standard install wizard. Do the following to install Oracle Studio.

- Run the installation file, either using the Run option in the Windows Start menu or through Windows Explorer. Follow the instructions on the wizard screen.

**Installing Oracle Studio on Linux**

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

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

**Installing Oracle Studio with the Wizard (SH installation)**

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

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

**Installing with a Silent Installation (RPM)**

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

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

**Configuring Oracle Connect**

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

To configure Oracle Connect, refer to the following sections:

- Setting Up the IBM z/OS Platform in Oracle Studio
- Securing Access to Oracle Connect
- Modeling Interactions for OracleAS Adapter for CICS
Configuring Oracle Connect

- 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 z/OS platform and that the Oracle Connect daemon is running on this computer.

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

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

Perform the following steps to configure the IBM 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, 2551.
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 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:

\[\text{Figure 2–2 The Preferences screen}\]

3. Select Studio.
4. Click Change Studio 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 z/OS Platform in Oracle Studio" on page 2-13.

---

**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:

![The Identities screen](image)

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

2. From the Design perspective, Configuration view, expand the **Machines** folder, then expand the machine where you want to set the user name and password.

3. Expand **Users**.

4. Right-click 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](image)

5. In the User editor, click **Add** to display the Add Authenticator screen.

6. Select **Remote Machine** from the **Resource type** list, as shown in the following figure:
Figure 2–5  The Add Authenticator screen

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

Modeling Interactions for OracleAS Adapter for 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 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 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 Design perspective, Configuration view, expand Machine folder.
3. Expand the machine defined in “Setting Up the IBM z/OS Platform in Oracle Studio” on page 2-13.
4. Expand the Bindings. The binding configurations available on this computer are listed.
5. Expand the NAV binding. The NAV binding configuration includes branches for data sources and adapters that are located on the computer.
6. Right-click Adapters and select New Adapter to start the New Adapter wizard.
7. Enter a name for the back-end adapter.
8. 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**

9. Select Create event queue for the adapter.

10. 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.

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

The Properties tab is shown in the following figure:
The Properties tab comprises fields, as listed in the following table:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>exciTransid</td>
<td>The CICS TRANSID. This value must be EXCI or a copy of this transaction</td>
</tr>
<tr>
<td>targetSystemApplid</td>
<td>The VTAM applid of the CICS target system.</td>
</tr>
<tr>
<td>transactionSupport</td>
<td>For future use. The level of transaction support for this adapter that is reported to the transaction manager</td>
</tr>
<tr>
<td>vtamNetname</td>
<td>The VTAM netname of the specific connection being used by EXCI (and MRO) to relay the program call to the CICS target system.</td>
</tr>
</tbody>
</table>

For example, if you issue to CEMP 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 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. From the Start menu, select Programs, Oracle, and then select Studio.
2. In the Design perspective, Configuration view, expand Machine folder.
3. Expand the machine defined in "Setting Up the IBM z/OS Platform in Oracle Studio" on page 2-13.
4. Expand the Bindings. The binding configurations available on this computer are listed.
5. Expand the NAV binding. The NAV binding configuration includes branches for data sources and adapters that are located on the computer.
6. Expand the Adapters folder.
7. Right-click the CICS adapter you defined in "Configuring an Oracle Connect Adapter" on page 2-17 and select Show Metadata View to display the Metadata view, with the CICS back-end adapter displayed under the Adapters list.
8. Right-click the Imports node and select New Import. The New Metadata Import screen is displayed.
9. Enter a name for the import. The name can contain letters and numbers and the underscore character only.
10. Select CICS Import Manager from the Import type list, as shown in the following figure:
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 2–8  The Metadata Import wizard**

![Image of Metadata Import wizard](image1)

---

**Figure 2–9  The Select Resources screen**

![Image of Select Resources screen](image2)
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.
9. Apply filters to the copybooks, as needed.

The following table lists the available filters:

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

10. In the import wizard, click Next. The Add Interactions screen is displayed, as shown in the following figure:
11. Click **Add** to add an interaction for the CICS adapter. Provide the information as listed in the following table:

**Table 2–14 Interaction Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>The name of the interaction. You can change the default name specified.</td>
</tr>
<tr>
<td><strong>Mode</strong></td>
<td>The interaction mode. You can select one of the following:</td>
</tr>
<tr>
<td></td>
<td>- sync-send-receive: The interaction sends a request and expects to receive a response. This is the default mode.</td>
</tr>
<tr>
<td></td>
<td>- sync-receive: The interaction expects to receive a response</td>
</tr>
<tr>
<td></td>
<td>- sync-send: The interaction sends a request and does not expect to receive a response.</td>
</tr>
<tr>
<td><strong>Input</strong></td>
<td>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. Note: You must specify an input record for each interaction before you can click <strong>Next</strong>. If the interaction does not require an input record, then the record specified here is ignored.</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>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. 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 <strong>Next</strong>.</td>
</tr>
</tbody>
</table>
Add as many interactions as necessary and click Next to generate the metadata definitions for the adapter.

Specify that you want to transfer the metadata from the Windows computer to the IBM z/OS platform and click Finish. The metadata is imported based on the options specified and it is stored on the IBM z/OS platform. An XML representation of the metadata is 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.

### 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 2-17, 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 2-19.

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 2-17.
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 z/OS 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 z/OS users, expand the Senders node and right click the users property to add the user.
8. Enter the name of the z/OS 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:

   ![The New Import screen](image)

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

<table>
<thead>
<tr>
<th>Table 2–15 Metadata Filters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Filter</strong></td>
</tr>
<tr>
<td>COMP_6 switch</td>
</tr>
<tr>
<td>Compiler source</td>
</tr>
<tr>
<td>Storage mode</td>
</tr>
<tr>
<td>Ignore after column 72</td>
</tr>
<tr>
<td>IgnoreFirst6</td>
</tr>
<tr>
<td>Replace hyphens (-) in record and field names with underscores (_)</td>
</tr>
<tr>
<td>Prefix nested columns</td>
</tr>
<tr>
<td>Case sensitive</td>
</tr>
<tr>
<td>Find</td>
</tr>
<tr>
<td>Replace with</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Table 2–16 Event Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Mode</td>
</tr>
<tr>
<td>Input</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td>Description</td>
</tr>
</tbody>
</table>

The Add Events screen is shown in the following figure:
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 z/OS platform, and click Finish, as shown in the following figure:
The metadata is imported based on the options specified and it is stored on the IBM z/OS platform. An XML representation of the metadata is 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.

**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 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.

Queueing 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:

```xml
<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
<?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_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>
  </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
```
The following provides an example of the final COBOL copybook. Note the first record has been copy `as is`, while the subsequent records have a REDEFINE appended to them.

```cobol
01 AR-RECORD.
   03 AR-RECORD2.
      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-RECORD3.
         03 AR-SELECTOR PIC 9(03).
         03 AR-RECKEY.
            05 AR-INST PIC X(02).

03 AR-RECORD2.
   03 AR-SELECTOR PIC 9(03).
   03 AR-RECKEY.
      05 AR-INST PIC X(02).

03 AR-RECORD3.
   03 AR-SELECTOR PIC 9(03).
   03 AR-RECKEY.
      05 AR-INST PIC X(02).

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).

03 AR-RECORD.
   03 AR-SELECTOR PIC 9(03).
   03 AR-RECKEY.
      05 AR-INST PIC X(02).

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.
** 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
z/OS platform:

CEDA DEFINE TDQUEUE

TDqueue ==> YTST
### EXTRA PARTITION PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databuffers</td>
<td>1-255</td>
</tr>
<tr>
<td>DDname</td>
<td></td>
</tr>
<tr>
<td>DSname</td>
<td></td>
</tr>
<tr>
<td>Sysoutclass</td>
<td></td>
</tr>
<tr>
<td>Erroroption</td>
<td>Ignore</td>
</tr>
<tr>
<td>Opentime</td>
<td>Initial</td>
</tr>
<tr>
<td>REWind</td>
<td>Leave</td>
</tr>
<tr>
<td>TYPEFile</td>
<td>Input</td>
</tr>
<tr>
<td>RECORDSize</td>
<td>0-32767</td>
</tr>
<tr>
<td>BLOCKSize</td>
<td>0-32767</td>
</tr>
<tr>
<td>RECORDFormat</td>
<td>Fixed</td>
</tr>
<tr>
<td>BLOCKFormat</td>
<td>Blocked</td>
</tr>
</tbody>
</table>

### INTRA PARTITION PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atifacility</td>
<td>File</td>
</tr>
<tr>
<td>RECOVstatus</td>
<td>Logical</td>
</tr>
<tr>
<td>Facilityid</td>
<td></td>
</tr>
<tr>
<td>TRAnsid</td>
<td></td>
</tr>
<tr>
<td>TRIggerlevel</td>
<td>00000</td>
</tr>
</tbody>
</table>

### NDOUBT ATTRIBUTES

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAIT</td>
<td>Yes</td>
</tr>
<tr>
<td>WAITAction</td>
<td>Reject</td>
</tr>
</tbody>
</table>

### NINDIRECT PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indirectname</td>
</tr>
<tr>
<td>EMOTE PARAMETERS</td>
</tr>
<tr>
<td>REMOTEName</td>
</tr>
<tr>
<td>REMOTESystem       =&gt;</td>
</tr>
</tbody>
</table>

---

**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.LOAD 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(ORACLE 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 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. Expand the Machines folder.

3. In the Design perspective, Configuration view expand the computer defined in “Setting Up the IBM z/OS Platform in Oracle Studio” on page 2-13.

4. Expand the Bindings. The binding configurations available on this computer are listed.

5. Expand the NAV binding. The NAV binding configuration includes branches for data sources, adapters, and events that are located on the computer.

6. Right-click Events and select New Event to open the New Event wizard.

7. Enter a name for the queue event.

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

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

   ![Figure 2–19 The New Adapter Screen](image)

9. 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. From the Start menu, select Programs, Oracle, and then select Studio.
2. Expand the Machines folder.
3. In the Design perspective, Configuration view expand the computer defined in “Setting Up the IBM z/OS Platform in Oracle Studio” on page 2-13.
4. Expand the Bindings. The binding configurations available on this computer are listed.
5. Expand the NAV binding. The NAV binding configuration includes branches for data sources, adapters, and events that are located on the computer.
6. Expand the Events folder and select the CICS Queue you created in “Configuring the CICS Queue Adapter” on page 2-37.
7. Expand the Bindings. The binding configurations available on this computer are listed.
8. Right-click the queue adapter and select Show in Metadata View.
   The Metadata tab is displayed, with the queue adapter displayed under the Adapters list.
9. Right-click Imports and select New Import.
   The New Import screen is displayed.
10. Enter a name for the import. The name can contain letters, numbers and the underscore character only.
11. Select CICS Queue Import Manager from the Import Type list.
12. 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.
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.
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**

![Apply Filters screen](image)

10. Apply filters to the copybooks. The available filters are listed in the table below. In addition, if required, apply the variant description filter.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replace hyphens (-) in record and field names with underscores (_).</td>
<td>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.</td>
</tr>
</tbody>
</table>

11. Click **Next**.
   
The Configure Adapter Specification screen is displayed, as shown in the following figure:
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 z/OS platform, and click **Finish**.
**Figure 2–24  The Import Metadata screen**

![The Import Metadata screen](image)

**Viewing the XML Schema**

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

**Creating XML Schemas**

The XML schema describing the adapter interactions and the input and output records for these interactions are created automatically during the import procedure, as described in "Generating Outbound Interactions" on page 2-19 and in "Generating Inbound Interactions" on page 2-25.
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.


## 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.

   ```xml
   <oc4j-connector-factories>
   <connector-factory location=" " connector-name="Oracle Legacy Adapter">
   <config-property name="userName" value="/"/>
   <config-property name="password" value="/"/>
   <config-property name="eisName" value="/"/>
   <config-property name="serverName" value="/"/>
   <config-property name="workspace" value="/"/>
   <config-property name="portNumber" value="/"/>
   <config-property name="persistentConnection" value="/"/>
   <config-property name="keepAlive" value="/"/>
   <config-property name="firewallProtocol" value="/"/>
   <config-property name="connectTimeout" value="/"/>
   <config-property name="encryptionProtocol" value="/"/>
   <config-property name="encryptionKeyName" value="/"/>
   <config-property name="encryptionKeyValue" value="/"/>
   <config-property name="fakeXA" value="/"/>
   <config-property name="useNamespace" value="/"/>
   <config-property name="networkXMLProtocol" value="/"/>
   <config-property name="exposeEventStreamMetadata" value="/"/>
   </connector-factory>
   </oc4j-connector-factories>
   ```
Table 3–1 provides a detailed description of these parameters.

### Table 3–1  OC4J Connection Properties for Outbound Interactions

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

   ```java
   AttuInteractionSpec iSpec = new AttuInteractionSpec("query",
   javax.resource.cci.InteractionSpec.SYNC_SECRET, 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);
   ```

   The following table describes the properties that can be specified:

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

The following is an `InteractionSpec` sample:

```java
AttuInteractionSpec iSpec = new AttuInteractionSpec("query",
javax.resource.cci.InteractionSpec.SYNC_SEND, 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>
<thead>
<tr>
<th>Message Listener Type</th>
<th>ActivationSpec Class</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>javax.resource.cci.MessageListener</code></td>
<td><code>com.oracle.adapter.AttuActivationSpec</code></td>
</tr>
<tr>
<td><code>oracle.tip.adapter.api.OracleMessageListener</code></td>
<td><code>com.oracle.adapter.AttuOracleActivationSpec</code></td>
</tr>
</tbody>
</table>

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

```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.AttuTestReceiverBean</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>
The following table lists the properties that are relevant for inbound interactions.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eisName</td>
<td>Required. Sets the name of the adapter to use. The adapter is defined in Oracle Connect server using Oracle Studio.</td>
</tr>
</tbody>
</table>
### Table 3–4  (Cont.) OC4J Connection Properties for Inbound Interactions

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

```java
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:**
  ```xml
  <eventStream>
  <EMPLOYEE>
  ...
  </EMPLOYEE>
  <DEPARTMENT>
  ...
  </DEPARTMENT>
  </eventStream>
  ```

- **When the `exposeEventStreamMetadata` property is set to false:**
  ```xml
  <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>
  ```
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
<?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:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
  <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>
  </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.oracle.AttuXMLRecordConver
terImpl"/>
    <operation name="add">
      <jca:operation FunctionName="add"
        InteractionSpec="com.oracle.adapter.AttuInteractionSpec" ExecutionTimeout="120"/>
      <input/>
      <output/>
    </operation>
  </binding>
</definitions>
```
Configuring Oracle BPEL Process Manager to interact with the OracleAS Adapter for CICS

4-4 Oracle Application Server Adapters for CICS User's Guide

Configuring the WSDL for Inbound Applications

When you build an inbound 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 inbound applications:

```xml
<?xml version = '1.0' encoding = 'UTF-8'?>
<definitions name="eventStream"
  targetNamespace="http://xmlns.oracle.com/pcbpel/testCICSQ1/eventStream"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns="http://xmlns.oracle.com/pcbpel/testCICSQ1" targetNamespace="http://xmlns.oracle.com/pcbpel/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:schema>
  </types>
</definitions>
```
<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>
</xsd:complexType>
</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="100"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
</xsd:element>
</xsd:complexType>
<xsd:maxLength value="150"/>
</xsd:restriction>
</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>
</schema>

<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.oracle.adapter.AttuOracleActivationSpec" 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" bindings="tns:eventStreamJCABinding">
<jca:address ResourceAdapterClassName="com.oracle.adapter.AttuOracleResourceAdapter" adapterInstanceJndi="eis/legacy/testCICSQ1"/>
</port>
</service>

<plt:partnerLinkType name="eventStreamPartnerLinkType">
<plt:role name="eventStreamRole">
<plt:portType name="tns:eventStreamPortType"/>
This chapter contains the following topics:

- Getting Started
- Deploy a BPEL Outbound Process

**Getting Started**

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

**Prerequisites**

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

- Personal computer running Microsoft Windows XP or Windows Server 2003 with one GB of RAM.
- The CICS adapter that you are using must be deployed to the Oracle Application Server.
- OracleAS CICS Adapter must be configured for outbound processing. See Configuring an OracleAS Adapter for CICS for Outbound Interactions.
- Oracle JDeveloper must be installed.
- Legacy adapter must be deployed.
- Oracle Studio with the necessary machines loaded. See Setting Up the IBM z/OS Platform in Oracle Studio.

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

- How to create J2CA connections. See Integrating OracleAS Adapters for CICS with OC4J for more information.
- How to configure the OracleAS adapters for CICS using Oracle Studio. See Configuring Oracle Connect for more information.

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

- **Design Time**: The OracleAS adapters for CICS are configured with Oracle Studio as described in Configuring Oracle Connect.
Deploy a BPEL Outbound Process

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

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

Beginning Tasks

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

Beginning Tasks

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

- Integrate the OracleAS Adapter for CICS with OC4J. See Integrating the J2CA 1.5 CICS Adapter for Outbound for more information.
- Configure the connection for the J2CA CICS resource adapter. You do this by creating a connection factory. See Configuring the J2CA 1.5 CICS Adapter for Outbound for information on how to do this.
- Write a schema file. First, create an interaction called findDoctor using Oracle Studio. See Configuring the J2CA 1.5 CICS Adapter for Outbound for a description on how to create the findDoctor interaction.
  
  Write a schema file called FINDDOCTORSchema.xsd to define the message schema. The following is an example of the findDoctor.xsd file:

  ```xml
  <schema targetNamespace="http://xmlns.oracle.com/Esb/findDoctorData"
  xmlns:cust="http://xmlns.oracle.com/Esb/findDoctorData"
  xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="findDoctorata" type="cust:findDoctorDataType"/>
  <complexType name="findDoctorDataType">
    <sequence>
      <element name="Record1" type="string"/>
    </sequence>
  </complexType>
  </schema>
  ```
- Create a BPEL PM Server Connection in JDeveloper.
- Create the WSDL for the interaction. You do this in JDeveloper Connections explorer before you begin to build your endpoint in JDeveloper. JDeveloper will create the WSDL automatically based on the interaction you create. In the Connections explorer, browse for the Oracle Connect server you configured in Oracle Studio (see Installing Oracle Connect on an IBM z/OS Series Platform), then double-click the findDoctor interaction to view and create the WSDL.
Deploy a BPEL Outbound Process

See Modeling Interactions for OracleAS Adapter for CICS for an explanation on how to create the findDoctor interaction and for an explanation on the WSDL.

Design-Time Configuration

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

- Create a BPEL Project for a BPEL Outbound Process
- Create a Database Adapter to Read the Doctor Data
- Design a BPEL Outbound Process to Read the String
- Create a Partner Link to the Database Adapter
- Edit the Invoke_1 Activity
- Edit the Assign_1 Activity

Create a BPEL Project for a BPEL Outbound Process

Do the following to create a BPEL project for outbound.

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

![Figure 5–1 Outbound Project Settings](image)

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

Create a Database Adapter to Read the Doctor Data
You create a database adapter to read the DOCTOR data. SOA executes the SQL statement defined in the interaction and passes it to a resource adapter. The resource adapter then processes the information and returns the data. Do the following to create the database adapter.

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

   **Figure 5–2 Partner Link for Database Adapter**
   ![Partner Link for Database Adapter](image)

   2. Click the Adapter Wizard button. This is the third button in the **WDSL Settings** section of the Create Partner Link dialog box, as shown in the figure above.
   3. Enter the following information in the Adapter Configuration wizard:
      a. Step 1, Adapter Type: Select **Database Adapter**.
      b. Step 2, Service Name: type **findDoctor**
      c. Step 3, Service Connection. In this step select the data you want to work with.
         Click **New** to open the **Create Connection** wizard.
         **Step 1**: Select the connection type:
         Connection Name: dbConnection1
         Connection Type: Oracle (JDBC)
         **Step 2**: Authentication.
         Enter the User Name, Password, and Role for the connection
         **Step 3**: Enter the connection information.
         Enter the driver type (thin, oci), the name of the computer where JDBC is deployed, and its port number. This information should be the same as defined in the JDBC connection factory. See **Configuring the J2CA 1.5 CICS Adapter for Outbound**.
         Select Service Name and enter **findDoctor**.
         Click **Finish** to go back to the Adapter Configuration wizard.
      d. Step 4, File Directories
         **Directory Names are Specified as**: Select **Physical Path**
Deploy a BPEL Outbound Process

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

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

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

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

4. Click Finish.

Design a BPEL Outbound Process to Read the String
In this step, you design the BPEL process. Do the following to design the BPEL outbound process:

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

Figure 5–3 Outbound Process

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

Create a Partner Link to the Database Adapter

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

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

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

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

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

**Edit the Invoke_1 Activity**

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

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

![Invoke Dialog Box](image)

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

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

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

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

6. The Output Variable should be **Invoke_1_FINDDOCTOR_OutputVariable**. Click OK.
Edit the Assign_1 Activity
You use the assign activity to assign a value to the resource adapter. Do the following to edit the Assign_1 activity.

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

![Assign Dialog Box]

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

![Create Copy]

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

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

The following figure shows how the final process looks:

Figure 5–9 Final Outbound Process

Runtime Configuration

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

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

```
<?xml version="1.0" encoding="UTF-8"?>
<CU:FindDoctorData
    xmlns:CU="http://xmlns.oracle.com/Esb/FindDoctorData"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Record1>CICS</Record1>
</CU:FindDoctorData>
```

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

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

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

Oracle Net Services Administrator's Guide BPEL Process Manager Examples  5-9
This chapter provides an example for creating an OracleAS adapter for CICS. In this case it describes the steps necessary to create an adapter that will get the names of a doctor and a patient from data under CICS. To work with an OracleAS adapter for CICS, you must also configure a BPEL endpoint connection. For information on how to create an endpoint, see Oracle Net Services Administrator’s Guide BPEL Process Manager Examples.

Creating Outbound Interactions with the OracleAS Adapter for CICS

To work with the OracleAS Adapter for CICS and create outbound interactions you must create a CICS adapter, import the metadata, then create interactions that will be used to access the data.

Requirements

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

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

Create the findDoctor and findPatient Interactions

You use a CICS adapter to create the interactions used to get the data that you need. The interactions are used as services when you set up the endpoint in JDeveloper. For more information on using JDeveloper to create an endpoint, see Oracle Net Services Administrator’s Guide BPEL Process Manager Examples.

Do the following to create the interactions.

- Prepare the System
- Set up Machine Access to Oracle Connect
- Add a CICS Adapter
- Import the Metadata
Prepare the System

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

Set up Machine Access to Oracle Connect

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

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

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

![Add Machine Dialog Box](image)

**Figure 6–1 Add Machine Dialog Box**

Add a CICS Adapter

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

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

For a detailed explanation on how to set up a CICS adapter, see Configuring an Oracle Connect Adapter.

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

**Figure 6–2 Add Adapter Dialog Box**

[Image of the Add Adapter Dialog Box]

---

**Import the Metadata**

You now import the metadata from the CICS data into the adapter. For information on how to import data, see Generating Outbound Interactions.

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

You will need the following information for the metadata import:

- In the **Get Input Files** step, add the file, hospital.cob.
- In the **Apply Filters** step, just click Next.
- In the **Add interactions** step, under the **Name** column, enter findDoctor, under the **Input** column, select DOCTOR and under the **Output** column, select DOCTOR.
- In the **Import Metadata** step, select Yes and then click Finish.
Create the findDoctor and findPatient Interactions
Troubleshooting OracleAS Adapter for CICS

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

This section contains the following topics:

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

Troubleshooting the Daemon

Troubleshooting the daemon and the communication between Oracle Application Server and OracleAS 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 7-2, you can restart the daemon as described in the following task.

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

Task: Starting the Daemon

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

'/s ATTDAEMN'

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

See Also: "Starting the Daemon" on page 2-9 for details about the ATTDAEMN JCL.

Shutting Down the Daemon

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

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

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 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>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit Daemon Configuration</td>
<td>Opens the daemon editor, enabling you to reconfigure the daemon.</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Advanced Tuning of the Daemon&quot; for details about the configuration settings.</td>
</tr>
<tr>
<td>Status</td>
<td>Checks the status of the daemon. The information about the daemon includes the daemon name, configuration used, the active client sessions, and logging information.</td>
</tr>
<tr>
<td>Reload Configuration</td>
<td>Reloads the configuration after any changes. Any servers currently started are not affected by the changed configuration.</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Advanced Tuning of the Daemon&quot; for details about the configuration settings.</td>
</tr>
<tr>
<td>View Log</td>
<td>Displays the daemon log. For details see &quot;Daemon Logs&quot; on page 7-4.</td>
</tr>
<tr>
<td>View Events</td>
<td>Displays the daemon events log.</td>
</tr>
</tbody>
</table>
**Troubleshooting the Daemon**

**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:

<p>| <strong>Table 7–2  Workspace Options</strong> |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit Workspace Configuration</td>
<td>Opens the daemon editor to enable you to reconfigure the workspace. See Also: &quot;Advanced Tuning of the Daemon&quot; for details about the configuration settings.</td>
</tr>
<tr>
<td>Status</td>
<td>Checks the status of the workspace, whether it is available or not.</td>
</tr>
<tr>
<td>View Log</td>
<td>Displays the log for all servers for the workspace. For details see &quot;Daemon Logs&quot; on page 7-4.</td>
</tr>
<tr>
<td>View Events</td>
<td>Displays the workspace events log.</td>
</tr>
<tr>
<td>Recycle Servers</td>
<td>Closes all unused servers and prepares all active servers to close when the client disconnects. New connection requests are allocated with new servers.</td>
</tr>
<tr>
<td>Kill Servers</td>
<td>Immediately closes all active and unused servers. <strong>Note</strong>: It is recommended to use this option with caution, as it may lead to data loss.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refreshes the display.</td>
</tr>
</tbody>
</table>

**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:
Troubleshooting the Daemon

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.

### Daemon Logs

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Checks the status of the server. The information about the server includes the server mode and the number of active client sessions for the server.</td>
</tr>
<tr>
<td>View Log</td>
<td>Displays the server log. For details see &quot;Daemon Logs&quot; on page 7-4.</td>
</tr>
<tr>
<td>View Events</td>
<td>This option is for future use.</td>
</tr>
<tr>
<td>Kill server</td>
<td>Ends the server process, regardless of its activity status. <strong>Note:</strong> It is recommended to use this option with caution, as it may lead to data loss.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refreshes the display.</td>
</tr>
</tbody>
</table>

### Table 7–3 Server Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Checks the status of the server. The information about the server includes the server mode and the number of active client sessions for the server.</td>
</tr>
<tr>
<td>View Log</td>
<td>Displays the server log. For details see &quot;Daemon Logs&quot; on page 7-4.</td>
</tr>
<tr>
<td>View Events</td>
<td>This option is for future use.</td>
</tr>
<tr>
<td>Kill server</td>
<td>Ends the server process, regardless of its activity status. <strong>Note:</strong> It is recommended to use this option with caution, as it may lead to data loss.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refreshes the display.</td>
</tr>
</tbody>
</table>

**Daemon Logs**

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

- **Daemon 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.
Resolving Communication Errors

- **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:** Workspace "General" on page A-7.

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:** Workspace "General" on page A-7.

Resolving Communication Errors

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

The following table describes the various scenarios that may exist when Oracle Application Server disconnects from the IBM z/OS computer.
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 z/OS platform. Use the Status option in the Runtime Manager perspective.

C00B: Invalid IRPCD client context.

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

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

C00C: Daemon request requires a server login.

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

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

C00D: Daemon request requires a client login.

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

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

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

See Also: "Setting Up Run-Time User Access to the IBM z/OS Platform" on page 2-16.

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 z/OS platform.

See Also: "Setting Up Run-Time User Access to the IBM z/OS Platform" on page 2-16.

C010: Anonymous server logins are not allowed.
Cause: Internal error.
Action: Contact Oracle Support Services.

C011: Client has already timed out.
Cause: A server process was started on behalf of a client and the client has timed out before the server completed its startup.
Action: Increase the Connect timeout value for the server workspace in the Workspace General tab of the daemon configuration.

See Also: Workspace "General" 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 z/OS platform.

See Also: "Setting Up Run-Time User Access to the IBM z/OS Platform" on page 2-16.
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 Server Mode tab of the workspace configuration.

See Also: "Server Mode" on page A-11.
Action: Try running the command later.

C015: Failed to start server process.
Cause: The Oracle Connect daemon failed to start a server process or the started server failed upon starting up.
Action: See the daemon and server logs for the reason the server did not start. For example, you might receive 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 z/OS computer. For example:

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

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

In the Workspace **Security** tab of the Navigator workspace under the daemon node in the Configuration Explorer, check **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 Workspace General tab.

See Also: Workspace "General" 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 Workspace Server Mode tab.

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

C023: Call made to unregistered module %d.
Cause: Internal error.
Action: Contact Oracle Support Services.

C024: Failed to create a socket.
Cause: An error occurred within the TCP/IP subsystem.
Action: Check whether you have sufficient system privileges.
Action: Check the TCP/IP subsystem on the computer by trying to ping it or run FTP or Telnet to or from it.

C025: Failed to set socket option %s
Cause: An error occurred within the TCP/IP subsystem.
Action: Check whether you have sufficient system privileges.
**Action:** Check the TCP/IP subsystem on the computer by trying to ping it or run FTP or Telnet to or from it.

**C026: Failed to bind server to port %s**
**Cause:** An Oracle Connect server or daemon was not able to bind to the specified port.
**Action:** Check whether another program is holding the port that was specified in the oc4j-ra-xml file for the adapter.
**Action:** Check whether you have sufficient system privileges.

**C027: Cannot create TCP service for %s**
**Cause:** An error occurred within the TCP/IP subsystem
**Action:** Check the TCP/IP subsystem on the computer by trying to ping it or run FTP or Telnet to or from it.

**C028: Unable to register (%s, %d, tcp)**
**Cause:** This error may happen when a portmapper is used (host:a) but the portmapper is not available.
**Action:** Enable the portmapper.
**Action:** Avoid using the portmapper (by not using :a when starting the daemon).

**C029: Failed to create a server thread**
**Cause:** Internal error.
**Action:** Contact Oracle Support Services.

**C02A: Server thread failed to start**
**Cause:** Internal error.
**Action:** Contact Oracle Support Services.

**C02B: Stopping the %s server - no client**
**Cause:** A server that was started by the Oracle Connect daemon to service a client did not get a client connection request within one minute. The server terminates.
**Action:** In most cases, the client was terminated by a user request, so no specific action is required.
**Action:** If no client can connect to the server, it may be that the server has multiple network cards and the Oracle Connect daemon is not aware of this. In this case, start the daemon with an IP address.

**C02C: Unexpected event - a termination signal intercepted**
**Cause:** Internal error.
**Action:** Contact Oracle Support Services.

**C02D: Modified transport, context unknown/lost**
**Cause:** Internal error.
**Action:** Contact Oracle Support Services.

**C02F: Corrupted arguments passed to procedure**
**Cause:** Internal error.
**Action:** Contact Oracle Support Services.
C030: Unable to free arguments for %s() of %s
  Cause: Internal error.
  Action: Contact Oracle Support Services.

C031: Cannot register a non-module RPC %s
  Cause: Internal error.
  Action: Contact Oracle Support Services.

C032: An IRPCD program is required
  Cause: Internal error.
  Action: Contact Oracle Support Services.

C033: An IRPCD super-server is required for module events
  Cause: Internal error.
  Action: Contact Oracle Support Services.

C034: An invalid super-server module ID was specified, %d
  Cause: Internal error.
  Action: Contact Oracle Support Services.

C035: Out of memory
  Cause: Not enough memory to service a client request.
  Action: Increase process memory quota or add memory to the system.

C036: Failed to register RPC procedure module %s
  Cause: Internal error.
  Action: Contact Oracle Support Services.

C037: Failed to register an invalid RPC procedure number %x
  Cause: Internal error.
  Action: Contact Oracle Support Services.

C038: Cannot reregister RPC procedure number %x
  Cause: Internal error.
  Action: Contact Oracle Support Services.

C042: Remote call to %s failed; %s
  Cause: Remote call to API failed.
  Action: Check the daemon log file.
  Action: If necessary, change the level of detail written to the log file to help resolve the problem.

  See Also: "Logging" on page 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 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 Workspace General tab.

   See Also: Workspace "General" 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 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
Resolving Specific Errors

**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 Workspace Security tabs in the daemon configuration.


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.
Resolving Specific Errors

**Action:** Contact Oracle Support Services.

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

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

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

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

*Cause:* Internal error.

*Action:* Contact Oracle Support Services.

**J0036: %s: %s**

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

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

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

*Cause:* Internal error.

*Action:* Contact Oracle Support Services.

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

*Cause:* Internal error.

*Action:* Contact Oracle Support Services.

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

*Cause:* Internal error.

*Action:* Contact Oracle Support Services.

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

*Cause:* A request for metadata was not fulfilled.

*Action:* Check the validity of the request.

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

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

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

**J0059: Value %s is invalid for attribute %s**

*Cause:* A request for metadata was not fulfilled.

*Action:* Check the validity of the request.

**J0068: Value must be of type string**

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

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

**J0069: Value must be of type MappedRecord**
Resolving Specific Errors

Troubleshooting OracleAS Adapter for CICS

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

This section contains the following topics:

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

### Configuring the Daemon for High Availability

The daemon workspace is responsible for allocating server processes to clients. You can configure a workspace to use a pool of server processes so that a server process is always available for a client request. Use Oracle Studio to maintain 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 Design perspective, Configuration view expand the Machines folder.
3. Expand the machine you are working with.
4. Expand the Daemons folder. The daemon configurations available on this computer are listed.
5. Right-click IRPCD and select New Workspace. The New Daemon Workspace screen is displayed.
6. Specify a name for the new workspace and optionally, provide a description.

7. Specify whether you want it to have default settings or copy the properties of an existing workspace.

To copy the properties of an existing workspace, click Ellipsis and select the workspace from which you want to copy the properties.

8. Click Next. The Select Scenario screen is displayed.

9. Select Application Server using connection pooling and click Next.

10. Continue through the wizard, specifying the required values for the workspace.

11. Click Finish.

The workspace is displayed under the IRPCD daemon node.

Editing the Workspace

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

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

Use Oracle Studio to access these tabs, as follows:

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

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

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

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

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

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

7. After editing the workspace, click Save.

Configuring the Server Mode

The server mode dictates how the daemon starts up new processes. The daemon supports the following server modes:
- **singleClient**: Each client receives a dedicated server process. The account in which a server process runs is determined either by the client login information or by the specific server workspace.

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

- **multiClient**: Clients share a server process and are processed serially. This mode has low overhead because the server processes are already initialized. However, because clients share the same process, they may impact one another, especially if they issue lengthy queries.

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

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

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

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

Set the server mode in the **Server Mode** tab of the daemon workspace editor as shown in the following figure:
Configuring the Daemon for High Availability

Figure 8–1 The Server Mode Tab

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

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

- **Port Range**: Select the range for specific firewall ports through which you access the workspace. Determines the range of ports available for this workspace when starting server processes. Use this option when you want to control the port number, so that Oracle Connect can be accessed through a firewall.

  Enter the port range in the following fields:
  - **From**: Enter the highest numbered port in the range
  - **To**: Enter the lowest numbered port in the range
  - **Use Default Port Range**: Select this to use the port range that is defined in the daemon.

- **Maximum number of server processes**: Enter the maximum number of server processes that can run at the same time.

- **Limit server reuse**: Select this if you want to limit the number of servers that can be reused. If this is selected, the **Reuse limit** parameter is available.

  If **Limit server reuse** is selected, in the field next to the check box, enter the maximum number of times a server can be reused. Select the maximum of clients accepted in a server process.
A one-client server can be reused after its (single) client has disconnected. Reuse of servers enhances startup performance because it avoids the need to repeat initialization.

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

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

- **Limit Concurrent clients per server**: Select this to limit the number of clients that a server can accept for the current workspace process. If this is not selected, the number of clients is unlimited. If Limit concurrent clients per server is selected, in the field next to the check box, enter the maximum number of clients that a server process for the current workspace accepts. The default for this field is None, indicating that the number of clients for each server is unlimited. This field is available if the server mode value is multiClient or multiThreaded.

- **Specify Server Priority**: Set the priority for servers. For example, a workspace for applications with online transaction processing can be assigned a higher priority than a workspace that requires only query processing. The lower the number, the higher the priority. For example, workspaces with a priority of 1 are given a higher priority than workspaces with a priority of 2.

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

- **Use default server priority**: Sets the priority to 0. There is no specific priority for this workspace. Clear this check box to set a priority in the Specify Server Priority parameter.

- **Keep when daemon ends**: Select this to kill all servers started by that daemon when a daemon is shutdown, even if they are active. Select this if you want the servers for the workspace to remain active, even after the daemon has been shut down. If selected, it is the responsibility of the system operator or manager to ensure that the servers are eventually killed. This must be done at the system level.

- **Number of prestarted servers in pool**: The number of server processes that are prestarted for this workspace when the daemon starts up. These are available for use by new client processes with minimal initialization time. Instead of starting a new server process each time one is requested by a client, the daemon immediately allocates (to the client) a server from a pool of available servers. When the number of available server processes drops lower than the value specified in the Minimum number of available servers field, the daemon again starts server processes until the specified number of available servers is reached. The default for this parameter is 0, meaning that no servers are prestarted for this workspace.

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

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

**Configuring a Binding Environment**

Each binding configuration includes the following:

- 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 z/OS Platform in Oracle Studio" on page 2-13.
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 8–2  The Binding Properties tab*

The binding environment is divided into the following categories:

- **Debug**
- **General**
The following table lists the parameters that define debugging and logging operations:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACX trace</td>
<td>Select this for the input xml sent to the back-end adapter and the output xml returned by the back-end adapter to be written to the log.</td>
</tr>
<tr>
<td>GDB Trace</td>
<td>This parameter is not applicable for use with OracleAS Adapter for CICS.</td>
</tr>
<tr>
<td>General trace</td>
<td>Select this to log general trace information. The default writes only error messages to the log.</td>
</tr>
<tr>
<td>Query warnings</td>
<td>This parameter is not applicable for use with OracleAS Adapter for CICS.</td>
</tr>
<tr>
<td>Add timestamp to traced events</td>
<td>Select this to add a timestamp on each event row in the log.</td>
</tr>
<tr>
<td>Query Processor trace</td>
<td>This parameter is not applicable for use with OracleAS Adapter for CICS.</td>
</tr>
<tr>
<td>Binary XML Log Level</td>
<td>Select the binary XML log level from the list. The following logging levels are available:</td>
</tr>
<tr>
<td>Log file</td>
<td>The high-level qualifier of the log file for messages. The following type of message are written to the log:</td>
</tr>
<tr>
<td>Trace Directory</td>
<td>This parameter is not applicable for use with OracleAS Adapter for CICS.</td>
</tr>
</tbody>
</table>
General

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAV_UTIL editor</td>
<td>This parameter is not applicable for use with OracleAS Adapter for CICS.</td>
</tr>
<tr>
<td>Temporary Dir</td>
<td>The directory where temporary files are written, including the temporary files created for use by hash joins and for sorting files. The default is the current high-level qualifier.</td>
</tr>
<tr>
<td>Year 2000 policy</td>
<td>Determines the way two-digit years are converted into four-digit years. When the year2000Policy parameter is not set, or when it is set to a value outside the range of values defined for the policy, as described in the following paragraphs, a default value of 5 and the Sliding Base Year policy is used. Two policies are provided: Fixed Base Year: year2000Policy is set to a value greater than, or equal to 1900. In this case, the value of year2000Policy is the first four-digit year after 1900 that can be represented by a two-digit year. For example, if year2000Policy is set to 1905, the years 2000-&gt;2004 will be represented by 00-&gt;04. All other two digits will map to 19xx. This solution is most required if there is live data at the low end (close to the year 1900), which the user wants to keep with the current two-digit format. The user will probably change the base date only after ensuring that these old dates have been deleted from the data source. Sliding Base Year: year2000Policy is set to a positive value less than 100. In this case, the value of year2000Policy represents the number of years ahead of the current year that can be represented by a two-digit number. With each passing year the earliest year that can be represented by a two-digit number changes to a year later.</td>
</tr>
<tr>
<td>Cache buffer size</td>
<td>Enter the number of bytes to be used for a memory buffer on a client machine, which is used by the Oracle Connect client/server to store read-ahead data. The default is 200000.</td>
</tr>
</tbody>
</table>

Language

The following table lists the parameters that define globalization support:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizer trace</td>
<td>This parameter is not applicable for use with OracleAS Adapter for CICS.</td>
</tr>
<tr>
<td>Transaction extended logging</td>
<td>Select this for the transaction manager to write additional information about transactions to the log.</td>
</tr>
</tbody>
</table>
Modeling

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

ODBC

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

OLEDB

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

Optimizer

The Optimizer parameters are not applicable for use with OracleAS Adapter for CICS.

Query Processor

The Query Processor parameters are not applicable for use with OracleAS Adapter for CICS.

Parallel Processing

The Parallel Processing parameters are not applicable for use with OracleAS Adapter for CICS.

Transactions

The following table lists the parameters that define transaction support:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction extended logging</td>
<td>Select this to write extended information about transactions to the transaction manager log files.</td>
</tr>
<tr>
<td>Commit on destroy</td>
<td>Select this to commit all single-phase commit transactions opened for a data source, if a connection closes while the transaction is still open.</td>
</tr>
</tbody>
</table>

Table 8–4 Language Settings Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Identifies the application language. A default codepage is selected based on the value specified for this parameter. See also: Appendix C, &quot;Globalization Settings&quot;.</td>
</tr>
<tr>
<td>Code Page</td>
<td>For use with globalization support to identify the codepage for the workspace. See also: Appendix C, &quot;Globalization Settings&quot;.</td>
</tr>
</tbody>
</table>
| NLS String| 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:  
  - The name of the codepage.  
  - Whether the character set reads from right to left (as in middle eastern character sets).  
    The default is false. |

Table 8–5 Transaction Parameters
The tuning parameters are not applicable for use with OracleAS Adapter for CICS.

**XML**

The following table lists the parameters that define XML support:

**Table 8-6 XML Category Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM maximum XML in memory</td>
<td>Specifies the maximum size of an XML document held in memory. The default is 65535 bytes.</td>
</tr>
</tbody>
</table>
| COM maximum XML size             | Specifies the maximum size of an XML document passed to another computer. The default is 65535 bytes.  
|                                  | **Note:** When you increase this value for this property, you may need to increase the value for the Maximum XML in memory property in the daemon. For more information on daemons, see “Control”. |
| COM XML transport buffer size     | Enter the maximum size of the internal communications buffer. The default value (-1) indicates there is no size limit. |
Migration Considerations

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

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

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 z/OS Platform in Oracle Studio" on page 2-13 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 z/OS Platform in Oracle Studio" on page 2-13.
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 z/OS computer with RACF installed, and with the workspace server mode set to multi-tasking, a RACROUTE VERIFY is performed for each task in the address space, according to the client connection.

In addition, Oracle Connect provides the following security 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 2-14.
- 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 Workspace Security tab in Oracle Studio.

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

To be allocated a server process, the client must be granted anonymous access to the workspace or be listed in the Workspace Users section of the Workspace Security tab in Oracle Studio.

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 Workspace Security tab in Oracle Studio.

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 theDaemon Security tab in Oracle Studio, as described in "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 Workspace Security tab.

### 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, select **Convert all to distributed** in the Transaction section of the binding environmental properties. See Transactions in the Configuring a Binding Environment section.

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

To define a DSN as APF-authorized, in the SDSF screen enter the 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, OracleAS Adapter for CICS can participate in a distributed transaction, as the only one-phase commit resource, if the Transaction log file environment property includes the keyword NORRS. See Transactions in the Configuring a Binding Environment section.

**Note:** If a log file is not specified, then the format for the logFile parameter when RRS is not running is: ,NORRS.
Advanced Tuning of the Daemon

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

- Control
- Logging
- Security
- Workspaces

Control

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

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

---

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

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

---

The Control tab is shown in the following figure:
The following table shows the parameters that can be set in the Control tab:

**Table 8–7 Daemon Control Tab Components**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daemon IP Address</td>
<td>Enter the IP address of the machine(s) where the daemon is listening. If no IP address is entered, the daemon will listen on all available IP addresses.</td>
</tr>
<tr>
<td>Daemon port</td>
<td>Enter the port where the daemon is listening. If no port is entered, the daemon listens on all available ports.</td>
</tr>
<tr>
<td>Automatically recover from failure</td>
<td>The daemon restarts automatically if it fails for any reason (any error that causes the daemon process to terminate, such as network process lost or the CPU running the daemon crashes and the backup daemon is defined on another CPU). All available and unconnected servers are terminated and any connected servers are marked and terminated on release. Also the backup starts a backup for itself. The backup appends a new log file to the log of the original daemon, adding a line indicating that a backup daemon was started.</td>
</tr>
<tr>
<td>Maximum XML request size</td>
<td>The maximum number of bytes that the daemon handles for an XML document.</td>
</tr>
<tr>
<td>Default language</td>
<td>The language that the daemon supports. This setting is used when working with a client with a code page different from the server code page.</td>
</tr>
<tr>
<td>Maximum XML in memory</td>
<td>The maximum amount of space reserved for the XML in memory.</td>
</tr>
</tbody>
</table>
Using the **Logging** tab, you define the daemon log file settings, the log file structure and the location where the log is saved. In addition, use it to define the data that is logged and traced in the file.

The following describes how to open the **Logging** tab.

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

---

**Table 8-7 (Cont.) Daemon Control Tab Components**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call timeout</td>
<td>The timeout period for short calls for all daemons. The definition of a short call is a call that should be completed in a few seconds. For example, most calls to a database such as <code>DESCRIBE</code> should be completed in a few seconds as opposed to call like a <code>GETROWS</code> call, which can take a long time. In heavily loaded or otherwise slow systems, even short calls such as calls to open a file, may take a significant amount of time. If a short call takes more than the specified time to complete, then the connection is stopped. The default value for this parameter is 60 seconds. Values of less than 60 seconds are considered to be 60 seconds. Specifying the timeout in a workspace overrides the value set in this field for that workspace.</td>
</tr>
<tr>
<td>Connect timeout</td>
<td>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. Specifying the timeout in a workspace overrides the value set in this field for that workspace.</td>
</tr>
<tr>
<td>Client idle timeout</td>
<td>The maximum amount of time any daemon client may be idle before the connection with the server is closed. Specifying the timeout in a Workspace overrides this setting for that workspace.</td>
</tr>
</tbody>
</table>

**Notes:**

- Entering the timeout in a workspace overrides the value set in this field for that workspace.
- Even if the XML source does not list this parameter in the workspace section, the workspace gets it using the default value. If you want to prevent a workspace from using the default value, you must enter a value of zero for this parameter in the workspace section.
7. Right-click the daemon and select **Recycle Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **Logging** tab for the daemon is shown in the following figure:

![Figure 8-4  The Logging tab](image)

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logging options</strong></td>
<td></td>
</tr>
<tr>
<td>Daemon log file location</td>
<td>Specifies the daemon produces its log data. The full path must be specified.</td>
</tr>
<tr>
<td>Server log filename format</td>
<td>Defines the name and location of the server log file. The field must specify the full path name. If no directory information is provided for the log file, then it will be located in the login directory of the account running Oracle Connect workstation.</td>
</tr>
<tr>
<td>Daemon operations</td>
<td>Select this to log all of the daemon operations.</td>
</tr>
</tbody>
</table>

**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 following tokens can appear in the log file template and will be replaced accordingly:

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

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

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

### Security

The following Security tab for the daemon is used to:

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

The following shows how to open the Daemon Security tab:

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

---

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

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

---

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

The Daemon Security tab is shown in the following figure:

**Figure 8–5 The Daemon Security tab**

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

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

**General**

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

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

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design perspective Configuration view, expand the **Machines** folder.

### Table 8–9 Daemon Security Tab Components

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrators privileges</td>
<td>Identifies the users (accounts) allowed to perform administrative tasks (tasks that require administrative login).</td>
</tr>
<tr>
<td>All users</td>
<td>Enables all users to access the daemon and change the settings.</td>
</tr>
<tr>
<td>Selected users only</td>
<td>Identifies the names of users (accounts) and groups that can be administrators. 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.</td>
</tr>
<tr>
<td>Machine access</td>
<td>Manages access to the computer.</td>
</tr>
<tr>
<td>Allow anonymous login</td>
<td>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.</td>
</tr>
<tr>
<td>Cached password</td>
<td>Enables login passwords to be cached. This enhances performance by reducing login times for future connections from the same client in a session.</td>
</tr>
<tr>
<td>Encryption methods</td>
<td>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.</td>
</tr>
</tbody>
</table>

1 The name is prefixed with '@', to utilize the operating system GROUP feature.
3. Right-click a computer and select **Open Runtime Perspective**.

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

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

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

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

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

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

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

**Figure 8–6** The General tab

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td></td>
</tr>
</tbody>
</table>
### Table 8–10 (Cont.) General Tab Components

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workspace name</td>
<td>The name used to identify the workspace. 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.</td>
</tr>
<tr>
<td>Description</td>
<td>A description of the workspace.</td>
</tr>
<tr>
<td>Startup script</td>
<td>The full path name of the script that starts the workspace server processes. The script specified here must always activate the nav_login procedure and then run the server program (svc). If you do not specify the directory, the startup procedure is taken from the directory where the daemon resides. Oracle Connect includes a default startup script, which it is recommended to use.</td>
</tr>
<tr>
<td>Server type</td>
<td>This field is not applicable for use with OracleAS Adapters for CICS.</td>
</tr>
<tr>
<td>Workspace binding name</td>
<td>This field is not applicable for use with OracleAS Adapters for CICS.</td>
</tr>
<tr>
<td>Timeout parameters</td>
<td>The time the client waits for the workspace server to start. If the workspace server does not start within this period, then the client is notified that the server did not respond. Specifying the timeout here overrides the default setting, specified in the Control section. See Also: Control for details about the Daemon Control section.</td>
</tr>
<tr>
<td>Client idle timeout</td>
<td>The maximum amount of time a workspace client can be idle before the connection with the server is closed.</td>
</tr>
<tr>
<td>Connect timeout</td>
<td>The time the client waits for a workspace server to start. If the workspace server does not start within this period, then the client is notified that the server did not respond.</td>
</tr>
<tr>
<td>Call timeout</td>
<td>The timeout period for short calls for all daemons. The definition of a short call is a call that should be completed in a few seconds. For example, most calls to a database such as DESCRIBE should be completed in a few seconds as opposed to call like a GETROWS call, which can take a long time. In heavily loaded or otherwise slow systems, even short calls such as calls to open a file, may take a significant amount of time. If a short call takes more than the specified time to complete, then the connection is stopped. The default value for this parameter is 60 seconds. Values of less than 60 seconds are considered to be 60 seconds. Specifying the timeout in a workspace overrides the value set in this field for that workspace.</td>
</tr>
</tbody>
</table>

### Logging and Trace Options
Specific log file format

Defines the name and location of the server log file if you want the data written to a file instead of SYSOUT for the server process. The parameter must specify the name and the high level qualifier.

The following tokens can appear in the log file template and will be replaced accordingly:

- %A: workspace name
- %D: date (yyymmdd)
- %I: instance number of the given workspace server
- %L: server account's login directory
- %P: server's process ID
- %T: time (hhmmss)
- %U: server's account name (username)

Trace options

Specifies the type of tracing to be performed. The following tracing options are available:

- **No timeout**: Select this to disable the standard RPC timeouts, setting them to a long duration (approximately an hour) to facilitate debugging.
- **Call trace**: Select this to generate a message in the server log file for each RPC function called. This is useful for troubleshooting the server.
- **RPC trace**: Select this to enable debugging messages on the server.
- **Sockets**: Select this to generate a message in the server log file for each socket operation. This is useful for troubleshooting client/server communication.
- **Extended RPC trace**: Select this to generate a verbose message in the server log file for each low-level RPC function called. This is useful for troubleshooting the server.
- **System trace**: Select this to generate operating system-specific tracing.
- **Timing**: Select this to generate a timestamp for every entry to the server log file.

Query governing restrictions

Max number of rows in a table that can be read

Select the maximum number of table rows that are read in a query. When the number of rows read from a table exceeds the number stated the query returns an error.

Max number of rows allowed in a table before scan is rejected

Select the maximum number of table rows that can be scanned. This parameter has different behavior for query optimization and execution.

- For query optimization, the value set is compared to the table cardinality. If the cardinality is greater than the value, the scan strategy is ignored as a possible strategy (unless it is the only available strategy).
- For query execution, a scan is limited to the value set. When the number of rows scanned exceeds the number entered, the query returns an error.

### Table 8–10 (Cont.) General Tab Components

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific log file format</td>
<td>Defines the name and location of the server log file if you want the data written to a file instead of SYSOUT for the server process. The parameter must specify the name and the high level qualifier. The following tokens can appear in the log file template and will be replaced accordingly:</td>
</tr>
<tr>
<td>Trace options</td>
<td>Specifies the type of tracing to be performed. The following tracing options are available:</td>
</tr>
<tr>
<td>Query governing restrictions</td>
<td>For query optimization, the value set is compared to the table cardinality. If the cardinality is greater than the value, the scan strategy is ignored as a possible strategy (unless it is the only available strategy). For query execution, a scan is limited to the value set. When the number of rows scanned exceeds the number entered, the query returns an error.</td>
</tr>
</tbody>
</table>
Server Mode

The Server Mode tab lets you configure the features that control the operation of the servers started up by the workspace and allocated to clients.

For example, you can configure the workspace to start up a number of servers for future use, prior to any client request, instead of starting each server when a request is received from a client.

Do the following to open the Server Mode tab:

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

---

Notes:

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

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

---

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

The Server Mode tab is shown in the following figure:
The **Server Mode** tab has the following fields:
### Table 8-11 Server Mode Tab Components

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

Specifies the number of servers in a pool of servers, available to be assigned to a client.

The following options are available:

- **Initial number of servers**: The number of server processes that are prestarted for this workspace when the daemon starts up. When the number of available server processes drops lower than the value specified in the Minimum number field, the daemon again starts server processes until this number of available server processes is reached. The default for this field is 0.

- **Minimum number**: The minimum number of server processes in the prestarted pool before the daemon resumes creating new server processes (to the value specified in the Initial number of servers field). If this field is set to a value higher than the Initial number of servers field, the daemon uses the value specified in the Initial number of servers field. The default for this field is 0.

- **Keep when daemon ends**: When a daemon is shutdown, all the servers started by that daemon are also killed, even if they are active. Set this field to true if you want the servers for the workspace to remain active, even after the daemon has been shut down. If this field is set to true, it is the responsibility of the system operator or manager to ensure that the servers are eventually killed. This must be done at the system level.

- **Set maximum number of servers**: The maximum number of available server processes. Once this number is reached, no new nonactive server processes are created for the particular workspace. For example, if a number of server processes are released at the same time, so that there are more available server processes than specified by this field, the additional server processes higher than this value are terminated. The default for this field is zero, meaning that there is no maximum.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server availability</td>
<td>Specifies the number of servers in a pool of servers, available to be assigned to a client.</td>
</tr>
<tr>
<td>Port range</td>
<td>Determines the range of ports available for this workspace when starting server processes. Use this option when you want to control the port number, so that Oracle Connect can be accessed through a firewall. Enter the port range in the following fields:</td>
</tr>
<tr>
<td>Use default port range</td>
<td>Select this to use the port range that is defined in the daemon. This is defined in the Port range for servers field in the daemon Control tab.</td>
</tr>
<tr>
<td>Maximum number of server processes</td>
<td>Enter the maximum number of server processes that can run at the same time.</td>
</tr>
<tr>
<td>Limit server reuse</td>
<td>Select this if you want to limit the number of servers that can be reused. If this is selected, the Reuse limit parameter is available.</td>
</tr>
</tbody>
</table>
If Limit server reuse is selected, in the field next to the check box, enter the maximum number of times a server can be reused. Select the maximum of clients accepted in a server process.

A one-client server can be reused after its (single) client has disconnected. Reuse of servers enhances startup performance because it avoids the need to repeat initialization.

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

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

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

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

If Limit concurrent clients per server is selected, in the field next to the check box, enter the maximum number of clients that a server process for the current workspace accepts. The default for this field is None, indicating that the number of clients for each server is unlimited. This field is available if the server mode value is multiClient or multiThreaded.

Specify Server Priority Set the priority for servers. For example, a workspace for applications with online transaction processing can be assigned a higher priority than a workspace that requires only query processing. The lower the number, the higher the priority. For example, workspaces with a priority of 1 are given a higher priority than workspaces with a priority of 2.

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

Use default server priority Sets the priority to 0. There is no specific priority for this workspace. Clear this check box to set a priority in the Specify server priority parameter.

Keep when daemon ends Select this to kill all servers started by that daemon when a daemon is shutdown, even if they are active. Select this if you want the servers for the workspace to remain active, even after the daemon has been shut down. If selected, it is the responsibility of the system operator or manager to ensure that the servers are eventually killed. This must be done at the system level.

Server Provisioning

Number of prestarted servers in pool Initial number of servers: The number of server processes that are prestarted for this workspace when the daemon starts up. When the number of available server processes drops lower than the value specified in the Minimum number field, the daemon again starts server processes until this number of available server processes is reached. The default for this field is 0.
**Security**

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

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

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

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

---

**Table 8–11 (Cont.) Server Mode Tab Components**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of spare servers</td>
<td>The minimum number of server processes in the prestarted pool before the daemon resumes creating new server processes (to the value specified in the Initial number of servers field). If this field is set to a value higher than the Initial number of servers field, the daemon uses the value specified in the Initial number of servers field. The default for this field is 0.</td>
</tr>
<tr>
<td>Prestarted server pool limit</td>
<td>The maximum number of available server processes. Once this number is reached, no new nonactive server processes are created for the particular workspace. For example, if a number of server processes are released at the same time, so that there are more available server processes than specified by this field, the additional server processes higher than this value are terminated. The default for this field is zero, meaning that there is no maximum.</td>
</tr>
</tbody>
</table>

**Resource limitations**

<table>
<thead>
<tr>
<th>Nber of sub-tasks</th>
<th>The number of sub-tasks for a server that are prestarted for this workspace when the daemon starts up. In addition to setting up a pool of server processes as described earlier, you can set additional server processes as sub-tasks by specifying this parameter. If you set 10 servers and 10 prestarted sub-tasks then 100 tasks are started (10 sub-tasks for each process).</th>
</tr>
</thead>
</table>
8. Right-click the daemon and select Recycle Servers. Any servers in the connection pool are closed and new servers start with the new configuration.

The Security tab is shown in the following figure:

*Figure 8–8  The Security Tab*

The Security tab has the following fields:
### Table 8–12 Security Tab

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server Account</strong></td>
<td>This section defines the users (accounts) allowed to access the workspace, firewall access ports, workspace account, and anonymous login permissions.</td>
</tr>
</tbody>
</table>
| Use specific workspace account | Select this if you want to define the operating system account used for the workspace.  
If selected, enter the name of the workspace account in the workspace account field.  
If not selected, the account name that was provided by the client is used. |
| Allow anonymous clients to use this workspace | Select this if you want to allow this workspace to be invoked without authentication.  
If selected, enter the name of the workspace account in the Server account to use with anonymous clients field. |
| Authorized Workspace users   | Indicate which users have permission to use the workspace. Select one of the following:  
- All users: Any user who has logged on to the daemon may use the workspace  
- Selected users only: Select this to allow only users (or accounts) with specific permission to use the workspace.  
  When this is selected, add the names of users (or accounts) and groups that can be use the workspace in the field below.  
  Note: If no user is specified, any user who has logged on to the daemon may use the workspace. |
| Authorized Administrators    | Identifies the users (accounts) with administrator privileges. Select one of the following:  
- All users: Indicates that anyone can access the workspace and change the settings.  
- Selected users only: Select this to allow only users (or accounts) with specific permission to be administrators.  
  When this is selected, add the names of users (or accounts) and groups that can be workspace administrators.  
  If no user is specified, any user who has logged on to the daemon may administrator this workspace. |
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 8–13 Data Type Mapping: COBOL and Oracle Connect Back-end Adapter

<table>
<thead>
<tr>
<th>COBOL Data Type</th>
<th>Oracle Connect Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BINARY (with fractional data)</td>
<td>string</td>
</tr>
<tr>
<td>BINARY (without fractional data)</td>
<td>int</td>
</tr>
<tr>
<td>COMP (with fractional data)</td>
<td>string</td>
</tr>
<tr>
<td>COMP (without fractional data)</td>
<td>int</td>
</tr>
<tr>
<td>COMP-2</td>
<td>double</td>
</tr>
<tr>
<td>COMP-3</td>
<td>string</td>
</tr>
<tr>
<td>COMP-4 (with fractional data)</td>
<td>string</td>
</tr>
<tr>
<td>COMP-4 (without fractional data)</td>
<td>int</td>
</tr>
<tr>
<td>COMP-5 (with fractional data)</td>
<td>string</td>
</tr>
<tr>
<td>COMP-5 (without fractional data)</td>
<td>int</td>
</tr>
<tr>
<td>COMP-X (with fractional data)</td>
<td>string</td>
</tr>
<tr>
<td>COMP-X (without fractional data)</td>
<td>int</td>
</tr>
<tr>
<td>INDEX</td>
<td>int</td>
</tr>
<tr>
<td>[SIGN [IS]] LEADING</td>
<td>string</td>
</tr>
<tr>
<td>[SIGN [IS]] LEADING SEPARATE [CHARACTER]</td>
<td>string</td>
</tr>
<tr>
<td>NATIVE-2</td>
<td>int</td>
</tr>
<tr>
<td>COBOL Data Type</td>
<td>Oracle Connect Data Type</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>NATIVE-4</td>
<td>int</td>
</tr>
<tr>
<td>NATIVE-8</td>
<td>string</td>
</tr>
<tr>
<td>PACKED-DECIMAL</td>
<td>string</td>
</tr>
<tr>
<td>POINTER</td>
<td>int</td>
</tr>
<tr>
<td>[SIGN [IS]] TRAILING</td>
<td>string</td>
</tr>
<tr>
<td>[SIGN [IS]] TRAILING SEPARATE [CHARACTER]</td>
<td>string</td>
</tr>
</tbody>
</table>
The OracleAS CDC Adapter for CICS provides the globalization support for the following languages:

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

This appendix describes how to define the language support.

**Defining the Language and Codepage**

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

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

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

- From the **Codepage** list, select the codepage that you want to use with this language. The code pages available are determined by the Language that is selected. If you have additional code pages available, you can manually enter them in this field.

  **Note**: If you change the language, the code page will also change. Check to be sure that you want to use the selected code page with the language you selected.

  If no codepage is selected, the default codepage for the selected language is used.

- From the **NLS string** list, select the NLS string for this language and codepage. The NLS strings available are determined by the code page that is selected. If you have additional NLS strings available, you can manually enter them in this field.

  The codepage is used by a field with a data type defined as `nlsString`. This parameter is used for a field with a codepage that is different than the machine’s codepage. This property includes values for the name of the codepage and whether the character set reads from right to left (as in middle-eastern character sets).

  For example, the following specifies a Japanese EUC 16-bit codepage:

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

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

The following table lists the codepages:

**Table 8–14 NLS Language Codes**

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
<th>Windows Default</th>
<th>ASCII Platforms (Default)</th>
<th>EBCDIC Platforms (Default)</th>
<th>Alternative Codepages (EBCDIC based unless noted otherwise)</th>
</tr>
</thead>
<tbody>
<tr>
<td>English UK</td>
<td>ENUK</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1146</td>
<td>IBM285, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>Language Name</td>
<td>Language Code</td>
<td>Windows Default</td>
<td>ASCII Platforms (Default)</td>
<td>EBCDIC Platforms (Default)</td>
<td>Alternative Codepages (EBCDIC based unless noted otherwise)</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>---------------------------</td>
<td>---------------------------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>French</td>
<td>FRE</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1147</td>
<td>IBM297, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>Latin International</td>
<td>LAT</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1148</td>
<td>IBM500, IBM037, IBM1140, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>Spanish</td>
<td>SPA</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1145</td>
<td>IBM284, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>German</td>
<td>GER</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1141</td>
<td>IBM273, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>Portuguese</td>
<td>POR</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1140</td>
<td>IBM037, IBM500, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>Italian</td>
<td>ITL</td>
<td>Windows-125 2</td>
<td>ISO-8859-15</td>
<td>IBM1144</td>
<td>IBM280, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)</td>
</tr>
<tr>
<td>Greek</td>
<td>GRK</td>
<td>Windows-125 3</td>
<td>ISO-8859-7</td>
<td>IBM875</td>
<td>-</td>
</tr>
<tr>
<td>Russian¹</td>
<td>RUS</td>
<td>Windows-125 1</td>
<td>ISO-8859-5</td>
<td>IBM1154</td>
<td>IBM1025</td>
</tr>
</tbody>
</table>
Defining the Language and Codepage

Table 8–14  (Cont.) NLS Language Codes

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
<th>Windows Default</th>
<th>ASCII Platforms (Default)</th>
<th>EBCDIC Platforms (Default)</th>
<th>Alternative Codepages (EBCDIC based unless noted otherwise)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turkish(^2)</td>
<td>TUR</td>
<td>Windows-1254</td>
<td>ISO-8859-9</td>
<td>IBM1155</td>
<td>IBM1026</td>
</tr>
<tr>
<td>Hebrew</td>
<td>HEB</td>
<td>Windows-1255</td>
<td>ISO-8859-8</td>
<td>IBM424</td>
<td>IBM 862</td>
</tr>
<tr>
<td>Arabic</td>
<td>ARA</td>
<td>Windows-1256</td>
<td>ISO-8859-6</td>
<td>IBM420</td>
<td></td>
</tr>
<tr>
<td>Chinese - Simplified</td>
<td>SCHI</td>
<td>GBK</td>
<td>GBK</td>
<td>IBM935</td>
<td>-</td>
</tr>
<tr>
<td>Chinese - Traditional</td>
<td>TCHI</td>
<td>BIG5</td>
<td>BIG5</td>
<td>IBM937</td>
<td>-</td>
</tr>
<tr>
<td>Korean</td>
<td>KOR</td>
<td>MS949</td>
<td>EUC-KR</td>
<td>IBM933</td>
<td>MS949</td>
</tr>
</tbody>
</table>

1 Russian users who use ANSI 1251 Cyrillic as their Windows codepage must edit the RUS.TXT file and compile it to RUS.CP using the NAV_UTIL CODEPAGE.

2 To work with solutions in Oracle Studio, when using Turkish, add the \(-nl en\) switch to the Target path in the Oracle Studio shortcut properties. For example: "C:\Program Files\Oracle\Studio1\studio.exe -nl en"
In many cases you must manually edit the metadata to configure parts of a solution or composition. Metadata is created in XML format. You define aspects of a solution by changing the values of the elements and attributes of the XML files that belong to the solution. Oracle Studio provides a graphical interface where you can define the various aspects of a solution. This interface lets you make changes easily without having to manually edit the XML file.

Preparing to Edit XML Files in Oracle Studio

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

- Machines
- Bindings.
- Daemons
- Users

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

Figure 8–9  XML Graphical Display
To edit an XML file in Oracle Studio
1. In the Design perspective, open the Navigator view.
2. In the Navigator view, find the item with the XML file that you want to edit. This can be a machine, binding, daemon, or user.
3. Right-click the item and select Open as XML. A graphical list of the file’s elements and attributes opens in the editor.
4. Find the element or attribute (property) that you want to change.
5. Click in the right column next to the property you are changing and edit or add the value.
6. Save the file, then select it again in the Project Explorer and press F5 to refresh. The XML file is updated automatically.

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

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

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

Add DTD Information
You can add DTD information to an element or attribute.

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

Table 8–15 Add DTD Information

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root element name</td>
<td>The name of the XML root element.</td>
</tr>
<tr>
<td>Public ID</td>
<td>The value in this field is the Public Identifier. It is used to associate the XML file (using an XML catalog entry) with a DTD file by providing a hint to the XML processor. Click Browse to select an XML catalog entry from a list. An XML Catalog entry contains two parts, a Key (which represents a DTD or XML schema) and a URI (which contains information about a DTD or XML schema's location). Select the catalog entry you want to associate with your XML file.</td>
</tr>
</tbody>
</table>
| System ID         | The value in this field is the DTD the XML file is associated with. You can change the DTD the file is associated with by editing this field. The XML processor will try to use the Public ID to locate the DTD, and if this fails, it will use the System ID to find it. Click Browse to select a system ID. You can do this in two ways:  
  ■ Select the file from the workbench. In this case, update the with the import dialog box.  
  ■ Select an XML catalog entry. |

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

Edit Namespaces

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

To edit namespaces

1. Right-click an element or attribute and select Edit namespaces. The Edit Schema Information dialog box opens.
2. Click on one of the buttons to make any changes to this information.

**To add a new namespace**

1. From the Schema Information dialog box, click **Add**.
2. The Add Namespace Definitions dialog box opens. Select one of the following:
   - **Select from registered namespaces**. This selection is available when the dialog box opens. Select from the list of registered namespaces and then click OK. If no registered namespaces are available, the list is empty.
   - **Specify new namespace**. Enter the information described in the following table:

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

**To edit a namespace**

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

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

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

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

Replace an Element

You can replace an element with another legal element.

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

**A**
- ACX trace parameter, 8-7
- add timestamp to traced events parameter, 8-7

**B**
- binary XML log level parameter, 8-7

**C**
- cache buffer size parameter, 8-8
- code page parameter, 8-9
- COM maximum XML size in memory parameter, 8-10
- COM maximum XML size parameter, 8-10
- COM XML transport buffer parameter, 8-10
- commit on destroy, 8-9

**D**
- daemon
  - logging, A-3
  - security, A-5
  - server modes, 8-2
  - shutting down, 7-2
  - starting, 7-1
  - timeout, 7-6
- data types
  - atomic metadata, B-1
  - NLS string parameter, 8-9
- debug parameters
  - ACX trace, 8-7
  - add timestamp to traced events, 8-7
  - environment, 8-7
  - GDB trace, 8-7
  - general trace, 8-7
  - binary XML log level, 8-7
  - log file, 8-7
  - optimizer trace, 8-8
  - query processor trace, 8-7
  - query warnings, 8-7
  - trace directory, 8-7
  - Transaction extended logging, 8-8
  - disable 2PC, 8-10
- disk space requirements
  - IBM mainframe, 2-2

**E**
- environment parameters
  - ACX trace, 8-7
  - add timestamp to traced events, 8-7
  - cache buffer size, 8-8
  - code page, 8-9
  - COM maximum XML size, 8-10
  - COM maximum XML size in memory, 8-10
  - COM XML transport buffer, 8-10
  - commit on destroy, 8-9
  - debug, 8-7
  - disable 2PC, 8-10
  - GDB trace, 8-7
  - general trace, 8-7
  - language, 8-9
  - log file, 8-7
  - miscellaneous, 8-8
  - nav utility editor, 8-8
  - NLS string, 8-9
  - odbc, 8-9
  - oledb, 8-9
  - optimizer, 8-9
  - optimizer trace, 8-8
  - parallel processing, 8-9
  - query processor trace, 8-7
  - query warnings, 8-7
  - queryProcessor, 8-9
  - recovery delay, 8-10
  - Replace invalid XML characters, 8-11
  - time limit, 8-10
  - trace directory, 8-7
  - transaction conversions parameters, 8-10
  - Transaction extended logging, 8-8
  - transaction extended logging, 8-9
  - transaction log file, 8-10
  - transactions, 8-9
  - temporary directory, 8-8
  - tuning, 8-10
  - user commit confirm table, 8-10
  - XML, 8-10
  - XML date format parameter, 8-11
  - XML trim char column, 8-11
  - year 2000 policy, 8-8
- error log, binary XML log level parameter, 8-7
- error log, log file parameter, 8-7
G
GDB trace parameter, 8-7
general parameters
cache buffer size, 8-8
nav utility editor, 8-8
temporary directory, 8-8
year 2000 policy, 8-8
general trace parameter, 8-7
generating interactions
inbound, 2-25
outbound, 2-38

H
hardware requirements
IBM mainframe disk space, 2-2
IBM mainframe memory, 2-2
UNIX, 2-2

I
IBM mainframe
post-installation, 2-8
preinstallation, 2-4
IBM mainframe
installing, 2-5
inbound interactions, generating, 2-25
installing
IBM mainframe, 2-5
interactions
inbound, 2-25
outbound, 2-38

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

M
memory requirements
IBM mainframe, 2-2
metadata
atomic data types, B-1
miscellaneous parameters
environment, 8-8

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

O
odbc environment parameters, 8-9
oledb environment parameters, 8-9
operating system requirements
UNIX, 2-2
optimizer
environment parameters, 8-9
trace directory parameter, 8-7
optimizer trace parameter, 8-8
Oracle Application Server requirements, 2-2
outbound interactions, generating, 2-38

P
parallel processing environment parameters, 8-9
post-installation, IBM mainframe, 2-8
preinstallation, IBM mainframe, 2-4
processor requirements
UNIX, 2-2

Q
query optimizer
logging strategy, 8-8
trace directory parameter, 8-7
query processor trace parameter, 8-7
query warnings parameter, 8-7
queryProcessor environment parameters, 8-9

R
recovery delay, 8-10
Replace invalid XML characters parameter, 8-11
requirements
UNIX hardware, 2-1
UNIX software, 2-2

S
security
daemon configurations, A-5
servers
configuring modes, 8-2
reusable, 8-3
Reuse limit daemon parameter, 8-3, A-13
SMS
See system managed storage
software requirements
Oracle Application Server, 2-2
UNIX operating system, 2-2
system managed storage, 2-4
temporary directory parameter, 8-8
temporary files, 8-8
time limit parameter, 8-10
timeout
timeout
client idle, 7-6
daemon, 7-6
trace directory parameter, 8-7
trace information, logging, 8-7
transaction conversions parameters, 8-10
transaction extended logging, 8-9
Transaction extended logging parameter, 8-8
transaction log file, 8-10
transactions environment parameters, 8-9
transactions parameters
commit on destroy, 8-9
disable 2PC, 8-10
recovery delay, 8-10
time limit, 8-10
transaction conversions parameters, 8-10
transaction extended logging, 8-9
transaction log file, 8-10
user commit confirm table, 8-10
tuning environment parameters, 8-10

UNIX
hardware requirements, 2-1
software requirements, 2-2
user commit confirm table, 8-10

Workspace server mode, A-13

XML
environment parameters, 8-10
XML date format parameter, 8-11
XML parameters
COM maximum XML size, 8-10
COM maximum XML size in memory, 8-10
COM XML transport buffer, 8-10
Replace invalid XML characters, 8-11
XML date format parameter, 8-11
XML trim char column, 8-11
XML trim char column parameter, 8-11

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