Component Broker for Windows NT and AIX
Oracle Application Adaptor Quick Beginnings
Release 2.0

Document Number GC09-2733-00

December 3, 1998

Owners:
IBM Corporation, http://www.ibm.com

Component Broker Home Page, http://www.software.ibm.com/ad/cb/

Component Broker for Windows NT and AIX

GC09-2733-00

Oracle Application Adaptor Quick Beginnings

Release 2.0

Component Broker for Windows NT and AIX

GC09-2733-00

Oracle Application Adaptor Quick Beginnings

Release 2.0

Note

Before using this information and the product it supports, be sure to read the general information under Appendix B, “Notices” on page 43.

Second Edition (December 1998)

This edition applies to Release 2.0 of Component Broker and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About This Book

The *Component Broker for Windows NT and AIX Oracle Application Adaptor Quick Beginnings* provides a brief technical overview of the Oracle Application Adaptor and provides information on how to:

- Install and configure the Oracle Application Adaptor portion of Component Broker with its prerequisite software.
- Write your first Component Broker Oracle applications using application development tools provided on the CBToolkit compact disc.

Note: The formal title for this package is the Component Broker Application Adaptor for Oracle. For brevity, this is often referred to as the Oracle Application Adaptor, and is often abbreviated OAA or oaa. These terms can be used interchangeably.

Do not use the either the *Component Broker for Windows NT and AIX Oracle Application Adaptor Quick Beginnings* or the *Component Broker for Windows NT and AIX Quick Beginnings* as a substitute for the Component Broker library. The library provides detailed information beyond the concepts introduced in this book. Before installing Component Broker, read the README file, located in the CBConnector compact disc root directory, for last minute product information. After completing the tasks and exercises in this book, review the rest of the product library.

Who Should Read This Book

The *Component Broker for Windows NT and AIX Oracle Application Adaptor Quick Beginnings* is intended for application programmers who want to:

- Understand the basics of the Oracle Application Adaptor run time.
- Plan an initial installation of the Oracle Application Adaptor run time on top of an existing Component Broker installation.
- Install the Oracle Application Adaptor portion of Component Broker.
- Develop basic Component Broker applications for Oracle backend systems.

How This Book is Organized

Chapter 1, “Introduction” on page 1 provides an introduction to the functions and features of the Component Broker application adaptor for Oracle as they apply to this document.

Chapter 2, “Planning the Installation” on page 3 describes the required system environment and software prerequisites for installing the Oracle application adaptor run-time environment. This chapter describes which installable units are part of this package.

Chapter 3, “Installing the Oracle Application Adaptor on Windows NT” on page 5 provides the procedures for installing, configuring, and uninstalling the Oracle application adaptor portion of Component Broker and its prerequisite software.

Chapter 4, “Installing the Oracle Application Adaptor on AIX” on page 15 provides the procedures for installing, configuring, and uninstalling the Oracle application adaptor portion of Component Broker and its prerequisite software.

Chapter 5, “Developing a Component Broker Oracle Application” on page 27 provides an end-to-end sample application using Oracle.

Appendix A, “Notes and Restrictions on the Oracle Application Adaptor” on page 39 provides additional information and restrictions for using Oracle.

Documentation Conventions

The following conventions distinguish different text elements.

plain Window titles, folder names, icon names, and method names.

`monospace` Programming examples, user input at the command line prompt or into an entry field, directory paths, and user output.

bold Menu choices, push buttons, check boxes, radio buttons, group-box controls, drop-down list boxes, combo-boxes, notebook tabs, and entry fields.

italics Programming keywords and variables, titles of documents, and initial use of unique terms.

The following shortcut conventions are used to abbreviate menu selections and object expansions within tree views of the System Manager User Interface and Object Builder.

→ The right arrow, when used within a tree, shows a series of folder (or object) expansions. For example, “**File** → **New**” is translated to mean: “On the **File** menu, click **New**”.

The right arrow, when used within a tree view, shows a series of folder (or object) expansions. For example, “Expand Management Zones → Sample Cell and Work Group Zone → Configuration” is translated to mean:

1. Expand Management Zones.
2. Expand Sample Cell and Work Group Zone.
3. Expand Configuration.

Note: An object in a view can be expanded when there is a plus (+) sign beside that object. After an object is expanded, the plus is replaced by a minus (-) sign.

+ Expands a tree structure to show more objects. To expand, click the plus (+) sign beside any object. If you double-click the object, a new tree structure is displayed with that object as the root of the tree.

- Collapses a tree structure. To collapse, click the minus (-) beside any object.

left mouse button
Used for all actions in the application except for opening the pop-up menu of an object. For example, if you click with the left mouse button on an object, it is selected. Click with the left mouse button on a menu option to perform that action.

right mouse button
Opens the pop-up menu of an object that contains a list of actions that can be performed on that object. The list varies depending on the type of object.

The Component Broker Documentation

The following information is part of Component Broker:

- Help information is available from Component Broker product panels.
- The Component Broker online library can be viewed using a frames-compatible Web browser.
- *Component Broker for Windows NT and AIX Quick Beginnings*, G04L-2375 explains how to easily create and verify a starter Component Broker environment. These instructions walk the user through a

typical server and client installation. Users can extend this configuration using the information in the *Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide*.

- *Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide*, SCO9-2798 provides a comprehensive overview of the Component Broker environment, then guides the user through planning considerations including capacity planning, performance tuning, prerequisites, and migration. It also leads the user through installation options for all Component Broker environments.
- *Component Broker for Windows NT and AIX CICS and IMS Application Adaptor Quick Beginnings*, GC09-2703 provides a brief technical overview of the CICS and IMS application adaptor and guides the user through its installation and configuration. Step-by-step instructions guide the user through creating an initial CICS and IMS application using application development tools included in the CBToolkit package.
- *Component Broker for Windows NT and AIX System Administration Guide*, SC09-2704 provides information about configuring and operating one or more hosts managed by Component Broker. It also provides general information about using the System Manager User Interface.
- *Component Broker Application Development Tools*, SC09-2705 explains how to create and test Component Broker applications using the tools provided in the CBToolkit with a focus on common development scenarios such as inheritance and team development.
- *Component Broker Programming Guide*, G04L-2376 describes the programming model including business objects, data objects, and information about MOFW, IDL, and C++ CORBA programming.
- *Component Broker Advanced Programming Guide*, SC09-2708 describes the Component Broker implementation for the CORBA Object Services and the Component Broker Object Request Broker (including remote method invocation and the Dynamic Invocation Interface (DII) procedures), Session Service, Cache Service, Notification Service, Interlanguage Object Model (IOM), and work-load management (WLM).
- *Component Broker Programming Reference*, SC09-2707 contains information about the APIs available to Component Broker application developers.
- *Component Broker for Windows NT and AIX Problem Determination Guide*, SCO9-2799 explains how to identify and resolve problems within a Component Broker environment using the tools provided with Component Broker. The book includes information on installation problems, run time errors, debugging of applications, and analysis of log messages.
- *Component Broker Glossary*, SC09-2710 contains terms and definitions relating to Component Broker.
- *OS/390 Component Broker Introduction*, GA22-7324 describes the concepts and facilities of Component Broker and the value it has on the OS/390 platform. The audience is a knowledgeable decision maker or a system programmer.
- *OS/390 Component Broker Planning and Installation*, GA22-7331 describes the planning and installation considerations for Component Broker on OS/390.
- *OS/390 Component Broker System Administration*, GA22-7328 describes system administration tasks and operations tasks, as provided in the system administration user interface for OS/390.
- *OS/390 Component Broker Programming: Assembling Applications*, GA22-7326 provides information for assembling applications using Component Broker on OS/390.
- *OS/390 Component Broker Operations: Messages and Diagnosis*, GA22-7329 provides diagnosis information and describes the messages associated with Component Broker on OS/390.

Chapter 1. Introduction

This chapter provides an introduction to the features of the Component Broker Application Adaptor for Oracle (also referred to as the Oracle Application Adaptor.)

The base Component Broker product includes a DB2 application adaptor that enables Component Broker access to data in DB2 databases. The Component Broker Oracle Application Adaptor package provides an additional application adaptor to enable Component Broker access to data in Oracle databases. The Component Broker Toolkit Sample package has been enhanced to provide tool support for creating Component Broker objects that can be used to access data in Oracle databases. A sample application that accesses an Oracle database using a Component Broker object is provided on the base Component Broker Toolkit Sample package.

Note: The Oracle Application Adaptor is intended to allow the connection of Component Broker applications to Oracle databases, not to Oracle applications. The intent is to allow Oracle data to become an integral part of Component Broker applications, and to be accessed using the Component Broker programming model.

The Component Broker Application Adaptor for Oracle Package

The Component Broker Application Adaptor for Oracle package contains the run time extensions needed to support access to Oracle databases from Component Broker. The samples for the Oracle Application Adaptor are included in the Component Broker Toolkit Sample package.

The software shipped with this package is:

- "Component Broker Application Adaptor for Oracle"

Component Broker Application Adaptor for Oracle

This application adaptor provides the "glue" that allows seamless access to Oracle databases from Component Broker applications. The Oracle Application Adaptor is integrated with the Component Broker object services:

- Transactions
- Security
- Connection management
- Cache
- Query
- Systems management

Chapter 2. Planning the Installation

This chapter provides information you may find useful in planning your Oracle Application Adaptor installation. This chapter contains the following information:

- “Packaging”
- “System Requirements”
- “Prerequisites”
- “The Component Broker Package”
- “Installation Considerations” on page 4
- “Installation Options” on page 4

Packaging

The Component Broker Application Adaptor for Oracle is packaged as follows:

- Component Broker Application Adaptor for Oracle on NT and AIX compact disc
- *Component Broker for Windows NT and AIX Oracle Application Adaptor Quick Beginnings*

System Requirements

The system requirements for installing the Component Broker Application Adaptor for Oracle support is the same as for installing the base Component Broker packages. For additional details, see the *Component Broker for Windows NT and AIX Quick Beginnings*.

Prerequisites

The following products are prerequisites for the Component Broker Application Adaptor for Oracle:

Windows NT Requirements:

- IBM Component Broker for Windows NT, Version 2.0 and all associated prerequisites.

  Note: As a minimum, the Component Broker server must be installed.

- Oracle8 Version 8.0.4.0.0 for Windows NT or later.
- Oracle8 RDBMS 8.0.4.1.1 Patch for Oracle8 Enterprise Edition for Windows NT or later.

AIX Requirements:

- IBM Component Broker for AIX, Version 2.0 and all associated prerequisites.

  Note: As a minimum, the Component Broker server must be installed.

- Oracle Server Enterprise Edition Version 8.0.5.0.0 for AIX-Based Systems or later.
- The Component Broker Application Adaptor for Oracle requires a patch for the Oracle8 Server Enterprise Edition. Version 8.0.5.0.0 for AIX-Based Systems. Please contact your IBM support person for more information on what patch is needed.

The Component Broker Package

The Oracle Application Adaptor run time is on the Component Broker Application Adaptor for Oracle compact disc. The samples are located on the Component Broker CBToolkit compact disc.

Installation Considerations

There are no identified restrictions applicable to the Component Broker Application Adaptor for Oracle.

Installation Options

There are no selectable options for installing the Component Broker Application Adaptor for Oracle.

Chapter 3. Installing the Oracle Application Adaptor on Windows NT

This chapter contains the following procedures for installing the Oracle Application Adaptor on Windows NT.

- “Preparing for the Install”
- “Installing and Configuring Oracle” on page 7
- “Installing and Configuring the Oracle Application Adaptor” on page 11
- “Installing and Configuring Component Broker Applications That Use Oracle” on page 13
- “Uninstalling Component Broker Applications that use Oracle” on page 13
- “Uninstalling the Oracle Application Adaptor” on page 14

See the “Installation and Configuration” section of the *Late Breaking News* provided with Component Broker for important setup information.

Preparing for the Install

- Choose where to install Oracle
- Choose authentication scheme
- Choose naming scheme

Choose Where to Install Oracle and the Oracle Application Adaptor

The Oracle server software can be installed on either the computer where you will install the Oracle Application Adaptor or on another Windows NT or AIX machine. If the Oracle server is installed on a different machine, then the Oracle client must be installed on the same machine that the Oracle Application Adaptor is installed on. The Oracle Application Adaptor must be installed on the same machine that the Component Broker server run time is installed on.

Only machines that have both the Component Broker server run time and the Oracle Application Adaptor installed can have Component Broker servers configured to run Component Broker applications that use Oracle. See the “Installing IBM Component Broker for Windows NT” topic, in the *Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide* for how to install Component Broker.

Regardless of where you install the Oracle Application Adaptor and the Oracle server, you will need to choose which authentication scheme for user ID's and which naming scheme for Oracle databases you will use.

Oracle Authentication Schemes

Users are defined to Oracle in two ways:

- “OS Authentication” on page 6
- “Oracle Authentication” on page 6

Each method has its advantages and disadvantages when used with Component Broker.

OS Authentication

OS authorization allows Oracle to use operating system authentication for authentication of users to Oracle. This method can be used if you have one user to verify on the machine that you have installed Component Broker on and this user owns the tables that are to be accessed. All Oracle applications running on the machine will use the same user ID and password for verification to Oracle databases. This is because:

- All servers run under the user ID under which the Component Broker Connector service starts up.
- The server ID is passed to Oracle, since the server is an Oracle client.

The Component Broker Connector service should be set up to run under this user. Note that under this method, the Oracle client directly passes the user ID and password of the user to the Oracle server. The user ID and password cannot be specified in Component Broker when Windows NT authorization is used.

Oracle Authentication

Oracle authorization allows users and passwords to be stored within Oracle. This method can be used if you have multiple users that you would like to use to verify access to Oracle. Each user ID and password must be defined to Oracle. The Component Broker Connector service can run under whatever user ID that you want to run under. For each RDBconnection, the open string must be modified to specify what user ID and password should be used to verify access to the table.

Access to Oracle Tables

Regardless of which authentication scheme is chosen, each user ID that will be used to access a table must have the proper privileges in order to work on the table. The user must also be able to connect and work with resources. Other privileges may be necessary. There are several ways to give a user these privileges:

- The user creates the table.
- The user is given privileges to work on all tables.
- The user is given privileges to work on specific tables.

See the Oracle documentation for how to specify privileges to tables.

Oracle NT Naming Schemes

Oracle 8.0 has two Network Configuration Models:

- Localized management

  Network addresses are mapped in a TNSNAMES.ORA file on each node (no Oracle Names Server). Naming Methods available are: Host Naming, Local Naming, and External Naming.

- Centralized management

  Networks where an Oracle Names Server performs network address resolution. An Oracle Names Server stores client configuration profiles in one location. Naming Methods available are: Oracle Names.

The instructions for setting up Component Broker are the same no matter what naming method is used because the service name defined in Net8 is always used to access the database.

Using Net8, the service name can be defined to a database on a remote machine. Component Broker uses this service name to access the database. The service name is a part of the “open string” that is part of the configuration information shipped with the Component Broker application.

See the Oracle documentation for more information on how to use each of the naming schemes and for using Net8 to set up service names.

Installing and Configuring Oracle

This section contains information about and procedures for installing the software prerequisites, including:

- “Setting Up Oracle”
- “Installing Oracle Patches” needed to run with OAA
- “Configuring the Windows NT Registry for XA”
- “Configuring an Oracle Database” on page 8
- “Creating Users” on page 10
- “Setting up Oracle Databases for Remote Access” on page 10

Setting Up Oracle

The Component Broker installation software does not verify the version of an existing Oracle installation on your server machine. As a result, the person doing the installation must verify that the correct version of Oracle has been installed on the system.

As part of Oracle Installation, you may have chosen to install the pre-configured database. If so, follow the rest of these instructions as is. If not, then you may need to modify these instructions to work with an alternate database that you have created.

Installing Oracle Patches

The Component Broker Application Adaptor for Oracle requires the Oracle RDBMS 8.0.4.1.1 for MSWindows NT or later patch for the Oracle8 Enterprise Edition for Windows NT version 8.0.4.0.0. At times, Oracle uses the same patch number for patches for different products. Please make sure that the patch you install is for RDBMS. Failure to install the correct patch will result in data not being committed to the Oracle database until the Component Broker server is shutdown.

You must download this patch from the Oracle Web Support page, at http://support.oracle.com/. A valid Oracle MetaLink account is required to download the patch. If you do not have one, you can use the registration page that is accessible from the Web Support page. You will need to enter one of your valid Support Identifiers (CSI, Access Code, and so on) as indicated on your invoice, so have it ready.

Once you have your MetaLink account set up, downloading and installing the patch (80411.exe) is easy and automatic. There are no special selections that you have to make. Instructions are included with the patch if you need any help.

Configuring the Windows NT Registry for XA

You need to make a change to the registry in order to configure Oracle as an XA Resource Manager. This change must be made using `regedt32`, not `regedit` (`regedit` does not let you assign the appropriate type to this value).

1. From a DOS command prompt enter `regedt32`, or select the regedt32 icon from your Windows NT environment.
2. In the HKEY_LOCAL_MACHINE on Local Machine window, double-click SOFTWARE and then double click ORACLE.
3. From the **Edit** menu, select **Add Value**.

4. Enter a **Value Name** of ORA_XA_REG_DLL.
5. Change the **Data Type** field to REG_EXPAND_SZ.
6. Click the **OK** button.
7. Enter `somtrxli` as the value of the **String** field.
8. Click the **OK** button.

Configuring an Oracle Database

Refer to Chapter 2, Creating an Oracle Database, of the *Oracle8 Administrator's Guide*, Release 8.0 that came with the Oracle compact disc for instructions on creating an Oracle database.

The following configuration steps are for configuring the default database that is created as part of the installation above. They need to be repeated for each new database you create. The expression *ORACLE_HOME* refers to the directory in which you installed Oracle (for example, C:\ORANT).

1. Check the *ORACLE_HOME*\NET80\ADMIN\LISTENER.ORA file to make sure that it has a SID_DESC stanza as illustrated in the following example. If it does not exist, add one. Stop and restart the OracleTNSListener80 service after making the change.

   ```
   (SID_LIST =
   (SID_DESC =
   (GLOBAL_DBNAME = machine_name)
   (SID_NAME = ORCL)
   ```

   Note:

   a. *machine_name* represents the IP name of the computer on which the Oracle database is installed. For example, if the IP name is `cbroker.austin.ibm.com`, you would use `cbroker`.

   b. ORCL is the system identifier for the default database.

2. The database needs to have a service name (also referred to in Oracle documentation as *host_string* or as database alias) defined so that the Oracle8 Client can connect to it. Use the Oracle Net8 Assistant to do this.

   a. On the **Start** menu, click **Programs** → **Oracle for Windows NT** → **Oracle Net8 Assistant**.

   b. Click the **Yes** button in the Net8 Assistant Warning pop-up window.

   c. Double-click the Service Names folder.

   d. Click **Edit** → **Create**.

   e. In the dialog:

      1) Type a string for the service name. If you are configuring the database for the Oracle sample application that is shipped with the Component Broker Toolkit, use `OPolicy`. This is the string that must be used as the Database Name in Object Builder's Schema definition panels.

      2) Click the **Next** button to continue.

   f. In this window:

      1) Select the **TCP/IP** (Internet Protocol).

      2) Click the **Next** button to continue.

   g. In this window:

      1) Type the host name portion of the IP name for the machine (for example, `cbroker` if the IP name is `cbroker.austin.ibm.com`).

2) Click the **Next** button to continue.

h. In this window:

1) Enter ORCL as the system identifier (SID).

Note: The SID for the database is specified when the database is created. ORCL is the system identifier for the default database.

2) Click the **Next** button to continue.

i. **Optional:** Click the **Test Service** button.

Note: This step is not required and could hang your computer when the configuration process is completing.

In this window:

1) Type SYSTEM as the **UserName**.
2) Type MANAGER as the **Password**.
3) Click the **Test** button. You should get a message indicating that the test was successful.
4) Click the **Done** button.

j. Click the **Finish** button. The new Service Name should appear in the Service Names folder.

k. Click **File** > **Save Network Configuration**.

3. Stop the Oracle services:

a. On the Control Panel, open Services.
b. Select the OracleStartORCL service.
c. Click the **Stop** button.
d. Select the OracleServiceORCL service.
e. Click the **Stop** button.
f. Click the **Close** button.

4. If you are using Windows NT authentication, then using Notepad or a similar editor, add the following lines at the end of the initORCL.ora file (*ORACLE_HOME*\database\initORCL.ora) to enable the Windows NT Operating System Authentication mechanism for logging on the Oracle database.

```
os_authent_prefix = ""
remote_os_authent = true
```

Note:

a. Refer to the *Oracle8 Administrator's Guide* for more details.

b. In the example, the quotes ("") are opening and closing double quotes with no space between (a null string). Using an editor other than Notepad could cause the double quotes to be entered with additional control characters that cause an

```
Oracle not available
```

error message when using parts of Oracle.

5. Restart the Oracle service:

a. On the Control Panel, open Services.
b. Select the OracleStartORCL service.
c. Click the **Start** button.
d. Click the **Close** button.

Creating Users

On the machine that has the Oracle server installed, do the following for each user to be created in Oracle:

1. Open a session of SQL Plus 8.0. On the **Start** menu, click **Programs** → **Oracle for Windows NT** → **SQL Plus 8.0**.
2. Login to SQL Plus 8.0. In this window, do the following:
   a. In the **User Name** field, type SYS.
   b. In the **Password** field, type CHANGE_ON_INSTALL (the default Oracle password). If you have changed the password, type in your new password.
   c. In the **Host String** field, type the name of the default database.
   d. Click the **OK** button to login.
3. Run the following SQL statements in the session. Replace USER_NAME with the user ID under which the Component Broker servers will run. (Alternatively, the Security Manager application can be used to accomplish the same task.)
   a. Create the user

   ```
   CREATE USER USER_NAME IDENTIFIED EXTERNALLY DEFAULT TABLESPACE "USER_DATA"
     TEMPORARY TABLESPACE "TEMPORARY_DATA" PROFILE DEFAULT ACCOUNT UNLOCK;
   ```

   b. Give the user connect and resource privileges

   ```
   GRANT "CONNECT" TO USER_NAME;
   GRANT "RESOURCE" TO USER_NAME;
   ```

   c. For convenience in testing, you may want to give the user unlimited tablespace.

   ```
   GRANT UNLIMITED TABLESPACE TO USER_NAME;
   ```

   d. In order to use applications which use XA interfaces, such as Component Broker applications, the user must be give SELECT privilege on the SYS.DBA_PENDING_TRANSACTIONS table.

   ```
   GRANT SELECT ON "SYS"."DBA_PENDING_TRANSACTIONS" TO USER_NAME;
   ```

   e. Turn on all the roles the user has been granted.

   ```
     ALTER USER USER_NAME DEFAULT ROLE ALL;
   ```

4. Exit the session.

Setting up Oracle Databases for Remote Access

See the Oracle publication, “Net8 Getting Started Release 8.0.4 for Windows NT and Windows 95,” Chapter 6 “Configuring the Network” and Chapter 7, “Connecting To a Database” for complete information on configuring Oracle for remote database access.

The following example assumes that the Local Naming naming method is used. In Local Naming, NTS authentication is used and the communication protocol is TCP/IP. Service names are associated with a TCP/IP address.

1. Set up the database on the host and configure it on the client.
2. Set up the user on the client and the host.
   a. The user ID and password must be the same on the client and the host.
   b. On the host, give the user ID appropriate privileges to insert, update, delete, select, and lock on the specific table or on all tables.

Use “Using the Local Naming Method” section of Chapter 6, to set up your system. Check the prerequisites to see if this fits your situation, otherwise, use another naming method.

A brief summary of the steps for using the host naming method follow. See Chapter 6 for more information.

1. Server configuration:
   a. LISTENER.ORA

      Check that GLOBAL_DBNAME = {hostname}, where {hostname} is the TCP/IP host name or alias of the server system. You can either look at the file or follow the instructions for using Net8.

      (During installation of Net8, the GLOBAL_DBNAME of the default database is set to the system name.)
2. Change the following files on the client:
   - TNSNAMES.ORA

     On the client, using Net8, set up the service with the host name for each remote database.
3. Change the following files on the server:
   - TNSNAMES.ORA

     Using Net8, set up the service name for the local database.

If Oracle authentication is being used, then a different user ID and password can be used to access the database than the one the Component Broker Connector service is running under. To set this up, the openString of the RDBConnection for the application needs to be changed to specify the user ID and password. For example, assuming the database is call OPolicy, the service name for the database ID OPolicy, the openString would normally look like this:

```
openString = "ORACLE_XA+ACC=P//+SesTm=0+Threads=True+DB=OPolicy+SqlNet=OPolicy"
```

To add a user ID of Harmon and a user ID of xyz123, change the open string to:

```
openString = "ORACLE_XA+ACC=P/harmon/xyz123+SesTm=0+Threads=True+DB=OPolicy+SqlNet=OPolicy"
```

Installing and Configuring the Oracle Application Adaptor

This section discusses the installation of the Oracle Application Adaptor run time.

- Verify Component Broker server run time is installed
- Install toolkit
- Install Oracle Application Adaptor

Verifying the Installation of the Component Broker Run Time

The Oracle Application Adaptor requires the Component Broker server run time.

To verify whether the Component Broker server is installed:

1. From the Windows NT **Start** menu, select **Run**. The Run dialog is displayed.
2. In this dialog, enter:

   ```
   regedit
   ```

   The Registry Editor window is displayed.

3. Expand HKEY_LOCAL_MACHINE → SOFTWARE → IBM → Component Broker, and select Packages.
4. Verify that the server is installed.

If the server run time is not installed, the Oracle Application Adaptor cannot be installed. For details on installing the Component Broker server run time, see the *IBM Component Broker Quick Beginnings*.

Install Toolkit

If you intend to develop Component Broker Oracle applications, you should also have installed the Component Broker Toolkit package. See *IBM Component Broker Quick Beginnings* for instructions on installing the Component Broker Toolkit Sample package. The Toolkit includes a sample of a Component Broker Oracle application and a step-by-step explanation of its development.

Installing the Oracle Application Adaptor for Windows

Perform the following steps to install the Oracle Application Adaptor option:

1. Insert the Oracle Application Adaptor compact disc into your CD-ROM drive. If your system autoloads the compact disc, you can skip to step 3.
2. Display the contents of the compact disc.
   a. Go to winnt directory:
   ```
   cd winnt
   ```
   b. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To install the Oracle Application Adaptor, you must click the Setup icon that looks like a computer. The Welcome window is displayed.
3. On the Welcome window, click the **Next** button. A message is displayed indicating that Setup is searching for Component Broker for Windows NT packages on your system. An informational message states that Component Broker for Windows NT packages were found and lists the directory in which they were found.
   a. Click the **OK** button.
   b. In the Verify Configuration Setting window, verify that the items to be installed are correct.
      - Click the **Back** button to review or change any information you previously entered.
      - Click the **Next** button to start the installation.
4. On completion of the installation, a window displays stating that the installation has completed. When you exit this window, your computer automatically restarts. Optionally in this window you can select check boxes to:
   - Automatically launch the configuration tool when the computer restarts.
   - Automatically display the readme file when the computer restarts.
5. Click the **Finish** button to close the installation program and to restart your computer.

Configuring the Oracle Application Adaptor

Perform the following steps to configure your Oracle Application Adaptor installation:

1. If your computer starts the configuration tool automatically when you restart your computer after performing the installation, skip to the next step. To start the configuration tool from the Windows NT **Start** menu, select **Programs** → **IBM Component Broker** → **Component Broker Configuration tool**. The Configure New Install window opens.

2. In the Configure New Install window:
   a. Select the **Yes** radio button to configure your additional Component Broker for Windows NT installation.
   b. Select the **Next** button to continue.
3. In the Verify Configuration Setting window, verify that the items to be installed are correct.
   - Click the **Back** button to review or change any information you previously entered.
   - Click the **Next** button to start the configuration.
4. The Configuration Status window displays information about the items being configured.
5. An informational window is displayed stating that the configuration has completed. Click the **OK** button to close the window.
6. Exit the Component Broker Configuration tool. In the Configuration Status window, click either the **Close** button or the **Cancel** button.

Installing and Configuring Component Broker Applications That Use Oracle

Installing Component Broker applications that use Oracle is similar to installing any other type of Component Broker application:

1. Create databases and tables if needed.
2. Authorize users to the tables.
3. Use Component Broker Systems Management to configured hosts on which you will install the application.
4. Install the application on the host.
5. Create or choose the system management configurations to which you will add the application.
6. Create or choose the server groups or stand-alone servers on which you will configure the application.
7. Add the iOAAServices application to those configurations to which you will add Oracle backed applications.
8. Configure the iOAAServices application on those server groups or stand-alone servers.
9. Install the application on each host as you would any other Component Broker application.
10. Update Systems Management if necessary for user ID and password if Oracle authentication is being used.

See the sample application for more information on installing applications.

Uninstalling Component Broker Applications that use Oracle

- Uninstall the applications.
- Delete tables and databases as appropriate.
- Remove iOAAServices application if no longer needed.

Uninstall Oracle Applications

Use the normal methods for uninstalling any Component Broker application. See the Component Broker Quick Beginnings for details on uninstalling Component Broker applications.

Delete Tables and Databases

If the tables and databases used by the applications that are being uninstalled are not needed by other applications, you may want to delete them. Perform appropriate backup procedures as you normally would for any important data that you keep in Oracle.

Remove iOAAServices from Servers

If after removing the applications from a server or server group, there are no more applications configured on the server or server group that use Oracle, the iOAAServices application can be removed from the server or server group configuration. This can improve the performance of DB2 applications if installed on the same server.

If after removing iOAAServices from server or server groups within a configuration, if there are no more applications in that configuration that need Oracle, iOAAServices can be deleted from the configuration. Do not delete iOAAServices from the set of Host applications.

Uninstalling the Oracle Application Adaptor

- Go through the steps to uninstall Component Broker applications that use Oracle
- Uninstall Oracle Application Adaptor
- Uninstall Component Broker server run time if no longer needed
- Uninstall Oracle if no longer needed

Note: If you uninstall the Component Broker run time, you must reinstall the Oracle Application Adaptor. You do not have to reinstall Oracle or patch.

Note: If you reinstall Oracle, you do not have to reinstall the Oracle Application Adaptor.

Uninstall the Oracle Application Adaptor

You can uninstall the Oracle Application Adaptor using one of the following methods:

- From the Windows NT **Start** menu, select **IBM Component Broker Application Adaptor for Oracle** → **Uninstall**.
- From the Control Panel, select Add/Remove Programs → IBM Component Broker Application Adaptor for Oracle.

The uninstall program confirms your intentions before the uninstall.

Chapter 4. Installing the Oracle Application Adaptor on AIX

This chapter contains the following procedures for installing the Oracle Application Adaptor on AIX.

- “Preparing for the Install”
- “Installing and Configuring Oracle” on page 17
- “Installing and Configuring the Oracle Application Adaptor” on page 21
- “Installing and Configuring Component Broker Applications That Use Oracle” on page 23
- “Uninstalling Component Broker Applications That Use Oracle” on page 24
- “Uninstalling the Oracle Application Adaptor” on page 24

See the "Installation and Configuration" section of the *Late Breaking News* provided with Component Broker for important setup information.

Preparing for the Install

- “Choose Where to Install Oracle and the Oracle Application Adaptor”
- Choose authentication scheme
- Choose naming scheme

Choose Where to Install Oracle and the Oracle Application Adaptor

The Oracle server software can be installed on either the computer where you will install the Oracle Application Adaptor or on another NT or AIX machine. If the Oracle server is installed on a different machine, then the Oracle client must be installed on the same machine that the Oracle Application Adaptor is installed on. The Oracle Application Adaptor must be installed on the same machine that the Component Broker server run time is installed on.

Only machines that have both the Component Broker server run time and the Oracle Application Adaptor installed can have Component Broker servers configured to run Component Broker applications that use Oracle. See the "Installing IBM Component Broker for Windows AIX" topic, in the *IBM Component Broker Quick Beginnings* for how to install Component Broker.

Regardless of where you install the Oracle Application Adaptor and the Oracle server, you will need to choose which authentication scheme for user ID and which naming scheme for Oracle databases you will use.

Oracle Authentication Schemes

Users are defined to Oracle in two ways:

- “OS Authentication” on page 16
- “Oracle Authentication” on page 16

Each method has its advantages and disadvantages when used with Component Broker.

OS Authentication

OS authorization allows Oracle to use operating system authentication for authentication of users to Oracle. This method can be used if you have one user to verify on the machine that you have installed Component Broker on and this user owns the tables that are to be accessed. All Oracle applications running on the machine will use the same user ID and password for verification to Oracle databases. This is because:

- All servers run under the user ID that the Component Broker Connector service starts up under.
- The server ID is passed to Oracle, since the server is an Oracle client.

The Component Broker Connector service should be set up to run under this user. Note that under this method, the Oracle client directly passes the user ID and password of the user to the Oracle server. The user ID and password cannot be specified in Component Broker when AIX authorization is used.

Oracle Authentication

Oracle authorization allows users and passwords to be stored within Oracle. This method can be used if you have multiple users that you would like to use to verify access to Oracle. Each user ID and password must be defined to Oracle. The Component Broker Connector service can run under whatever user ID that you want to run under. For each RDB connection, the open string must be modified to specify what user ID and password should be used to verify access to the table.

Access to Oracle Tables

Regardless of which authentication scheme is chosen, each user ID that will be used to access a table must have the proper privileges in order to work on the table. The user must also be able to connect and work with resources. Other privileges may be necessary. There are several ways to give a user these privileges:

- The user creates the table.
- The user is given privileges to work on all tables.
- The user is given privileges to work on specific tables.

See the Oracle documentation for how to specify privileges to tables.

Oracle Naming Schemes

Oracle 8.0 has two Network Configuration Models:

- Localized Management

  Network address are mapped in a TNSNAMES.ORA file on each node (no Oracle Names Server). Naming Methods available are: Host Naming, Local Naming, and External Naming.

- Centralized Management

  Networks where an Oracle Names Server performs network address resolution. An Oracle Names Server stores client configuration profiles in one location. Naming Methods available are: Oracle Names.

The instructions for setting up Component Broker are the same no matter what naming method is used because the service name defined in Net8 is always used to access the database.

Using Net8, the service name can be defined to a database on a remote machine. Component Broker uses this service name to access the database. The service name is a part of the "open string" that is part of the configuration information shipped with the Component Broker application.

See the Oracle documentation for more information on how to use each of the naming schemes and for using Net8 to set up service names.

Installing and Configuring Oracle

This section contains information about and procedures for installing the software prerequisites, including:

- “Setting Up Oracle”
- Installing any Oracle patches needed to run with OAA
- “Configuring an Oracle Database”
- “Creating Users” on page 19
- “Setting Up Oracle Databases for Remote Access” on page 20

Setting Up Oracle

The Component Broker installation software does not verify the version of an existing Oracle installation on your server machine. As a result, the person doing the installation must verify that the correct version of Oracle has been installed on the system.

As part of Oracle Installation, you may have chosen to install the pre-configured database. If so, follow the rest of these instructions as is. If not, then you may need to modify these instructions to work with an alternate database that you have created.

Do the following while logged on with the user ID that was used to install Oracle.

Installing Oracle Patches

The Component Broker Application Adaptor for Oracle requires a patch for the Oracle8 Server Enterprise Edition. version 8.0.5.0.0 for AIX - Based Systems. Please contact your IBM support person for more information on what patch is needed.

You must download this patch from the Oracle Web Support page, at http://support.oracle.com/. A valid Oracle MetaLink account is required to download the patch. If you do not have one, you can use the registration page that is accessible from the Web Support page. You will need to enter one of your valid Support Identifiers (CSI, Access Code, and so on) as indicated on your invoice, so have it ready.

Once you have your MetaLink account set up, downloading and installing the patch is easy and automatic. There are no special selections that you have to make. Instructions are included with the patch if you need any help.

Configuring an Oracle Database

Refer to Chapter 2, Creating an Oracle Database, of the *Oracle8 Adminstrator's Guide, Release 8.0* that came with the Oracle compact disc for instructions on creating an Oracle database.

The following configuration steps are for configuring the default database that is created as part of the installation above. They need to be repeated for each new database you create. The expression:

1. Check the $ORACLE_HOME/network/admin/listener.ora file to make sure that it has a SID_DESC stanza as illustrated in the following example. If it does not exist, add one. Stop and restart the OracleTNSListener80 service after making the change.

```
(SID_LIST =
(SID_DESC =
(GLOBAL_DBNAME = machine_name)
(SID_NAME = ORCL)
```

Notes:

a. *machine_name* represents the IP name of the computer on which the Oracle database is installed. For example, if the IP name is `cbroker.austin.ibm.com`, you would use `cbroker`.

b. ORCL is the system identifier for the default database.

The database needs to have a service name (also referred to in Oracle documentation as *host_string* or as database alias) defined so that the Oracle8 Client can connect to it. Use the Oracle Net8 Assistant to do this.

a. Make sure that the environment variable, LD_LIBRARY_PATH, contains $ORACLE_HOME/network/lib. You can check this by using:

   ```
   echo $LD_LIBRARY_PATH
   ```

b. Start Net8 Assistant

   ```
   $ORACLE_HOME/bin/net8asst.sh
   ```

c. Click the **Yes** button in the Net8 Assistant Warning pop-up window.

d. Double-click the Service Names folder.

e. Click **Edit** > **Create**.

f. In the dialog:

   1) Type a string for the service name. If you are configuring the database for the Oracle sample application that is shipped with the Component Broker Toolkit, use `OPolicy`. This is the string that must be used as the Database Name in Object Builder's Schema definition panels.

   2) Click the **Next** button to continue.

g. In this window:

   1) Select the **TCP/IP** (Internet Protocol).
   2) Click the **Next** button to continue.

h. In this window:

   1) Type the host name portion of the IP name for the machine (for example, `cbroker` if the IP name is `cbroker.austin.ibm.com`).

   2) Click the **Next** button to continue.

i. In this window:

   1) Enter ORCL as the system identifier (SID).

      Note: The SID for the database is specified when the database is created. ORCL is the system identifier for the default database.

   2) Click the **Next** button to continue.

j. **Optional:** Click the **Test Service** button.

   Note: This step is not required and could hang your computer when the configuration process is completing.

k. In this window:

   1) Enter SYSTEM as the **UserName**.
   2) Specify MANAGER as the **Password**.

        3) Click the **Test** button. You should get a message indicating that the test was successful.
        4) Click the **Done** button.

    l. Click the **Finish** button. The new Service Name should appear in the Service Names folder.

    m. Click **File** > **Save Network Configuration**.

2. Stop the Oracle databases

    a. From a command window enter Server Manager

    ```
    svrmgrl
    ```

    b. Connect as INTERNAL

    ```
    connect internal
    ```

    c. Stop all the databases

    ```
    shutdown
    ```

    d. Exit Server Manager

    ```
    exit
    ```

3. If you are using OS authentication, then using *dtpad* or a similar editor, add the following lines at the end of the init*SID*.ora file ($ORACLE_BASE/admin/*SID*/pfile/init*SID*.ora) to enable the NT Operating System Authentication mechanism for logging on the Oracle database. *SID* is the oracle system identifier for the database. For example, for the default database, the SID is ORCL, so you would need to change

    ```
    $ORACLE_BASE/admin/ORCL/pfile/initORCL.ora
    ```

    ```
      os_authent_prefix = ""
      remote_os_authent = true
    ```

    Note: Refer to the *Oracle8 Administrator's Guide* for more details.

    a. In the example, the quotes ("") are opening and closing double quotes with no space between (a null string). Using an editor other than *dtpad* could cause the double quotes to be entered with additional control characters that cause an `Oracle not available` error message when using parts of Oracle.

    Note: Restart the Oracle databases

    a. From a command window enter Server Manager

    ```
    svrmgrl
    ```

    b. Connect as INTERNAL

    ```
    connect internal
    ```

    c. Start all the databases

    ```
    startup
    ```

    d. Exit Server Manager

    ```
    exit
    ```

Creating Users

On the machine that has the Oracle server installed, do the following for each user to be created in Oracle:

1. Open a session of SQL Plus 8.0 from a command window.

    ```
    sqlplus
    ```

2. Login to SQL Plus 8.0:

   Enter the following information:

   ```
   Enter user-name:sys
   Enter password:change_on_install
   ```

   Note: *CHANGE_ON_INSTALL* is the default password supplied with Oracle. If you have already changed the password you must use the new password.

3. Run the following SQL statements in the session. Replace USER_NAME with the user ID under which the Component Broker servers will run. (Alternatively, the Security Manager application can be used to accomplish the same task.)

   a. Create the user:

   ```
   CREATE USER USER_NAME IDENTIFIED EXTERNALLY DEFAULT TABLESPACE "USER_DATA"
   TEMPORARY TABLESPACE "TEMPORARY_DATA" PROFILE DEFAULT ACCOUNT UNLOCK;
   ```

   b. Give the user connect and resource privileges:

   ```
   GRANT "CONNECT" TO USER_NAME;
   GRANT "RESOURCE" TO USER_NAME;
   ```

   c. For convenience in testing, you may want to give the user unlimited tablespace.

   ```
   GRANT UNLIMITED TABLESPACE TO USER_NAME;
   ```

   d. In order to use applications which use XA interfaces, such as Component Broker applications, the user must be give SELECT privilege on the SYS.DBA_PENDING_TRANSACTIONS table.

   ```
   GRANT SELECT ON "SYS"."DBA_PENDING_TRANSACTIONS" TO USER_NAME;
   ```

   e. Turn on all the roles the user has been granted.

   ```
   ALTER USER USER_NAME DEFAULT ROLE ALL;
   ```

4. Exit the session.

   ```
   EXIT
   ```

Setting Up Oracle Databases for Remote Access

See the Oracle publication, "Oracle Net8 Administrator's Guide", for complete information on configuring Oracle for remote database access.

The following example assumes that the Local Naming naming method is used. In Local Naming, OS authentication is used and the communication protocol is TPC/IP. Service names are associated with a TCP/IP address.

1. Set up the database on the host and configure it on the client.
2. Set up the user on the client and the host.

   a. The user ID and password must be the same on the client and the host.

   b. On the host, give the user ID appropriate privileges to insert, update, delete, select and lock on the specific table or on all tables.

Use the Oracle documentation to set up your system. Check the prerequisites to see if this fits your situation, otherwise, use another naming method.

A brief summary of the steps for using the host naming method follows.

1. Server configuration

   a. LISTENER.ORA

Check that GLOBAL_DBNAME = {hostname}, where {hostname} is the TCP/IP host name or alias of the server system. You can either look at the file or follow the instructions for using Net8.

(During installation of Net8, the GLOBAL_DBNAME of the default database is set to the system name.)

2. Change the following files on the client:

   a. TNSNAMES.ORA

      On the client, using Net8, set up the service with the host name for each remote DB.

   b. SQLNET.ORA

      Includes the names resolution method. On the client, set:

      ```
      SQLNET.AUTHENTICATION_SERVICES = (???)
      ```

      The server does not need this set unless it too is being used as a client.

3. Change the following files on the server:

   a. TNSNAMES.ORA

      Using Net8, set up the service name for the local database.

   b. SQLNET.ORA The server doesn't need SQLNET.AUTHENTICATION_SERVICES set unless it too is being used as a client.

If Oracle authentication is being used, then a different user ID and password can be used to access the database than the one the Component Broker Connector service is running under. To set this up, the openString of the RDBConnection for the application needs to be changed to specify the user ID and password. For example, assuming the database is called OPolicy, the service name for the database ID OPolicy, the openString would normally look like this:

```
openString = "ORACLE_XA+ACC=P//+SesTm=0+Threads=True+DB=OPolicy+SqlNet=OPolicy"
```

To add a user ID of Harmon and a user ID of xyz123, change the open string to:

```
openString = "ORACLE_XA+ACC=P/harmon/xyz123+SesTm=0+Threads=True+DB=OPolicy+SqlNet=OPolicy"
```

Installing and Configuring the Oracle Application Adaptor

This section discusses the installation of the Oracle Application Adaptor run time.

- Verify Component Broker server run time is installed
- Install Oracle Application Adaptor

Verifying the Installation of the Component Broker Run Time

The Oracle Application Adaptor requires the Component Broker server run time. Verify that the server run time is installed by running:

```
lslpp -l | grep CB
```

If the Component Broker server run time is installed, then you should see:

```
CBConnector.custom.server
```

in the list that is displayed.

If the server run time is not installed, the Oracle Application Adaptor cannot be installed. For details on installing the Component Broker server run time, see the *Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide*.

Install Toolkit

If you intend to develop Component Broker Oracle applications, you should also have installed the Component Broker Toolkit package. See *Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide* for instructions on installing the Component Broker Toolkit package. The Toolkit includes a sample of a Component Broker Oracle application and a step-by-step explanation of its development.

Installing the Oracle Application Adaptor

These procedures assume the korn shell (ksh) as the default shell. Component Broker for AIX supports only the korn shell.

To complete the instructions in this chapter, you must be logged in as root (unless otherwise specified). Additionally, they assume installation, configuration, and uninstallation using the ASCII version of SMIT (using the smitty command from the shell prompt). To use the GUI version of SMIT (using the smit command from the shell prompt), modify the instructions as appropriate.

Perform the following steps to install the Oracle Application Adaptor option:

1. Insert the Oracle Application Adaptor compact disc into your CD-ROM drive.
2. Mount the file system. To mount, enter:

   ```
   mount /cdrom
   ```

3. At a shell prompt, enter:

   ```
   smitty install_latest
   ```

   The Install and Update from LATEST Available Software screen is displayed.

4. On this screen:

   a. In the INPUT device / directory for software field, press the **F4** key to display a list of available devices from which you can select the appropriate input device (/dev/cd0) and press the Enter key. This screen is redisplayed with additional fields.

   b. In the SOFTWARE to install field, press the **F4** key to display a list of the software packages contained on the compact disc.

   c. In the SOFTWARE to install list, scroll down to the entry containing:

      ```
      CBOAA.TYPICAL     ALL
      ```

   d. Press the **F7** key to select the packages you want to install.

   e. Press the **Enter** key. The Install and Update from LATEST Available Software screen is redisplayed, but the software packages selected are in the SOFTWARE to install field.

   f. Ensure that the AUTOMATICALLY install requisite software? and EXTEND file systems if space needed? fields are **yes**.

   g. Press the **Enter** key. The ARE YOU SURE? screen is displayed.

   h. On the ARE YOU SURE? screen, press the **Enter** key to continue.

   The installation process begins. After the files are copied, the command status, as indicated in the upper left corner of the screen should state OK. When you receive this OK status, all files were copied and the system is ready to be configured. Press the **F3** key until the System Management main screen is displayed.

The files installed during the run-time installation are in the /usr/lpp/CBConnector directory.

Note: If any errors occurred, view the $HOME/smit.log file. If the cause of the error is not clear, save a copy of the smit.log file and report the problem to IBM.

Configuring the Oracle Application Adaptor

1. Login as root user.
2. At a command prompt, enter:

   ```
   smitty apps
   ```

3. Select configure CBOAA (Oracle Application Adaptor) and press the **Enter** key.
4. On the next screen, select '**y**' to configure Application Adaptor for Oracle. If the field is not set to '**y**', use the Tab key to toggle the value.
5. Press the **Enter** key to begin configuration. The configuration is completed when the command status, as indicated in the upper left corner of the screen should state **OK**.
6. After configuration, press F10 to exit smit.
7. Log out as root.

Installing and Configuring Component Broker Applications That Use Oracle

Installing Component Broker applications that use Oracle is similar to installing any other type of Component Broker application.

1. Create databases and tables if needed.
2. Authorize users to the tables.
3. Use Component Broker Systems Management to configure hosts that you will install the application on.
4. Install the application on the host.
5. Create or choose the system management configurations that you will add the application to.
6. Create or choose the server groups or stand-alone servers that you will configure the application on.
7. Add the iOAAServices application to those configurations that you will add Oracle backed applications to.
8. Configure the iOAAServices application on those server groups or stand-alone servers.
9. Install the application on each host as you would any other Component Broker application.
10. Update Systems Management if necessary for user ID and password if Oracle authentication is being used.

See the sample application for more information on installing applications.

Uninstalling Component Broker Applications That Use Oracle

- “Uninstall Oracle Applications” on page 24
- “Delete Tables and Databases” on page 24
- “Remove iOAAServices from Servers” on page 24

Uninstall Oracle Applications

Use the normal methods for uninstalling any Component Broker application. See the Quick Beginnings for Component Broker for details on uninstalling Component Broker applications.

Delete Tables and Databases

If the tables and databases used by the applications that are being uninstalled are not needed by other applications, you may want to delete them. Perform appropriate backup procedures as you normally would for any important data that you keep in Oracle.

Remove iOAAServices from Servers

If after removing the applications from a server or server group there are no more applications configured on the server or server group that use Oracle, the iOAAServices application can be removed from the server or server group configuration. This can improve the performance of DB2 applications if installed on the same server.

If after removing iOAAServices from server or server groups within a configuration, if there are no more applications in that configuration that need Oracle, iOAAServices can be deleted from the configuration. Do not delete iOAAServices from the set of Host applications.

Uninstalling the Oracle Application Adaptor

1. Go through the steps to “Uninstalling Component Broker Applications That Use Oracle”
2. “Uninstall the Oracle Application Adaptor”
3. Uninstall Component Broker server run time if no longer needed
4. Uninstall Oracle if no longer needed

Uninstall the Oracle Application Adaptor

1. Login as root.
2. At a command prompt, enter:

   ```
   smitty remove
   ```

   The **Remove Installed Software** screen is displayed.
3. On the **Remove Installed Software** screen:

   a. In the SOFTWARE name field, type CBOAA to remove the run time for Oracle Application Adaptor.

   b. In the **PREVIEW only** field, change to '**n**'. If the field is not set to '**n**', use the Tab key to toggle the value.

   c. In the **REMOVE dependent software** field, change to '**y**'.

   d. Press the **Enter** key to remove Application Adaptor for Oracle for AIX from your system. The installation process begins. After the files are removed, the command status, as indicated in the upper left corner of the screen should state **OK**. When you receive this **OK** status, all files were removed. Press the **F10** key to exit smit.

4. Log out as root.

Environment Setup

Use the same user name to run Component Broker on AIX to run Component Broker Application Adaptor for Oracle on AIX. Ensure the steps listed in the section titled "*Configuring Your Component Broker User ID*" in the Chapter titled "*Installing Component Broker for AIX*" in the *Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide* to properly configure the environment.

Chapter 5. Developing a Component Broker Oracle Application

This chapter provides information for building a sample Component Broker application with an Oracle backend.

Note: To walk-through this sample, the following Component Broker software must be installed on your system:

- The CBToolkit and Samples
- The Component Broker Application Adaptor for Oracle run time
- A Component Broker server

Important Information: Before working through this sample, please refer to the Late Breaking News provided with Component Broker before performing the exercise in this chapter. The Late Breaking News provides the latest information about the Oracle application adaptor samples. These instructions may differ from those included in this document.

Oracle Sample Application

The Oracle sample application is similar to the Policy Application Sample which is part of the base CBToolkit installation.

Installing the Sample Application

The Oracle sample application is installed when you install the Component Broker Toolkit and Samples. The sample can be found in:

NT *CBroker*\samples\InstallVerification\OAA\.

AIX *CBroker*/samples/InstallVerification/OAA/.

There is a README file in that directory that reiterates the instructions for building and running the sample application.

Configuring Oracle for the Sample Application

Instructions for creating the OPolicy database and OPolicy table are in:

NT Chapter 3, “Installing the Oracle Application Adaptor on Windows NT” on page 5

AIX Chapter 4, “Installing the Oracle Application Adaptor on AIX” on page 15

Developing the Sample Application

The following sections walk you through the development of the Oracle Sample Application:

- “Creating the Object” on page 28
- “Building the Test Application” on page 32
- “Creating a Table” on page 33
- “Java Client” on page 37

The sample application supplied with the Oracle application adaptor is located in the:

NT *CBroker*\samples\InstallVerification\OAA\BusinessObjects\OPolicy

AIX *CBroker*/samples/InstallVerification/OAA/BusinessObjects/OPolicy

directory. You can use Object Builder to view the model, then build your own object using the following procedures.

Note: When viewing the sample application, do **not** make any changes.

If you do not want to view the model using object builder, but want to immediately generate, build, and run the sample application, skip the “Creating the Object” section and proceed to the “Building the Test Application” section.

If you do not wish to view the model using object builder, but instead want to immediately generate, build and run the sample application, skip the section on Creating the Object and instead go to the section Building the Test Application.

Creating the Object

The procedure for building a CBroker object to access an Oracle database follows. The names used in the following procedure are from the sample, but are for illustrative purposes only. You will use your own names for the objects you create. When you have completed the procedure, you can compare the contents of your directory with the contents of the sample application directory.

1. Start Object Builder.

   NT From the Windows NT **Start** menu, select **Programs** → **IBM Component Broker for Windows NT** → **Object Builder**.

   AIX To start Object Builder from the command line, enter `ob`.

2. Specify a project directory:

   a. Enter the name of the directory in which you intend to keep the object model and code. For example, you could specify:

      NT *CBroker*\oaaNewObject

      AIX *CBroker*/oaaNewObject

      Note: Do **not** specify the name of the directory in which the sample application is located to avoid overwriting the files.

   b. Click the **Finish** button.

3. Define the business object file:

   a. Right-click on “User-Defined Business Objects” to display its pop-up menu.

   b. On the pop-up menu, click **Add File...**.

   c. Type `OPolicy` as the name of the file to contain the business objects. (For the sample application, the file name is OPolicy. You can use the same name only because you are building the object in a different directory.)

   d. Click the **Finish** button.

4. Define the Business Object interface:

   a. Expand “User-Defined Business Objects.” Right-click on the file name you specified to display its pop-up menu. (For the sample application, the file name is OPolicy.)

   b. On the pop-up menu, click **Add Interface...**.

   c. In the dialog:
      1) Type `OPolicy` for the interface name.
      2) Select **The interface is queryable**. This must be selected for all Oracle backed objects.
      3) Click the **Next** button until the Attributes window displays.
   d. In the Attributes window, define the user-defined attributes:
      1) Click the **Add Another** button.
      2) Type `policyNo` as the name of the attribute in the **Attribute Name** field, and ensure that `long` is selected as the attribute type.
      3) Click the **Add Another** button. Your attribute is added to the tree, and the fields are filled with default values.
      4) Type `amount` as the name of the attribute in the **Attribute Name** field, and ensure that `float` is selected as the attribute type.
      5) Click the **Add Another** button. Your attribute is added to the tree, and the fields are filled with default values.
      6) Type `premium` as the name of the attribute in the **Attribute Name** field, and ensure that `float` is selected as the attribute type.
      7) Press the **Next** button to continue.
   e. In the Methods window, define the user-defined methods:
      1) Click the **Add Another**.
      2) Type `addBeneficiary` in the **Method Name** field, and ensure that `void` is selected as the return type. The case of the attribute name is important. Be sure to enter the attribute name exactly as shown.
      3) Click the **Add Another** button. The `addBeneficiary` method is added to the tree, and the fields are filled with default values.
      4) Type `delBeneficiary` in the **Method Name** field, and ensure that `void` is selected as the return type. The case of the attribute name is important. Be sure to enter the attribute name exactly as shown.
      5) Click the **Finish** button.

5. Define a key object:
   a. Right-click the interface name to display its pop-up menu.
   b. On the pop-up menu, click **Add Key...**.
   c. Select `policyNo` as the attribute that represents the key field of the database.
   d. Click the **Finish** button.
6. Define a copy helper object:
   a. Right-click the interface name to display its pop-up menu.
   b. On the pop-up menu, click **Add Copy Helper...**.
   c. Select the attributes that you want the copy helper to know about. For the sample application, all the attributes are selected.
   d. Click the **Finish** button.
7. Define a business object implementation:
   a. Right-click the interface name to display its pop-up menu.

    b. On the pop-up menu, click **Add Implementation...**.

    c. Click the **Next** button until the Data Object Interface window displays.

    d. In the Data Object Interface window, select the attributes you want in the data object. For the sample application, all the attributes are selected.

    e. Click the **Finish** button.

8. Define a managed object:

    a. Right-click the business object name to display its pop-up menu. For the sample application, the business object name is OPolicyBO.

    b. On the pop-up menu, click **Add Managed Object...**.

    c. Click the **Finish** button.

9. Define a Data Object Interface:

    a. Right-click the Data Object interface to display its pop-up menu.

    b. On the pop-up menu, click **Add Implementation...**.

    c. In the dialog, select **BOIM with any key** and **Oracle Caching services**.

    d. Click the **Finish** button.

    and

10. Define a persistent object and schema:

    a. Right-click the Direct Object Implementation to display its pop-up menu.

    b. On the pop-up menu, click **Add Persistent Object and Schema**.

    c. In the dialog, type `OPolicy` as the group name and as the database name.

    d. Click the **Next** button.

    e. Expand the policyNo attribute and select the OPolicyPO.policyNo mapping attribute. Under the **Schema Column**, change the **SQL Type** to `NUMBER` and change the **length** to `8`.

    f. Expand the amount attribute and select the OPolicyPO.amount mapping attribute. Under the **Schema Column**, change the **SQL Type** to `NUMBER` and change the **length** to `0`.

    g. Expand the premium attribute and select the OPolicyPO.premium mapping attribute. Under the **Schema Column** change the **SQL Type** to `NUMBER` and change the **length** to `0`.

    h. Click the **Finish** button.

11. Generate the code:

    a. Right-click the **OPolicy File** to display its pop-up menu.

    b. On the pop-up menu, click **Generate** → **All**. This generates the code for the object.

12. Create the client DLL file:

    a. Right-click on Build Configuration to display its pop-up menu.
    b. On the pop-up menu, click **Add Client DLL...**.
    c. In the dialog, type `OPolC.dll` as the name of the client DLL.
    d. Click the **Next** button.
    e. Select all the items for **Client Source Files**.
    f. Click the **Next** button.
    g. Select all the items for **Libraries to Link With**.
    h. Click the **Finish** button.

13. Create the Server DLL file:

   a. Right click on Build Configuration to display its pop-up menu.
   b. On the pop-up menu, click **Add Server DLL...**.
   c. In the dialog, type `OPols.dll` as the name of the server DLL.
   d. Click the **Next** button.
   e. Select all the items for **Server Source Files**.
   f. Click the **Next** button.
   g. Select all the items for **Libraries to Link With**.
   h. Click the **Finish** button.

14. Build the DLL files:

   a. Right-click on Build Configuration to display its pop-up menu.

   b. On the pop-up menu, click **Generate** → **All** → **C++ Default Targets**. This generates the make files to build the object.

15. Define a container instance:

   a. Right-click on Container Definition to display its pop-up menu.

   b. On the pop-up menu, click **Add Container Instance...**.

   c. In the dialog, type `OPolicyContainer` as the name of the container.

   d. Click the **Next** button until you get to the Services window.

   e. In the Services window, select **Use RDB Transaction Services**.

   f. Click the **Next** button.

   g. In the Services Details window, select **Throw an exception and abandon the call**.

   h. Click the **Next** button.

   i. In the Data Access Patterns window, select **Caching** for the Business Object and select **Caching services** for the Data Object.

   j. Click the **Finish** button.

16. Create the application family:

   a. Right-click Application Configuration to display its pop-up menu.
   b. On the pop-up menu, click **Add Application Family...**.
   c. In the dialog, type `OracleSampleApplication` as the application family name.
   d. Click the **Finish** button.

17. Define the application:

   a. Right-click the application family to display its pop-up menu.
   b. On the pop-up menu, click **Add Application...**.
   c. In the dialog, type `OPolicy` as the application name.
   d. Click the **Finish** button.

18. Define the managed object:

   a. Right-click the application to display its pop-up menu.

   b. On the pop-up menu, click **Add Managed Object...**.

   c. Choose the managed object you want added to the application. All the fields should be filled in.

   d. Click the **Next** button.

   e. In the Data Object Implementations window, click the **Add Another** button. The correct data object implementation should be preselected.

   f. Click the **Finish** button.

19. Generate the DDL files:
    a. Right-click on the application family to display its pop-up menu.
    b. On the pop-up menu, click **Generate**. This generates the DDL files needed to configure the application on a Component Broker Server.

Building the Test Application

For the sample, the model has already been defined and generated for an OPolicy object. The following instructions are also found in the README file provided with the sample application. To build the OPolicy object and the OPolicyApp.exe test program:

1. Go to the *Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide*, chapter“Installing Component Broker,” section “Information About Component Broker Samples,” subsection “Generating the Business Objects for the Samples” and follow the instructions for generating the OAA sample.
2. After generating the sample, you should still be in the directory:`NT...\samples\InstallVerification\OAAAIX.../samples/InstallVerification/OAA`
3. Switch to the directory of the platform you generated the code for Windows NT.

   ```
   cd BusinessObjects\OPolicy\Working\NT
   ```

   or AIX

   ```
   cd BusinessObjects/OPolicy/Working/AIX
   ```

4. Compile the files. Enter:(for Windows NT)

   ```
   nmake -c -f all.mak
   ```

   or AIX

   ```
   make -c all.mak
   ```

5. Copy the DLL files for Windows NT. Enter:

   ```
   copy OPolC.dll CBroker\bin\
   copy OPolS.dll CBroker\bin\
   ```

   Copy the run time files for AIX. Enter:

   ```
   cp OPolC.a CBroker/bin/
   cp OPolC.so CBroker/bin/
   cp OPolS.a CBroker/bin/
   cp OPolS.so CBroker/bin/
   ```

6. Change directories to the samples directory. Enter:

NT	cd ..\..\..\..\Applications\OPolicyApp
AIX	cd ../../../../Applications/OPolicyApp

7. Compile the files for Windows NT. Enter:

   ```
   nmake -c -f OPolicyApp.mak
   ```

   Compile the files for AIX. Enter:

   ```
   make -f OPolicyApp.mak
   ```

8. Copy the compiled file. Enter:

NT	copy OPolicyApp.exe CBroker\bin\
AIX	cp OPolicyApp CBroker/bin/

Creating a Table

The following should be done under the user ID that will own the table.

1. Copy the SQL file to another directory. At a command prompt, enter:

   ```
   NT
   copy x:\CBroker\samples\InstallVerification\OAA\BusinessObjects\OPolicy\Working
         \NT\OPolicy.sql x:\CBroker\sample
   ```

   ```
   AIX
   cp CBroker/samples/InstallVerification/OAA/BusinessObjects/OPolicy/Working
         /AIX/OPolicy.sql /$HOME
   ```

2. Start a new session of SQL Plus 8.0.

   NT **Start** menu, click **Programs** → **Oracle for Windows NT** → **SQL Plus 8.0**

   AIX Open a session of sql plus 8.0 from a command window.

   ```
   splplus
   ```

3. Login to the SQL Plus 8.0 session and enter the following information:

   a. NT

      1) In the **User Name** field, type /.

      2) Leave the **Password** field blank.

      3) In the **Host String** field, type OPolicy.

         Note: You should use OPolicy when setting up the database for the sample application. This is the service name you defined in step 2.

      4) Click the **OK** button to login. SQLPlus connects to the target database as the current OS user.

   b. AIX

      1) Enter the following information:

         ```
         Enter user-name: /
         Enter password:
         ```

      2) Connect to OPolicy:

         ```
         connect @opolicy
         ```

4. Create any required tables in the user schema. For the Oracle sample application, create the OPolicy table in the OPolicy database:

   a.

      ```
      NT
      @x:\CBroker\samples\OPolicy.sql
      ```

      ```
      AIX
      @$HOME/OPolicy.sql
      ```

5. Exit the SQL Plus 8.0 session.

   ```
   quit
   ```

6. If the tables to be accessed from Component Broker are in a different user's schema, the appropriate privileges must be “GRANTED” to the Component Broker user.

Choice of user schema

The name of the table is used in Object Builder when defining the Persistent Object (PO). This name may need to be qualified by the name of the user schema in which it resides. The Component Broker service runs under the name of a particular user. For OS authentication, if the Component Broker service is running under the same user ID that was used to create the table, then the name of the user schema can be omitted from the table name. If the user is different, then the name of the user schema must be specified in the table name. For example,

```
CBroker service is running under userA
OClaim table was created by userA
```

then either OClaim or userA.OClaim can be used for the table name.

However if,

```
CBroker service is running under userB
OClaim table was created by userA
```

then userA.OClaim must be used for the table name.

From this it can be seen that specifying the user schema as part of the table name is always guaranteed to work, regardless of what authority CBroker service is running under. The user schema name can either be filled in for the Persistent Object using Object Builder or it can be updated after the application is installed by using system management to change the databaseTable field of the TableAlias for the application.

System Management

The Oracle Application Adaptor must be configured on all server groups that will be configured with Oracle applications. An Oracle application must be loaded on the host image and configured on each server group on which it is to be available. These procedures are presented in the following sections.

Configure the Oracle application adaptor on a Server Group

Note: Steps 2, 3, and 4 can be omitted for the "Sample Configuration".

This section assumes that configuration has been previously done using the instructions in IBM Component Broker Quick Beginnings and that “Sample Cell and Workgroup Zone” exists as a management zone, “Sample Configuration” exists as a configuration under “Sample Cell and Workgroup Zone,” “myservergroup” exists as a server group under “Sample Configuration,” and a server named “myserver” exists under the server group “myservergroup.”

To configure the Oracle application adaptor on a server or a server group, perform the following steps:

1. Expand Available Applications and select `iOAAServices`.
2. Open the pop-up menu for iOAAServices and select **Drag**.
3. Expand Management Zones → Sample Cell and Workgroup Zone → Configurations, and select Sample Configuration.
4. Open the pop-up menu for Sample Configuration and select **Add Application**.
5. Expand Sample Configuration → Applications, and select `iOAAServices`.
6. Open the pop-up menu for iOAAServices and select **Drag**.
7. Expand Server Groups, and select `myservergroup`.

8. Open the pop-up menu for myservergroup and select **Configure Application**.
9. Optional: To verify that the application has been configured on the server group:
   a. Expand Management Zones → Sample Cell and Workgroup Zone → Configurations → Sample Configuration → Server Groups → myservergroup → Configured Applications.
   b. Verify that iOAAServices is one of the configured applications listed under Configured Applications.
10. If necessary, repeat the previous steps to configure iDB2IMServices (expand Available Applications) on the same server group. To determine if iDB2IMServices is already configured, perform the optional procedure in step 9. If configured, iDB2IMServices is listed under Configured Applications.

Important: Every Oracle application server must have the iDB2IMServices configured on it, even if the server is not going to run any DB/2 applications.

Configure an Oracle Application on a Server Group

To configure the Oracle application on a server group, perform the following steps:

1. From the System Management User Interface, change to the Expert User mode.
2. Expand Host Images and select **myhost** (where *myhost* represents the name of your host computer).
3. Open the pop-up menu for **myhost** and select **Load Application**.
4. In the dialog:
   a. Click the **Browse** button and navigate to:

      ```
      NTCBroker\samples\InstallVerification\OAA\BusinessObjects\OPolicy\Working\NT
               \OracleSampleApplication

      AIX
      CBroker/samples/InstallVerification/OAA/BusinessObjects/OPolicy/Working/AIX
               /OracleSampleApplication
      ```

      (Where *CBroker* represents the directory where Component Broker is installed.)
   b. Select `OracleSampleApplication.ddl` as the name of the application.
   c. Click the **OK** button.
5. Repeat steps 3 and 4 to load the `SpecificOracleSampleApplication.ddl` application.
6. From the System Manager User Interface menu, select **Wizards**→ **Configure Servers** to open the Configure Server Smartguide window:
   a. In the Select Applications to Configure window:
      1) Under **Available Applications**, select `OPolicy` and click the **Add** button. Repeat the procedure for `Specific OPolicy`.
      2) Click the **Next** button to continue.
   b. In the Management Zone window:
      1) Under **Available Items**, select `Sample Cell and Workgroup Zone`.
      2) Click the **Next** button to continue.
   c. In the Configuration window:
      1) Select `Sample Configuration`.
      2) Click the **Next** button to continue.
   d. In the Select Servers to Configure Applications on window:

1) Select `myservergroup`.

2) Click the **Add** button.

3) Click the **Finish** button.

7. Verify that the application has been configured on the server group:

   a. Expand Management Zones → Sample Cell and Workgroup Zone → Configurations → Sample Configuration → Server Groups → myservergroup → Configured Applications.

   b. Verify that OPolicy and Specific OPolicy are listed under Configured Applications.

8. Activate the new configuration:

   a. Expand Management Zones → Sample Cell and Workgroup Zone → Configurations, and select `Sample Configuration`.

   b. Open the pop-up menu for Sample Configuration and select **Activate**.

Note: If all the Oracle based Component Broker applications are removed from a server or server group configuration, then you should remove iOAAServices from the server or server group configuration. If you remove all Component Broker Oracle based applications from a machine and remove the Oracle Application Adaptor, this procedure is not necessary. This procedure can be used to clean up a particular server or server group when Oracle based Component Broker applications are no longer configured on it. Assuming that all Oracle applications have been removed from myservergroup, perform the following steps:

1. Expand `Management Zones → Sample Cell and Workgroup Zone → Configurations → Sample Configuration → Server Groups → myservergroup → Configured Applications`.

2. Expand `Configured Applications`.

3. From the pop-up menu for iOAAApplications, select **Delete**.

4. Activate the new configuration:

   a. Expand `Management Zones → Sample Cell and Workgroup Zone → Configurations`.

   b. Select `Sample Configuration`.

   c. From the pop-up menu for `Sample Configuration`, select **Activate**.

Running the Application

If you configured the OPolicy application on myservergroup and it has a server configured for it called myserver, then run the application by entering:

```
OPolicyApp
```

If you configured the application on another server group, or if myservergroup has servers with a name other than myserver, you must specify the name of the server. For example, if you have a server group called testservergroup and there is a server under testservergroup called testserver, run the application by entering:

```
OPolicyApp commit testserver
```

You can view help for running OPolicyApp by entering:

```
OPolicyApp ?
```

Java Client

There are no special considerations for the Java Client usage of a managed object whose data resides in Oracle.

Appendix A. Notes and Restrictions on the Oracle Application Adaptor

The following sections discuss some things that you need to know about developing your own Component Broker Oracle application:

- “All Oracle Applications Must Utilize Caching”
- “No Support for Callbacks”
- “Query Pushdown”
- “Cache Service and Oracle Locking”
- “Type Mapping” on page 40
- “Limited Support for Suspending and Resuming Transactions” on page 42
- “Maximum Number of Databases for a Server” on page 42
- “Performance Considerations” on page 42

All Oracle Applications Must Utilize Caching

Component Broker Application Adaptor for Oracle does not support Embedded SQL in POs. All Oracle backed Data Objects must use the Caching Service.

Note: Object Builder does not allow any other selections if Oracle is chosen as the database type.

No Support for Callbacks

Callbacks or recursive calls (for example, server A invoking a method on an object in server B which calls a method on an object in Server A) are not supported when the Oracle application adaptor is installed. Such scenarios result in the transaction timing out.

Query Pushdown

In general, the rules for query pushdown are the same for the Oracle application adaptor as they are for the DB2 Application Adaptor. Given that the typesets are different, the specific instances where one can exploit query pushdown are different, although the premise remains the same.

Cache Service and Oracle Locking

When reading records from an Oracle database, the cache service obtains an exclusive lock when “lock confidence=pessimistic.” When “lock confidence=optimistic,” no lock is obtained on the data record. For data retrieved using an OO-SQL query, the record in the database is never locked. The setting of the “access” configuration attribute has no effect on the type of locking used for an Oracle database.

An example of how pessimistic locking can affect your proram is when, within a transaction, a findByPrimaryKeyString () call has returned a reference to an object. While this transaction is running, another transaction is started that does a findByPrimaryKeyString () using the same key to the same table. The second findByPrimaryKeyString () will wait until it can get the lock to the record in the database. One of three thing can happen:

1. The first transaction will complete before any timeouts occur, and the findByPrimaryKeyString () in the second transaction will get a reference to the object.

2. Oracle will time out on the attempt to get the lock, which results in a CORBA::PERSIST_STORE being returned.
3. The transaction will time out, which results in a CORBA::TRANSACTION_ROLLEDBACK.

An attempt to use an object reference for a record that is currently locked by another transaction results in a CORBA::INV_OBJREF.

For applications that use atomic transactions (i.e. use objects that are associated with a container that has the default Transaction Policy set to start a new transaction and complete it for each method), CORBA::TRANSACTION_ROLLEDBACK will be returned if, when the method is invoked, the object is locked by another transaction, and either the Oracle lock times out, or the transaction times out waiting to get the lock.

In any case where Oracle times out, there should be an appropriate error log from SQL showing that a time out has occurred.

Lock time outs and transaction time outs can be avoided by designing programs so that the transaction completes before either the Oracle lock time out occurs, or the transaction time out occurs. Where it is necessary to have long running transaction, either adjust the Oracle lock time out, the transaction time out, or both to be longer.

For more information, see the “Cache Service” topic in the *Component Broker Advanced Programming Guide*.

Type Mapping

When creating a PO and Schema top-down from an Oracle caching services Data Object Implementation, the Object Builder presents DB2 column types as the defaults instead of Oracle types.

To select appropriate column types, do the following:

Before finishing the "Add Persistent Object and Schema" Wizard, go to the "Attributes Mapping" Page. For each attribute, expand the attribute, then click on the item underneath the attribute. The mapping information will be shown. In the "schema column" for the mapping information, select the desired Oracle column type from the "SQL Type" combo-box and change the length and scale as needed.

Typical Oracle alternatives to DB2 types are:

DB2	Oracle
INTEGER	NUMBER, length 9, scale 0
SMALLINT	NUMBER, length 4, scale 0
DECIMAL(p,s)	NUMBER, length p, scale s

When creating an object that has an attribute whose type is that of an object, one of the choices for storing the reference to the object is to use a stringafied object reference. In the Persistent Object mapping, Object Builder selects VARCHAR2(2000) for the attribute type. This type cannot be remapped to the correct Oracle type using Object Builder. Go ahead and generate the SQL file. It will need to be modified to change the attribute type to either RAW(2000) or LONG RAW. For example, if an OClaim object is defined as:

```
OClaim.idl file contains:
interface OClaim;
interface OPolicy;
#include <IManagedClient.idl>
#include <OPolicy.idl>

interface OClaim : IManagedClient::IManageable         {
  readonly attribute long claimNo;
  attribute long amount;
  attribute ::OPolicy aPolicy;
}; // end interface OClaim
```

then Object Builder will generate the following SQL file:

```
CREATE TABLE  OClaim
(
   claimNo NUMBER(8,0) NOT NULL,
   amount  NUMBER(8,0),
   aPolicy  VARCHAR2(2000)   FOR BIT DATA,
   PRIMARY KEY
   ( claimNo )
);
```

The OClaim sql file needs to be modified to change the aPolicy definition to either:

```
 1. CREATE TABLE  OClaim
     (
      claimNo  NUMBER(8,0) NOT NULL,
      amount  NUMBER(8,0),
      aPolicy  RAW(2000),
      PRIMARY KEY
      ( claimNo )
     );


OR

2.   CREATE TABLE  OClaim
  (
   claimNo  NUMBER(8,0) NOT NULL,
   amount  NUMBER(8,0),
   aPolicy  LONG RAW,
   PRIMARY KEY
   ( claimNo )
  );
```

For more information about mapping Oracle data types to IDL data types, see Programs → IBM Component Broker → Component Broker Information Library → References → Programming Reference → OO-SQL → Constants → Character String Constants. There is a table called "Evaluation of ORACLE, DB2 MVS and DB2 CS data types in OOSQL queries. Also see the Programs → IBM Component Broker → Component Broker Information Library → Reference → Tools Reference → Object Builder → Attribute Mapping Properties → Oracle Data Type Mappings.

Limited Support for Suspending and Resuming Transactions

Due to differences in the implementation of the XA specification between Oracle and DB2, there are limitations in the Oracle application adaptor support for suspending and resuming transactions.

CosTransaction::Current::resume() and CosTransaction::Current::suspend() can be used in client code that is using a business object that uses an Oracle database as its backing store.

CosTransaction::Current::resume() and CosTransaction::Current::suspend() should not be used in business object code that uses an Oracle database as its backing store. This includes all code generated by Object Builder for a business object including the MO, DO, and PO code.

Maximum Number of Databases for a Server

At most, 32 Oracle databases can be configured to a given Component Broker server. This is because the maximum number of connections (xa_opens) supported by Oracle per thread is now 32.

Performance Considerations

Better performance for Business Objects that use DB2 as a backing store can be obtained by configuring DB2 based and Oracle based Business Objects on different servers and not configuring iOAAServices on those servers that are configured for Business Objects that use DB2 as a backing store.

Restriction on Use of Oracle Service Names

Normally, the configuration for an Oracle client would only have one service name set up for each database on each Oracle server that will be accessed from the same client. However, Oracle allows multiple service names to be set up for the same database on a server. If more than one service name is set up for a database on the same server, only one of the service names can be used for configuration purposes on a Component Broker server. The same service name does not need to be used for all Component Broker servers, a different one could be chosen for each server, just as long as only one is chosen for each server. For example, if you have defined the following service names in your TNSNAMES.ORA file:

```
ORACLEDB1.WORLD =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = hostOne)(PORT = 1521))
    (CONNECT_DATA = (SID = ORCL))
)

ORACLEDB2.WORLD =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = hostOne)(PORT = 1521))     (CONNECT_DATA = (SID = ORCL))
)

ORACLEDB3.WORLD =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = hostOne)(PORT = 1521))     (CONNECT_DATA = (SID = ORCL))
)
```

you could use ORACLEDB1 and ORACLEDB3 as database names defined for Component Broker server, but then not ORACLEDB2. ORACLEDB2 could be used as a database name on a different Component Broker server as long as ORACLEDB1 was not used.

Appendix B. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department LZKS
11400 Burnet Road
Austin, TX 78758
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

CICS
AIX
DB2
IBM
MVS/ESA
OS/2
OS/390
PowerPC
VisualAge

AFS and DFS are trademarks of Transarc Corporation in the United States, or other countries, or both.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Oracle and Oracle8 are registered trademarks of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Part Number: C092733

Printed in U.S.A.

C092733

GC09-2733-00