Contents

Send Us Your Comments ...................................................................................................................... ix

Preface ..................................................................................................................................................... xi

1 Introduction

Overview ............................................................................................................................................. 1-2
Architecture for Oracle Access Manager for AS/400 ................................................................. 1-3
Components ..................................................................................................................................... 1-5
Features ............................................................................................................................................. 1-6
  Powerful SQL Language ............................................................................................................. 1-6
  Leverage Existing Applications ............................................................................................... 1-6
  Universal Access ....................................................................................................................... 1-7
  Datatype Conversions .............................................................................................................. 1-7
  Support for Distributed Applications ...................................................................................... 1-7
Calling Oracle Access Manager for AS/400 .............................................................................. 1-8
IBM Distributed Relational Databases ....................................................................................... 1-9
Activation Groups ...................................................................................................................... 1-10

2 Release Information

Product Set ......................................................................................................................................... 2-2
Changes and Enhancements ......................................................................................................... 2-2
  Release 9.2.0.1.0 ....................................................................................................................... 2-2
  Release 9.2.0.1.0 ....................................................................................................................... 2-2
Known Problems and Restrictions .......................................................................................... 2-4
Problems .......................................................................................................................... 2-4
Restrictions ....................................................................................................................... 2-6

3 System Requirements

Hardware Requirements ........................................................................................................ 3-2
Processor ............................................................................................................................. 3-2
CD-ROM Drive .................................................................................................................. 3-2
Disk Space .......................................................................................................................... 3-2

Software Requirements ..................................................................................................... 3-3
Operating System .............................................................................................................. 3-3
Communications ............................................................................................................... 3-3
Oracle Server ..................................................................................................................... 3-3
PTF Requirements ............................................................................................................. 3-3
Additional Software .......................................................................................................... 3-3

Recommended Documentation ......................................................................................... 3-4

4 Installing Oracle Access Manager

Installation ........................................................................................................................... 4-2
Checklist for Installation .................................................................................................... 4-2
Step 1: Mount the Product CD .......................................................................................... 4-3
Step 2: Start the Installation Process .................................................................................. 4-3
Step 3: Select the Installation Library Name ....................................................................... 4-3
Step 4: Verify Installation Parameters ................................................................................ 4-4
Step 5: Finish the Installation ............................................................................................ 4-5

Post-Installation ................................................................................................................... 4-6
Checklist for Post-Installation ............................................................................................ 4-6
Step 1: Add Oracle Access Install Library to Library List .................................................. 4-7
Step 2: Change Password for User Profile ......................................................................... 4-7
Step 3: Configure Oracle Net ............................................................................................. 4-7
Step 4: Test Connectivity to Oracle Server ......................................................................... 4-7
Step 5: Configure Oracle Access Manager ......................................................................... 4-8
Step 6: Configure Two-Phase Commit (Optional) ............................................................... 4-8
Step 7: Retest Connectivity to Oracle Server ..................................................................... 4-8
5 Oracle Net

Oracle Net File name Structure ................................................................. 5-2
Oracle Net Tracing Members ................................................................. 5-2
Connecting to Oracle through Oracle Net ............................................... 5-3
Defining Oracle Access Manager to the Oracle Server ....................... 5-4
  ORA(TNSNAMES) ................................................................................... 5-4
  Remote Database Directory Entry ......................................................... 5-4
  TCP/IP Entries ..................................................................................... 5-5
Oracle Net for TCP/IP AS/400 Configuration ........................................ 5-6
  Checklist for TCP/IP AS/400 Configuration ......................................... 5-6
  Step 1: Define a Physical Line ............................................................. 5-7
  Step 2: Define TNSNAMES Entry ......................................................... 5-7
  Step 3: Create Relational Database Directory Entry ........................... 5-9

6 Configuring Oracle Access Manager

Oracle Access Manager Commands ......................................................... 6-2
  CHGSQLDFLT ....................................................................................... 6-4
  CHGSQLNET ....................................................................................... 6-8
  CHGSQLDBG ..................................................................................... 6-10
  CRTORAPKG ..................................................................................... 6-12
  CHG2PCPRM ..................................................................................... 6-15

7 Distributed Transaction Coordination

Two-Phase Commit Overview ................................................................. 7-2
  Terminology ....................................................................................... 7-2
Two-Phase Commit and Oracle Access Manager for AS/400 ............... 7-3
Two-Phase Commit Process ................................................................. 7-4
  Phase 1: Prepare .................................................................................. 7-4
  Phase 2: Commit .................................................................................. 7-4
In-Doubt Transactions ........................................................................ 7-5
  Resolving In-Doubt Transactions ....................................................... 7-5
## Enabling Two-Phase Commit Processing

- Checklist for Enabling Two-Phase Commit ................................................................. 7-7
- Step 1: Change Oracle Access Manager Two-Phase Commit Default ....................... 7-8
- Step 2: Change the AM400_RID Options ................................................................. 7-8
- Step 3: Add or Remove AM400_COM Privileged Userids ........................................... 7-11
- Step 4: Start the Subsystem that Runs the AM400_RID Program ......................... 7-12
- Step 5: Start the AM400_COM Program (Optional) .................................................. 7-13

### Oracle Required Privileges

- AM400_COM Program ...................................................................................................... 7-14
- AM400_COM Commands ............................................................................................. 7-15

### 8 Using Oracle Access Manager

- Connecting to the Oracle Server ................................................................................. 8-2
- Connecting through the STRSQL (Interactive SQL) Commands ............................... 8-2
- Connecting from an AS/400 Precompiled Application ............................................ 8-3
- Connection Management ............................................................................................ 8-4
- Describing Tables ......................................................................................................... 8-6
- Stored Procedures ........................................................................................................ 8-6
- Advantages of Stored Procedures ............................................................................. 8-6
- Declaring Procedure Parameters .................................................................................. 8-7
- Declaring Procedures .................................................................................................. 8-7
- Conversion of Oracle Datatypes to DB2/400 Datatypes ............................................ 8-9
- Message and Error Code Processing ......................................................................... 8-10

### A Sample Programs

- Sample Oracle Access Manager C Program ................................................................. A-2
- Sample Oracle Access Manager COBOL Program ...................................................... A-5
- Sample Oracle Access Manager RPG Program .......................................................... A-7

### B National Language Support

- Oracle Access Manager NLS Support .......................................................................... B-2
- Oracle Character Set Names ...................................................................................... B-3
Index
Send Us Your Comments

Oracle Access Manager for AS/400 Installation and User’s Guide, Release 9.2.0.1.0 for IBM AS/400
Part No. A97616-01

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

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

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

infoibm_us@oracle.com

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

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

This manual describes how to install and use the Oracle Access Manager for AS/400 software.

About this Guide

Read this guide if you are responsible for performing tasks such as:

- installing and configuring the Oracle Access Manager gateway
- administering the Oracle Access Manager gateway
- using the Oracle Access Manager gateway

Understand the fundamentals of the Oracle Access Manager and AS/400 before using this guide for installation or system administration.

Intended Audience

This guide is intended for anyone installing, configuring and administering the Oracle Access Manager gateway, as well as for developers writing applications that access remote host databases through the Oracle Access Manager gateway.
Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Product Name

The complete name for this product is Oracle Access Manager for AS/400, also abbreviated as Oracle Access Manager and AM4AS/400 in this document.
Documents Referenced in this Guide

**Oracle Books:**

- Oracle9i Administrator’s Reference
- Oracle Advanced Security Administrator’s Guide
- Oracle9i Application Developer’s Guide - Fundamentals
- Oracle Call Interface Programmer’s Guide
- Oracle9i Database Administrator’s Guide
- Oracle9i Database Error Messages
- Oracle9i Database Reference
- Oracle9i Database Utilities
- Oracle9i Heterogeneous Connectivity Administrator’s Guide
- Oracle9i Net Services Administrator’s Guide
- Oracle9i Net Services Reference Guide
- PL/SQL User’s Guide and Reference
- Oracle9i SQL Reference
- SQL*Plus User’s Guide and Reference

**IBM Books:**

Refer to the IBM documents for your platform and operating system.
Conventions Used in this Guide

Examples of input and output to the system are shown in a special font:

`WRKCFGSTS *DEV device_name`

All output is shown as it actually appears. For input, the following conventions apply:

- The use of monospaced uppercase font, such as `WRKCFGSTS`, indicates a word or phrase that must be entered exactly as spelled.
- The use of monospaced Italic font, such as `device_name`, indicates a variable, a word or phrase for which you must substitute your own word or phrase, such as an actual device name.
- Square brackets indicate that the enclosed arguments are optional. Do not enter the brackets themselves.
- Angle brackets enclosed user-supplied names. Do not enter the brackets themselves.
- Vertical lines separate choices.
- Ellipses indicate that the preceding item can be repeated. You can enter an arbitrary number of similar items.
- Other punctuation, such as commas and quotes, must be entered as shown unless otherwise specified.

Commands, reserved words, and keywords appear in uppercase in both examples and text. In plain text, these words may appear in **BOLD** face font. A file ID can appear with both uppercase and lowercase text. The use of **Italic font** indicates that those portions of a file ID that appear in **Italic font** can vary. Reserved words and keywords must always be entered as is and have reserved meanings within Oracle software products.

Storage Measurements

Storage measurements use these abbreviations:

- KB for kilobyte, which equals 1024 bytes
- MB for megabyte, which equals 1 048 576 bytes
- GB for gigabyte, which equals 1 073 741 824 bytes
Oracle Access Manager for AS/400 enables SQL applications on the AS/400 to access and modify data in an Oracle database.

Read this chapter for information about the architecture and features of Oracle Access Manager for AS/400. Topics include:

- **Overview** on page 1-2
- **Architecture for Oracle Access Manager for AS/400** on page 1-3
- **Features** on page 1-6
- **Calling Oracle Access Manager for AS/400** on page 1-8
- **IBM Distributed Relational Databases** on page 1-9
- **Activation Groups** on page 1-10
Overview

Oracle Open Gateway technology combined with Oracle Access Manager products allow you to fully integrate your AS/400 and Oracle systems. The Oracle Transparent Gateway for DB2/400 and the Oracle Transparent Gateway for IBM DRDA provide Oracle users access to AS/400 data as demonstrated in Figure 1–1. Oracle Access Manager provides AS/400 applications access to Oracle data through SQL statements or stored procedures as shown in Figure 1–2.

Figure 1–1 Oracle Transparent Gateway

GUI → Oracle Server → Oracle Transparent Gateway → AS/400 data

Figure 1–2 Oracle Access Manager

AS/400 → Oracle Access Manager → Oracle Server
Oracle Access Manager for AS/400 provides AS/400 users with access to data that is residing on any supported Oracle platform via DB2/400 SQL or stored procedures. Oracle Access Manager for AS/400 connects to your Oracle environment through Oracle Net. The connection from your AS/400 to an Oracle server uses TCP/IP.

Oracle Access Manager takes advantage of the IBM Application Requester Driver (ARD) interface to access an Oracle server. Refer to the IBM document for System API Reference for a description of this interface.

Oracle Access Manager allows direct access to Oracle data from existing AS/400 applications written in C, COBOL, or RPG or from the AS/400 Interactive Query Manager product (STRSQL). Oracle Access Manager intercepts SQL statements during program execution and directs them to the Oracle server.

Figure 1–3 shows the architecture of Oracle Access Manager for AS/400.
Figure 1–3 Oracle Access Manager for AS/400 Architecture

AS/400

AS/400 Application

SQL Client manager

Application Requester

Driver Interface

Oracle Access Manager

Oracle Net

Oracle Net

Oracle Server

Oracle Database

Oracle Server

Oracle Database

Oracle Transparent Gateway

Oracle

IBM

TCP/IP

TCP/IP

Informix

INGRES

DB2/MVS

Sybase

VSAM

IMS

IDMS

RDB

RMS

...
Components

The components interact in the following ways:

<table>
<thead>
<tr>
<th>Table 1–1 Access Manager Component Interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AS/400 application</strong></td>
</tr>
<tr>
<td><strong>SQL Client Manager</strong></td>
</tr>
<tr>
<td><strong>Application Requester Driver Interface</strong></td>
</tr>
<tr>
<td><strong>Oracle Access Manager</strong></td>
</tr>
<tr>
<td><strong>Oracle server</strong></td>
</tr>
<tr>
<td><strong>Oracle Net</strong></td>
</tr>
</tbody>
</table>
Features

Powerful SQL Language

AS/400 applications can take advantage of the powerful Oracle SQL language to analyze data stored in an Oracle database. Virtually any SQL command can be embedded in an AS/400 application, including INSERT, DELETE, UPDATE, DDL, and SELECT. The SQL commands may contain any functions or clauses that are allowed by Oracle SQL. AS/400 applications can also transparently access procedures and packages stored in an Oracle database, thereby leveraging business logic centralized in the Oracle server. If you have an existing application that uses the IBM CALL command to execute an AS/400 procedure, you can redirect this procedure to call Oracle PL/SQL stored procedure using the same IBM CALL command.

Note:  With two-phase commit enabled, only DML statements (SELECT, INSERT, DELETE, and UPDATE) and calls to stored procedures can be used. If two-phase commit is not enabled, then any Oracle DDL or DML SQL command may be used.

Leverage Existing Applications

Because Oracle Access Manager uses the IBM ARD interface, existing AS/400 applications written in C, COBOL, or RPG can directly access Oracle data. Dynamic SQL access to Oracle data is also available through the IBM DB2/400 Query Manager (STRSQL). AS/400 applications written using ANSI SQL might not require recompilation to access an Oracle server. Oracle Access Manager not only allows you to leverage your investment in existing AS/400 products, such as DB2/400 Query Manager and SQL Development Kit, it also gives you the opportunity to redirect AS/400 applications with little or no change to your application. Your application need not write to a vendor’s proprietary Application Program Interface (API). You simply embed standard SQL statements in your AS/400 application.
Universal Access

AS/400 applications can now benefit from the fully distributed Oracle architecture. Oracle Access Manager allows access to data stored in any Oracle database on your network. When used with Oracle Open Gateway technology, AS/400 applications also have access to over 30 non-Oracle datastores, including DB2/400, VSAM, RDB, Informix, Sybase, and other DB2/400 systems.

Datatype Conversions

Oracle Access Manager ensures that the appropriate datatype conversions are performed between Oracle and the AS/400. This includes support for AS/400-specific datatypes such as zoned decimal. Oracle Access Manager can also translate Oracle date format to the IBM AS/400 default format or to any other date format you select.

Support for Distributed Applications

Because Oracle Access Manager gives your AS/400 application direct access to Oracle data, it is not necessary to download and upload large quantities of data to other processors. Instead of moving data between machines and risking unsynchronized and inconsistent data, you can access data where it resides. From a single program, your AS/400 application can read AS/400 data and insert it into an Oracle database. You can also read Oracle data and insert it into DB2/400 tables.
Calling Oracle Access Manager for AS/400

The AS/400 system calls the Oracle Access Manager for AS/400 during the following operations:

- **package creation**
  During package creation, initiated by any of the AS/400 CRTSQLxxx commands (for example, CRTSQLPKG), the AS/400 calls Oracle Access Manager for AS/400 when the RDB parameter matches the RDB name in a relational database directory entry that is associated with the Oracle Access Manager. The AS/400 system passes SQL statement information from the user program or package. Oracle Access Manager for AS/400 can use this information to plan SQL access for the program. However, SQL statements that are passed to Oracle Access Manager for AS/400 are currently ignored when the package is being created. If the connection fails, then the package creation also fails.

- **CONNECT processing**
  Oracle Access Manager for AS/400 is called when a CONNECT statement is issued specifying an RDB name that matches the RDB name in the relational database directory entry that is associated with the Oracle Access Manager. Environment information is passed to Oracle Access Manager for AS/400 so it can create an environment in which to run the subsequent SQL statements.

- **SQL statement processing**
  Oracle Access Manager for AS/400 is called during SQL processing when the current connection is to an RDB name in the relational database directory entry that is associated with the Oracle Access Manager. The AS/400 passes information about the statement being run so Oracle Access Manager for AS/400 can process the statement.
IBM Distributed Relational Databases

DB2/400 provides support for Distributed Relational Database Architecture (DRDA) to allow an application requester to communicate with IBM application servers. In addition, Oracle application servers can be accessed using the IBM ARD interface. Oracle Access Manager utilizes this interface to integrate Oracle data into your distributed database environment.

DB2/400 supports two levels of distributed databases:

- remote unit of work
- distributed unit of work

Remote unit of work entails preparing and running SQL statements at only one application server during a unit of work. Distributed unit of work entails preparing and running SQL statements at multiple application servers during a unit of work. However, a single SQL statement can refer only to objects located at a single application server.

Oracle Access Manager supports only distributed unit of work and extends its functionality by allowing a single statement to refer to objects located on more than one database server. For more information about distributed relational databases, refer to the appropriate IBM documentation.

A unit of work is a sequence of SQL commands that the database manager (DB2/400) treats as a single entity. The database manager ensures the consistency of data by verifying that all the data changes made during a unit of work are performed or that none of them are performed. A unit of work is ended by a COMMIT or ROLLBACK operation.
Activation Groups

An activation group provides:

- runtime data structures to support running programs
- addressing protection
- a logical boundary for message creation
- a logical boundary for application cleanup processing

All SQL connections are managed at the activation group level. The activation group and RDB name are used together to uniquely identify a connection. It is not possible to simultaneously have more than one active connection with the same RDB name in the same activation group. However, it is possible to have multiple connections with different RDB names in the same activation group and to have multiple connections with the same Oracle server through different activation groups.

Oracle Access Manager is created using ACTGRP(*CALLER). Therefore, it runs in the same activation group as the caller when it is invoked. When called from STRSQL or by the CRTSQLxxx commands, it may run in the default activation group. For more information about connection management and activation groups, refer to the IBM guide for DB2/400 SQL Programming.
This chapter contains release information for Oracle Access Manager for AS/400. It includes:

- Product Set on page 2-2
- Changes and Enhancements on page 2-2
- Known Problems and Restrictions on page 2-4
Product Set

The following products and component versions are included on the product tape. All components are at production level.

- Oracle Access Manager for AS/400: 9.2.0.1.0
- Oracle Net, TCP/IP adapter: 9.2.0

Changes and Enhancements

Release 9.2.0.1.0

LONG and LONG RAW Datatypes

Oracle LONG and LONG RAW columns can now be retrieved, subject to the IBM VARCHAR length limit (32,740 characters). Refer to "LONG and LONG RAW Datatypes" on page 2-6.

Encryption

Saved passwords are encrypted.

Release 9.2.0.1.0

The following bugs have been fixed for this release of the Oracle Access Manager for AS/400:

Bugs Fixed as of This Release

- 2663341
  Defaults not being used when Oracle Number has precision and scale of zero.
- 2189517
  C2M1212 error messages in JobLog.
- 1947812
  MCH3601 error in QRWSARDB (apply IBM PTF SE02609 for OS/400 V5R1.
  Refer to "PTF Requirements" on page 3-3.
Changes and Enhancements

- 1670446
  Internal errors reported with SQL0969 do not have SQLSTATE field (in SQLCA) correctly set.
- 1630400
  MCH3601 in AM400_RID program.
- 1340863
  Lengths of default userid and password are limited to ten chars (they have been expanded to 31 chars).
- 1325197
  Enhance CRTORAPKG to include activation group specification.
- 1318943
  MCH3601 at statement 29 in ARDBIND.
- 1282791
  ORA-3106 when attempting to connect to server with UTF8 charset.
- 1219774
  MCH3601 in AM400_RIDI program.
- 1170151
  Procedure OUT parm incorrect.
- 1168479
  TGTRLS parameter added to CRTORAPKG panel.
- 1116738
  No data returned from final block fetch if rows < blksize.
- 970972
  CEE9901 MCH3601 WHEN USING STRSQL : LANGUAGE SET TO ES.
- 916836
  ZONED DECIMALS ARE LIMITED TO 16 DIGITS -- SHOULD BE 31.
Known Problems and Restrictions

The following problems and restrictions are known to exist in the Oracle Access Manager for AS/400 products on the product CD. The description of problems includes suggestions for dealing with them, when possible.

A current list of problems is available online. Contact your local Oracle Corporation office for information about gaining access to this list.

Problems

STRSQL on Columns of Type NUMBER

When using STRSQL, an ORA-1457 might occur on columns of type NUMBER where no precision or scale is specified. Oracle Access Manager supplies default datatypes *precision and *scale, for these situations. Even the use of these datatypes, however, may not be sufficient, and you could still receive the ORA-1457 error. To check the default datatype, precision and scale values use the Oracle Access Manager for AS/400 CHKSQLDFLT command.

The AS/400 is shipped with the following default settings:

- Datatype *PKDEC (packed decimal)
- Precision 7
- Scale 2

For example, when using STRSQL, a column of type NUMBER with a value of 123456 results in an ORA-1457 error. To avoid this error, you can change the default datatype to *FLOAT or keep the datatype as *PKDEC and change the precision to 8 while keeping the default scale of 2.
General Limitations

Oracle hints cannot be specified in a SQL statement that is explicitly part of any EXEC SQL statement. The IBM preprocessor strips out the Oracle hints, as if they are comments.

If you want Oracle hints in a DELETE, SELECT, or UPDATE statement, you must:

1. Copy the relevant DELETE, SELECT, or UPDATE statement to a host variable.
2. PREPARE a statement from that host variable.
3. For a SELECT, DECLARE a cursor from that prepared statement. Open the cursor, and then FETCH from that cursor.
4. For an UPDATE or DELETE, EXECUTE the prepared statement.

Stored Procedure Limitations

The following limitations are known to exist with Oracle stored procedure:

- The TIMESTAMP and TIME datatypes return the date only.
- You must use the DB2/400 DECLARE PROCEDURE statement in preparation to call an Oracle stored procedure from an AS/400 program.
- The DECLARE PROCEDURE statement must have the same number of arguments as the Oracle PL/SQL procedure. The input and output modes must also be the same.
- The SIMPLE CALL WITH NULLS phrase is not supported. The SIMPLE CALL phrase is supported.
Known Problems and Restrictions

Restrictions

**PREPARE with a COMMIT or ROLLBACK Command**

When using two-phase commit processing, COMMIT or ROLLBACK commands PREPARED as dynamic SQL statements are rejected with this error:

SQL0969: Error occurred while passing request to application requester driver program.

This information is available in the job log:

ARDARPS: cannot PREPARE a COMMIT or ROLLBACK

For additional information on two-phase commit processing, refer to Chapter 8, "Using Oracle Access Manager".

**RDBCNNMTH Parameter set to *DUW**

Oracle Access Manager requires that the RDBCNNMTH parameter be set to *DUW for any SQL package that is preprocessed with any of the CRTSQLxxx commands.

**Non-DML SQL Statements**

When using two-phase commit processing, CONNECT is the only non-Data Manipulation Language (DML) SQL statement supported by Oracle Access Manager. Use of such non-DML statements results in this error:

ORA-2089 COMMIT is not allowed in a subordinate session

**COMMIT HOLD**

The HOLD parameter on an EXEC SQL COMMIT HOLD command is not honored. All cursors are closed at each logical unit of work boundary (COMMIT and ROLLBACK).

**LONG and LONG RAW Datatypes**

Only the first 32,740 bytes in a LONG or LONG RAW column can be retrieved. This is because of the DB2/400 length limit on a LONG VARCHAR column. Also note that the total data length on a SELECTed row is 32,760 bytes. The maximum length for a LONG or LONG RAW column will be further diminished by the amount of data in the OTHER columns retrieved.
This chapter describes the hardware and software requirements of the Oracle Access Manager for AS/400. Topics include:

- **Hardware Requirements** on page 3-2
- **Software Requirements** on page 3-3
- **Recommended Documentation** on page 3-4
Hardware Requirements

Processor
The Oracle Access Manager for AS/400 requires an AS/400 processor that supports any current version of OS/400.

CD-ROM Drive
The Oracle Access Manager for AS/400 is distributed on a CD.

Disk Space
During installation, 100 MB of disk space may be consumed (but given back). After installation, you will note that approximately 110 MB of disk space has been used. You should have at least 200 MB of free disk space available before beginning installation of this product.
Software Requirements

Operating System

OS/400, any version supported by IBM.

Communications

The Oracle Access Manager for AS/400 requires a TCP/IP connection to the target Oracle server.

Oracle Server

Any current release of the Oracle server is supported. The Oracle server must be configured with TCP/IP access.

PTF Requirements

Refer also to "Bugs Fixed as of This Release" and to Bug "1947812" on page 2-2. For OS/400 V4R5, you should have the latest DB2/400 Group PTF installed. It must be dated December 21, 2001 or later. Do a DSPDTAARA QSYS/SF99105 to verify the date.

Additional Software

No additional software is needed for OS/400 V5. With OS/400 V4R5, you must have the SQL Development Kit (IBM Program Number 5769-ST1).
The following documents might be useful in providing additional information about products used with the Oracle Access Manager for AS/400:

- IBM Guide for DB2/400 SQL Programming
- *Oracle Call Interface Programmer’s Guide*
- *Oracle9i Net Services Administrator’s Guide*
- *Oracle9i Net Services Reference Guide*
- *Oracle9i SQL Reference*
The installation of the Oracle Access Manager for AS/400 is divided into two parts:

1. installation steps
2. post-installation steps

Before you begin installation, ensure that you have the required hardware and software that are described in Chapter 3, “System Requirements”.

This chapter contains the following topics:

- Installation on page 4-2
- Post-Installation on page 4-6
Installation

Before installing Oracle Access Manager for AS/400, you must log on with a user profile that has the *SECADM special authority. The user profile QSECOFR normally has that authority.

---

**Caution:** Oracle Corporation recommends that you back up any existing Oracle Access Manager library before upgrading to a new release.

---

The panels are shown as they appear when first displayed. The values documented in this installation guide are the default values that are provided by Oracle Corporation at the time that you install the Oracle Access Manager for AS/400. The default values are used unless they are manually changed before continuing.

Checklist for Installation

1. **Step 1:** Mount the Product CD
2. **Step 2:** Start the Installation Process
3. **Step 3:** Select the Installation Library Name
4. **Step 4:** Verify Installation Parameters
5. **Step 5:** Finish the Installation
Step 1: Mount the Product CD

Insert the product CD into the AS/400 CD drive.

Step 2: Start the Installation Process

Enter the command:

LODRUN CD_drive

where CD_drive is the name of the CD drive where you mounted the product CD. The default is normally OPT01.

Step 3: Select the Installation Library Name

When the Install Oracle Access Manager Product Panel is displayed, use the default name of ORAAM400 or enter a name from 1 to 10 characters long.

The specified library name must denote an existing, but empty, library, or must denote a nonexistent library. The installation procedure will not install into a library with existing entries.

For auxiliary storage pool ID, use the default of 1, or enter another ID.

After entering your information on the panel, press [Enter] to continue.
**Step 4: Verify Installation Parameters**

The following panel appears:

*Figure 4–2 Verify Oracle Access Manager Install Panel*

Verify Oracle Access Manager Install at V9.2.0.1.0

System: HQAS400

Verify that you want to install the product. Once this process has started, it will create a library and other objects. Enter *YES to verify this operation before it is started.

New Install Library name ........ ORAAM400
Enter *YES to verify ............... *NO *NO, *YES

Enter *YES to continue the installation process. Press [Enter] to continue.
Step 5: Finish the Installation

If you entered *YES, then status messages will be displayed on the bottom of the screen as the installation proceeds. After a few minutes, a screen will appear, stating:

Install complete. Press ENTER. Check Job log for any errors.

When the installation is finished, check the job log for error messages with the following command:

DSPJOBLOG

Installation of Oracle Access Manager for AS/400 is complete.

---

**Note:** During installation, an AS/400 userid is created with the same name and password as the install library name. By default, the userid and the password are ORAAM400. You need to change the password using the following command syntax:

```
CHGUSRPRF USRPRF(install_lib) PASSWORD(new_password)
```

where:

*install_lib* is the installation library. The default is ORAAM400.

*new_password* is the new userid password that you want to use.
Post-Installation

This section describes the steps to be performed after installing Oracle Access Manager for AS/400.

Checklist for Post-Installation

- Step 1: Add Oracle Access Install Library to Library List
- Step 2: Change Password for User Profile
- Step 3: Configure Oracle Net
- Step 4: Test Connectivity to Oracle Server
- Step 5: Configure Oracle Access Manager
- Step 6: Configure Two-Phase Commit (Optional)
- Step 7: Retest Connectivity to Oracle Server
Step 1: Add Oracle Access Install Library to Library List

Before attempting to use any of the Oracle Access Manager configuration utilities, you must add the Oracle Access Manager library to your library list. To do this, enter the following command on any command line:

```
ADDLIBLE library_name
```

where `library_name` is the name of your Oracle Access Manager installation library.

The default position is *FIRST*.

You do not need to manipulate the library list when running the Oracle Access Manager itself. It will determine which libraries need to be added to the library list, and it will add those libraries to the top of the user portion of the library list.

Step 2: Change Password for User Profile

Use the CHGUSRPRF command to change the user profile. The user profile name is identical to the installation library name.

Step 3: Configure Oracle Net

Configure Oracle Net on the Oracle integrating server. Refer to Chapter 5, "Oracle Net" for more information.

Step 4: Test Connectivity to Oracle Server

Refer to “Connecting through the STRSQL (Interactive SQL) Commands” on page 8-2 in Chapter 8, "Using Oracle Access Manager" for information about connecting to an Oracle server.

---

**Note:** You should test connectivity before changing any configuration parameters.
Step 5: Configure Oracle Access Manager

Configure Oracle Access Manager for AS/400. All options have defaults that are used unless you change them. Refer to Chapter 6, “Configuring Oracle Access Manager” for more information.

**Note:** Steps 6 and 7 need to be performed only if you will be using two-phase commit.

Step 6: Configure Two-Phase Commit (Optional)

If you want to use two-phase commit processing, then you need to configure it now. Refer to Chapter 7, “Distributed Transaction Coordination” for the steps to use in configuring two-phase commit processing.

Step 7: Retest Connectivity to Oracle Server

At this point, you need to retest your connectivity to the Oracle server. Refer to “Connecting through the STRSQL (Interactive SQL) Commands” on page 8-2 for information about connecting to an Oracle server.
In the client/server model, a client application accesses a database server to satisfy database operations initiated by an application. The Oracle Access Manager for AS/400 is based on the client/server model. All application processing, such as the user interface and data presentation, occurs at the client workstation on an AS/400. All Oracle database processing occurs at the Oracle server, which, when using Oracle Access Manager for AS/400, is not on an AS/400 computer.

Interaction between the client and server is executed through a common dialogue or messaging system designed for the SQL language. With Oracle Net, databases and their applications can reside on different computers and communicate as peer applications in a Transparent Networking Substrate (TNS) network.

Read this chapter to learn how to configure and use Oracle Net with Oracle Access Manager for AS/400. Topics include:

- Oracle Net File name Structure on page 5-2
- Connecting to Oracle through Oracle Net on page 5-3
- Defining Oracle Access Manager to the Oracle Server on page 5-4
- Oracle Net for TCP/IP AS/400 Configuration on page 5-6
Oracle Net File name Structure

Oracle9i Net Services Administrator’s Guide and Oracle9i Net Services Reference Guide refer to files in the form:

basename.extension

where:

basename is the base portion of the name
extension is the extension portion of the name.

An example of this form is TNSNAMES.ORA.

On the AS/400, the Oracle Net parameter files are installed in the Oracle Access Manager library using this file name mapping:

■ the base portion of the file name is mapped to a member in an AS/400 file
■ the extension portion of the file name is mapped to an AS/400 file

For example, the TNSNAMES.ORA parameter file is mapped to member TNSNAMES in the ORA file located in the Oracle Access Manager library.

Member names are referred to as:

file(member_name)

Oracle Net Tracing Members

When Oracle Net tracing is enabled with the CHGSQLNET command, members with names similar to CLNTnnnnnn are created in the TRC file in the Oracle Access Manager library. These members contain Oracle Net trace records that are to be used by Oracle Support Services for diagnosing communications problems. The nnnnnn corresponds to the OS/400 job number.

For more information on Oracle Access Manager commands, see Chapter 6, "Configuring Oracle Access Manager".
Connecting to Oracle through Oracle Net

When an AS/400 application communicates with a remote IBM database server, it uses the IBM DRDA connection that is defined through the AS/400 ADDRDBDIRE command.

Figure 5–1 Native IBM Networking

When an Oracle SQL application communicates with a remote Oracle server, it uses an Oracle Net connection defined in the ORA(TNSNAMES) file.

Figure 5–2 Native Oracle Networking

Connecting through Oracle Access Manager involves combining the IBM DRDA connection with the Oracle Oracle Net connection.

Figure 5–3 Oracle Access Manager Networking

In Figure 5–3, the remote database name is used as the key for both the RDBDIRE lookup and the TNSNAMES lookup.
Defining Oracle Access Manager to the Oracle Server

To connect to Oracle through Oracle Net, you must define an Oracle Net TNSNAMES entry with a name that matches your AS/400 RDBDIRE entry. This section provides a general discussion of these definitions. To create these definitions, refer to "Oracle Net for TCP/IP AS/400 Configuration" on page 5-6 for specific instructions. Figure 5-4 shows how these matching entries interact:

**Figure 5–4 Defining Oracle Access Manager to the Oracle Server**

![Diagram showing the interaction between Oracle Access Manager, RDBDIRE, TNSNAMES, and Oracle Net]

**ORA(TNSNAMES)**

To connect to the target Oracle server through Oracle Net, you must add an entry to the ORA(TNSNAMES) file in the Oracle Access Manager installation library. The entry name must match the RDB directory entry relational database name.

**Remote Database Directory Entry**

Using the AS/400 ADDRDBDIRE command, you must define a remote database directory entry on your AS/400 for each Oracle Oracle Net TNSNAMES entry. The relational database name must have a matching entry in the ORA(TNSNAMES) file.
TCP/IP Entries

The following examples map the relationship between the AS/400 TCP/IP configuration and Oracle Access Manager. The "OracleTCP" entries (in bold font) are the matching entry names.

- ORA(TNSNAMES)

  OracleTCP = (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=TCP)
      (PORT=1521)
      (HOST=UNIX9)
    )
    (CONNECT_DATA=(SID=Oracle92)
  )

- AS/400 Relational Database Directory Entry (DSPRDBDIRE)

  Relational database . . . . . . : ORACLETCP
  Remote location : . . . . . . . : *ARDPGM
  Application requester driver:
  Program . . . . . . . . . . . : ORAARD
  Library . . . . . . . . . . . : ORAAM400
  Interface level . . . . . . . : 1
  Text. . . . . . . . . . . . . . : 

The TNSNAMES entry may be in any case but will be mapped to uppercase by Oracle Access Manager.
Oracle Net for TCP/IP AS/400 Configuration

Perform these steps to configure your client machines for TCP/IP. Refer to the networking documentation for your client machine for more information about configuring it.

Checklist for TCP/IP AS/400 Configuration

- Step 1: Define a Physical Line
- Step 2: Define TNSNAMES Entry
- Step 3: Create Relational Database Directory Entry
Step 1: Define a Physical Line

This step is required only if no current physical connection exists between the AS/400 and the Oracle server. If both the AS/400 and the Oracle Server computers are already connected to a LAN, for example, and if both machines can successfully communicate over that LAN, then you can skip this step.

To define the physical connection, use any of the following AS/400 commands:

- CRTLINETH creates an ethernet connection.
- CRTLINS DLC creates an SDLS connection.
- CRTLINTRN creates a token ring connection.
- CRTLINX25 creates an X.25 connection.

When you use one of these commands, you can set the AUTOCREATE CONTROLLER parameter to *YES (the AS/400 system default is *NO). This enables the AS/400 auto-configuration feature.

If a line is already defined, then you can use the DSPLIND line_name command to display the line description parameters.

These line description parameters are used to configure the network on the machine where the Oracle Access Manager resides. For example, when using a token ring system, you must know the AS/400 token ring address.

Step 2: Define TNSNAMES Entry

Edit member TNSNAMES in the ORA file in the Oracle Access Manager installation library using this command:

WRKMBRPDM library_name/ORA

where library_name is the name of your Oracle Access Manager installation library. Enter a "2" on the line for TNSNAMES to edit the TNSNAMES member. If you do not have the Program Development Manager, then you can use the EDTF command to edit the member.
The TNSNAMES entry name must match the AS/400 RDB directory entry that will be defined in Step 3.

```
server_name = (DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=TCP)
    (PORT=port_number)
    (HOST=host_name)
  )
  (CONNECT_DATA=(SID=instance_name)
  )
)
```

where:

- **server_name** is the name of an Oracle server that matches an entry in the RDB directory. An entry in the RDB directory can be added using the ADDRDBDIRE command.

- **TCP** is the TCP protocol used for TCP/IP connections.

- **port_number** is the port number of the Oracle Net listener. This is usually port number 1521.

- **host_name** is the name that defines the system where the target Oracle server resides. This name must be in the local host definition on the AS/400 or in a name server on your network. The host name can also be entered as an IP address, for example, 161.14.10.12.

- **instance_name** is the instance name of the Oracle server, usually specified as the system identifier (SID).

Here is an example:

```
OracleTCP = (DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=TCP)
    (PORT=1521)
    (HOST=UNIX9)
  )
  (CONNECT_DATA=(SID=Oracle92)
  )
)
```
Step 3: Create Relational Database Directory Entry

Using the AS/400 ADDRDBDIRE command, define a relational database directory entry (Panel 1, Figure 5–5). Enter the name of the entry on the line entitled "Relational database". Enter *ARDPGM on the line entitled "Name or address". Do not change the data on the line entitled "Type". You may change the text parameters to whatever you choose. Press [Enter]. The screen will change to add a section entitled "Application Requester Driver" (Panel 2, Figure 5–6).

Figure 5–5 ADDRDBDIRE Panel 1
In Panel 2 (Figure 5–6), enter ORAARD on the line entitled "Program", and enter the name of the install library for the Access Manager on the line entitled "Library".

**Figure 5–6 ADDRDBDIRE  Panel 2**

Add RDB Directory Entry (ADDRDBDIRE)

Type Choices, press Enter.

Relational database ............... > ORACLETCP__________________
Remote location:
   Name or address ............... > *ARDPGM_____________________________________

Type ......................... ____ *SNA, *IP
Text .......................... > 'TCP/IP connection to oracle'_____________________

Application requester driver:
   Program ........................ ORAARD____ Name, *DRDA
   Library ........................ ORAAM400__ Name, *LIBL, *CURLIB

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
(C) Copyright Oracle Corporation, 2002
The following parameters must be entered:

- **Relational database** is the name that you are assigning to the Oracle server relative to the AS/400. This must match the entry name of the ORA(TNSNAMES) entry.
- **Remote location** must be *ARDPGM for the ARD interface.
- **Program** must be ORAARD, the name of the Oracle Access Manager executable program.
- **Library** must be the name of the Oracle Access Manager installation library.

You can also change the **text** parameter to whatever you choose, where *text* is a form of string text. However, Oracle Corporation recommends that you do not change any other parameters on the ADDRDBDIRE screen.
After installing Oracle Access Manager, you can:

- create a SQL package
- use the STRSQL command to interactively access data in an Oracle server

All Oracle Access Manager parameters or system values requiring modification can be changed using Oracle Access Manager commands or through a menu system. These commands and their menus are described in this chapter. Topics include:

- **Oracle Access Manager Commands** on page 6-2
- **CHGSQLDFLT** on page 6-4
- **CHGSQNLNET** on page 6-8
- **CHGSQLDBG** on page 6-10
- **CRTORAPKG** on page 6-12
- **CHG2PCPRM** on page 6-15
Oracle Access Manager Commands

Before attempting to enter any of the Oracle Access Manager commands, you must add the Oracle Access Manager library to your library list. To do this, enter the following command on any command line:

`ADDLIB library_name`

where `library_name` is the name of your Oracle Access Manager installation library. The default position is `FIRST`.

Then enter:

`GO CMDORAAM`

The following command displays the Oracle Access Manager commands menu.

**Figure 6–1  Oracle Access Manager Commands Panel**

<table>
<thead>
<tr>
<th>CMDORAAM</th>
<th>Oracle Access Manager Commands</th>
<th>System: HQAS400</th>
</tr>
</thead>
</table>

Select one of the following:

1. Change Oracle Access Manager Default settings
2. Change Oracle Access Manager Oracle Net settings
3. Change Oracle Access Manager Debug settings
4. Create an Oracle Access Manager SQL Package
5. Change Oracle Access Manager Two-phase commit parameters

Selection or command

```
===>
```

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel

(C) Copyright Oracle Corporation, 2002
Each of the options on the Oracle Access Manager commands menu can be accessed directly by entering the command name from any command line. The commands are listed in the following table:

<table>
<thead>
<tr>
<th>Menu Choice</th>
<th>Command Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CHGSQDLDFLT</td>
<td>Changes the Oracle Access Manager default settings.</td>
</tr>
<tr>
<td>2</td>
<td>CHGSQNLNET</td>
<td>Changes the Oracle Net trace settings.</td>
</tr>
<tr>
<td>3</td>
<td>CHGSQDBG</td>
<td>Changes Oracle Access Manager debugging parameters.</td>
</tr>
<tr>
<td>4</td>
<td>CRTORAPKG</td>
<td>Creates a SQL package using the specified source and Oracle server name.</td>
</tr>
<tr>
<td>5</td>
<td>CHG2PCPRM</td>
<td>Changes Oracle Access Manager two-phase commit parameters.</td>
</tr>
</tbody>
</table>
After selecting option 1 on the Oracle Access Manager commands menu or issuing the CHGSQLDFLT command, the first panel that is displayed confirms which library is to be changed. This should match the library where you installed Oracle Access Manager for AS/400.

**Figure 6–2  Oracle Access Manager Default Options Panel**

Type choices for default option, press Enter.

Existing Install library name ....ORAAM400    Name

The next panel displays the default userid and default password that are needed for the Oracle server to which you are connecting. The defaults supplied on the distribution tape are SCOTT/TIGER. You will probably want to change this default for security reasons.
The ten fields on the panel in Figure 6–3 are described as follows:

**Existing Install library name** identifies the library containing the Oracle Access Manager program. The default is the Oracle Access Manager installation library.

**Access Manager default user id** specifies the default userid. The default userid and default password are used if the USER or PASSWORD parameter is not specified in the CRTSQLxxx command or if the CONNECT TO statement is issued without the USER or USING options. If you are concerned about security, you can change this field to BLANK. This forces you to always explicitly enter a userid and password in order to CONNECT.
Access Manager default password specifies the default password for the default userid. The default userid and default password are used if the USER or the PASSWORD parameter is not specified in the CRTSQLxxx command, or if the CONNECT TO statement is issued without the USER or USING options.

Oracle Corporation recommends that you use the default password only for testing, and that in your application, you imbed the user ID and password. A password, if given, will be encoded.

Access Manager default blocksize specifies the default block size. The Oracle Access Manager uses the blocksize data area to determine the number of rows to retrieve in a single fetch. The default is 100. Any SELECT statement must contain the FOR FETCH ONLY clause in order for block fetching to be performed.

Access Manager default number format specifies the DB2/400 datatype for an Oracle returned NUMBER field with precision and scale equal to zero.

*FLOAT specifies floating point notation. *PKDEC specifies packed decimal notation with precision set to the default precision and scale set to the default scale (refer to "Access Manager default scale" in this list). Oracle functions, such as AVG, COUNT(*), MAX, and SUM, return NUMBER values with precision and scale set to zero. For example, using the default precision and scale values of 9 and 2, respectively, the following SQL statements return the following values:

```
SELECT AVG(DEPTNO) FROM DEPTNO;
25.00 if *PKDEC is specified
2.50E+001 if *FLOAT is specified
SELECT 1/3 FROM DUAL;
.33 if *PKDEC is specified
3.33E-001 if *FLOAT is specified
```

*PKDEC is the default.

Access Manager default precision specifies the default precision to be used if *PKDEC is specified for Access Manager default number format. The default is 9.

Access Manager default scale specifies the default scale to be used if *PKDEC is specified for Access Manager default number format. The default is 2.
Access Manager NLS language specifies the National Language Support (NLS) language. NLS enables Oracle applications to interact with users in their native language using their conventions for displaying data. The default NLS language can be changed to any valid value. Refer to Appendix B, "National Language Support" for a list of valid values.

Access Manager Two-Phase Commit specifies if two-phase commit processing is to be enabled. Two-phase commit ensures transactional consistency in a heterogeneous environment. If set to *NO, data might not be synchronized between the local and remote database. If set to *NO, then DDL statements, such as CREATE and DROP, may be used.

If set to *YES, then DDL statements cannot be used.

Commitment-Control Resource Release specifies when the commitment control resource, allocated to perform a job, is released. Oracle Access Manager for AS/400 is required (by IBM Access Requester specifications) to use an AS/400 commitment control resource.

If you intend to use Oracle Access Manager within the confines of a user-defined commitment control definition, then you need Oracle Access Manager to release its commitment control resource when a transactional unit of work ends at a commitment control boundary. You reach a commitment control boundary when a COMMIT or ROLLBACK operation is performed. When Oracle Access Manager starts a new unit of work, it then reacquires the required commitment control resource. For this scenario, set Commitment-Control Resource Release to *RELEASE.

If you are not planning to use commitment control definitions external to Oracle Access Manager for AS/400, then you can specify *KEEP for this parameter.
After selecting option 2 on the Oracle Access Manager commands menu or issuing the CHGSQLNET command, the first panel that is displayed confirms which library is to be changed. This should match the library where you installed Oracle Access Manager for AS/400.

Figure 6–4  Oracle Access Manager Network Parameters Panel

Change Oracle Access Manager Network Parameters

Type choices for default option, press Enter.

Existing Install library name ........... ORAAM400

F3=Exit    F4=Prompt    F9=Retrieve    F12=Cancel

On the next panel, you can change the settings of the Oracle Net trace options. Because running a Oracle Net trace can adversely affect AS/400 and Oracle Access Manager performance, do not change these settings unless you are asked to do so by Oracle Support Services.
The fields on the panel in Figure 6–5 are described as follows:

Existing Install library name identifies the library containing the Oracle Access Manager program. The default is the Oracle Access Manager installation library.

Client trace level specifies the tracing level for the AS/400 client. Valid values are *OFF, *USER, *ADMIN, and 16. The number 16 gives full trace information. Additional values can be supplied by Oracle Support Services for debugging purposes.
After selecting option 3 on the Oracle Access Manager commands menu or issuing the CHGSQLDBG command, the first panel displayed confirms which library is to be changed. This matches the library where you installed Oracle Access Manager for AS/400.

**Figure 6–6  Oracle Access Manager Debugging Options Panel**

Change Oracle Access Manager Debugging Options

System: HQAS400

Type choices for debugging options, press Enter.

Existing Install library name ........ ORAAM400

F1=Help  F4=Prompt  F9=Retrieve  F12=Cancel

(C) Copyright Oracle Corporation, 2002

On the next panel, you can change the debugging settings. The debugging function helps diagnose Oracle Access Manager problems. Do not change these settings unless you are asked to do so by Oracle Support Services.
The fields on the panel in Figure 6–7 are described as follows:

**Existing Install library name** identifies the Oracle Access Manager installation library.

**Access Manager debugging control string** specifies the level of tracing. A 0 turns off all tracing and a 1 turns on full tracing. The trace output always goes to a member in the file determined by the **Access Manager trace file** and the **Access Manager trace library**.

**Access Manager trace file** specifies the qualified file name for trace output under control of the Oracle Access Manager debugging control string. The file needs to be created as a SOURCE PHYSICAL FILE with a maximum record length of 240.
Access Manager trace library specifies the library portion of a qualified file name. This field is used along with the Access Manager trace file field to designate a file into which trace members are written. If specified as blank, then a value of QTEMP is used.

This field is relevant only if the Access Manager debugging control string field contains a nonzero value.

After selecting option 4 on the Oracle Access Manager commands menu, or after issuing the CRTORAPKG command, the Oracle SQL Package panel is displayed. Oracle Access Manager provides CRTORAPKG to facilitate the development of distributed applications that are targeted for execution against an Oracle server. Oracle Access Manager issues the necessary commands to precompile, compile, and bind your application program. Refer to Chapter 8, "Using Oracle Access Manager" for information on sample programs.
Figure 6–8  Oracle SQL Package Panel

Create Oracle SQL Package

Type choices, press Enter.

File ............................... QSQLCSRC  name
   Library ............................ ARDQA  name
Member ............................... XI1  name
Language Type ....................... *C  *C, *RPG, *COBOL
Oracle server .................... ORACLETCP
Activation Group .................. *New
Object Type ........................ *PGM
Batch/Terminal Compile .......... *BATCH
Target Release ..................... *CURRENT
Additional pre-compile options ....

====>

F1=Help  F4=Prompt  F9=Retrieve  F12=Cancel
(C) Copyright Oracle Corporation, 2002

The required input fields on the panel in Figure 6–8 are as follows:

File  is the name of a file containing the program source.
Library  is the name of the library containing the file.
Member  is the source member in the file.
Language type  is the language type. Valid values are *C, *COBOL, and
*RPG.
Oracle server  is the name of the target Oracle server. This must match an
entry name in ORA(TNSNAMES) and an entry in the RDB directory.
After filling in the data, the file, library, and member entries are verified to ensure that they exist. A batch job is submitted to create the package.

---

**Caution:** During compilation, CRTORAPKG attempts to connect to the specified Oracle server. Ensure that your Oracle server and your network connection are active before running CRTORAPKG. You can use STRSQL to check the status of your connection.

---

After the batch job is complete, check the job log to determine if the package creation was completed successfully. Check the output spool queue for each of the submitted jobs. If package creation failed, then check for the following problems:

- syntax errors in the program source
- missing RDB directory entry
- missing or incorrect Oracle server entry name in ORA(TNSNAMES)
- unavailable target Oracle server
- nonfunctional Oracle Net system

The CRTORAPKG command uses the relevant IBM CRTSQLxxx command to pre-process your file. Failure will occur only if the severity level of an error is greater than 30. Some severity level 30 messages are issued because the preprocessor does not recognize some Oracle SQL constructs or functions. For this reason, you must examine the spooled output to determine the validity of any errors of severity level 30.

The IBM precompiler flags any reference to Oracle server sequence numbers (error SQL5001, severity 30) and any reference to non-IBM functions, such as TO_CHAR and TO_DATE (error SQL0105, severity 30), as errors.

After finding and correcting the error, resubmit the job from the CRTORAPKG panel. Ensure that the input information is correct before pressing [Enter].
CHG2PCPRM

This command allows you to change Oracle Access Manager two-phase commit parameters. The following panel is displayed:

**Figure 6–9  Change RID (Resolve In Doubt) Options Panel**

```
Change RID (Resolve In Doubt) Options

Installation Library: ORAAM920

DataQueue Timeout (in seconds) ........ 90
CleanUp Delay (in seconds) ............ 30
Trace Level ........................... 49
Dead Check Counter .................... 100

Type options, press Enter.
1-Add  2=Change  4=Remove

Opt     Oracle Server Name     Description
1-VMS_SRV1               Connect to VMS1X system
2-MVS03                  MVS system 3
3-MVS_92                 MVS running Oracle 9.2
4-PC_GTW_92              WinNT 4.0 server - 9.2
5-H920_130               SUN running Oracle 9.2.0

Bottom

F1=Help  F4=Prompt  F9=Retrieve  F10=Switch to PrivID screen  F12=Cancel

(C) Copyright Oracle Corporation, 2002
```

For more information about this panel, refer to Chapter 7, "Distributed Transaction Coordination".
Distributed Transaction Coordination

Read this chapter to learn about two-phase commit processing. Topics include:

- Two-Phase Commit Overview on page 7-2
- Two-Phase Commit and Oracle Access Manager for AS/400 on page 7-3
- Two-Phase Commit Process on page 7-4
- In-Doubt Transactions on page 7-5
- Enabling Two-Phase Commit Processing on page 7-7
- Oracle Required Privileges on page 7-13
- AM400_COM Program on page 7-14
Two-Phase Commit Overview

Oracle Access Manager for AS/400 supports two-phase commit processing to ensure transaction consistency in a heterogeneous environment. Two-phase commit processing is an optional feature in Oracle Access Manager for AS/400.

During installation, the Oracle Access Manager parameter for two-phase commit processing defaults to *NO. To enable transaction consistency in a heterogeneous environment, the two-phase commit parameter must be changed to *YES on the Change Oracle Access Manager Default Options panel or by using the CHGSQDLFT command. For additional information, refer to Chapter 6, "Configuring Oracle Access Manager".

Note: If the two-phase commit parameter is set to *NO, then data might not be synchronized between local and remote databases. Synchronization is needed ONLY when data changes (changes in files or tables) are being made both on the AS/400 and on the remote Oracle server.

Terminology

**AM400_RID program** automatically resolves in-doubt transactions. This program must be active to enable two-phase commit processing.

**AM400_COM program** provides a mechanism for privileged users to dynamically communicate with the AM400_RID program.

**Two-phase commit parameter** enables two-phase commit processing in Oracle Access Manager for AS/400. The default setting is *NO. It must be changed to *YES to enable two-phase commit processing.

**Commit point site** is the node that initiates a commit or rollback. The AS/400 that is participating in a two-phase commit transaction is always designated as the commit point site. The commit point site is also known as the "coordinator" site.

**Prepare phase** is the phase during which the coordinator asks participants to prepare to commit or to rollback.
**Commit phase** is the phase during which the coordinator asks all nodes to commit the transactions, if all participants have responded to the coordinator that they are prepared. If any participants cannot prepare, then the coordinator asks all nodes to rollback the transaction.

**Implied commit** When a program (activation group) ends without error, default behavior for the AS/400 system triggers all pending updates to be committed. This commitment of all pending updates would include updates to Oracle data as part of the AS/400 transactions through the Oracle Access Manager for AS/400.

---

**Two-Phase Commit and Oracle Access Manager for AS/400**

Oracle Access Manager for AS/400 enables AS/400 applications to update data on both the AS/400 and remote Oracle servers during a single transaction. Any time that updates are allowed in a distributed environment, a new level of complexity is introduced. When multiple users are sharing and accessing data that exists at many sites, rather than at a single site, these distributed transactions require careful management.

With Oracle Access Manager for AS/400, you now have the option to ensure that all participants (nodes) in a distributed transaction react unanimously to the action to take on a transaction. That is, they all commit, or they all rollback.

For a database to fully participate in a two-phase commit transaction, it must support both a prepare phase and a commit phase for communicating transactions. The prepare phase ensures that all participating nodes that are referenced in a distributed transaction are prepared to commit or to abort the transaction, regardless of intervening failures. The prepare phase is performed when a COMMIT command is issued at the end of a distributed transaction. The commit phase is performed when all transaction participants have responded to the coordinator that they are prepared to commit.

The Oracle server supports two-phase commit transactions. Therefore, any number of Oracle servers can participate in a distributed two-phase commit transaction.

---

**Note:** If the two-phase commit parameter has been set to *YES*, and if the AM400_RID program is active, then the prepare phase occurs automatically when an application COMMIT is executed. No other action is necessary.
Two-Phase Commit Process

There are two phases for two-phase commit processing:

1. prepare phase
2. commit phase

Phase 1: Prepare

When DB2/400 receives a commit request from an application or STRSQL transaction, it asks all participants in the distributed system to enter a prepared state (to promise to commit or rollback the transaction, even if there is a failure). The participating sites reply either that they are prepared to commit or that they are not prepared to commit. When Oracle Access Manager receives such a commit request from DB2/400, it broadcasts an informational message to the AM400_RID program. This communication is one way. The AM400_RID program sends no communication to the Oracle Access Manager.

Phase 2: Commit

If the AS/400 receives a prepared to commit message from all the participating sites, then the AS/400 broadcasts a COMMIT command to all sites. If even one site has replied that it is not prepared to commit, then the AS/400 aborts the transaction and broadcasts a ROLLBACK command. When an Oracle Access Manager site has committed, an information signal is sent to the AM400_RID program to announce the end of the transaction.

Note: Oracle Access Manager operates under AS/400 commitment control, including implied commits (defined on page 7-3). Refer to the appropriate IBM AS/400 documentation for additional information.
In-Doubt Transactions

An in-doubt transaction occurs when a program or operation terminates abnormally before committing or rolling back the transaction. This creates an outstanding transaction, which must be resolved in order to free the involved resources and to ensure transaction consistency.

Resolving In-Doubt Transactions

In-doubt transactions can be resolved automatically by Oracle Access Manager for AS/400. Oracle Access Manager for AS/400 uses a separate task to resolve in-doubt transactions. This task runs in its own subsystem and must be active to enable two-phase commit processing. The task runs the AM400_RID program. On the WRKACTJOB display, the AM400_RIDI job will be seen to be active in the "lib/lib" system (where "lib" is the name of the install library).

The AM400_RID program controls the automatic resolve in-doubt processing of Oracle Access Manager for AS/400. If the AM400_RID program is not running, then you cannot sign onto any Oracle server that is defined in the Change RID (Resolve In Doubt) Options panel when two-phase commit processing is enabled for that server.
As a transaction progresses, the Oracle Access Manager sends status messages to the AM400_RID program. The AM400_RID program does not send anything back to the Oracle Access Manager.
Enabling Two-Phase Commit Processing

To enable two phase commit processing, follow these steps.

Checklist for Enabling Two-Phase Commit

1. Step 1: Change Oracle Access Manager Two-Phase Commit Default
2. Step 2: Change the AM400_RID Options
3. Step 3: Add or Remove AM400_COM Privileged Userids
4. Step 4: Start the Subsystem that Runs the AM400_RID Program
5. Step 5: Start the AM400_COM Program (Optional)
Enabling Two-Phase Commit Processing

Step 1: Change Oracle Access Manager Two-Phase Commit Default

Change the Oracle Access Manager two-phase commit parameter to *YES. For more information about this panel and its settings, refer to Chapter 6, "Configuring Oracle Access Manager".

Step 2: Change the AM400_RID Options

After issuing the CHG2PCPRM command, the first panel displayed confirms which library is to be changed and the five resolve in-doubt (RID) options that can be manipulated. The library name should match the library where you installed the Oracle Access Manager for AS/400.

Figure 7–2 Change RID (Resolve In Doubt) Options Panel

<table>
<thead>
<tr>
<th>Opt</th>
<th>Oracle Server Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMS_SRV1</td>
<td>Connect to VMS1X system</td>
<td></td>
</tr>
<tr>
<td>MVS_920</td>
<td>MVS running Oracle 9.2.0</td>
<td></td>
</tr>
<tr>
<td>MVS_901</td>
<td>MVS running Oracle 9.0.1</td>
<td></td>
</tr>
<tr>
<td>PC_GTW_92</td>
<td>WinNt 4.0 server - 9.2</td>
<td></td>
</tr>
<tr>
<td>H920_130</td>
<td>SUN running Oracle 9.2.0</td>
<td></td>
</tr>
</tbody>
</table>

(C) Copyright Oracle Corporation, 1995
The parameters are described below:

**DATAQUEUE_TIMEOUT** is a tuning parameter requiring a single numeric value (in seconds). With two-phase commit enabled, every Oracle Access Manager for AS/400 user task transmits a status signal to the AM400_RID program for every connect and disconnect and at the beginning and end of every transaction. If a status signal is not received by the AM400_RID program after a specified interval, then a timeout occurs.

The AM400_RID program will check the status of all jobs about which it knows at least one time in every ‘n’ seconds, where ‘n’ is specified by the **DATAQUEUE_TIMEOUT** parameter. If a connection has already been terminated when the timeout occurs, then the AM400_RID program attempts to perform resolve in-doubt tasks.

**CLEANUP_DELAY** is a tuning parameter requiring a single numeric value (in seconds). It specifies the amount of time that the AM400_RID program waits (after it has determined that resolve-in-doubt processing might be needed) before attempting to perform resolve in-doubt tasks.

**TRACE_LEVEL** determines the trace level for the AM400_RID program. You must change this parameter only when requested to do so by Oracle Support Services.

**DEAD_CHECK_COUNTER** is a tuning parameter requiring a single numeric value. It specifies the minimum (or threshold) number of status signals that will be received by the AM400_RID program before it attempts to perform resolve in-doubt tasks.

---

**Note:** The **CLEANUP_DELAY**, **DATAQUEUE_TIMEOUT**, and **DEAD_CHECK_COUNTER** are tuning parameters that are used to govern how often and when the AM400_RID program attempts to perform resolve in-doubt tasks. Do not change these parameters unless directed to do so by Oracle Support Services.
Enabling Two-Phase Commit Processing

Note: The DATAQUEUE_TIMEOUT and DEAD_CHECK_COUNTER together determine how often the AM400_RID program checks to see if an in-doubt transaction might be present. DATAQUEUE_TIMEOUT specifies a value in "time space", and DEAD_CHECK_COUNTER specifies a value in "event-count space".

ORACLE_SERVER defines the name of any Oracle server that is to be accessed by Oracle Access Manager. If a server is not found in this list of servers, then it cannot be accessed by Oracle Access Manager when two-phase commit is enabled.

To add a new Oracle server, select option 1. To change the attributes of an existing Oracle server, select option 2. After selecting either option, the following panel () is displayed:

Figure 7–3 Change or Enter New Server Attributes Panel

<table>
<thead>
<tr>
<th>Change or Enter New Server Attributes</th>
<th>System: HQAS400</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation Library . . . . . . . .</td>
<td>ORAAM130</td>
</tr>
<tr>
<td>Oracle Server Name . . . . . . . . .</td>
<td>VMS_SRV1</td>
</tr>
<tr>
<td>Privileged Oracle Id . . . . . . . .</td>
<td>VMS1Z</td>
</tr>
<tr>
<td>Password for Oracle Id . . . . . . .</td>
<td>zilchnut</td>
</tr>
<tr>
<td>Startup Parameter . . . . . . . . .</td>
<td>startup</td>
</tr>
<tr>
<td>Description . . . . . . . . . . . .</td>
<td>Connect to VMS1X system</td>
</tr>
<tr>
<td>Dead Check Counter . . . . . . . . .</td>
<td>100</td>
</tr>
</tbody>
</table>

F1=Help  F4=Prompt  F9=Retrieve  F12=Cancel

(C) Copyright Oracle Corporation, 1995

The fields in the panel are described as follows:
Enabling Two-Phase Commit Processing

Oracle Server Name  defines the name of the Oracle server that is to be accessed by Oracle Access Manager for AS/400.

Privileged Oracle Id  designates the Oracle userid associated with the defined server name. For additional information, see "Oracle Required Privileges.”

Password for Oracle Id  is the password of the Privileged Oracle Id that is associated with the defined server name. This parameter will be saved in an encrypted form. For additional information, refer to "Oracle Required Privileges” on page 7-13.

Startup Parameter  specifies when the AM400_RID program can attempt to connect to the designated server. If the STARTUP keyword is specified, then the AM400_RID program attempts to connect to the designated server when the AM400_RID program is started. If no keyword is specified, then the AM400_RID program attempts to connect to the designated server when any applications try to connect to the server for the first time.

Description  is a description of the defined oracle_server.

Step 3: Add or Remove AM400_COM Privileged Userids

The options found on the screen in Figure 7-2 can also be dynamically changed using the AM400_COM program. Use of the AM400_COM program to change these options is restricted to the set of AS/400 userids found in the panel in Figure 7-4.

The AM400_COM program allows specified AS/400 user profiles to dynamically add or remove Oracle server parameters, to designate AM400_RID connections to Oracle servers, and to shutdown connections to those Oracle servers. For example, if an Oracle server is not designated as an ORACLE_SERVER parameter in the Change RID (Resolve In Doubt) Options panel, it can be defined dynamically, for the current session, with the AM400_COM program. Refer to "AM400_COM Program” for additional information about the AM400_COM program.

To access the Change RID (Resolve In Doubt) Privileged IDs panel, first issue the CHG2PCPRM command. The first panel displayed confirms which library is to be changed and the five RID options that can be used. Then select [PF10] to display the Change RID (Resolve In Doubt) Privileged IDs panel.
Enabling Two-Phase Commit Processing

Figure 7–4  Change Privileged IDs (Relative to AM400_COM) Panel

Change Privileged IDs (Relative to AM400_COM)  System: HQAS400

Installation Library: ORAAM130

Time options, press Enter.
1=Add  4=Remove

Opt       Privileged ID
QSECOFR
AM400ADMIN

Bottom

F1=Help  F4=Prompt  F9=Retrieve  F10=Switch to OracleServers screen  F12=Cancel

(C) Copyright Oracle Corporation, 1995

The userid in the Privileged ID field has full access to the AM400_COM program.
The Change Privileged IDs (Relative to AM400_COM) panel is distributed with the default of QSECOFR in the Privileged ID field. QSECOFR is the AS/400 security administration userid.

Step 4: Start the Subsystem that Runs the AM400_RID Program

To enable two-phase commit processing, the subsystem that runs the AM400_RID program must be started using:

STRSBS install_lib/install_lib

where install_lib is the library in which Oracle Access Manager is installed.
Step 5: Start the AM400_COM Program (Optional)

You can now start the AM400_COM program to dynamically change the current options for the AM400_RID program. These options are valid only for the current instance of the AM400_RID program. Refer to "AM400_COM Program" on page 7-14 for more information about starting the AM400_COM program.

Oracle Required Privileges

For the AM400_RID program to be able to perform resolve in-doubt processing, it must be able to connect to any Oracle server to which typical users connect with STRSQL or with a SQL-program package. The AM400_RID program connects to the ORACLE_SERVER (see Figure 7-2) using the Oracle userid and password specified for that server on the CHG2PCPRM panel. The privileged userid must have the following:

- read access to the SYS.PENDING_TRANS$ table
- update access to the SYS.PENDING_SESSIONS$ table
- privileges to COMMIT or ROLLBACK any transaction with the FORCE option.
  For more information about the FORCE option, refer to the Oracle7 Server SQL Language Reference Manual.

To grant these privileges, the database administrator can issue these GRANT commands in Oracle using SQL*Plus. This can be done for every server designated as an ORACLE_SERVER parameter in the Change RID (Resolve In Doubt) Options panel.

```
GRANT CREATE SESSIONS, FORCE ANY TRANSACTION TO userid;
GRANT SELECT ON SYS.PENDING_TRANS$ TO userid;
GRANT UPDATE ON SYS.PENDING_SESSIONS$ TO userid;
```

where `userid` is the Oracle userid for the associated server as set or changed in the Change or Enter New Server Attributes panel.
The AM400_COM program is used to communicate dynamically with the AM400_RID program. The AM400_RID program controls the automatic resolve in-doubt processing of Oracle Access Manager for AS/400. The AM400_COM program can query the AM400_RID program for this information:

- the servers to which the AM400_RID program is connected
- the user tasks connected to the various Oracle servers

The AM400_COM program can also be used to dynamically alter the environment of the AM400_RID program during an AS/400 user session. The AM400_COM program can be used to:

- add Oracle servers to the list of servers to which normal AS/400 tasks can connect
- remove Oracle servers from the list of available servers so that no more connections can be made from AS/400 user tasks to those servers
- change the DATAQUEUE_TIMEOUT parameter
- change the CLEANUP_DELAY parameter

To access the AM400_COM program, type the following on the AS/400 command line:

```
CALL AM400_COM PARM(install_lib)
```

where `install_lib` is the library to which Oracle Access Manager is installed.
**AM400_COM Commands**

The AM400_COM program recognizes the following commands:

**STATUS** requests global connection status of the AM400_RID program. Examples:

1. Request the status of all servers to which the AM400_RID program has a connection:
   ```
   STATUS SERVERS
   or
   STATUS
   ```

2. Request the status of the connection between AM400_RID and Oracle server `server_name`:
   ```
   STATUS SERVER server_name
   ```

3. Request the status of what is known about a given job: The `numeric_value` is the six digit AS/400 job number.
   ```
   STATUS JOB numeric_value
   ```

**ADD** adds a server to the list of available servers. For example:
```
ADD SERVER server_name oracle_id oracle_password
```

**REMOVE** removes a server from the list of available servers. For example:
```
REMOVE SERVER server_name
```

**STARTUP** starts a connection between the AM400_RID program and an Oracle server. For example:
```
STARTUP SERVER server_name
```

**SHUTDOWN** shuts down a connection between the AM400_RID program and an Oracle server. For example:
```
SHUTDOWN SERVER server_name [FORCE]
```

**MODIFY** modifies the Oracle userid and the password to be used when AM400_400 attempts to connect to a specific Oracle server. For example:
```
MODIFY SERVER server_name oracle_id oracle_password
```
**QUERY**  queries various items within the AM400_RID environment. Examples:

1. Query whether the AM400_RID program knows about a given server or has a connection to a given server. If the STARTUP parameter is given and the AM400_RID program knows about the server, but there is no connection from the AM400_RID program to that server, then an attempt is made to start the connection. For example:

   ```query_server server_name [STARTUP]```

2. Query the value set for the DATAQUEUE_TIMEOUT parameter. For example:

   ```query_dataqueue_timeout```

3. Query the value set for the CLEANUP_DELAY parameter. For example:

   ```query_cleanup_delay```

4. The value set for the DEAD_CHECK_COUNTER parameter is queried. For example:

   ```query_dead_check_counter```

5. Query the value set for the TRACE_LEVEL parameter. For example:

   ```query_trace_level```

6. Query the value set for the SERVER_STARTUP_TIMEOUT parameter. This value is used by the AS/400 user task when it is attempting to ensure the AM400_RID program has a connection to the same Oracle server to which the user task is attempting to connect. If the user task does not get a positive response from the AM400_RID program within the timeout value set by SERVER_STARTUP_TIMEOUT, then the user task is unable to CONNECT to the Oracle server. For example:

   ```query_server_startup_timeout```

**SET**  sets various items within the AM400_RID environment. The following items can be set:

```set_cleanup_delay numeric_value```
```set_dataq_timeout numeric_value```
```set_dead_check_counter numeric_value```
```set_trace_level numeric_value```
```set_server_startup_timeout numeric_value```
LIST displays various items in the AM400_RID environment. Examples:

1. A list of all the servers known to the AM400_RID program and whether or not there is a connection from the AM400_RID program to that server. For example:

```
LIST AVAILABLE SERVERS
```

2. A list of various items about the shared data space through which the AM400_COM program, the AM400_RID program, and the AS/400 user tasks communicate. Use of this command might be requested by Oracle Support Services. An example:

```
LIST DATASPACE HEADER
```

3. A list of all used entries in the shared data space. Use of this command might be requested by Oracle Support Services. An example:

```
LIST ALL DATASPACE ENTRIES
```

**FORCE** forces a scan of all AS/400 user jobs known to the AM400_RID program and marks any jobs that need to use resolve in-doubt processing. For example:

```
FORCE SCAN
```
To use Oracle Access Manager for AS/400, you need to understand the following topics:

- **Connecting to the Oracle Server** on page 8-2
- **Connection Management** on page 8-4
- **Describing Tables** on page 8-6
- **Stored Procedures** on page 8-6
- **Conversion of Oracle Datatypes to DB2/400 Datatypes** on page 8-9
- **Message and Error Code Processing** on page 8-10
Connecting to the Oracle Server

When the network definition is completed, you can connect to Oracle from interactive SQL or your precompiled AS/400 program.

Connecting through the STRSQL (Interactive SQL) Commands

From the AS/400 command line, enter:

```
STRSQL
```

Ensure that your interactive STRSQL session uses the SQL syntax `library.file`. To ensure this, use STRSQL NAMING("SQL"). Oracle Access Manager does not support AS/400 `library/file` syntax.

From the interactive SQL command line, enter the IBM DB2/400 CONNECT TO command to connect to the Oracle server:

```
CONNECT TO remote_oracle USER userid USING 'pw'
```

where:

```
remote_oracle  is your RDB directory entry that has a matching TNSNAMES entry.
userid        is a valid Oracle userid.
pw            is the password for userid.
```

If no userid is specified, Oracle Access Manager uses the default userid and password. The default userid and password are SCOTT/TIGER. You can change these with the CHGSQDLDFLT command. Refer to Chapter 6, "Configuring Oracle Access Manager" for more information.

After you are connected, you can issue SQL statements to the remote Oracle server.

---

**Note:** Interactive SQL sends a CREATE TABLE `xxx` command to Oracle Access Manager for each connection. Consequently, the job log contains the following message, which can be ignored:

```
ORA-00901: invalid CREATE command
```
Connecting from an AS/400 Precompiled Application

Three sample programs are distributed with Oracle Access Manager. They are SAMP1C, SAMP1CBL, and SAMP1RPG. Use the CRTORAPKG command to precompile, compile, and bind these programs. Refer to Chapter 6, "Configuring Oracle Access Manager" for more information about the CRTORAPKG command.

After using the CRTORAPKG command, you can execute the sample programs using the CALL command from the AS/400 command line. For example:

```
CALL SAMP1RPG
```

Each program explicitly connects to a server called GENERIC using an Oracle user name of SCOTT and a password of TIGER. You can edit these sample programs if you want to test them against a different server or use a different username or password.

After execution, you can issue a SELECT statement for the DEPT table through interactive SQL (STRSQL) to verify that the rows were inserted successfully.

These sample programs (included in Appendix A, "Sample Programs") are in the Oracle Access Manager installation library in the following files:

<table>
<thead>
<tr>
<th>Program</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMP1C</td>
<td>QSQLCSRC</td>
</tr>
<tr>
<td>SAMP1CBL</td>
<td>QCBLLESRC</td>
</tr>
<tr>
<td>SAMP1RPG</td>
<td>QRPGLESRC</td>
</tr>
</tbody>
</table>

Table 8–1 Sample Program Files
Connection Management

Connection management consists of connecting to and disconnecting from one or more servers, and using COMMIT and ROLLBACK to coordinate transactions.

Oracle Access Manager connects to an Oracle server using the DB2/400 CONNECT syntax. The Oracle CONNECT syntax is not supported by Oracle Access Manager.

An application can connect to an Oracle server explicitly or implicitly. Explicit connection occurs when the DB2/400 CONNECT TO statement is issued. Implicit connection occurs when the first SQL statement in an application program is not a CONNECT TO statement. If this occurs, the connection is made to the server specified on the RDB parameter in the IBM CRTSQL.xxx command.

After a connection is established, the server becomes active, and all SQL statements are processed on that server.

Oracle Access Manager imposes no restrictions on the number of connections that can be opened at one time. In a single session, you can connect to multiple Oracle servers and multiple IBM DRDA application servers. When multiple servers are connected, you can switch between servers using the DB2/400 SET CONNECTION command.

For example:

```
CONNECT TO LOCAL
INSERT INTO DEPT VALUES (50, 'LOCAL', 'ROW 1')
CONNECT TO ORACLE1 USER SCOTT USING 'TIGER'
INSERT INTO DEPT VALUES (50, 'ORACLE1', 'ROW 1')
CONNECT TO ORACLE2 USER JOHN USING 'SMITH'
INSERT INTO DEPT VALUES (50, 'ORACLE2', 'ROW 1')
SET CONNECTION ORACLE1
INSERT INTO DEPT VALUES (50, 'ORACLE1', 'ROW 2')
SET CONNECTION LOCAL
INSERT INTO DEPT VALUES (50, 'LOCAL', 'ROW 2')
```

If multiple connections are open and a COMMIT or ROLLBACK statement is issued, all database servers are requested to COMMIT or ROLLBACK.

You cannot have multiple connections open to the same RDB name within a single session. For example, the following would be invalid:

```
CONNECT TO ORACLE1 USER X USING 'PWDX'
CONNECT TO ORACLE1 USER Y USING 'PWDY'
```
All applications, including interactive SQL (STRSQL), keep all connections open until you do one of the following:

- sign off from your AS/400 system
- issue a RELEASE command for each open connection followed by a COMMIT statement
- issue a RELEASE ALL command followed by a COMMIT statement, thereby ending all connections

Exiting from Interactive SQL (STRSQL) does not affect the status of any connections that were obtained (and are still active) within Interactive SQL. All Oracle server connections that are left open are available on subsequent invocations of Interactive SQL within the same job. However, exiting from a program that uses embedded SQL in the EXEC SQL format will cause any Oracle server connection that was created within that program to be released.

---

**Note:** The DB2/400 RELEASE command followed by a ROLLBACK statement does not end the connection.

---

Oracle Access Manager connects to an Oracle server using CONNECT type 5 and establishes a protected conversation, as documented in the IBM DB2/400 SQL Programming Guide.

---

**Note:** Because Oracle Access Manager runs as a protected conversation, an attempt to use the DISCONNECT command results in the following message for both TCP/IP and APPC/LU6.2 connections:

SQL0858 "Cannot disconnect relational database <server_name> due to LU6.2 protected conversation"
Describing Tables

Interactive SQL (STRSQL) does not allow the DESCRIBE TABLE to be issued interactively. However, column and datatype information can be obtained through interactive SQL (STRSQL) by using the prompt option (PF4), when issuing a SELECT from the command line. This information can be used by application programmers to understand how Oracle datatypes are mapped to AS/400 datatypes.

Stored Procedures

On the AS/400, you call a stored procedure by using the IBM DB2/400 CALL statement. The CALL statement can be executed interactively from interactive SQL or embedded in an application within the appropriate "EXEC-SQL END-EXEC" delimiters for the host language.

Oracle Access Manager allows access from AS/400 applications to Oracle stored procedures or packages by providing mapping from the IBM CALL statement to an Oracle stored procedure.

The results of a CALL command can be committed or rolled back. Oracle Access Manager always assumes that an invocation of stored procedures has changed something on the Oracle server.

Advantages of Stored Procedures

As defined on the AS/400, a stored procedure is a programming construct that can be called from within the SQL environment to perform a set of operations. The operations can include host language statements, SQL statements, or logic. When connected to an Oracle server, the host language statements are packaged within a PL/SQL stored procedure on the Oracle server.

Stored procedures in PL/SQL provide the same benefits as procedures in a host language such as C or COBOL. A PL/SQL procedure that is stored in the Oracle database can be called from several programs. The use of stored procedures can also enhance the performance of a distributed application. For example, assume that you want to execute several SQL statements at an Oracle server using interactive SQL. If stored procedures are not used, then the AS/400 must send a separate request to the Oracle server for each SQL statement. If the same SQL statements are stored in a stored procedure at the Oracle server, then a single CALL statement on the AS/400 executes the entire block of SQL statements.
Stored Procedure Parameters

Parameters that are used only to pass data to a stored procedure are called IN parameters. Constant values used as parameters can be used only as IN parameters. Parameters that are used only to return data from a stored procedure are called OUT parameters. Parameters that can be used in both directions are called IN OUT parameters. You must use a host variable to have data returned from an Oracle stored procedure.

Calls to a stored procedure on the Oracle server from within an AS/400 application can use IN, OUT, and IN OUT parameters. This enables parameters to be passed to the stored procedure and data to be returned. Calls to a stored procedure from within the interactive SQL environment can only use IN parameters. You can pass constant parameters to the stored procedure on the Oracle server, but no data can be returned.

Declaring Procedures

When calling an Oracle stored procedure, you must use the DB2/400 DECLARE PROCEDURE statement. This statement is used to specify the IN, OUT, or IN OUT capability and the datatype of each parameter. For example:

```sql
DECLARE Sample PROCEDURE (IN :arg1 INTEGER, IN OUT :arg2 CHAR(20),
  OUT :arg3 SMALLINT) (LANGUAGE C SIMPLE CALL)
```

**Caution:** If a DECLARE PROCEDURE statement is not found for a CALL to a stored procedure, then each parameter defaults to IN OUT, and the datatype for each parameter defaults to the datatype of the associated host variable.

Ensure that the datatypes specified in the DECLARE PROCEDURE statement are compatible with those found in the stored procedure specification of the Oracle stored procedure. In addition, the input or output modes of each parameter (specified by IN, OUT, or IN OUT in the DECLARE PROCEDURE statement) and the number of parameters must match those of the Oracle stored procedure.

The following example is an extract from an AS/400 C program that calls the Oracle stored procedure GETRESULTS. The datatypes of parameters match, the input and output modes match, and the number of parameters match (nine parameters are in the DECLARE PROCEDURE statement on the AS/400, in the CALL to the procedure, and in the declaration of the Oracle stored procedure).
EXEC SQL BEGIN DECLARE SECTION;
char p1[10], p3[10], p5[10];
struct {short len; char value[20];} p1a; /* A VARCHAR */
struct {short len; char value[20];} p3a; /* A VARCHAR */
struct {short len; char value[20];} p5a; /* A VARCHAR */
long p2, p4, p6;
EXEC SQL END DECLARE SECTION;

EXEC SQL DECLARE GETRESULTS PROCEDURE
(IN :par1 CHAR(10), IN :par1a VARCHAR(20), IN :par2 INTEGER,
INOUT :par3 CHAR(10), INOUT :par3a VARCHAR(20),
INOUT :par4 INTEGER,
OUT :par5 CHAR(10), OUT :par5a VARCHAR(20),
OUT :par6 INTEGER);
EXEC SQL CALL GETRESULTS(:p1, :p1a, :p2, :p3, :p3a, :p4, :p5, :p5a,
:par6);

The following is the file "getresults.sql" on the ORACLE server.
create or replace procedure GETRESULTS
(IN :par1 CHAR(10), IN :par1a VARCHAR2, IN :par2 INTEGER,
INOUT :par3 CHAR(10), INOUT :par3a VARCHAR2,
INOUT :par4 INTEGER,
OUT :par5 CHAR(10), OUT :par5a VARCHAR2,
OUT :par6 INTEGER)
LANGUAGE C SIMPLE CALL;
EXEC SQL CALL GETRESULTS(:p1, :p1a, :p2, :p3, :p3a, :p4, :p5, :p5a,
:par6);
Conversion of Oracle Datatypes to DB2/400 Datatypes

Oracle Access Manager for AS/400 converts Oracle datatypes to DB2/400 datatypes to allow application host variables to store and represent the Oracle data they receive into a format native to the AS/400 environment.

The following table describes these datatype conversions. In the table, "p" refers to precision, and "s" refers to scale. On DB2/400, VARCHAR can contain 32 740 characters at most.

<table>
<thead>
<tr>
<th>Oracle Server Datatype</th>
<th>DB2/400 Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR</td>
<td>CHAR</td>
</tr>
<tr>
<td>DATE</td>
<td>DATE</td>
</tr>
<tr>
<td>LONG</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>LONG RAW</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>NUMBER(p,s)</td>
<td>SMALLINT 1&lt;=p&lt;=4, s=0 or s&lt;&gt;0, p&lt;=31</td>
</tr>
<tr>
<td></td>
<td>INTEGER 5&lt;=p&lt;=9, s=0</td>
</tr>
<tr>
<td></td>
<td>DECIMAL 10&lt;=p&lt;=31, s=0</td>
</tr>
<tr>
<td></td>
<td>or s&lt;&gt;0, p&lt;=31</td>
</tr>
<tr>
<td></td>
<td>FLOAT anything else</td>
</tr>
<tr>
<td>RAW</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>VARCHAR2</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>ROWID</td>
<td>VARCHAR</td>
</tr>
</tbody>
</table>

LONG datatypes or combinations of LONG and LONG RAW datatypes are limited by the DB2/400 restriction that the entire length of data returned for a row must be less than 32 767 bytes. This restricts LONG and LONG RAW datatypes to a maximum of 32 740 bytes for all the LONG and LONG RAWs returned in a row.
Message and Error Code Processing

Oracle Access Manager communicates all error conditions to the AS/400 application program through the SQLCA. Because Oracle error codes differ from DB2/400 error codes, a one-to-one mapping is not possible for all error conditions. Therefore, Oracle Access Manager uses the SQLSTATE field of the SQLCA as a mechanism to communicate Oracle errors to DB2/400.

When a SQLSTATE other than 00000 is returned from Oracle Access Manager, DB2/400 attempts to map the SQLSTATE to a DB2/400 SQLCODE and message. If it is unable to recognize the SQLSTATE, DB2/400 issues the common message for that class code. For a list of SQLSTATE class codes, refer to the IBM DB2/400 SQL programming guide.

In some situations, Oracle Access Manager translates a SQLSTATE value into a valid DB2/400 SQLCODE. This is called a mapped error code. When mapping occurs, a valid DB2/400 SQLCODE is returned, and all applicable substitution tokens are placed in the SQLERRMC field of the SQLCA. This enables you to see valid text in the replacement variables of the DB2/400 SQL error message. Oracle Access Manager currently maps these error codes as indicated in the following table:

<table>
<thead>
<tr>
<th>Oracle Error Code</th>
<th>DB2/400 SQLSTATE</th>
<th>DB2/400 SQLCODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORA-0904 invalid column name</td>
<td>42703</td>
<td>-206</td>
</tr>
<tr>
<td>ORA-0932 inconsistent datatypes</td>
<td>42884</td>
<td>-440</td>
</tr>
<tr>
<td>ORA-0936 missing expression</td>
<td>42601</td>
<td>-104</td>
</tr>
<tr>
<td>ORA-00901 invalid create command - 104</td>
<td>42601</td>
<td>-104</td>
</tr>
<tr>
<td>ORA-00911 invalid character</td>
<td>42601</td>
<td>-7</td>
</tr>
<tr>
<td>ORA-00923 from keyword not found - 104</td>
<td>42601</td>
<td>-104</td>
</tr>
<tr>
<td>ORA-00942 table or view does not exist</td>
<td>24704</td>
<td>-204</td>
</tr>
<tr>
<td>ORA-01095 DML statement process 0 rows</td>
<td>02000</td>
<td>+100</td>
</tr>
<tr>
<td>ORA-01400 cannot insert NULL into string</td>
<td>23502</td>
<td>-407</td>
</tr>
<tr>
<td>ORA-01401 inserted value too large for column</td>
<td>22001</td>
<td>-404</td>
</tr>
<tr>
<td>ORA-01403 no data found</td>
<td>02000</td>
<td>+100</td>
</tr>
<tr>
<td>ORA-01405 fetched column value is NULL</td>
<td>22002</td>
<td>-305</td>
</tr>
</tbody>
</table>
If DB2/400 does not recognize a SQLSTATE, then the following message is issued:

---

**Table 8–3  Error Code Mapping (Cont.)**

<table>
<thead>
<tr>
<th>Oracle Error Code</th>
<th>DB2/400 SQLSTATE</th>
<th>DB2/400 SQLCODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORA-01406 fetched column value was truncated</td>
<td>22001</td>
<td>-404</td>
</tr>
<tr>
<td>ORA-01422 exact fetch returns more than requested number of rows</td>
<td>21000</td>
<td>-811</td>
</tr>
<tr>
<td>ORA-01424 missing or illegal character following the escape character</td>
<td>22025</td>
<td>-130</td>
</tr>
<tr>
<td>ORA-01425 escape character must be character string of length 1</td>
<td>22019</td>
<td>-130</td>
</tr>
<tr>
<td>ORA-01427 single row subquery returns more than one row</td>
<td>21000</td>
<td>-811</td>
</tr>
<tr>
<td>ORA-01438 value larger than specified precision for this column</td>
<td>22001</td>
<td>-404</td>
</tr>
<tr>
<td>ORA-01455 converting column overflows integer datatype</td>
<td>22003</td>
<td>-304</td>
</tr>
<tr>
<td>ORA-01457 converting column overflows decimal datatype</td>
<td>22003</td>
<td>-304</td>
</tr>
<tr>
<td>ORA-01476 divisor is equal to zero</td>
<td>22012</td>
<td>-802</td>
</tr>
<tr>
<td>ORA-01479 last character in buffer is not NULL</td>
<td>22024</td>
<td>-302</td>
</tr>
<tr>
<td>ORA-01480 trailing null missing from STR bind variable</td>
<td>22024</td>
<td>-302</td>
</tr>
<tr>
<td>ORA-01488 invalid nibble or byte in input data</td>
<td>22023</td>
<td>-302</td>
</tr>
<tr>
<td>ORA-12154 TNS: could not resolve service name</td>
<td>08003</td>
<td>-30000</td>
</tr>
<tr>
<td>ORA-12533 TNS: illegal ADDRESS parameters</td>
<td>08003</td>
<td>-30000</td>
</tr>
<tr>
<td>ORA-12541 TNS: no listener</td>
<td>08003</td>
<td>-30000</td>
</tr>
<tr>
<td>ORA-12154 TNS: could not resolve service name</td>
<td>08003</td>
<td>-30000</td>
</tr>
<tr>
<td>ORA-12560 TNS: protocol adapter error</td>
<td>08003</td>
<td>-30000</td>
</tr>
</tbody>
</table>
SQL7940 SQLSTATE class unknown, SQLSTATE xxxxx...

Examine the job log for the relevant Oracle error that caused the SQLSTATE xxxxx to be issued.

If Oracle Access Manager is unable to map an error code, then it returns a SQLCODE of -969 and a SQLSTATE which applies to the class code of the error. For example, all syntax errors have a SQLSTATE class code of 42. Therefore, if Oracle Access Manager receives a syntax parsing error from the Oracle server (which it cannot map), then it returns SQLCODE of -969 and SQLSTATE 42000.

If you are unable to determine the root cause of the error based on the SQLCODE and SQLSTATE returned to the application, then check the job log for more details. All Oracle errors and message text are displayed in the job log at the time that the error occurs.
This appendix includes sample Oracle Access Manager programs written in C, COBOL, and RPG. The following sample programs are included:

- Sample Oracle Access Manager C Program on page A-2
- Sample Oracle Access Manager COBOL Program on page A-5
- Sample Oracle Access Manager RPG Program on page A-7
Sample Oracle Access Manager C Program

#include <string.h>
#include <stdlib.h>
#include <stdio.h>
/*---------------------------------------------------------------*/
/*                                                               */
/*  Program Name:  SAMP1C                                        */
/*                                                               */
/*  Function: Insert a sample row into the DEPT table            */
/*                                                               */
/*  Author:   Oracle Corporation                                 */
/*  Date:     8/25/95                                            */
/*                                                               */
/*---------------------------------------------------------------*/
#ifdef TRUE
#  undef TRUE
#endif
#define TRUE 1
EXEC SQL INCLUDE sqlca;
struct sqlca *sca;
main()
{
EXEC SQL BEGIN DECLARE SECTION;
    char server??(10??);
    char username??(10??);
    char password??(10??);
    long int deptno;
    char dname??(14??);
    char loc??(13??);
    char sql_stmt??(80??);
EXEC SQL END DECLARE SECTION;
long int rc;
EXEC SQL WHENEVER SQLERROR GOTO big_error;
EXEC SQL WHENEVER SQLWARNING GOTO big_warn;
printf("Entering test SAMP1C...\n");/*
strcpy(server, "GENERIC");
strcpy(username, "SCOTT");

A-2   Oracle Access Manager for AS/400 Installation and User's Guide
strcpy(password, "TIGER");
EXEC SQL CONNECT TO :server USER :username USING :password;
*/
printf("After implicit connect.\n");

strcpy(sql_stmt,
'INSERT INTO DEPT VALUES (?, ?, ?)');
EXEC SQL PREPARE S FROM :sql_stmt;
printf("After prepare.\n");

deptno = 88;
strcpy(dname, "SAMP1 C");
strcpy(loc, "Success");
EXEC SQL EXECUTE S USING :deptno, :dname, :loc;
printf("After execute.\n");
EXEC SQL RELEASE CURRENT;
printf("After release current.\n");
EXEC SQL COMMIT;
printf("After commit.\n");
printf("Exiting test SAMP1C...\n");
exit(0);

big_warn:
EXEC SQL WHENEVER SQLWARNING CONTINUE;
printf("Big warn\n");
big_error:
EXEC SQL WHENEVER SQLERROR CONTINUE;
printf("Big error!: SQLCA...
");
printf("sqlcaid =%0.8s.\n",sqlca.sqlcaid);
printf("sqlcabc =%d.\n",sqlca.sqlcabc);
printf("sqlcode =%d.\n",sqlca.sqlcode);
printf("sqlerrml =%d.\n",sqlca.sqlerrml);
printf("sqlerrmc =%0.70s.\n",sqlca.sqlerrmc);
printf("sqlerrp =%0.8s.\n",sqlca.sqlerrp);
printf("sqlerrd =%d %d %d %d %d %d
",sqlca.sqlerrd??(0??),
                               sqlca.sqlerrd??(1??),sqlca.sqlerrd??(2??),
                               sqlca.sqlerrd??(3??),sqlca.sqlerrd??(4??),
                               sqlca.sqlerrd??(5??));
printf("sqlwarn =%0.11s.\n",sqlca.sqlwarn);
printf("sqlstate =%0.5s.\n",sqlca.sqlstate);
Sample Oracle Access Manager C Program

    exit(1);
    not_found:
        exit(0);
    }

A-4  Oracle Access Manager for AS/400 Installation and User's Guide
IDENTIFICATION DIVISION.
PROGRAM-ID. SAMP1CBL.
  AUTHOR. Oracle.
  INSTALLATION. Oracle Corporation.
  DATE-Compiled.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
  SOURCE-COMPUTER. IBM-AS400.
  OBJECT-COMPUTER. IBM-AS400.
DATA DIVISION.
WORKING-STORAGE SECTION.
  EXEC SQL BEGIN DECLARE SECTION END-EXEC.
  77 SERVER-NAME PIC X(10).
  77 USER-NAME PIC X(10).
  77 USER-PASSWORD PIC X(10).
* Anything going into a NUMERIC, DECIMAL, SMALLINT, INTEGER
* column MUST have a sign.
* DEPTNO in the following is a four-byte zoned decimal number
* with the sign in the 'zone' portion of the right hand character
  77 DEPNO PIC S9(4).
  77 DEPTNAME PIC X(14).
  77 MISC PIC X(13).
  77 SQL-STATEMENT PIC X(100).
  EXEC SQL END DECLARE SECTION END-EXEC.
  EXEC SQL INCLUDE SQLCA END-EXEC.
  77 CONV1 PIC S999999999 SIGN IS LEADING SEPARATE.
  77 CONV2 PIC S999999999 SIGN IS LEADING SEPARATE.
  77 CONV3 PIC S999999999 SIGN IS LEADING SEPARATE.
PROCEDURE DIVISION.
P1NTVLE-INIT.
  MOVE "GENERIC" TO SERVER-NAME.
  MOVE "SCOTT" TO USER-NAME.
  MOVE "TIGER" TO USER-PASSWORD.
  MOVE SPACES TO SQLERRMC.
  EXEC SQL WHENEVER SQLERROR GO TO BIG-ERROR END-EXEC.
  EXEC SQL WHENEVER SQLWARNING GO TO BIG-WARNING END-EXEC.
P1NTVLE-CONNECT.
  DISPLAY "Connected implicitly...".
P1NTVLE-PREPARE.
  MOVE 'Insert into DEPT VALUES(?,?,?)' to SQL-STATEMENT.
  DISPLAY ' ' SQL-STATEMENT.
  EXEC SQL PREPARE S FROM :SQL-STATEMENT END-EXEC.
DISPLAY "Prepared ...".

PINTVLE-EXECUTE.
  MOVE 88 TO DEPNO.
  MOVE "SAMP1 CBL" TO DEPTNAME.
  MOVE "Success" to MISC.
  EXEC SQL EXECUTE S USING :DEPNO, :DEPTNAME, :MISC END-EXEC.
  DISPLAY "Executed...".
PILE-RELEASE.
  EXEC SQL RELEASE CURRENT END-EXEC.
  DISPLAY "Released current...".
PILE-COMMIT.
  EXEC SQL COMMIT END-EXEC.
  DISPLAY "Committing ...".
PILE-END.
  DISPLAY "Exiting ...".
  STOP RUN.
BIG-ERROR.
  Display "SQL Error ...".
  PERFORM DUMP-SQLCA.
  GO TO PILE-END.
BIG-WARNING.
  Display "SQL Warning ...".
  PERFORM DUMP-SQLCA.
  GO TO PILE-END.
DUMP-SQLCA.
  *Move binary fields to displayable fields for DISPLAY stmt.
    MOVE SQLCODE TO CONV1. MOVE SQLSTATE TO CONV2.
    DISPLAY " SQLCODE=" CONV1 ", SQLSTATE=" CONV2.
    DISPLAY " SQLERRMC on following line ...".
    DISPLAY " " SQLERRMC.
    DISPLAY " SQLERRP=" SQLERRP.
    MOVE SQLERRD(1) TO CONV1.
    MOVE SQLERRD(2) TO CONV2.
    MOVE SQLERRD(3) TO CONV3
    DISPLAY "SQLERRD(1)..SQLERRD(3)=" CONV1 " " CONV2 " " CONV3.
    MOVE SQLERRD(4) TO CONV1.
    MOVE SQLERRD(5) TO CONV2.
    MOVE SQLERRD(6) TO CONV3.
    DISPLAY "SQLERRD(4)..SQLERRD(6)=" CONV1 " " CONV2 " " CONV3.
    DISPLAY "SQLWARN=" SQLWARN.
Sample Oracle Access Manager RPG Program

H
F* FILE DECLARATION FOR QPRINT
FQPRINT O F 132 PRINTER
DVARS DS
DDEPTNO 1 4b 0
DHOSTDN 5 8b 0
DSERVER 9 18
DUSERNM 19 28
DPASSWD 29 38
DSTMSQL 39 118
DSTR 119 198 DIM(80)
DSTMT1 C 'Insert into DEPT - VALUES(88,'SAMPI RPG','Success')'
C MOVE *BLANKS STR
C CLEAR SQLCA
C*PUT OUT "ENTERING TEST P1..." MESSAGE
C MOVEL(P) 'GENERIC' SERVER
C MOVEL(P) 'SCOTT' USERNM
C MOVEL(P) 'TIGER' PASSWD
C MOVEL(P) STMT1 STMSQL
C EXCEPTION NTRP1
C EXCEPTION NTRP2
C/EXEC SQL
C+ WHENEVER SQLERROR GOTO BIGERR
C/END-EXEC
C/EXEC SQL
C+ WHENEVER SQLWARNING GOTO BIGWRN
C/END-EXEC
C*EXEC SQL
C* CONNECT TO :SERVER USER :USERNM USING :PASSWD
C* CONNECT
C*END-EXEC
C* PUT OUT "CONNECTED IMPLICITLY" MESSAGE...
C EXCEPTION CNCTD
C/EXEC SQL PREPARE S FROM :STMSQL
C/END-EXEC
C* PUT OUT "AFTER PREPARE" MESSAGE
C EXCEPTION PRPAF
C* NOW DO THE EXECUTE OF THE SQL STATEMENT AND PUT OUT MESSAGE
C/EXEC SQL EXECUTE S
C/END-EXEC
C EXCEPTION XEQAF
C* NOW DO THE RELEASE OF THE SERVER AND PUT OUT MESSAGE
C/EXEC SQL RELEASE CURRENT
C/END-EXEC
C                  EXCEPT    RLSAF
C*  NOW DO THE COMMIT AND PUT OUT MESSAGE ... 
C/EXEC SQL COMMIT
C/END-EXEC
C                  EXCEPT    CMTAF
C*  AND NOW WE ARE GOING TO EXIT ... 
C                  EXCEPT    XITNW
C                  GOTO      FINISH
C*
C               BIGWRN    TAG
C/EXEC SQL WHENEVER SQLWARNING CONTINUE
C/END-EXEC
C                  EXCEPT    WRNMSG
C                  EXSR      PSQLCA
C                  GOTO      FINISH
C*
C               BIGERR   TAG
C/EXEC SQL WHENEVER SQLERROR CONTINUE
C/END-EXEC
C                  EXCEPT    ERMSG
C                  EXSR      PSQLCA
C                  GOTO      FINISH
C*
C               FINISH    TAG
C                  SETON                                        LR
C*
CSR   PSQLCA        BEGSR
C                  EXCEPT    SQLCA1
C                  EXCEPT    SQLCA2
C                  EXCEPT    SQLCA3
C                  EXCEPT    SQLCA4
C                  EXCEPT    SQLCA5
C                  EXCEPT    SQLCA6
CSR                 ENDSR
C*
OQPRINT    E            NTRP1       1
O          E            NTRP2       1
O          E            CNCTD       1
O          E            PRPAP       1

'Entering SAMP1RPG...'
'STMSQL='
'CONNECTED IMPLICITLY'
Sample Oracle Access Manager RPG Program

O  E  XEQAF  1  'AFTER PREPARE'
O  E  RLSAF  1  'AFTER EXECUTE'
O  E  CMTAF  1  'AFTER RELEASE CURRENT'
O  E  XITNW  1  'AFTER COMMIT'
O  E  WRNMSG  1  'EXITING SAMP1 RPG'
O  E  ERMXG  1  'BIG WARN:SQLCA…'
O  E  SQLCA1  1  'BIG ERROR:SQLCA…'
O  E  SQLCA2  1  8 'SQLAID='
O  E  SQLCA3  1  8 'SQLRM='
O  E  SQLCA4  1  16 'SQLER1..SQLER3='
O  E  SQLCA5  1  16 'SQLER4..SQLER6='
O  E  SQLCA6  1  8 'SQLWRN='
O  E  SQLWRN  +0  +0 ', SQLSTT='
O  E  SQLSTT  +0
This appendix documents the National Language Support (NLS) for the Oracle Access Manager. The following topics are included:

- Oracle Access Manager NLS Support on page B-2
- Oracle Character Set Names on page B-3
- Oracle NLS_LANG Specifications on page B-5
Oracle Access Manager NLS Support

The Oracle Access Manager is designed to support a multitude of single-byte character sets. Source files containing SQL packages can be written in any of the coded character set identifiers (CCSID) that are listed in Table B–1.
Oracle Character Set Names

In the table are the Oracle character set names that can be included as the third part of an Oracle NLS language specification. The Oracle language specification can be explicitly set for an invocation of the Oracle Access Manager by having access to a data area named NLS_LANG.

**Table B–1 Oracle Character Set Names**

<table>
<thead>
<tr>
<th>CCSID</th>
<th>Oracle Character Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>WE8EBCDIC37</td>
</tr>
<tr>
<td>273</td>
<td>D8EBCDIC273</td>
</tr>
<tr>
<td>277</td>
<td>DKEBCDIC277</td>
</tr>
<tr>
<td>278</td>
<td>S8EBCDIC278</td>
</tr>
<tr>
<td>280</td>
<td>I8EBCDIC280</td>
</tr>
<tr>
<td>284</td>
<td>WE8EBCDIC284</td>
</tr>
<tr>
<td>285</td>
<td>WE8EBCDIC285</td>
</tr>
<tr>
<td>297</td>
<td>F8EBCDIC297</td>
</tr>
<tr>
<td>500</td>
<td>WE8EBCDIC500</td>
</tr>
<tr>
<td>838</td>
<td>TH8TISEBCDIC</td>
</tr>
<tr>
<td>870</td>
<td>EE8EBCDIC870</td>
</tr>
<tr>
<td>875</td>
<td>EL8EBCDIC875</td>
</tr>
<tr>
<td>1140</td>
<td>WE8EBCDIC1140</td>
</tr>
<tr>
<td>1141</td>
<td>D8EBCDIC1141</td>
</tr>
<tr>
<td>1142</td>
<td>DK8EBCDIC1142</td>
</tr>
<tr>
<td>1143</td>
<td>S8EBCDIC1143</td>
</tr>
<tr>
<td>1144</td>
<td>I8EBCDIC1144</td>
</tr>
<tr>
<td>1145</td>
<td>WE8EBCDIC1145</td>
</tr>
<tr>
<td>1146</td>
<td>WE8EBCDIC1146</td>
</tr>
<tr>
<td>1147</td>
<td>F8EBCDIC1147</td>
</tr>
<tr>
<td>1148</td>
<td>WE8EBCDIC1148</td>
</tr>
</tbody>
</table>
The NLS_LANG data area that is shipped with the Oracle Access Manager defaults to AMERICAN_AMERICA.WE8EBCDIC37. All data emanating from a SQL package is data to be transmitted from the AS/400 to the Oracle database. This data is translated from the character set that is associated with the CCSID of the package to the character set that is denoted explicitly by NLS_LANG or by the default for NLS_LANG. Data that is returned from the Oracle server to the AS/400 is translated by the IBM Application Requester Driver API.
Oracle NLS_LANG Specifications

An Oracle NLS_LANG specification includes three parts:

- a language
- a territory
- a character set

For example:

```
language_territory.character_set
AMERICAN_AMERICA.D8EBCDIC1141
```

For the Oracle Access Manager, the language is AMERICAN, and the territory is AMERICA. The error messages are currently in American English only. The third part is the Oracle character set name from Table E-1. The NLS_LANG values in Table E-2 are currently supported:

**Table B–2 Oracle NLS_LANG Values**

<table>
<thead>
<tr>
<th>Language</th>
<th>Setting for language_territory.charset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austrian/German</td>
<td>AMERICAN_AMERICA.D8EBCDIC273 and AMERICAN_AMERICA.D8EBCDIC1141</td>
</tr>
<tr>
<td>Danish/Norwegian</td>
<td>AMERICAN_AMERICA.D8EBCDIC277 and AMERICAN_AMERICA.D8EBCDIC1142</td>
</tr>
<tr>
<td>Eastern European</td>
<td>AMERICAN_AMERICA.EE8EBCDIC870</td>
</tr>
<tr>
<td>Finnish/Swedish</td>
<td>AMERICAN_AMERICA.S8EBCDIC278 and AMERICAN_AMERICA.S8EBCDIC1143</td>
</tr>
<tr>
<td>French</td>
<td>AMERICAN_AMERICA.F8EBCDIC297 and AMERICAN_AMERICA.F8EBCDIC1147</td>
</tr>
<tr>
<td>Greek</td>
<td>AMERICAN_AMERICA.EL8EBCDIC875</td>
</tr>
<tr>
<td>Italian</td>
<td>AMERICAN_AMERICA.8EBCDIC280 and AMERICAN_AMERICA.8EBCDIC1144</td>
</tr>
<tr>
<td>Thai</td>
<td>AMERICAN_AMERICA.TH8T1EBCDIC</td>
</tr>
<tr>
<td>Western European</td>
<td>AMERICAN_AMERICA.WE8EBCDIC37 and AMERICAN_AMERICA.WE8EBCDIC1140</td>
</tr>
<tr>
<td>Western European</td>
<td>AMERICAN_AMERICA.WE8EBCDIC500 and AMERICAN_AMERICA.WE8EBCDIC1148</td>
</tr>
</tbody>
</table>
Each user can have their own NLS_LANG data area. To change the NLS_LANG data area, change the Oracle Access Manager NLS language parameter by using the CHGSQLDFLT command as described in Chapter 6, "Configuring Oracle Access Manager".
Activation groups, 1-10
ACTGRP, 1-10
commands
CRTSQLxxx, 1-10
STRSQL, 1-10
ADD command, 7-15
ADDLIBLE command, 4-7, 6-2
ADDRDBDIRE command, 5-3, 5-4, 5-8
AM400_COM program
 defined, 7-2
commands, 7-15
 ADD, 7-15
 FORCE, 7-17
 LIST, 7-17
 MODIFY, 7-15
 QUERY, 7-16
 REMOVE, 7-15
 SET, 7-16
 SHUTDOWN, 7-15
 STARTUP, 7-15
 STATUS, 7-15
 Privileged ID field, 7-12
privileged userids, 7-11
with LIST command, 7-17
AM400_RID
with LIST command, 7-17
AM400_RID program
 defined, 7-2
altering the environment, 7-14
starting, 7-12
startup parameter, 7-11
with DATAQUEUE_TIMEOUT parameter, 7-9
with resolve-in-doubt transactions, 7-5
with two-phase commit, 7-4
Architecture, 1-3
architecture components, 1-5
ARD interface, B-4
AUTOCREATE CONTROLLER parameter, 5-7
AVG function, 6-6

B
Blocksize data area, 6-6
bugs fixed as of this release, 2-2

C
CALL command, 8-3, 8-6
CALL procedure, 8-7
CALL statement, 8-6
Change or Enter New Server Attributes panel, 7-10
Change Privileged Ids (Relative to AM400_COM) panel, 7-12
Change RID (Resolve In Doubt) Options panel, 7-8
Changes and enhancements, 2-2
CHAR datatype, 8-9
Checklists
 enabling two-phase commit, 7-7
 installation, 4-2
 Oracle Net
 TCP/IP AS/400 configuration, 5-6
 post-installation, 4-6
CHG2PCPRM command, 6-3, 6-15, 7-8, 7-11
CHGSQLDBG command, 6-3, 6-10
CHGSQLDFLT command, 6-3, 6-4, 7-2, 8-2, B-6
CHGSQLNET command
  Access Manager commands, 6-3
  enabling Oracle Net tracing, 5-2
  network parameters panel, 6-8
CHGUSRPRF command, 4-5
CHGSQDFLT command, 2-4
Clauses
  FOR FETCH ONLY, 6-6
CLEANUP_DELAY parameter, 7-9, 7-14, 7-16
Commands
  ADDLIBLE, 4-7
  AM400.COM program, 7-15
    ADD, 7-15
    FORCE, 7-17
    LIST, 7-17
    MODIFY, 7-15
    QUERY, 7-16
    REMOVE, 7-15
    SET, 7-16
    SHUTDOWN, 7-15
    STARTUP, 7-15
    STATUS, 7-15
  CHG2PCPRM, 7-8, 7-11
  CHGSQDFLT, 7-2
  CHGSQLNET, enabling Oracle Net tracing, 5-2
  CHGUSRPRF, 4-5
  CHGSQDFLT, 2-4
  COMMIT, 1-9, 2-6, 7-3, 7-4, 7-13
  COMMIT HOLD, 2-6
  CREATE TABLE, 8-2
  CRTLINETH, 5-7
  CRTLINS DLC, 5-7
  CRTLINTRN, 5-7
  CRTLINX25, 5-7
  CRTSQLXXX, 2-6
  CRTSQLxxx, 1-10
  DB2/400
    CONNECT TO, 8-2
    SET CONNECTION, 8-4
  DSPJOBLOG, 4-5
  DSPLIND, 5-7
  EDITF, defining TNSNAMES entry, 5-7
  GRANT, 7-13
  IBM
    CALL, 1-6
  CRTSQL, 8-4
  LODRUN, 4-3
  Oracle Access Manager, 6-2, 6-3
    ADDLIBLE, 6-2
    CALL, 8-3, 8-6
    CHG2PCPRM, 6-15
    CHGSQDBG, 6-10
    CHGSQDFLT, 6-4, 8-2, B-6
    COMMIT, 6-7
    CRTORAPKG, 6-12, 8-3
    CRTSQLxxx, 6-5
    DESCRIBE TABLE, 8-6
    DISCONNECT, 8-5
    RELEASE ALL, 8-5
    ROLLBACK, 6-7
    SELECT, 8-6
    RELEASE, 8-5
    ROLLBACK, 1-9, 2-6, 7-4, 7-13
  SQL*Net
    ADDRDBDIRE, 5-3, 5-4, 5-8
    WRKMBRPDM, 5-7
  STRSBS, 7-12
  STRSQL, 1-10
  COMMIT command, 1-9, 2-6, 6-7, 7-3, 7-4, 7-13
  COMMIT HOLD command, 2-6
  commit phase, defined, 7-3
  commit point site, defined, 7-2
  COMMIT statement, 8-4, 8-5
  Commitment control resource, 6-7
Components, 1-5
  Application Requester Driver interface, 1-5
  AS/400 application, 1-5
  Oracle Access Manager, 1-5
  Oracle server, 1-5
  SQL client manager, 1-5
  SQL*Net, 1-5
Configuring Oracle Access Manager, 6-1
Configuring SQL*Net TCP/IP, 5-6
  TNSNAMES entry, 5-7
  CONNECT statement, 1-8
  CONNECT TO command, 8-2
  CONNECT TO statement, 8-4
Connecting
  through Oracle Net, 5-3
  through TCP/IP, 5-5
to Oracle server from precompiled application, 8-3
Connection management, 8-4
COUNT(*) function, 6-6
CREATE statement, 6-7
CREATE TABLE command, 8-2
CRTLINETH command, 5-7
CRTLINS DLC command, 5-7
CRTLINTRN command, 5-7
CRTLINX25 command, 5-7
CRTORAPKG command, 6-3, 6-12, 8-3
CRTSQL command, 8-4
CRTSQLxxx command, 1-8, 2-6, 6-5

D
Data areas
  blocksize, 6-6
Data Definition Language (DDL), 2-6
Data Manipulation Language (DML), 2-6
DATACLUE_TIMEOUT parameter, 7-9, 7-14, 7-16
Datatypes
  conversion of Oracle to DB2/400, 8-9
  DB2/400, 8-9
  LONG, 2-6
  LONG RAW, 2-6
  TIME, 2-5
  TIMESTAMP, 2-5
DATE datatype, 8-9
DB2/400
  commands
    CONNECT TO, 8-2
    SET CONNECTION, 8-4
    CONNECT syntax, 8-4
  statements
    CONNECT TO, 8-4
    DECLARE PROCEDURE, 8-7
DEAD_CHECK_COUNTER parameter, 7-9, 7-16
DECIMAL datatype, 8-9
DECLARE PROCEDURE statement, 8-7
default datatypes precision and scale, 2-4
Defining Oracle Access Manager, 5-4
  RDB entry, 5-4
  TNSNAMES entry, 5-4
DELETE statement, 2-5
DESCRIBE TABLE command, 8-6
Describing tables, 8-6
Description field, 7-11
Diagnosing problems
  Oracle Access Manager, 6-10, 6-14
DISCONNECT command, 8-5
Distributed database, 1-9
Distributed unit of work, 1-9
Documentation
  recommended, 3-4
DROP statement, 6-7
DSPJOBLOG command, 4-5
DSPLIND command, 5-7

E
EDTF command, defining TNSNAMES entry, 5-7
error
  ORA-0090, invalid CREATE command, 8-2
  ORA-1457, when using STRSQL, 2-4
Error code processing
  Oracle Access Manager, 8-10
EXEC SQL statement, 2-5

F
Features, 1-6
  datatype conversions, 1-7
  distributed applications, 1-7
  leveraging existing applications, 1-6
  SQL language, 1-6
  universal access, 1-7
Fields
  SQLERRMC, 8-10
  SQLSTATE, 8-10
Files
  ORA(TNSNAMES), 5-4
  Oracle Access Manager
    QCBLLESRC, 8-3
    QRFQLESRC, 8-3
    QSQLCSRC, 8-3
  SQL*Net
    ORA(TNSNAMES), 5-4
FLOAT datatype, 8-9
FOR FETCH ONLY clause, 6-6
FORCE option, 7-13
Functions
  AVG, 6-6
  COUNT(*), 6-6
  MAX, 6-6
  SUM, 6-6
  TO_CHAR, 6-14
  TO_DATE, 6-14

G
  GETRESULTS procedure, 8-7
  GRANT command, 7-13

H
  Hardware requirements, 3-2
  HOLD parameter, 2-6

I
  IBM
    commands
      CALL, 1-6
      CRTSQL, 8-4
      Distributed Relational Databases, 1-9
  implied commit, defined, 7-3
  IN OUT parameter, 8-7
  IN parameter, 8-7
  In-doubt transactions, 7-5
    resolving, 7-5
  Install Oracle Access Manager Product panel, 4-3
  Installation, 4-1
    checklist, 4-2
    choosing the library, 4-3
    default values, 4-2
    finish installation, 4-5
    logging on to AS/400, 4-2
    mount product CD, 4-3
    select installation library name, 4-3
    start install process, 4-3
    verify installation parameters, 4-4
    verify successful installation, 4-5
  INTEGER datatype, 8-9
  Introduction
    Oracle Access Manager, 1-1

K
  known problems and restrictions, 2-4

L
  Language types
    values
      *C, 6-13
      *COBOL, 6-13
      *RPG, 6-13
  Libraries
    Oracle Access Manager, 6-2
    Oracle Access Manager installation, 6-2
  LIST command, 7-17
  LODRUN command, 4-3
  LONG datatype, 2-6, 8-9
  LONG RAW datatype, 2-6, 8-9

M
  MAX function, 6-6
  Members
    name format, 5-2
  Messages
    Oracle Access Manager
    processing, 8-10
  MODIFY command, 7-15

N
  Network
    requirements, 3-3
  NLS, 6-7
    languages, B-5
  NUMBER datatype, 8-9

O
  Operating system
    requirements, 3-3
  Options
FORCE, 7-13
Oracle Access Manager
   USER, 6-5
   USING, 6-5
ORA(TNSNAMES) file, 5-3, 5-4
ORA-0090 error, invalid CREATE command, 8-2
ORA-1457 error, when using STRSQL, 2-4
Oracle Access Manager
   architecture, 1-3
calling, 1-8
commands, 6-2
   ADDLIBLE, 6-2
   CALL, 8-3, 8-6
   CHG2PCPRM, 6-15
   CHGSQDBG, 6-10
   CHGSQDFLT, 6-4, 8-2, B-6
   COMMIT, 6-7
   CRTORAPKG, 6-12, 8-3
   CRTSQLxxx, 6-4, 8-2, B-6
   DESCRIBE TABLE, 8-6
   DISCONNECT, 8-5
   ROLLBACK, 6-7
   SELECT, 8-6
   commitment control resource, 6-7
   configuring, 6-1
   connect processing, 1-8
   connecting
      from AS/400 precompiled application, 8-3
to Oracle server, 8-2
   connection management, 8-4
   conversion of Oracle datatypes to DB2/400
datatypes, 8-9
   DB2/400
      CONNECT syntax, 8-4
   statements
      CONNECT TO, 8-4
debbuging, 6-10
defining to Oracle server, 5-4
describing tables, 8-6
diagnosing problems, 6-10, 6-14
derror code processing, 8-10
   explicit connection, 8-4
   features, 1-6
files
   QCBLLESRC, 8-3
QRPGLESRC, 8-3
QSQLCSRC, 8-3
implicit connection, 8-4
installation library, 6-2
introduction, 1-1
library, 6-2
message processing, 8-10
NLS support, B-2
options
   USER, 6-5
   USING, 6-5
   package creation, 1-8
parameters
   PASSWORD, 6-5
   RDB, 8-4
   stored procedures, 8-7
      IN, 8-7
      IN OUT, 8-7
      OUT, 8-7
performance
   SQL*Net trace, 6-8
programs
   SAMP1C, 8-3
   SAMP1CBL, 8-3
   SAMP1RPG, 8-3
   sample
      C, A-2
      COBOL, A-5
      RPG, A-7
RELEASE ALL command, 8-5
SQL statement processing, 1-8
   statements
      CALL, 8-6
      COMMIT, 8-4, 8-5
      CONNECT TO, 6-5
      DDL
         CREATE, 6-7
         DROP, 6-7
      ROLLBACK, 8-4, 8-5
      SELECT, 6-6, 8-3
   stored procedures, 8-6
      advantages, 8-6
debugging, 8-7
   Index-5
GETRESULTS, 8-7
tracing values
*ADMIN, 6-9
*OFF, 6-9
*USER, 6-9
16, 6-9
two-phase commit
default options panel, 6-7
Oracle Access Manager Debugging Options panel, 6-10
Oracle Access Manager for AS/400, 1-8
Oracle Access Manager Network Parameters panel, 6-8
Oracle Net
connecting through, 5-3
trace options, 6-8
tracing members, 5-2
Oracle server
connecting to, 8-2
defining Oracle Access Manager to, 5-4
Oracle Server Name field, 7-11
ORACLE_SERVER parameter, 7-10, 7-11, 7-13
OUT parameter, 8-7

P
Panels
Change or Enter New Server Attributes, 7-10
Change Privileged Ids (Relative to AM400_COM), 7-12
Change RID (Resolve In Doubt) options, 7-8
Install Oracle Access Manager Product, 4-3
Oracle Access Manager Debugging Options, 6-11
Oracle Access Manager Network Parameters, 6-8
Verify Oracle Access Manager Install, 4-4
Parameters
AUTOCREATE CONTROLLER, 5-7
CLEANUP_DELAY, 7-9, 7-14, 7-16
DATAQUEUE_TIMEOUT, 7-9, 7-14, 7-16
DEAD_CHECK_COUNTER, 7-9, 7-16
HOLD, 2-6
Oracle Access Manager PASSWORD, 6-5
RDB, 8-4
USER, 6-5
ORACLE_SERVER, 7-10, 7-13
ROBCNMMTH, 2-6
SERVER_STARTUP_TIMEOUT, 7-16
SQL*Net
TNSNAMES.ORA, 5-2
STARTUP, 7-11, 7-16
stored procedures, 8-7
IN, 8-7
IN OUT, 8-7
OUT, 8-7
TRACE_LEVEL, 7-9, 7-16
two-phase commit
ORACLE_SERVER, 7-11
verify installation parameters, 4-4
Password for Oracle Id field, 7-11
PASSWORD parameter, 6-5
Phrases
SIMPLE CALL WITH NULLS, 2-5
Post-installation, 4-6
add Oracle Access install library to library list, 4-7
checklist, 4-6
configure SQL*Net, 4-7
configure two-phase commit (optional), 4-8
retest connectivity to Oracle server, 4-8
test connectivity to Oracle server, 4-7
precision, default datatype, 2-4
prepare phase, defined, 7-2
Privileged Oracle Id field, 7-11
Privileges
GRANT command, 7-13
required, 7-13
Procedures
CALL, 8-7
GETRESULTS, 8-7
Processor
requirements, 3-2
Product set, 2-2
Programs
AM400_COM, 7-11, 7-12, 7-14, 7-17
AM400_RID, 7-5, 7-14, 7-17
Oracle Access Manager
SAMP1C, 8-3
SAMP1CBL, 8-3
SAMP1RG, 8-3
Sample programs
  Oracle Access Manager
    C, A-2
    COBOL, A-5
    RPG, A-7
scale, default datatype, 2-4
SELECT command, 8-6
SELECT statement, 2-5, 6-6, 8-3
SERVER_STARTUP_TIMEOUT parameter, 7-16
SET command, 7-16
SET CONNECTION command, 8-4
SHUTDOWN command, 7-15
SID, system identifier, 5-8
SIMPLE CALL WITH NULLS phrase, 2-5
SMALLINT datatype, 8-9
Software requirements, 3-3
SQL statements
  processing, 1-8
SQL*Net
  checklists
    TCP/IP AS/400 configuration, 5-6
  commands
    ADDRDBDIRE, 5-3, 5-4, 5-8
    CRTLINETH, 5-7
    CRTLNSDLCC, 5-7
    CRTLINTRN, 5-7
    CRTLINX25, 5-7
    DGPLIND, 5-7
    WRKMBRPDM, 5-7
  configuration, 4-7
  TCP/IP, 5-6
  file names, 5-2
  files
    ORA(TNSNAMES), 5-4
  parameters
    TNSNAMES.ORA, 5-2
    token ring address, 5-7
SQLERRMC field, 8-10
SQLSTATE field, 8-10
STARTUP command, 7-15
STARTUP parameter, 7-11, 7-16
Startup Parameter field, 7-11
Statements

Q
QCBLLESRC file, 8-3
QRPGLESRC file, 8-3
QSECOFR security administration userid, 7-12
QSQLCSRC file, 8-3
QUERY command, 7-16

R
RAW datatype, 8-9
RDB entry, 5-4
RDB parameter, 8-4
RELEASE ALL command, 8-5
RELEASE command, 8-5
Release information, 2-1
Remote unit of work, 1-9
REMOVE command, 7-15
Requirements
  communications, 3-3
  hardware, 3-2
    CD-ROM drive, 3-2
    disk space, 3-2
  processor, 3-2
  software, 3-3
  additional, 3-3
  communications, 3-3
  operating system, 3-3
  Oracle server, 3-3
  PTF, 3-3
ROBCNNMTH parameter, 2-6
ROLLBACK command, 1-9, 2-6, 6-7, 7-4, 7-13
ROLLBACK statement, 8-4, 8-5
ROWID datatype, 8-9

S
SAMP1C program, 8-3
DB2/400
CONNECT TO, 8-4
DECLARE PROCEDURE, 8-7
DELETE, 2-5
EXEC SQL, 2-5
Oracle Access Manager
CALL, 8-6
COMMIT, 8-4, 8-5
CONNECT TO, 6-5
DDL
CREATE, 6-7
DROP, 6-7
ROLLBACK, 8-4, 8-5
SELECT, 6-6, 8-3
SELECT, 2-5
SQL
processing, 1-8
UPDATE, 2-5
STATUS command, 7-15
Stored procedures, 2-5, 8-6
advantages, 8-6
declaring procedures, 8-7
GETRESULTS procedure, 8-7
parameters, 8-7
IN, 8-7
IN OUT, 8-7
OUT, 8-7
STRBS command, 7-12
STRESS, 6-14, 8-2
STRESS SQL command, 1-10
SUM function, 6-6
SYS.PENDING_SESSIONS$ table, 7-13
SYS.PENDING_TRANS$ table, 7-13
system identifier (SID), 5-8
System requirements
hardware, 3-2
disk space, 3-2
processor, 3-2
software
additional software, 3-3
DB2/400 Query Manager, 3-3
network, 3-3
operating system, 3-3

T
Tables
SYS.PENDING_SESSIONS$, 7-13
SYS.PENDING_TRANS$, 7-13
TCP/IP
AS/400 configuration
define physical line, 5-7
define TNSNAMES entry, 5-7
configuring SQL*Net, 5-6
Terminology
two-phase commit
AM400_COM program, 7-2
AM400_RID program, 7-2
commit phase, 7-3
commit point site, 7-2
implied commit, 7-3
prepare phase, 7-2
two-phase commit parameter, 7-2
TIME datatype, 2-5
TIMESTAMP datatype, 2-5
TNS, 5-1
TNSNAMES
defining for TCP/IP, 5-7
tnsnames.ora, 5-2
entry, 5-4
tnsnames.ora parameter, 5-2
TO_CHAR function, 6-14
to_DATE function, 6-14
Token ring address, 5-7
TRACE_LEVEL parameter, 7-9, 7-16
Tracing
values
*ADMIN, 6-9
*OFF, 6-9
*USER, 6-9
16, 6-9
Transparent Networking Substrate See TNS
Two-phase commit
AM400_COM program, 7-11, 7-12, 7-14, 7-17
AM400_RID program, 7-4, 7-5, 7-9, 7-11, 7-14, 7-17
altering, 7-14
starting, 7-12
commands
AM400_COM program, 7-15
ADD, 7-15
FORCE, 7-17
LIST, 7-17
MODIFY, 7-15
QUERY, 7-16
REMOVE, 7-15
SET, 7-16
SHUTDOWN, 7-15
STARTUP, 7-15
STATUS, 7-15

CHG2PCPRM, 7-8, 7-11
CHGSQLDFLT, 7-2
COMMIT, 7-3, 7-4, 7-13
GRANT, 7-13
ROLLBACK, 7-4, 7-13
STRSBS, 7-12
default options panel, 6-7
enabling, 7-7
add or remove AM400_COM privileged
userids, 7-8
change Oracle Access Manager commit
default, 7-8
change the AM400_RID options, 7-8
checklist, 7-7
start the AM400_COM program
(optional), 7-13
start the subsystem that runs AM400_RID
program, 7-12
fields
description, 7-11
Oracle server name, 7-11
password for Oracle id, 7-11
privileged Oracle id, 7-11
startup parameter, 7-11
overview, 7-2
panels
Change or Enter New Server Attributes, 7-10
Change Privileged Ids (Relative to AM400_COMBO
Change RID (Resolve In Doubt) Options, 7-8
parameters
CLEANUP_DELAY, 7-9, 7-14, 7-16
DATAQUEUE_TIMEOUT, 7-9, 7-14, 7-16
DEAD_CHECK_COUNTER, 7-9, 7-16
ORACLE_SERVER, 7-10, 7-11, 7-13
SERVER_STARTUP_TIMEOUT, 7-16
STARTUP, 7-11, 7-16
TRACE_LEVEL, 7-9, 7-16
process, 7-4
commit phase, 7-4
phase 1, 7-4
phase 2, 7-4
prepare phase, 7-4
resolve in-doubt transactions, 7-5
tables
SYS.PENDING_SESSIONS$, 7-13
SYS.PENDING_TRANS$, 7-13
terminology, 7-2
AM400_COM program, 7-2
AM400_RID program, 7-2
commit phase, 7-3
commit point site, 7-2
implied commit, 7-3
prepare phase, 7-2
two-phase commit parameter, 7-2
two-phase commit parameter, defined, 7-2
U
UPDATE statement, 2-5
USER option, 6-5
USER parameter, 6-5
V
VARCHAR datatype, 8-9
VARCHAR2 datatype, 8-9
Verify Oracle Access Manager Install panel, 4-4
W
WRKMBRPDM command, 5-7