Oracle® Database Gateway for APPC

Installation and Configuration Guide

11*g* Release 2 (11.2) for Microsoft Windows

E12079-02

February 2010

ORACLE®

Oracle Database Gateway for APPC Installation and Configuration Guide, 11*g* Release 2 (11.2) for Microsoft Windows

E12079-02

Contents

4 Installing the Gateway

5 Configuring Your Oracle Network

6 Configuring the SNA Communication Package on Windows

7   Configuring the OLTP

8   Gateway Configuration Using SNA Communication Protocol

9   Gateway Configuration Using TCP/IP Communication Protocol

10   Security Requirements

11   Migration from Existing Gateways

A   Gateway Initialization Parameters for SNA Protocol

B   Gateway Initialization Parameters for TCP/IP Communication Protocol

C   Gateway Terminology

D   Configuration Worksheet

Index

Preface

The Oracle Database Gateway for APPC provides Oracle applications with seamless access to IBM mainframe data and services through Remote Procedure Call (RPC) processing.

Refer to the *Oracle Database Installation Guide* for Microsoft Windows and to the certification matrix on My Oracle Support for the most up-to-date list of certified hardware platforms and operating system versions. The My Oracle Support Web site can be found at:

http://metalink.oracle.com

Audience

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

- determining hardware and software requirements

- installing, configuring, or administering an Oracle Database Gateway for APPC

- developing applications that access remote host databases through the gateway using either the SNA communication protocol or the TCP/IP for IMS Connect communication protocol.

- determining security requirements

- determining and resolving problems

Before using this guide to administer the gateway, you should understand the fundamentals of the Windows operating system, SNA Server, the Database Gateways, PL/SQL and the Oracle database .

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at http://www.fcc.gov/cgb/consumerfacts/trs.html, and a list of phone numbers is available at http://www.fcc.gov/cgb/dro/trsphonebk.html.

Related Documents

The Oracle Database Gateway for APPC Installation and Configuration Guide for Microsoft Windows is included as a part of your product shipment. Also included is:

- *Oracle Database Gateway for APPC User's Guide*

You might also need Oracle database and Oracle Net documentation. The following is a useful list of the Oracle publications that may be referenced in this book:

- *Oracle Database Installation Guide for Microsoft Windows*
- *Oracle Database Administrator's Guide*
- *Oracle Database Advanced Application Developer's Guide*
- *Oracle Database Concepts*
- *Oracle Database Error Messages*
- *Oracle Database Net Services Administrator's Guide*
- *Oracle Database PL/SQL Language Reference*
- *Oracle Call Interface Programmer's Guide*

In addition to the Oracle documentation, ensure that you have required documentation for your platform, for your operating system, and for your communications packages.

The following IBM documentation may be useful:

- *IMS Connect Guide and Reference*

The title of the IBM publication was accurate at the time of the publication of this guide. Titles and structure of these documents are subject to change. For other operating system, SNA communication package, and TCP/IP package references, refer to the appropriate vendor documentation for your system.

Conventions

Examples of input and output for the gateway and Oracle environment are shown in a special font:

```
C:\> mkdir \ORACLE\your_name
```

All output is shown as it appears. For input, the list of conventions and their meanings are as follows:

- `example text`:   Words or phrases, such as `mkdir` and `ORACLE`, must be entered exactly as spelled and in the letter case shown. In this example, `mkdir` must be entered in lowercase letters and `ORACLE` in uppercase letters.

- *italic text*:   Italicized uppercase or lowercase, such as *your_name*, indicates that you must substitute a word or phrase, such as the actual directory name.

- **BOLD text** or ***bold italic TEXT***:   Bold words or phrases refer to a file or directory structure, such as a directory, path, or file ID.

- ... :   Ellipses indicate that the preceding item can be repeated. You can enter an arbitrary number of similar items.

- { }:   Curly braces indicate that one of the enclosed arguments is required. Do not enter the braces themselves.

- |:   Vertical lines separate choices.

- [ ]:   Square brackets enclose optional clauses from which you can choose one or none. Do not enter the brackets themselves.

Other punctuation, such as commas, quotation marks or the pipe symbol (|) must be entered as shown unless otherwise specified. Directory names, file IDs and so on appear in the required letter case in examples. The same convention is used when these names appear in text, and the names are highlighted in **bold**. The use of *italics* indicates that those portions of a file ID that appear in *italics* can vary.

Gateway commands, file IDs reserved words, MS-DOS commands, keywords and environment variables appear in uppercase in examples and text. Reserved words must always be entered as shown; they have reserved meanings within the Oracle system.

SQL*Plus Prompts

The SQL*Plus prompt, `SQL>`, appears in SQL statement and SQL*Plus command examples. Enter your response at the prompt. Do not enter the text of the prompt, `SQL>`, in your response.

MS-DOS Prompts

The MS-DOS prompt, `C:\>`, appears in MS-DOS command examples. Enter your response at the prompt. Do not enter the text of the prompt, `C:\>`, in your response.

PGAU Prompts

The PGAU prompt, `PGAU>`, appears in PGAU command examples. Enter your response at the prompt. Do not enter the text of the prompt, `PGAU>`, in your response.

Directory Names

Throughout this document, there are references to the directories in which product-related files reside. `%ORACLE_HOME%` is used to represent the Oracle home directory. This is the default location for Oracle products. If you have installed into a

location other than `%ORACLE_HOME%`, replace all references to `%ORACLE_HOME%` with the drive and path specification you have used.

1

Introduction to Oracle Database Gateway for APPC

The Oracle Database Gateway for APPC (the "gateway") enables users to initiate transaction program execution on remote online transaction processors (OLTPs.) The Oracle Database Gateway for APPC can establish connection with OLTP using the SNA communication protocol. The gateway can also use TCP/IP for IMS Connect to establish communication with the OLTP through TCP/IP. The gateway provides Oracle applications with seamless access to IBM mainframe data and services through Remote Procedural Call (RPC) processing. The gateway can access any application capable of using the CPI-C API either directly or through a TP monitor such as CICS.

This chapter discusses the architecture, uses, and features of the gateway. It contains the following sections:

- Overview of the Gateway

- Features of the Gateway

- Terms

- Architecture of the Gateway

- Implementation of the Gateway

- Communication with the Gateway

- Remote Procedural Call Functions

- Transaction Types for Gateway Using SNA

- Transaction Types for Gateway Using TCP/IP

Overview of the Gateway

The Oracle Database Gateway for APPC extends the Remote Procedural Call (RPC) facilities available with the Oracle database . The gateway enables any client application to use PL/SQL to request execution of a remote transaction program (RTP) residing on a host. The gateway provides RPC processing to systems using the SNA APPC (Advanced Program-to-Program Communication) protocol and to systems using TCP/IP for IMS Connect protocol. This architecture allows efficient access to data and transactions available on the IBM mainframe and IMS, respectively.

The gateway requires no Oracle software on the remote host system. Thus, the gateway uses existing transactions with little or no programming effort on the remote host.

For gateways using SNA:

The use of a generic and standard protocol, APPC, allows the gateway to access a multitude of systems. The gateway can communicate with virtually any APPC-enabled system, including IBM Corporation's CICS on any platform and IBM Corporation's IMS and APPC/MVS. These transaction monitors provide access to a broad range of systems, allowing the gateway to access many datastores, including VSAM, DB2 (static SQL), IMS, and others.

The gateway can access any application capable of using the CPI-C API either directly or through a TP monitor such as CICS.

Features of the Gateway

The Oracle Database Gateway for APPC provides the following benefits:

- TCP/IP support for IMS Connect

   This release of the gateway includes TCP/IP support for IMS Connect, giving users a choice of whether to use the SNA or TCP/IP communication protocol. IMS Connect is an IBM product which allows TCP/IP clients to trigger execution of IMS transactions. The gateway can use a TCP/IP communication protocol to access IMS Connect, which triggers execution of IMS transactions. If you choose to use TCP/IP, then there is no SNA involvement with this configuration. Related to this new feature of the gateway is:

   - The `pg4tcpmap` tool. This release of the gateway includes a new tool whose purpose is to map the information from your side profile name to TCP/IP and IMS Connect. For more information about the gateway mapping tool, refer to Chapter 6, of the *Oracle Database Gateway for APPC User's Guide*, and to Chapter 9, "Gateway Configuration Using TCP/IP Communication Protocol" in this guide.

- Fast interface

   The gateway is optimized so that remote execution of a program is achieved with minimum network traffic. The interface to the gateway is an optimized PL/SQL stored procedure specification (called the TIP or transaction interface package) precompiled in the Oracle database . Because there are no additional software layers on the remote host system, overhead occurs only when the program processes.

- Location transparency

   Client applications need not be operating system-specific. For example, your application can call a program on a CICS Transaction Server for z/OS. If you move the program to a CICS region on pSeries, then you need not change the application.

- Application transparency

   Users calling applications that execute a remote transaction program are unaware that a request is sent to a host.

- Flexible interface

   You can use the gateway to interface with existing procedural logic or to integrate new procedural logic into an Oracle database environment.

- Oracle database integration

The integration of the Oracle database with the gateway enables the gateway to benefit from existing and future Oracle features. For example, the gateway can be called from an Oracle stored procedure or database trigger.

- Transactional support

  The gateway and the Oracle database allow remote transfer updates and Oracle database updates to be performed in a coordinated fashion.

- Wide selection of tools

  The gateway supports any tool or application that supports PL/SQL.

- PL/SQL code generator

  The Oracle Database Gateway for APPC provides a powerful development environment, including:

  - a data dictionary to store information relevant to the remote transaction

  - a tool to generate the PL/SQL Transaction Interface Package, or TIP

  - a report utility to view the information stored in the gateway dictionary

  - a complete set of tracing and debugging facilities

  - a wide set of samples to demonstrate the use of the product against datastores such as DB2, IMS, and CICS.

- Site autonomy and security

  The gateway provides site autonomy, allowing you to do such things as authenticate users. It also provides role-based security compatible with any security package running on the mainframe computer.

- Automatic conversion

  Through the TIP, the following conversions are performed:

  - ASCII to and from EBCDIC

  - remote transaction program datatypes to and from PL/SQL datatypes

  - national language support for many languages

Terms

The following terms and definitions are used throughout this guide. Refer to Appendix C, "Gateway Terminology" for a more elaborate list of terms and definitions pertaining to the gateway, its components and functions.

Oracle Database

This is any Oracle database instance that communicates with the gateway for purposes of performing remote procedural calls to execute remote transaction programs (RTP). The Oracle database can be on the same system as the gateway or on a different system. If it is on a different system, then Oracle Net is required on both systems. Refer to Figure 1–2, "Gateway Architecture" for a view of the gateway architecture.

OLTP

Online Transaction Processor (OLTP) is any of a number of online transaction processors available from other vendors, including CICS Transaction Server for z/OS, IMS/TM, and z/OS

PGAU

Procedural Gateway Administration Utility (PGAU) is the tool that is used to define and generate PL/SQL Transaction Interface Packages (TIPs.) Refer to Chapter 2, "Procedural Gateway Administration Utility" in the *Oracle Database Gateway for APPC User's Guide* for more information about PGAU.

PG DD

Procedural Gateway Data Dictionary (PG DD) is a repository of remote host transaction (RHT) definitions and data definitions. PGAU accesses definitions in the Data Dictionary (PG DD) when generating TIPs. The PG DD has datatype dependencies because it supports the PGAU and is not intended to be directly accessed by the customer. Refer to Appendix A, "Database Gateway for APPC Data Dictionary" in the *Oracle Database Gateway for APPC User's Guide* for a list of PG DD tables.

RPC

Remote Procedural Call (RPC) is a programming call that executes program logic on one system in response to a request from another system. Refer to "Gateway Term Definitions" on page C-1 for more information, and refer to Appendix C, "Gateway RPC Interface" in the *Oracle Database Gateway for APPC User's Guide* as well.

RTP

A remote transaction program (RTP) is a customer-written transaction, running under the control of an OLTP, which the user invokes remotely using a PL/SQL procedure. To execute a RTP through the gateway, you must use RPC to execute a PL/SQL program to call the gateway functions.

TIP

A Transaction Interface Package (TIP) is an Oracle PL/SQL package that exists between your application and the remote transaction program. TIP is a set of PL/SQL stored procedures that invoke the remote transaction program through the gateway. TIPs perform the conversion and reformatting of remote host data using PL/SQL and `UTL_RAW` or `UTL_PG` functions.

Figure 1–1, "Relationship of Gateway and Oracle Database on Windows" illustrates where the terminology discussed in the preceding sections apply within the gateway's architecture.

Figure 1–1   Relationship of Gateway and Oracle Database on Windows
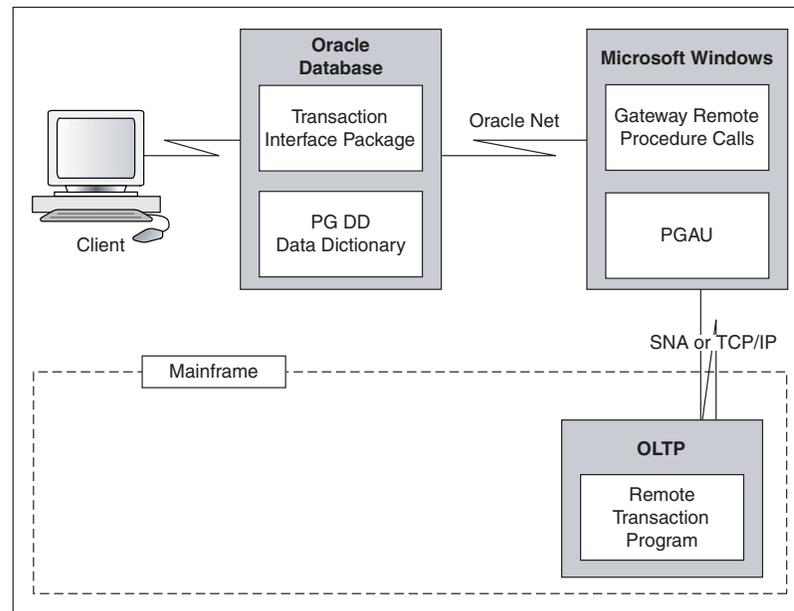
TCP/IP

****

Architecture of the Gateway

The architecture of Oracle Database Gateway for APPC consists of several components:

1. Oracle database

2. The gateway

   Oracle Database Gateway for APPC must be installed on a server that can run the required version of the operating system.

3. An OLTP

   The OLTP must be accessible from the gateway using the SNA or TCP/IP communication protocol. Multiple Oracle database s can access the same gateway. A single system gateway installation can be configured to access more than one OLTP.

   - **For a** gateway **using TCP/IP support for IMS Connect:** The only OLTP that is supported through TCP/IP is IMS through IMS Connect.

   The OLTP must be accessible to the system using the TCP/IP protocol. Multiple Oracle database s can access the same gateway. A single-system gateway installation can be configured to access more than one OLTP. Multiple IMS can be accessed from an IMS Connect. If you have a number of IMS Connect systems available, then any of these may be connected to one or more IMS systems.

Figure 1–2 illustrates the architecture of Oracle Database Gateway for APPC using either SNA or TCP/IP, as described in the preceding section.

Figure 1–2   Gateway Architecture
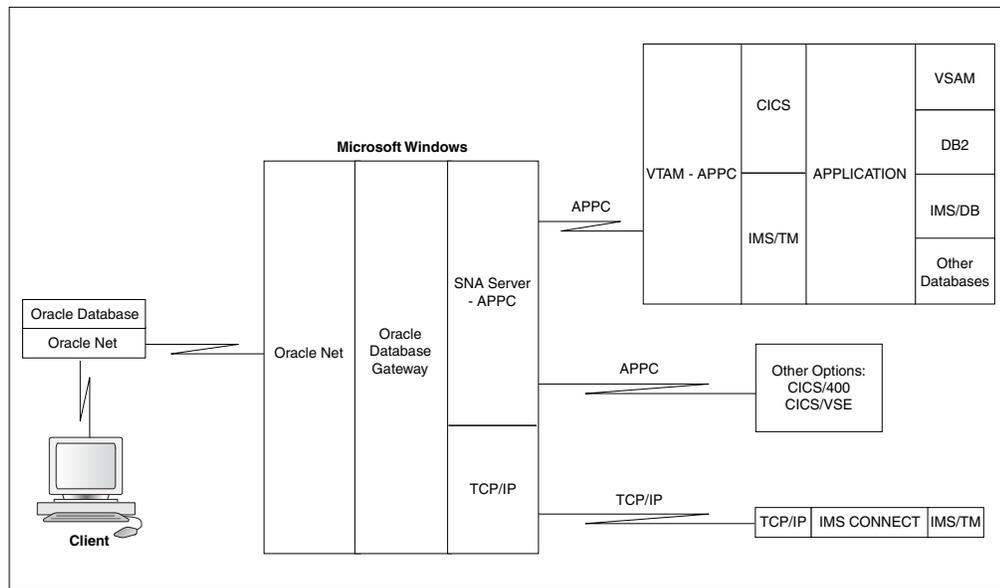
Implementation of the Gateway

The basic structure of the gateway is the same whether your communications protocol is SNA or TCP/IP support for IMS Connect. The gateway has some of the same components as an Oracle database instance on Microsoft Windows. It has the following components:

- a home directory, similar to the one associated with an Oracle instance's *ORACLE_HOME* environment variable;

- a system identifier, identified as *sid* or *ORACLE_SID*;

- an initialization parameter file, similar to the Oracle database 's init*sid*.ora file.

The gateway does not have:

- control, redo log, or database files;

- the full set of subdirectories and ancillary files associated with an installed Oracle database .

Because the gateway has no background processes and does not need a management utility such as Oracle Enterprise Manager, you do not need to start the gateway product. Each Oracle database user session that accesses a particular gateway creates an independent process on Windows, which, in turn, runs the gateway server and executes either the SNA or TCP/IP functions to communicate with an OLTP.

Communication with the Gateway

All of the communication between the user or client program and the gateway is handled through a TIP which executes on an Oracle database . The TIP is a standard PL/SQL package that provides the following functions:

- declares the PL/SQL variables that can be exchanged with a RTP;

- calls the gateway packages that handle the communications for starting the conversation, exchanging data, and terminating the conversation;

- handles all datatype conversions between PL/SQL datatypes and the target program datatypes.

The PGAU, provided with the gateway, automatically generates the TIP specification.

The gateway is identified to the Oracle database using a database link. The database link is the same construct used to identify other Oracle databases. The functions in the gateway are referenced in PL/SQL as:

```
function_name@dblink_name
```

Remote Procedural Call Functions

The Oracle Database Gateway for APPC provides a set of functions that are invoked by the client through remote procedural call (RPC). These functions direct the gateway to initiate, transfer data with, and terminate RTPs running under an OLTP on another system.

Table 1–1 lists the RPC functions and the correlating commands that are invoked in the gateway and remote host.

Table 1–1   RPC Functions and Commands in the Gateway and Remote Host

Applications	Oracle TIP	Gateway	Remote Host
call tip_init	tip_init call pgainit@gateway	PGAINIT	Initiate program
call tip_main	tip_main call pgaxfer@gateway	PGAXFER	Exchange data
call tip_term	tip_term call pgaterm@gateway	PGATERM	Terminate program

Description of RPC Functions

The following sections describe how the RPC functions perform on gateways using SNA or TCP/IP communication protocols.

Remote Transaction Initiation

The TIP initiates a connection to the remote host system, using one of the gateway functions, PGAINIT.

When the communication protocol is SNA: PGAINIT provides, as input, the required SNA parameters to start a conversation with the target transaction program. These parameters are sent across the SNA network, which returns a conversation identifier to PGAINIT. Any future calls to the target program use the conversation identifier as an INPUT parameter.

When the communication protocol is TCP/IP: PGAINIT provides, as input, the required TCP/IP parameters. Use the pg4tcpmap tool to map the parameters. These parameters are sent across the TCP/IP network to start the conversation with the target transaction program, the TCP/IP network returns a socket file descriptor to PGAINIT. Future calls to the target program made by PGAXFER and PGATERM use the socket file descriptor as an input parameter.

Refer to Appendix B, "Gateway Initialization Parameters for TCP/IP Communication Protocol" in this guide, and to Chapter 6 in the *Oracle Database Gateway for APPC User's Guide*, for more information about the function and use of the `pg4tcpmap` tool.

Data Exchange

After the conversation is established, a database gateway function called `PGAXFER` can exchange data in the form of input and output variables. `PGAXFER` sends and receives buffers to and from the target transaction program. The gateway sees a buffer as only a RAW stream of bytes. The TIP that is residing in the Oracle database is responsible for converting the application's PL/SQL datatypes to RAW before sending the buffer to the gateway. It is also responsible for converting RAW to the PL/SQL datatypes before returning the results to the application.

Remote Transaction Termination

When communication with the remote program is complete, the gateway function `PGATERM` terminates the conversation between the gateway and the remote host.

When the communication protocol is SNA:, `PGATERM` uses the conversation identifier as an `INPUT` parameter to request conversation termination.

When the communication protocol is TCP/IP:, `PGATERM` uses the socket file descriptor for TCP/IP as an `INPUT` parameter to request conversation termination.

Transaction Types for Gateway Using SNA

The Oracle Database Gateway for APPC supports three types of transactions that read data from and write data to remote host systems:

- **one-shot**

  In a one-shot transaction, the application initializes the connection, exchanges data, and terminates the connection, all in a single call.

- **persistent**

  In a persistent transaction, multiple calls to exchange data with the remote transaction can be executed before terminating the conversation.

- **multiconversational**

  In a multiconversation transaction, the database gateway server can be used to exchange multiple records in one call to the RTP.

Refer to "Remote Host Transaction Types" in Chapter 4, "Client Application Development" of the *Oracle Database Gateway for APPC User's Guide* for more information about transaction types.

The following list demonstrates the power of the Oracle Database Gateway for APPC:

- You can initiate a CICS transaction on the mainframe to retrieve data from a VSAM file for a PC application.

- You can modify and monitor the operation of a remote process control computer.

- You can initiate an IMS/TM transaction that executes static SQL in DB2.

- You can initiate a CICS transaction that returns a large number of records in a single call.

Transaction Types for Gateway Using TCP/IP

The Oracle Database Gateway for APPC using TCP/IP for IMS Connect supports three types of transaction socket connections:

- **transaction socket**

  The socket connection lasts across a single transaction.

- **persistent socket**

  The socket connection lasts across multiple transactions.

- **nonpersistent socket**

  The socket connection lasts across a single exchange consisting of one input and one output.

> **Note:** Do not use the nonpersistent socket type if you plan on implementing conversational transactions because multiple connects and disconnects will occur.

Refer to the section about `pg4tcpmap` commands in Chapter 6 of the *Oracle Database Gateway for APPC User's Guide* and to Chapter 9, "Gateway Configuration Using TCP/IP Communication Protocol" in this guide for more information about how to enter these parameters.

You can initiate an IMS/TM transaction that executes static SQL in DB2, this illustrates the power of the Oracle Database Gateway for APPC feature supporting TCP/IP for IMS Connect.

2

Release Information

This chapter provides information about this release of the Oracle Database Gateway for APPC. It contains the following section:

- Product Set
- Changes and Enhancements
- Known Restrictions

Product Set

The following product components are included in the product installation media:

- Oracle Database Gateway for APPC, release 11.2
- Oracle Net, release 11.2

Changes and Enhancements

The following sections describe the changes and enhancements unique to the 11g release of the gateway:

- Gateway Password Encryption Tool
- Partial IPv6 Support

Gateway Password Encryption Tool

The Gateway Password Encryption tool (`g4drpwd`) has been replaced by a generic feature that is now part of Heterogeneous Services. Refer to Chapter 15, "Security Considerations" in Oracle Database Gateway Installation and Configuration Guide for Microsoft Windows for details.

Partial IPv6 Support

There is full IPv6 support between Oracle databases and the gateway but IPv6 is not yet supported between the gateway and IMS Connect.

Known Restrictions

The following sections list the known restrictions for the Oracle Database Gateway for APPC and PGAU.

Known Restrictions for the Gateway

The following restrictions are known to exist in this release of the Oracle Database Gateway for APPC.

Multibyte Character Sets Are Not Supported for Numeric Data and Clauses

The Oracle Database Gateway for APPC has supported multibyte character set data for COBOL `PIC G` datatypes since version 3.4. However, the non-numeric character data (such as $, (,), +, -,.) that is allowed in `DISPLAY` datatypes and `PIC 9` edit masks must still be specified in EBCDIC. The non-numeric character data is not subject to MBCS translation.

CICS Transactions Do Not Allow PF Key Emulation

When performing a CICS transaction using the Oracle Database Gateway for APPC, you cannot emulate CICS PF keys.

APPC PIP Data Is Not Supported

You cannot define and transmit APPC `PIP` data in this release of the Oracle Database Gateway for APPC.

Floating Point Data Type Conversion Is Not Supported

Conversion of the floating point datatype is not supported by the Oracle Database Gateway for APPC.

Transaction Programs Are Responsible for All Data Compression and Decompression

The Oracle Database Gateway for APPC does not provide exits for compression and decompression facilities. All data exchanged between the gateway and the transaction must be in uncompressed format.

IBM VS COBOL II Compiler Desupported

The IBM VS COBOL II compiler has been desupported. However, the string `"IBMVSCOBOLII"` is still used as the value of the compiler name parameter to represent any COBOL compiler you choose to use. The value `IBMVSCOBOLII` should still be used and does not create a dependency on any specific version of the compiler.

Known Restrictions for PGAU

Restrictions with PGAU COBOL COPY REPLACE

When COBOL input to the PGAU `DEFINE DATA` statement contains a `COPY REPLACE` clause, only the first replacement is made.

3

System Requirements

This chapter describes the system requirements of the gateway. It contains the following sections:

- Hardware Requirements
- Software Requirements

Hardware Requirements

The hardware requirements for this release of the gateway on Microsoft Windows are described in the following sections.

Processor Requirements

Refer to the *Oracle Database Installation Guide for Microsoft Windows* for Microsoft Windows and to the certification matrix on My Oracle Support for the most up-to-date list of certified hardware platforms and operating system version requirements to operate the gateway for your system. The My Oracle Support Web site can be found at:

http://metalink.oracle.com

The Oracle Database Gateway for APPC requires an Intel or 100% compatible personal computer based on a Pentium processor that can run Microsoft Windows 2000 and higher.

Memory Requirements

For most installations, Oracle recommends a minimum of 256 MB of real memory for the system running the Oracle Database Gateway for APPC.

Each concurrent use of the gateway requires a minimum of 10 MB.

The following factors affect the virtual memory requirements of the gateway server process:

- number of concurrent gateway connections opened by each user;
- number of data items being transferred between the gateway and the RTP;
- additional factors such as configured network buffer size;
- the Oracle Net protocol adapters that were included during the gateway installation.

Network Attachment Requirements

The gateway requires any network attachment supported by either the SNA Server for your platform or the TCP/IP Networking Facility for TCP/IP communication.

However, if you are using solely the new TCP/IP support for IMS Connect feature, you will not need an SNA package. The Windows operating system comes with TCP/IP installed, you will need to configure it.

Disk Space Requirements

The gateway disk space requirement is a minimum of 700 MB.

Software Requirements

The system software configuration described in these requirements is supported by Oracle, provided that the underlying system software products are supported by their respective vendors. Verify the latest support status with your system software vendors.

Operating System Requirements

Refer to the *Oracle Database Installation Guide for Microsoft Windows* and to the certification matrix on My Oracle Support for the most up-to-date list of certified operating system version requirements to operate the gateway for your Windows Intel system. The My Oracle Support Web site can be found at:

http://metalink.oracle.com

Communication Protocol Requirements

A Microsoft Host Integration Server 2000 is required, or IBM Communication Server V6.1.1 (or higher) for Windows. In this document, either of these communications servers will be referred to generically as SNA Server.

> **Note:** If you choose to use TCP/IP support for IMS Connect as your communications protocol, then you will not need to use an SNA Server. Your operating system comes with a TCP/IP protocol automatically installed. If you choose to use the TCP/IP protocol, you will need to configure it to work properly with the gateway then instructions for doing so are located in Chapter 9, "Gateway Configuration Using TCP/IP Communication Protocol".

Oracle Database Requirements

The Microsoft Windows platform requires that the Oracle database which is to act as the Oracle database be up to date with the latest patch set for supported Oracle database releases.

Oracle Networking Product Requirements

Oracle Net is automatically installed on the system where the Oracle database is installed and on the system where the gateway is installed. Refer to Chapter 5, "Configuring Your Oracle Network" in this guide for configuration information. Additionally, you may refer to the *Oracle Database Net Services Administrator's Guide*.

IBM Mainframe Requirements

In addition to the other software requirements of the gateway and your particular platform, the following list outlines other requirements necessary on the IBM mainframe:

OLTP for SNA

The OLTP must support mapped APPC conversations. If the OLTP transaction programs to be executed through the gateway perform database updates, then the APPC verbs `CONFIRM`, `CONFIRMED`, and `SEND_ERR` must be supported by the OLTP. These verbs implement APPC `SYNCLEVEL 1`.

All resources controlled by an OLTP that can be updated by transaction programs invoked through the gateway must be defined as recoverable resources to the OLTP and host system if `COMMIT/ROLLBACK` capability is required for those resources. For example, a VSAM file updated by a CICS transaction must be defined to CICS as a recoverable file for `COMMIT/ROLLBACK` to control the updates.

The gateway is compatible with all supported releases of SNA-enabled products such as CICS, IMS/TM, and z/OS.

Important: For a list of known restrictions, read the "Known Restrictions" section on page 2-1 before proceeding with installation of the gateway.

OLTP for TCP/IP

IMS/TM: Release 7.1 or later is required, as well as any APARs (patches) listed in the IBM *IMS Connect Guide and Reference*.

IMS Connect: Release 1.2 or higher is required.

4

Installing the Gateway

The following topics in this chapter describe how to install and configure the Oracle Database Gateway for APPC:

- Before You Begin
- Planning to Upgrade or Migrate the Gateway
- Performing Preinstallation Procedures
- Installing the Gateway Software
- Installation Steps
- Removing Your Oracle Database Gateway for APPC

Before You Begin

Configuring an online transaction processor to allow access by the gateway requires actions on the OLTP and on certain components of the host operating system. Although no Oracle software is installed on the host system, access to, and some knowledge of the host system and the OLTP are required. Although this chapter includes some information about host system and OLTP installation steps, you must ensure that you have the applicable OLTP and host system documentation available.

Some of the configuration actions on the OLTP might require you to restart the OLTP. In preparation for this, have your host system programmer or DBA review the instructions for your OLTP to allow for any necessary preparations.

To install and configure the gateway with a single Oracle database and a single OLTP, perform the procedures described in this chapter.

> **Note:**   **If your** gateway **uses the SNA communication protocol**, then you will follow the instructions for installation and configuration in this chapter, in Chapter 5, "Configuring Your Oracle Network" and in Chapter 8, "Gateway Configuration Using SNA Communication Protocol".
>
> **If your** gateway **uses the TCP/IP communication protocol**, then you will follow the instructions for installation and configuration in this chapter, in Chapter 5, "Configuring Your Oracle Network" and in Chapter 9, "Gateway Configuration Using TCP/IP Communication Protocol".

Planning to Upgrade or Migrate the Gateway

If you have an earlier version of the Oracle Database Gateway for APPC installed on your system, then you may be upgrading or migrating to the current release. A gateway upgrade represents a minor software upgrade within a release, (for example, moving from Release 9.0 to Release 9.2.0) a gateway migration represents a significant change from one version number to another, (for example, migrating from Release 9.0 to 10.2).

This section is only for customers who have a previous release of Oracle Database Gateway for APPC. If you have a previous gateway installation, then you will need to carry out specific tasks before you can install Release 11.2 of the Oracle Database Gateway for APPC.

- After reading this section, you will need to read Chapter 11, "Migration from Existing Gateways" to determine the specific actions you must take to prepare for upgrade or migration of your gateway. If you are migrating to Oracle Database Gateway for APPC Release 11.2 from version 4.01 or earlier, then you will find specific material related to migration of the gateway in Chapter 11, "Migration from Existing Gateways".

If you are installing for the first time, then begin with "Performing Preinstallation Procedures" on page 4-3.

Performing Pre-Upgrade Procedures

Perform the following steps to prepare for upgrading the Oracle Database Gateway for APPC to current versions:

1. Make backups of altered PGA shipped files.

2. Remove or rename any old gateway directories.

Upgrade and Migration Considerations

Upgrade considerations are as follows:

- PGAU control files from Gateway Release 8 or 9 are upward-compatible, and you do not need to change them.

- After upgrade, the PG DD contains all of its earlier entries without modification. New PGAU control information has been added along with some columns to support new features, but no customer entries are altered by the upgrade.

- All TIPs from Oracle Database Gateway for APPC Release 4.0.1 or earlier must be recompiled, due to changes in the following:
  - PL/SQL compatibility
  - gateway server RPC interface
  - UTL_PG interface

- **TCP/IP only:** If you have existing TIPs that were generated previously on a gateway using the SNA communication protocol and you want to use the new TCP/IP feature, then the TIPs will have to be regenerated by PGAU with mandatory NLS_LANGUAGE and Side Profile Settings. Specify the appropriate ASCII character set in the DEFINE TRANSACTION command.

  This is due to the fact that the gateway assumes that the appropriate user exit in IMS Connect is being used, which would translate between the appropriate ASCII and EBCDIC character sets.

> **Caution:** An upgraded PG Data Dictionary (PG DD) cannot be accessed by an earlier release of PGAU.

Restoration

If you want to restore a previous release of gateway, then you must restore the following components to their previous versions:

- PGAU
- PG DD
- gateway server

Performing Preinstallation Procedures

Before you install the gateway, perform the following preinstallation procedures:

- Ensure that your system meets all the hardware and software requirements specified in Chapter 3, "System Requirements".

- Ensure that your security requirements are met.

  Refer to Chapter 3, "System Requirements" for more information about the security requirements for connections and data access on your OLTP.

- Fill out the worksheet identifying unique parameter names needed to configure your system and your chosen communication protocol (either SNA or TCP/IP), which is located in Chapter D, "Configuration Worksheet".

- Decide on an SID (system identifier) for your gateway. This SID is used in Section , "Configuring the Gateway" on page 8-10.

  The SID must be unique and must not be used by any other gateway or Oracle database on the system.

- **SNA only**: Your SNA package must be installed and configured before you can proceed with installation of the gateway. Ensure that your system can communicate with the OLTP using the SNA Server required for your platform. Refer to Chapter 6, "Configuring the SNA Communication Package on Windows" for more information about setting up and configuring SNA for Windows.

- **TCP/IP only**: Your TCP/IP package must be installed and configured before you can proceed with installation of the gateway.

  Ensure that your system can communicate with the OLTP using the TCP/IP communication package for your platform.

If you need general information about installing Oracle products and using the Oracle Universal Installer, then refer to the *Oracle Database Installation Guide for Microsoft Windows*

Gateway Installation Methods

You can install the gateway in either of the following ways:

1. On the same system as the existing Oracle database but in a different directory.

   All tasks for this type of installation or upgrade are discussed in this section.

2. On a system different from a local Oracle database .

3. On the same system as the Oracle database , and in the same Oracle home directory. Note that in this case, the Oracle database and the gateway must be at the same release level.

Installing the Gateway Software

For general information about installing Oracle products and how to use the Oracle Universal Installer, refer to the *Oracle Database Installation Guide for Microsoft Windows* and perform all necessary tasks there first.

If your server release is different than your gateway release, then *do not* install the gateway in the same Oracle home directory as the Oracle database . This is required to isolate the gateway from the Oracle database upgrades that might cause incompatibilities if the gateway executables were relinked with later versions of the Oracle database libraries.

Installation Steps

If you wish to install the gateway in the same Oracle home as the Oracle database , then the release number of both products must be the same.

Step 1: Log On to Windows System

Logon to your Windows system as a member of the Administrators group. If you are not currently a DBA user, then contact your system administrator to create a DBA login user ID. Refer to the *Oracle Database Installation Guide for Microsoft Windows* for login information.

Step 2: Ensure Minimum Amount of Disk Space

If you are installing the gateway for the first time, then ensure that there is enough space on the disk where the gateway will reside, as specified in"Disk Space Requirements" on page 3-2.

Step 3: Stop All Oracle Services

Before beginning the gateway installation process, you must stop all Oracle services that are currently running. Follow these steps:

1. Click **Start**, then **Settings**, then **Control Panel**.

2. Select **Services**. A list of all Windows services appears.

3. Select an Oracle service (those services begin with Oracle).

4. Click **Stop**.

5. Continue to select and stop Oracle services until all active Oracle services are stopped.

Step 4: Insert the Gateway Product Installation Media

Verify that the drive is assigned to the logical drive you selected and that you can access files on the installation media.

> **Note:** The installation steps that follow assume that the installation media location is mapped to the D: drive.

The installation media package contains the Oracle Database Gateway for APPC and the Oracle Universal Installer.

Step 5: Start the Oracle Universal Installer

To start the Oracle Universal Installer, run `setup.exe`:

1. From the Start Menu, select **Run**.

2. Enter the path of the executable file name. For example:

   `D:\Disk1\setup.exe`

Step 6: Step Through the Oracle Universal Installer

> **Caution:** Oracle Universal Installer automatically installs the Oracle-supplied version of the Java Runtime Environment (JRE). This version is required to run the Oracle Universal Installer and several Oracle assistants. Do not modify the JRE except by using a patch provided by Oracle Support Services. The Oracle Universal Installer also installs JDK.

Oracle Universal Installer is a menu-driven utility that guides you through installing the gateway by prompting you with action items. The action items and the sequence in which they appear depend on your platform.

The following section describes how to use the Oracle Universal Installer to install the gateway on your platform.

Oracle Universal Installer on Windows Platforms

Use Table 4–1 as a guide to step through the Oracle Universal Installer. At each prompt from the Oracle Universal Installer, perform the actions described in the Response column of the table to install the gateway on your Windows platform.

Table 4–1   The Oracle Universal Installer: Steps for Installing the Gateway

Prompt	Response
Oracle Universal Installer: Welcome	Click **Next**
Oracle Universal Installer: Specify Home Details	a. Specify the name of the installation.
	Specify the full path where you want to install the product
	Click **Next**
Oracle Universal Installer: Available Product Components	**a.** Deselect the checked products
	b. Select **Oracle Database Gateway 11.2**, open up this row
	c. Select **Oracle Database Gateway for APPC 11.2**. (Note that you may also choose to select other products to install besides the Oracle Database Gateway for APPC).
	d. Click **Next**
Oracle Universal Installer: Network Software	Specify your network package and click **Next**
Oracle Universal Installer: Summary	Click **Install**

Table 4–1   (Cont.) The Oracle Universal Installer: Steps for Installing the Gateway

Prompt	Response
Oracle Net Configuration Assistance: Welcome	Click **Next**
Oracle Net Configuration Assistance: Listener Configuration, Listener Name	Specify the name of Listener you want to create and click **Next**
Oracle Net Configuration Assistance: Listener Configutration, Select Protocols	Select the protocols and click **Next**
Oracle Net Configuration Assistance: Listener Configuration, TCP/IP Protocol	Specify a port number and click **Next**
Oracle Net Configuration Assistance:Listener Configuration, More Listeners?	Click **No** and then click **Next**
Oracle Net Configuration Assistance: Listener Configuration Done	Click **Next**
Oracle net Configuration Assistance: Naming Methods Configuration	Click **No** and then click **Next**
Oracle Net Configuration Assistance:Done	Click **Yes**
Oracle Universal Installer: End of Installation	Click **Finish**
Exit	Click **Exit**

Your gateway is now installed.

When the Oracle Universal Installer confirms that the installation is complete, verify that the installation procedure was successful. To do this, read the contents of the installation log file, which is located in the `C:\Program Files\Oracle\Inventory\logs` directory.

The default file name is `InstallActionsYYYY-MM-DD_HH-mm-SS-AM/PM.log`, where:

> *YYYY* is year;
> *MM* is month
> *DD* is day
> *HH* is hour
> *mm* is minute
> *SS* is seconds
> *AM/PM* is daytime or evening

Each of these variables in the log file name represents the date and time the product was installed.

> **Attention:** Print the contents of the
> `%ORACLE_HOME%\dg4appc\doc\README.doc` file and read the
> entire document, it contains important information about the
> installation. After reading the `README.doc` file, proceed with the
> configuration of the gateway.

Removing Your Oracle Database Gateway for APPC

This chapter describes how to remove Oracle Database Gateway from an Oracle home
directory. It contains information about the following topics:

- About the Deinstallation Tool
- Removing Oracle Software

About the Deinstallation Tool

he Deinstallation Tool (`deinstall`) is available in the installation media before
installation, and is available in Oracle home directories after installation. It is located
in `ORACLE_HOME\deinstall`.

The `deinstall` command stops Oracle software, and removes Oracle software and
configuration files on the operating system.

The script uses the following syntax, where variable content is indicated by italics:

```
deinstall -home complete path of Oracle home [-silent] [-checkonly] [-local]
[-paramfile complete path of input parameter property file] [-params name1=value
name2=value . . .] [-o complete path of directory for saving files] [-help | -h]
```

The options are:

- `-silent`

  Use this flag to run the command in noninteractive mode. This option requires a
  properties file that contains the configuration values for the Oracle home that is
  being deinstalled or deconfigured.

  To create a properties file and provide the required parameters, see the template
  file `deinstall.rsp.tmpl`, located in the response folder. If you prefer, instead
  of using the template file, you can generate a properties file by using the
  `-checkonly` option to have deconfig discover information from the Oracle home
  that you want to deinstall and deconfigure. The tool will generate the properties
  file, which you can then use with the `-silent` option.

- `-checkonly`

  Use this flag to check the status of the Oracle software home configuration.
  Running the command with the `-checkonly` flag does not remove the Oracle
  configuration.

- `-local`

  Use this flag on a multinode environment to deconfigure Oracle software in a
  cluster.

  When you run deconfig with this flag, it deconfigures and deinstalls the Oracle
  software on the local node (the node where deconfig is run). On remote nodes, it
  deconfigures Oracle software, but does not deinstall the Oracle software.

- `-paramfile` *complete path of input parameter property file*

Use this flag to run deconfig with a parameter file in a location other than the default. When you use this flag, provide the complete path where the parameter file is located.

The default location of the parameter file depends on the location of deconfig:

- From the installation media or stage location: `ORACLE_HOME\response`

- From a unzipped archive file from OTN: `ziplocation\response`

- After installation from the installed Oracle home: `ORACLE_HOME\deinstall\response`

- `-params [name1=value name 2=value name3=value . . .]`

  Use this flag with a parameter file to override one or more values that you want to change in a parameter file you have already created.

- `-o complete path of directory for saving files`

  Use this flag to provide a path other than the default location where the properties file is saved. The default location is `\response\deinstall.rsp.tmpl`.

  The default location of the parameter file depends on the location of deconfig:

  - From the installation media or stage location before installation: `ORACLE_HOME\`

  - From an unzipped archive file from OTN: `\ziplocation\response\`

  - After installation from the installed Oracle home: `ORACLE_HOME/deinstall/response`

- `-help | -h`

  Use the help option (`-help` or `-h`) to obtain additional information about the optional flags

Removing Oracle Software

Complete the following procedure to remove Oracle software:

1. Log in as a member of the Administrators group.

2. Run the `deinstall` command, providing information about the Oracle System Identifier (SID), when prompted.

5

Configuring Your Oracle Network

Now that you have installed the gateway, this chapter will instruct you about how to configure the network.

It contains the following section:

- Configuring the Oracle Network for the Gateway

Configuring the Oracle Network for the Gateway

The instructions in this section will show you how to configure the network whether your gateway is using the SNA or TCP/IP communication protocol.

The gateway must be defined to the Oracle Net Listener, and a service name must be defined for accessing the gateway. To do this, perform the following steps:

1. Add an entry for the gateway to the listener.ora file:

    - **If you are using SNA**:

      ```
      (SID_DESC=
            (SID_NAME=PGA)
            (ORACLE_HOME= C:\oracle\pga\11.2)
            (PROGRAM=pg4asrv)
      )
      ```
      where: `C:\oracle\pga\11.2` is your gateway Oracle home, and `PGA` is the gateway SID name.

    - **Or, if you are using TCP/IP**:

      ```
      (SID_DESC=
            (SID_NAME=PGA)
            (ORACLE_HOME= C:\oracle\pga\11.2)
            (PROGRAM=pg4t4ic)
      )
      ```
      where: `C:\oracle\pga\11.2` is your gateway Oracle home, and `PGA` is the gateway SID name.

2. Add a service name for the gateway to the tnsnames.ora file on the system where your Oracle database is located. The service name is specified in the `USING` parameter of the database link defined for accessing the gateway from the Oracle database . For example, if you are using the IPC protocol adapter and your gateway *sid* is PGA, then add the following entry to tnsnames.ora:

   ```
   pgaipc=
          (DESCRIPTION =
                  (ADDRESS = (PROTOCOL = ipc) (KEY=key))
                  (CONNECT_DATA = (SID=PGA))
                  (HS=)
   ```

```
        )
```

In this example, `key` is the IPC key defined in the listener.ora file for the IPC protocol. You can use the IPC protocol only if the Oracle database and the gateway are on the same system.

If you are using the TCP/IP protocol adapter and if your gateway `sid` is PGA, then add the following entry to tnsnames.ora:

```
pgatcp=
        (DESCRIPTION =
                (ADDRESS = (PROTOCOL= TCP)(Host= gateway)(Port= port))
                  (CONNECT_DATA = (SID=PGA))
                  (HS=)
        )
```

In this example, `port` is the TCP port defined in the listener.ora file for the TCP protocol, and gateway is the TCP/IP host name of the system where the gateway is located.

> **Note:** Under the following circumstances:
>
> - If your gateway and Oracle database are not on the same system,
>
> - or, if the gateway and the Oracle database are on the same system but the Oracle database Listener is different than the gateway listener,
>
> then you must define the Oracle database to PGAU by adding a service name to tnsnames.ora on the system where your gateway resides. For example:
>
> ```
> ora_server =
>     (DESCRIPTION=
>       (ADDRESS =
>         (PROTOCOL= TCP)
>         (PORT= port)
>         (HOST= ora_srv)
>       )
>       (CONNECT_DATA= (SID= ora_server))
>     )
> ```
>
> In this example,
>
> - `port` is the TCP port defined in the Oracle database listener.ora for the TCP protocol;
>
> - `ora_srv` is the TCP/IP host name of the system where the Oracle database resides; and
>
> - `ora_server` is the SID of the Oracle database .

Make sure to start your defined listeners. Refer to the *Oracle Database Net Services Administrator's Guide* for more information about configuring the network.

To see a sample listener.ora file refer to "Sample listener.ora File for a Gateway Using SNA" and "Sample tnsnames.ora File for a Gateway Using SNA" in Appendix A (if your communication protocol is SNA).

To see a sample listener.ora file refer to"Sample listener.ora File for a Gateway Using TCP/IP" and "Sample tnsnames.ora File for a Gateway Using TCP/IP" in Appendix B (if your communication protocol is TCP/IP).

Proceed with Configuring Your Communication Package for the Gateway

- **If your communication protocol is SNA**, you must now configure the SNA Server profiles for APPC connections.

  Proceed to Chapter 6, "Configuring the SNA Communication Package on Windows"

- **If your communication protocol is TCP/IP**, proceed to Chapter 7, "Configuring the OLTP".

6

Configuring the SNA Communication Package on Windows

The Oracle Database Gateway for APPC uses the SNA Advanced Program to Program Communication (APPC/LU6.2) protocol to communicate with an OLTP.

APPC support on Microsoft Windows can be provided by using either of the following tools:

- Microsoft Host Integration Server 2000(HIS), or

- IBM Communications Server v6.1.1 (or higher).

Read this chapter to learn how to configure the SNA Server on a Windows system to run the Oracle Database Gateway for APPC, using either a Microsoft Host Integration Server or IBM Communications Server.

This chapter contains the following sections:

- Using SNA Security Validation

- Processing Inbound Connections

- Configuring Your Microsoft Host Integration Server

- Creating SNA Server Definitions on Microsoft Host Integration Server

- Configuring an IBM Communications Server

- Creating IBM Communications Server Definitions for the Gateway

- Testing the Connection

- Resume Configuration of the Gateway

Using SNA Security Validation

When a Remote Procedure Call (RPC) request to start a remote transaction program (RTP) is received by the gateway, the gateway attempts to start an APPC conversation with the OLTP. Before the conversation can begin, a session must start between the Windows Logical Unit (LU) and the OLTP LU.

SNA and its various access method implementations (including HIS and IBM Communication Server and VTAM) provide security validation at session initiation time, allowing each LU to authenticate its partner. This validation is carried out entirely by network software before the gateway and OLTP application programs begin their conversation and process conversation-level security data. If session-level security is used, then correct password information must be established in the Windows SNA Server definitions and in similar parameter structures in the OLTP to

be accessed. Refer to suitable communications software product documentation for detailed information about this subject.

Processing Inbound Connections

Many OLTPs provide options for manipulating the security conduct of an inbound (client) APPC session request. Refer to suitable OLTP documentation for detailed information about this topic.

Note that for CICS, one security option is not supported by the gateway:

- `ATTACHSEC=PERSISTENT`, specified on the CICS `CONNECTION` definition, requires capability that is not yet available in the gateway.

However, `ATTACHSEC=LOCAL`, `ATTACHSEC=IDENTIFY`, `ATTACHSEC=VERIFY`, and `ATTACHSEC=MIXIDPE` are fully supported by the gateway.

Configuring Your Microsoft Host Integration Server

The following sections discuss general information about the SNA Server and how to configure a Microsoft Host Integration Server.

> **Note:** If you are using the IBM Communication Server to configure your Windows system for the gateway, then proceed to Section , "Configuring an IBM Communications Server".

Independent Versus Dependent LUs

Oracle recommends independent LUs for the Oracle Database Gateway for APPC because they support multiple parallel sessions or conversations. This means that multiple Oracle client applications can be active simultaneously with the same OLTP through the independent LU.

Dependent LUs support only a single active session. The CP (Control Point for the Node, which is SNA Server for Windows in this case) queues additional conversation requests from the gateway server behind an already active conversation. In other words, conversations are single-threaded for dependent LUs.

If a dependent LU is correctly defined, then no alterations to the Oracle Database Gateway for APPC configuration are needed, nor should any changes be needed to the host transaction or how the OLTP is started.

The operational impact of dependent LUs is that the first client application can initiate a conversation through the database gateway with the OLTP. While that transaction is active (which could be seconds to minutes to hours, depending on how the client application and transaction are designed), any other client application initiating a conversation with the same OLTP instance appears to hang as it waits behind the previous conversation.

If a production application really uses only a single conversation or transaction at any one time, then there should be no impact.

However, additional concurrent conversations or transactions might be required for testing or for other application development. Each requires that additional dependent LUs be defined on the remote host, plus additional SNA Server configuration entries, which define the additional dependent LUs on the Windows system. The TIP that initiates the conversation must specify the different Partner LU through a different Side Information Profile or by overriding the LU name. Refer to PGAU `DEFINE`

TRANSACTION , SIDEPROFILE, and LUNAME parameters in Chapter 2, "Procedural Gateway Administration Utility," in the *Oracle Database Gateway for APPC User's Guide*.

Location of Sample SNA Server Definitions

The `%ORACLE_HOME%\dg4appc\sna` subdirectory contains a sample set of gateway SNA Server definitions created with the `SNACFG` command. The `snacfg.ctl` file contains sample definitions for SNA Server.

Before building the SNA Server definitions, examine the snacfg.ctl file to determine the definitions needed, their contents, and their inter-relationships. The file format is text-oriented, and each field of each definition is clearly labelled. You can print a copy of the file to use while working with your definitions in a Microsoft Host Integration Server (formerly SNA Server Manager) session.

HIS Definition Types

Several types of SNA Server definitions are relevant to gateway APPC/LU6.2 operation. Each definition can be created and edited using a corresponding Microsoft Host Integration Server (formerly SNA Server Manager) menu.

The definitions relevant to the gateway are presented here in hierarchical order. Those definition types that are lowest in the hierarchy are discussed first. This matches the logical sequence in which to create the definitions.

Refer to the Windows HIS online documentation for a complete discussion of HIS definitions. This section provides an overview of HIS definitions in relation to the Oracle Database Gateway for APPC.

Methods of Creating SNA Server Definitions for the Gateway

HIS definitions can be created and modified in two ways:

- You can create the definitions using menus in the Microsoft Host Integration Server (formerly SNA Server Manager).

  Using the Microsoft Host Integration Server is the recommended method for creating the definitions. You should be able to accept most of the defaults. The default values assigned to many of the fields in a new set of definitions are acceptable to the gateway.

  Step through the tasks for creating SNA Server definitions in Section , "Creating SNA Server Definitions on Microsoft Host Integration Server" on page 6-4.

- Alternatively, you can install the definitions directly on your system using the `SNACFG` command.

  For information on using the `SNACFG` command, refer to your vendor documentation.

  If you use the `SNACFG` command method, then you must use Microsoft Host Integration Server (formerly SNA Server Manager) to review and modify the installed definitions. Because of configuration and naming differences, it is unlikely that they will work without modification.

Maintenance of SNA definitions is normally done by a user with Administrator authority. The following information is intended for the person creating SNA definitions for the gateway. You should have some knowledge of SNA before reading the following sections.

Creating SNA Server Definitions on Microsoft Host Integration Server

This section describes the process of creating your SNA definitions in the Microsoft Host Integration Server (formerly called the SNA Server Manager). All of the tasks described in this section are performed from within the Microsoft Host Integration Server.

Server Selection

Select the appropriate folder to ensure that definitions created are for that server. When HIS Manager is started, a dialog box appears.

Select the Servers folder under your local system and select the local SNA Server. From a list of services for that server, select the SNA Service.

Link Service Definition

You must install and configure a link service for SNA Server to use the network adapter installed in your workstation, as follows:

1. From the **Insert** menu, select **Link Service**.

2. From the **Insert Link Service** dialog box, select the Link Service you want to use from the selection list and click **Add**. For example, select **DLC 802.2 Link Service** and click **Add**.

Now, the **Link Service Properties** dialog box is displayed. Note that the contents of this dialog box vary depending on which link service was selected. In this example, the **DLC 802.2 Link Service Properties** dialog box is used.

Select the suitable network adapter from the Adapter drop-down menu and click **OK**. In the **Insert Link Service** dialog box, click **Finish**. The system now updates your network bindings.

Connection Definition

You must create a connection definition to define the devices which HIS uses to perform SNA communication. From the **Insert** menu, select **Connection**. The **Connection Properties** dialog box appears.

Select the **General** tab. Enter a connection name. This is the name used by HIS to name the connection. This example names the connection TOKEN1. From the Link Service drop-down menu, select a link service for the connection. All other settings can be left set to their default values.

Select the **Address** tab in the **Connection Properties** box. Enter the **Remote Network Address** and the **Remote SAP Address**.

Now, select the **System Identification** tab. Under Local Node Name, enter the **Network Name**, **Control Point Name**, and **Local Node ID**.

Under Remote Node Name, enter the **Network Name**, **Control Point Name**, and optionally, the **Remote Node ID**. The XID Type should be set to Format 3.

Next, select the **DLC** tab in the **Connection Properties** box. In this example, the **802.2 DLC** (Token Ring) is being used. For the **802.2 DLC**, all of the defaults are usually acceptable. If you need to change any values, then do so now.

Now, all the connection properties are set. Click **OK** to continue.

Local LU Definition

You must create a local logical unit (LU) definition. The local LU definition describes the SNA LU through which the gateway communicates with OLTP systems.

From the **Insert** menu, select **Local APPC LU**. The **Local LU Properties** dialog box appears.

Select the **General** tab. Enter the **LU Alias**, **Network Name**, and **LU Name**. Ensure that the **APPC SyncPoint Support** box is not checked.

Select **Advanced** tab of the **Local APPC LU Properties** box. Check the Member of Default Outgoing Local APPC LU Pool check box. Set the LU 6.2 Type to Independent to allow parallel sessions.

Now, the Local LU properties are all set. Click **OK** to continue.

Mode Definition

This definition describes an SNA mode entry to be used when establishing sessions between LUs. The mode defined here must match a mode defined on the target system.

From the **Insert** menu, select **APPC Mode Definition**. The **APPC Mode Properties** dialog box appears.

Select the **General** tab. Enter the **Mode Name**. The mode name that you specify must be defined to the OLTP communications software. Choose the mode name in addition to other mode parameters after consulting the person responsible for configuring the OLTP communications software.

Next, select the **Limits** tab. Enter the **Parallel Session Limit**, **Minimum Contention Winner Limit**, **Partner Min Contention Winner Limit**, and **Automatic Activation Limit** values.

The Parallel Session limit determines the maximum number of concurrent conversations allowed between the gateway instance and the OLTP. This equates to the maximum number of concurrently active RTP calls through the gateway instance.

Now, select the **Characteristics** tab. Enter the **Pacing Send Count**, **Pacing Receive Count**, **Max Send RU Size**, and **Max Receive RU Size**. For optimal performance, check the **High Priority Mode** check box.

The pacing and RU size parameters are performance-related and should be tuned to suit your application. For most installations, the values set in the example are sufficient.

After you finish entering the values called for in this box, all the APPC mode properties will be set. Click **OK** at the bottom of the box to continue.

Remote LU Definition

This definition describes the SNA LU of the OLTP system with which the gateway communicates. You must create a remote LU definition for the remote OLTP system. Determine the link with which to associate the LU (in the example, it is TOKEN1). From the **Insert** menu, select **APPC Remote LU**. The **Remote APPC LU Properties** dialog box appears.

Select the **General** tab. Use the Connection drop-down menu to select the connection used to access this LU. Enter the **LU Alias**, **Network Name**, **LU Name**, and **Uninterpreted LU Name**. You should contact the person responsible for your SNA network to determine the correct LU and network names. Note that you can use the

LU Alias to define a name known only to SNA Server, and that name can remain the same even if the remote LU name changes. This helps to reduce the amount of maintenance required when network changes occur.

Now, select the **Options** tab. Check the **Supports Parallel Sessions** check box. Use the Implicit Incoming Mode drop-down menu to select the mode. Set any security options you need.

Once you have filled in the suitable values in the **Remote APPC LU Properties** box, the APPC LU properties will be set. Click **OK** to continue.

CPI-C Symbolic Destination Name

When the Local and Remote Partner definitions and Mode definitions have been created, you can create CPI-C Symbolic Destination Names, also called side information. The side information is used to identify target OLTP systems to be accessed through the gateway. From the **Insert** menu, select **APPC CPIC Symbolic Name**. The **CPIC Name Properties** dialog box appears.

Select the **General** tab. Enter a **Name** for the side information. From the Mode Name drop-down list, select the suitable mode.

Now, select the **Partner Information** tab. Select **Application TP** and enter the TP name. If you plan to define one CPI-C Symbolic Destination Name for accessing multiple transaction programs at an OLTP, then you can enter a dummy TP Name at this time. The TP Name is overridden by the gateway at run time.

Enter the Partner LU Name **Alias**.

Click **OK** to save the side information.

Configuration of the Microsoft Host Integration Server is now complete. Proceed to

Configuring an IBM Communications Server

If you are using IBM Communications Server as your SNA Server, then read the following sections to learn how to configure it to communicate with the Oracle Database Gateway for APPC.

Independent And Dependent LUs

Oracle recommends independent LUs for the Oracle Database Gateway for APPC because they support multiple parallel sessions or conversations. This means that multiple Oracle client applications can be active simultaneously with the same OLTP through the independent LU.

Dependent LUs support only a single active session. The CP (Control Point for the node, which is IBM Communications Server, in this case) queues additional conversation requests from the gateway server behind an already active conversation. In other words, conversations are single-threaded for dependent LUs.

If a gateway LU is correctly defined, then no alterations to the Oracle Database Gateway for APPC configuration are needed, nor should any changes be needed to the gateway server.

The operational impact of dependent LUs is that the first client application can initiate a conversation through the database gateway with the gateway server. While that transaction is active (which could be seconds to minutes to hours, depending on how the client application and transaction are designed), any other client application

initiating a session with the same gateway server appears to hang as it waits behind the previous session.

If a production application really uses only a single conversation at any one time, then there should be no impact.

However, additional concurrent conversations might be required for testing or for other application development. Each requires that additional dependent LUs be defined on the remote host, plus additional IBM Communication Server configuration entries, which define the additional dependent LUs on the host Windows system.

Additional side information profiles should be defined to use the new dependent LUs. New gateway instances should be created and configured to use these new side information profiles.

Refer to PGAU DEFINE TRANSACTION, SIDEPROFILE, and LUNAME parameters in Chapter 2, "Procedural Gateway Administration Utility," in the *Oracle Database Gateway for APPC User's Guide*.

Definition Types

Several types of IBM Communications Server definitions are relevant to gateway APPC/LU6.2 operation. Each definition can be created and edited using a corresponding SNA Node Configuration menu.

The definitions relevant to the gateway are presented here in hierarchical order. Those definition types that are lowest in the hierarchy are discussed first. This matches the logical sequence in which to create the definitions.

Refer to the IBM Communications Server online documentation for a complete discussion of IBM Communications Server definitions. This section provides an overview of IBM Communications Server definitions in relation to the Oracle Database Gateway for APPC.

Creating IBM Communications Server Definitions for the Gateway

IBM Communications Server definitions are created using the SNA Node Configuration tool, while the actual operation of the server is done using the SNA Node Operations tool, both of which are provided with IBM Communications Server.

Maintenance of SNA definitions is normally done by a user with Administrator privileges.

The following sections describe the process of creating your SNA definitions for IBM Communications Server using the SNA Node Configuration tool. All the tasks described in this section are performed within SNA Node Configuration.

Creating the Configuration

SNA Node Configuration will first ask you if you are creating a new configuration or loading an existing configuration. These tasks are based on the assumption that a new configuration is being created.

SNA Node Configuration will next prompt you for a configuration scenario.

Creating the Node

Each SNA Server must have a control point defined. This is typically called the Node definition. To define the node:

1. Click **Node**.

2. Click **Create**.

3. In the **Define the Node** dialog box, select the **Basic** tab. Enter the **Control Point**, **Local Node ID**, and **Node Type** information.

   You can select **Advanced** tab options, depending on your SNA network configuration.

4. Click **OK**.

Creating Devices

Configure Communication Devices next. Follow these instructions:

1. Click **Devices**.

2. Click **Create**.

Choosing the Device Type

Select the type of device to use for communication. The LAN type is typical for either Ethernet or Token Ring attached network devices.

Configuring a LAN Device

Configure a LAN device next. Follow these instructions:

1. Select the **Basic** tab. Select the adapter to use and the Local SAP.

   The other tabs provide options on network tuning parameters.

2. Click **OK**.

Creating Peer Connections

Configure Peer Connections next. Follow these instructions:

1. Click **Peer Connections**.

2. Click **Create**.

Defining the Link Station

Define the Link Station next. Follow these instructions:

1. In the **Basic** tab, enter a Link Station name for this connection.

2. Choose the device for the connection.

3. Enter the Destination address and Remote SAP.

Defining the Adjacent Node

Define the Adjacent Node next. Follow these instructions:

1. Select the **Adjacent Node** tab.

2. Enter the Adjacent CP name of the remote system and pick its CP type. You may have to choose a different transmission group (TG) than the default. Consult your SNA network administrator for details.

   Other tabs provide options on tuning and reactivation.

3. Click **OK**.

Creating Local LUs

Create Local LUs for this node next. Follow these instructions:

1. Click **Local LU 6.2 LU**s.

2. Click **Create**.

Defining Local LUs

Define Local LUs next. Follow these instructions:

1. In the **Basic** tab, enter the name of the Local LU, and, optionally, an alias. The name must match the Local LU definition of the remote host for this node.

   You can examine the other tab for synchronization support and for LU session limits.

2. Click **OK**.

Creating Partner LUs

Create remote Partner LUs next. Follow these instructions:

1. Click **Partner LU 6.2 LUs**.

2. Click **Create**.

Defining Partner LUs

Define Partner LUs next. Follow these instructions:

1. In the **Basic** tab, enter the name of the Remote or Partner LU, and, optionally, an alias.

2. Select the Fully Qualified CP from the Existing list.

   You can examine the other tab for logical record limits and security support.

3. Click **OK**.

Creating the CPI-C Side Information Profile

Next, define the CPI-C profile that will be used to create the gateway. Follow these instructions:

1. Click the **CPI-C Side Information Definitions**.

2. Click **Create**.

Testing the Connection

Before proceeding with the gateway configuration tasks, ensure that your connection is working. You can do this by using the Microsoft Host Integration Server (SNA Server Manager).

Figure 6–1 shows the relationship between SNA Server definitions and the VTAM definitions on the host.

Figure 6–1   Relationship Between SNA Server Definitions and Host VTAM
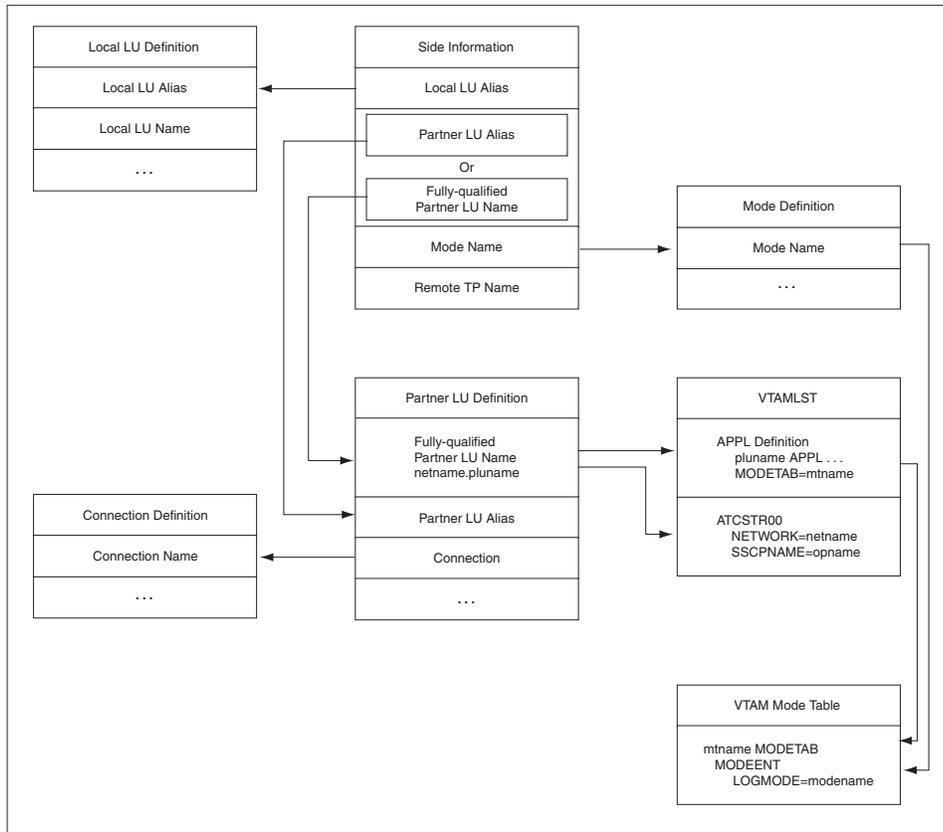
Resume Configuration of the Gateway

When you have finished configuring the SNA Server for your Windows system, proceed to Chapter 7, "Configuring the OLTP" to continue configuring the network.

7

Configuring the OLTP

The following sections detail how to configure the OLTP.

- **If your communications protocol is SNA**: proceed to Section , "Configuring the OLTP for Your SNA Environment" on page 7-1.

- **If your communications protocol is TCP/IP:** proceed to Section , "Configuring the OLTP for Your TCP/IP Environment" on page 7-4.

> **Note:** On a gateway using TCP/IP support for IMS Connect, you must specify `EDIT=ULC` in the IMS `TRANSACT` macro if you need input case sensitivity. When using SNA support, you need not specify `EDIT=ULC` in the IMS `TRANSACT` macro.

Configuring the OLTP for Your SNA Environment

The steps for configuring your OLTP to communicate with the Oracle Database Gateway for APPC vary, depending on which OLTP you are using and on which platform the OLTP is running. CICS Transaction Server for z/OS, IMS/TM, APPC/MVS, and z/OS are the currently supported OLTPs. Choose the instructions suitable to your OLTP from the following sections:

- "Configuring CICS Transaction Server for z/OS" on page 7-1

- "Configuring IMS/TM" on page 7-2

- "Configuring APPC/MVS" on page 7-3

> **Note:** You need only perform the configuration steps for an OLTP if this is a first-time configuration for that OLTP.

Configuring CICS Transaction Server for z/OS

If your OLTP is CICS Transaction Server for z/OS, then perform the following steps to configure it for communication with the gateway:

1. Configure MVS VTAM for the SNA Server that will make the APPC connection to your system. At least one independent LU must be available to the gateway.

2. Check the VTAM logmode table used by the CICS Transaction Server for z/OS. (The table name is specified in the `MODETAB` parameter in the VTAM APPL definition for CICS.) Ensure that an entry exists for APPC sessions with parallel session and sync-level support.

The `oraplu62.asm` file in the `%ORACLE_HOME%\dg4appc\sna` directory contains a sample mode entry, including comments that indicate the required values in the mode entry.

3. Using your file transfer facility, transfer the following files from the `%ORACLE_HOME%\dg4appc\demo\CICS` directory to the z/OS system on which you run CICS Transaction Server for z/OS:

   ■ `dfhcsdup.jcl` - JCL to run the CICS `DFHCSDUP` utility

   ■ `pgaflip.asm` - assembler source for the CICS `FLIP` transaction

   ■ `pgaflip.jcl` - JCL to assemble and linkedit the CICS `FLIP` transaction

4. Using the comments in the `dfhcsdup.jcl` file, tailor the JCL and input statements to match your system setup, and submit it for batch execution. Performing this step updates your Transaction Server for z/OS system definitions.

5. Using the instructions in the `pgaflip.jcl` file comments, tailor the JCL to match your system setup, and submit it for batch execution. Performing this step assembles and linkedits the `pgaflip.asm` file into a load module library accessible to your Transaction Server for z/OS through the `DFHRPL` DD statement in the CICS startup procedure.

6. Log on to your CICS Transaction Server for z/OS and enter the following transaction:

```
CEDA INSTALL GROUP(ORAPGA)
```

This transaction installs the CICS connection and session definitions for APPC communication with the gateway on Windows. It also installs definitions for the sample CICS programs and transactions provided with the gateway.

Your CICS Transaction Server for z/OS configuration is now complete.

Configuring IMS/TM

If your OLTP is IMS/TM, then perform the following steps to configure IMS/TM and z/OS for communication with the gateway:

1. Configure the IMS system for use with APPC.

2. Configure MVS VTAM for the SNA APPC connection to Windows. At least one independent LU must be available for use by the gateway, unless you are using the IMS LU6.1 Adapter for LU6.2 applications. In this case, you must have one dependent LU defined for each concurrent session. For example, if you want to support 10 concurrent sessions, then you must have 10 dependent LUs defined.

3. Check the VTAM logmode table used by IMS/TM. The table name is specified by the `MODETAB` parameter in the VTAM APPL definition.

   For APPC/IMS, ensure that an entry exists for APPC sessions with sync-level support and parallel session support. The `oralu62.asm` and `oraplu62.asm` files in the `%ORACLE_HOME%\dg4appc\sna` directory contain sample mode entries for single session and parallel session support, respectively. The samples include comments that indicate the required values in the mode entries.

4. Using your file transfer facility, transfer the following files from the `%ORACLE_HOME%\dg4appc\demo\IMS` directory to the z/OS system on which you run IMS/TM:

   ■ `pgaflip.asm` is assembler source for IMS `FLIP` transaction;

   ■ `pgaflip.jcl` is JCL to assemble and linkedit IMS `FLIP` transaction;

- `imsgen.asm` is IMS stage 1 gen definitions for the IMS `FLIP` transaction.

5. Add the statements in the `imsgen.asm` file to your IMS stage 1 gen and run your IMS stage 1 and stage 2 gens. Use the online change utility to enable the new transaction definition.

6. Using the comments in the `pgaflip.jcl` file, tailor the JCL to match your system setup and submit it for batch execution. This assembles and linkedits the `pgaflip.asm` file into a load module library that is accessible to your IMS/TM system and creates a PSB and an ACB for the `FLIP` transaction.

7. Perform the tasks necessary on your system to make the new transaction available to IMS/TM. Depending on your system setup, you might have to restart IMS.

   The IMS/TM configuration is now complete.

Configuring APPC/MVS

If your OLTP is APPC/MVS, then perform the following steps to configure APPC/MVS for communication with the gateway:

1. Configure MVS VTAM for the SNA APPC connection to Windows. At least one independent LU must be available for use by the gateway.

2. Check the VTAM logmode table used by APPC/MVS. (The table name is specified by the `MODETAB` parameter in the VTAM APPL definition for APPC/MVS.) Ensure that an entry exists for APPC sessions with `SYNCLEVEL` and parallel session support. The `oraplu62.asm` file in the `%ORACLE_HOME%\dg4appc\sna` directory contains a sample mode entry, including comments that indicate the required values in the mode entry.

3. Allocate a partitioned dataset (PDS) on your z/OS system where the sample files are placed. The PDS should be allocated with `RECFM=FB`, `LRECL=80`, and a `BLKSIZE` suitable for the device type on which it resides. Approximately two tracks of `3390` disk space are required with one directory block. Oracle suggests naming this partitioned dataset (PDS) `ORAPGA.APPCMVS.SAMPLIB`.

4. Using the file transfer facility, transfer the following files from the `%ORACLE_HOME%\dg4appc\demo\MVS` directory to the z/OS PDS you allocated in the previous step, using the following specified member names:

   - `pgaflip.jcl` is JCL to add an APPC/MVS TP profile and to define the execution environment for the transaction. Store this file in your z/OS PDS as member `PGAFLIPJ`.

   - `pgaflip.rex` is the REXX source for the APPC/MVS `PGAFLIP` transaction. Store this file in your z/OS PDS as member `PGAFLIP`.

5. Using the comments in the `pgaflip.jcl` file, tailor the JCL to match your system setup and submit it for batch execution. Performing this step defines the APPC/MVS TP profile for the `PGAFLIP` transaction and stores it in the APPC/MVS profile dataset. Ensure that you change the dataset name in the JCL to match the name of the z/OS PDS allocated in Step 3.

The APPC/MVS configuration is now complete.

- Now that you have completed configuration of the network on a gateway using the SNA protocol, proceed to Chapter 8, "Gateway Configuration Using SNA Communication Protocol".

- If you wish to configure commit-confirm, then instructions for this configuration can be found in Section , "Configuring Commit-Confirm".

Configuring the OLTP for Your TCP/IP Environment

These are the steps for configuring the OLTP to communicate with Oracle Database Gateway for APPC using TCP/IP for IMS Connect. IMS/TM, through IMS Connect, is the only supported OLTP for this release of the gateway.

Perform the following steps to configure IMS/TM and z/OS for communication with the gateway:

1. Configure the IMS system.

2. Configure IMS Connect.

    For information on how to configure IMS Connect, refer to the IBM *IMS Connect Guide and Reference*.

3. Using your file transfer facility, transfer the following files from the `%ORACLE_HOME%\dg4appc\demo\IMS` directory to the z/OS system on which you run IMS/TM:

    ■ `pgaflip.asm` is assembler source for IMS `FLIP` transaction;

    ■ `pgaflip.jcl` is JCL to assemble and linkedit IMS `FLIP` transaction;

    ■ `imsgen.asm` is IMS stage 1 gen definitions for the IMS `FLIP` transaction.

4. Add the statements in the `imsgen.asm` file to the IMS stage 1 gen and run your IMS stage 1 and stage 2 gens. Use the online change utility to enable the new transaction definition.

5. Using the comments in the `pgaflip.jcl` file, tailor the JCL to match your system setup and submit it for batch execution. This assembles and linkedits the pgaflip.asm file into a load module library that is accessible to your IMS/TM system and creates a PSB and an ACB for the `FLIP` transaction.

6. Perform the tasks necessary on your system to make the new transaction available to IMS/TM. Depending on your system setup, you might have to restart IMS.

The IMS/TM configuration is now complete.

■ At this point, proceed to Chapter 9, "Gateway Configuration Using TCP/IP Communication Protocol" to complete configuration of the gateway and its components.

8

Gateway Configuration Using SNA Communication Protocol

This chapter outlines the steps needed to configure the Oracle database for a gateway using the SNA protocol on the Microsoft Windows platform. It also shows you how to configure commit-confirm, should you choose to implement it.

This chapter also provides the steps necessary to verify installation and configuration of the gateway components, including optional commit-confirm. It contains the following sections:

- Before You Begin
- Preparing to Configure a Gateway Installation/Upgrade
- Integrating Server Configuration: First-Time Gateway Installations
- Upgrading or Migrating the Oracle Database from Previous Gateways
- Configuring the Oracle Database for Gateways to Coexist
- Optional Configuration Steps to Permit Multiple Users
- Configuring the Gateway
- Configuring Commit-Confirm
- Verifying the Gateway Installation and OLTP Configuration
- Performing Postinstallation Procedures

Configuring the Oracle Database Gateway for APPC using SNA involves working with the following components:

- the Oracle database
- your Microsoft Windows system
- your network
- the OLTP

Before You Begin

This chapter requires you to input parameters unique to your system in order to properly configure the gateway and SNA communications interface.

Refer to Appendix D, "Configuration Worksheet" for a worksheet listing the installation parameters you will need to know before you can complete the configuration process. Ask your network administrator to provide you with these unique parameter names before you begin.

Preparing to Configure a Gateway Installation/Upgrade

There are three ways to establish the gateway-Oracle database relationship when you are installing or upgrading/migrating the gateway:

- When the Oracle Database and the gateway are installed in the same ORACLE_HOME on the same machine

- When the Oracle Database and the gateway are installed on separate machine

- When the Oracle Database and the gateway are on the same machine but in different directories

Depending on the location of your gateway and your Oracle database , you may need to transfer some of the gateway administrative files to the location where your Oracle database is installed.

Follow the instructions suitable to your combination of the gateway-Oracle database locations listed below.

When the Oracle Database and the gateway are installed in the same ORACLE_HOME on the same machine

You do not need to transfer files. Proceed to "Integrating Server Configuration: First-Time Gateway Installations" on page 8-3.

When the Oracle Database and the gateway are installed on separate machine

1. Locate the gateway administrative files in the gateway `%ORACLE_HOME%\dg4appc\admin` directory. All files in this directory that have the suffix `.sql`, `.pkh`, and `.pkb` must be copied into a similarly-named directory in the Oracle database 's Oracle home directory.

2. Now, locate the gateway demo files and subdirectories in the gateway's `%ORACLE_HOME%\dg4appc\demo` directory. Copy the `pgavsn.sql` and `pgaecho.sql` files into a similarly named directory in the Oracle database .

3. Copy the other subdirectories and files related to your installed OLTP on your remote host. For example, if you have CICS as your only OLTP, then copy the gateway `%ORACLE_HOME%\dg4appc\demo\CICS` files into a similarly named directory in the Oracle database .

> **Note:** Before transferring the files from the `%ORACLE_HOME%\dg4appc\demo` directory, make sure that you have generated your required TIPs. You need to transfer the TIPs as well.
>
> Refer to the *Oracle Database Gateway for APPC User's Guide* for information about generating TIPs using PGAU.

When the Oracle Database and the gateway are on the same machine but in different directories

You must change your gateway's Oracle home to the Oracle database 's Oracle home directory.

1. For example, if your gateway's Oracle home is set as follows:

```
C:\> echo %ORACLE_HOME%
C:\oracle\pga\11.2
```

and your server's Oracle home is located in the `\oracle\pga\11.2` directory, then you need to do the following:

```
C:\> SET ORACLE_HOME=C:\oracle\pga\11.2
```

2. Now create the directories with the following commands:

```
C:\> cd %ORACLE_HOME%
C:\> mkdir dg4appc
C:\> mkdir dg4appc\admin
```

3. Use whatever file transfer mechanism is available on your system to copy all the `.sql`, `.pkh`, and `.pkb` files from the gateway Oracle home `%ORACLE_HOME%\dg4appc\admin` directory to the Oracle database 's Oracle home `%ORACLE_HOME%\dg4appc\admin` directory.

4. You may also transfer the demo files from the gateway directory to the Oracle database directory. Copy the files and the directory recursively from the gateway Oracle home `%ORACLE_HOME%\dg4appc\demo` directory to the Oracle database `%ORACLE_HOME%\dg4appc\demo` directory.

   For example:

```
C:\> cd %ORACLE_HOME%\dg4appc
C:\> mkdir\demo
C:\> cd demo
C:\> xcopy C:\oracle\pga\11.2\demo /E /I /Q
```

Note: Before transferring the files from the `%ORACLE_HOME%\dg4appc\demo` directory, make sure that you have generated the required TIPs. You need to transfer the TIPs as well.

Refer to the *Oracle Database Gateway for APPC User's Guide* for information about generating TIPs using PGAU.

If this is a first-time installation, then proceed with "Integrating Server Configuration: First-Time Gateway Installations" on page 8-3.

If this is an upgrade, then proceed with "Upgrading or Migrating the Oracle Database from Previous Gateways" on page 8-6.

Following those steps, you may want to perform the Optional Configuration Steps to Permit Multiple Users, on page 8-8.

Integrating Server Configuration: First-Time Gateway Installations

Follow these steps to configure your Oracle database if you have installed Oracle Database Gateway for APPC for the first time:

1. Ensure that the `UTL_RAW` PL/SQL package has been installed on your Oracle database . All PGAU-generated TIP specifications use `UTL_RAW`, which provides routines for manipulating raw data.

   a. Use SQL*Plus to connect to the Oracle database as user `SYS`.

   b. From SQL*Plus, enter the following command:

```
SQL> DESCRIBE UTL_RAW
```

The `DESCRIBE` statement produces output on your screen. If you browse through the output, then you should see some functions, including a compare function. If you do not see this output, then continue the `UTL_RAW` installation by performing Step d below.

If the `DESCRIBE` statement indicates success, then your Oracle database has `UTL_RAW` installed and you can proceed to Step 2.

c. Use SQL*Plus to connect to the Oracle database as user `SYS`.

d. From SQL*Plus, run the utlraw.sql and prvtrawb.plb scripts in the Oracle database 's `%ORACLE_HOME%\rdbms\admin` directory, in the following order:

```
C:\> cd %ORACLE_HOME%\rdbms\admin
SQL> @utlraw.sql
SQL> @prvtrawb.plb
```

2. Ensure that the `DBMS_OUTPUT` standard PL/SQL package is enabled on your Oracle database . The sample programs and installation verification programs on the distribution media use this standard package.

a. If necessary, then use SQL*Plus to connect to the Oracle database as user `SYS`.

b. From SQL*Plus, enter the following command:

```
SQL> DESCRIBE DBMS_OUTPUT
```

The `DESCRIBE` statement produces output on your screen. If you browse through that output, you should see some functions, including a put_line function.

If you do not see this output, then you must create the `DBMS_OUTPUT` package. Refer to the *Oracle Database Advanced Application Developer's Guide* for more information about creating the `DBMS_OUTPUT` package. After successful installation of the `DBMS_OUTPUT` package, issue the `DESCRIBE` statement.

If the `DESCRIBE` statement indicates success, then your Oracle database has `DBMS_OUTPUT` created, and you can proceed to Step 3.

3. Install the `UTL_PG` PL/SQL package. All PGAU-generated TIP specifications use `UTL_PG`, which provides routines for performing numeric conversions to and from raw data.

a. If necessary, then use SQL*Plus to connect to the Oracle database as user `SYS`.

b. From SQL*Plus, run the `utlpg.sql` and `prvtpgb.plb` scripts in the Oracle database 's `%ORACLE_HOME%\rdbms\admin` directory, in the following order:

```
C:\> cd %ORACLE_HOME%\rdbms\admin
SQL> @utlpg.sql
SQL> @prvtpgb.plb
```

4. Install the Heterogeneous Services (HS) catalogs.

a. If necessary, use SQL*Plus to connect to the Oracle database as user `SYS`.

b. Enter the following command:

```
SQL> DESCRIBE HS_FDS_CLASS
```

The `DESCRIBE` statement produces output on your screen. If the `DESCRIBE` statement indicates success, then heterogeneous services catalogs have been created on your Oracle database and you can proceed to Step 5. otherwise

follow the next step only if the DESCRIBE statement does not indicate success. The next step will create the Heterogeneous Services catalog.

c. If it is necessary to create the Heterogeneous Services catalog, then enter the following command:

```
C:\> cd %ORACLE_HOME%\rdbms\admin
SQL> @caths.sql
```

5. Create a public database link to access Oracle Database Gateway for APPC:

Use SQL*Plus to connect to the Oracle database as user SYSTEM. You can use the following SQL*Plus sample whether the Oracle database and the gateway are on the same system or on different systems. In the following sample, pgasrv is the *tns_name_entry* that will be assigned to the gateway when you modify the tnsnames.ora file later in this chapter.

```
SQL> CREATE PUBLIC DATABASE LINK PGA USING 'PGASRV'
```

6. Create the gateway administrator user PGAADMIN and install the PG DD.

a. Use SQL*Plus to connect to the Oracle database as user SYSTEM.

b. From SQL*Plus, run the pgacr8au.sql script in the %ORACLE_HOME%\dg4appc\admin directory. This script creates the PGAADMIN user ID.

The initial password defined for PGAADMIN is PGAADMIN. Use the ALTER USER command to change the password. For further information about password issues, refer to the *Oracle Database SQL Language Reference*.

```
C:\> cd %ORACLE_HOME%\dg4appc\admin
SQL> @pgacr8au.sql
```

c. Use SQL*Plus to connect to the Oracle database as user PGAADMIN.

d. From SQL*Plus, run the pgddcr8.sql script in the %ORACLE_HOME%\dg4appc\admin directory. This script installs the PG DD.

```
C:\> cd %ORACLE_HOME%\dg4appc\admin
SQL> @pgddcr8.sql
```

e. From SQL*Plus, connect to the Oracle database as user SYS.

f. Grant execution privileges on DBMS_PIPE to PGAADMIN:

```
SQL> GRANT EXECUTE ON DBMS_PIPE TO PGAADMIN
```

7. Install the TIP trace access PL/SQL routines. These routines require that the DBMS_PIPES standard PL/SQL package is installed and that PGAADMIN has execute privileges on it. For more information on DBMS_PIPES, refer to the *Oracle Database Advanced Application Developer's Guide*.

a. If necessary, then use SQL*Plus to connect to the Oracle database as user PGAADMIN.

b. From SQL*Plus, run the pgatiptr.sql script in the %ORACLE_HOME%\dg4appc\admin directory. This script creates PL/SQL routines that can be called to read and purge trace information created by PGAU-generated TIP specifications. It also creates public synonyms for these routines. The script prompts you for the necessary user IDs and passwords.

```
C:\> cd %ORACLE_HOME%\dg4appc\admin
SQL> @pgatiptr.sql
```

8. Install the `GPGLOCAL` package. This package is required for compilation and execution of all PGAU-generated TIP specifications. TIP developers should be granted execute privileges on `GPGLOCAL` (refer to Step 3 of "Optional Configuration Steps to Permit Multiple Users" on page 8-8).

   a. Use SQL*Plus to connect to the Oracle database as user `PGAADMIN`.

   b. From SQL*Plus, run the `gpglocal.pkh` script in the `%ORACLE_HOME%\dg4appc\admin` directory. This script compiles the `GPGLOCAL` package specification.

   ```
   C:\> cd %ORACLE_HOME%\dg4appc\admin
   SQL> @gpglocal.pkh
   ```

   c. From SQL*Plus, run the `gpglocal.pkb` script in the `%ORACLE_HOME%\dg4appc\admin` directory. This script compiles the `GPGLOCAL` package body.

   ```
   C:\> cd %ORACLE_HOME%\dg4appc\admin
   SQL> @gpglocal.pkb
   ```

Upgrading or Migrating the Oracle Database from Previous Gateways

Follow these instructions *only* if you have a previous version of the Oracle Database Gateway for APPC installed on your system and need to configure it for Release 11.2 of the gateway.

1. Upgrade your Oracle Database Gateway for APPC to current version levels as follows:

   a. Use SQL*Plus to connect to the Oracle database as user `SYS`.

   b. Install the `UTL_RAW` package body. From SQL*Plus, run the `prvtrawb.plb` script from the `%ORACLE_HOME%\rdbms\admin` directory. This script upgrades the `UTL_RAW` package body.

   ```
   C:\> cd %ORACLE_HOME%\rdbms\admin
   SQL> @prvtrawb.plb
   ```

   c. Install the `UTL_PG` package body. From SQL*Plus, run the `prvtpgb.plb` script from the `%ORACLE_HOME%\rdbms\admin` directory. This script upgrades the `UTL_PG` package body.

   ```
   C:\> cd %ORACLE_HOME%\rdbms\admin
   SQL> @prvtpgb.plb
   ```

   The `prvtrawb.plb` and `prvtpgb.plb` scripts should complete successfully. If they fail because specifications do not exist or were invalidated, then consider reinstalling the package specifications as directed in Section , "If You Must Reinstall Package Specifications".

If You Must Reinstall Package Specifications

If the `UTL_RAW` or `UTL_PG` package has been invalidated or removed, then the `prvtrawb.plb` and `prvtpgb.plb` scripts may not complete successfully, then, you might have to reinstall the package specifications.

If you do reinstall the package specifications, then any dependent objects (such as existing user TIPs and client applications) are invalidated and will subsequently need to be recompiled. The impact of this is a one-time performance delay while recompilation of the TIPs and dependent client applications proceeds.

> **Important:** Before proceeding with reinstallation of the package scripts, make sure that you are in the `%ORACLE_HOME%\dg4appc\admin` directory.

TIPs were split into separate specification and body files in release 3.3 to avoid cascaded recompilations in later releases.

Step 1 Run the following scripts before proceeding with the PGAU upgrade

From SQL*Plus, run the `utlraw.sql` script:

1. If necessary, then use SQL*Plus to connect to the Oracle database as user `SYS`.

2. From SQL*Plus, run the `utlraw.sql` and `utlpg.sql` scripts in the Oracle database `%ORACLE_HOME%\rdbms\admin` directory, in the following order, to upgrade their respective package specifications:

   ```
   C:\> cd %ORACLE_HOME%\rdbms\admin
   SQL> @utlraw.sql
   SQL> @utlpg.sql
   ```

Step 2 Repeat installation of UTL_RAW and UTL_PG package body

After the scripts have run, repeat Steps 1b and c in Section , "Upgrading or Migrating the Oracle Database from Previous Gateways" on page 8-6. Then, proceed to the section titled "Upgrading PGAU from Previous Gateway Releases" on page 8-7.

> **Note:** TIPs and dependent client applications must be recompiled after reinstallation of the package specifications. Refer to the "Compiling a TIP" section in Chapter 3 of the *Oracle Database Gateway for APPC User's Guide* for information about compiling TIPs.

Upgrading PGAU from Previous Gateway Releases

1. Upgrade the PG DD as follows before executing the new PGAU:

   a. If necessary, then use SQL*Plus to connect to the Oracle database as user `PGAADMIN`.

   b. From SQL*Plus, run the `pgddupgr.sql` script in the `%ORACLE_HOME%\dg4appc\admin` directory. This script upgrades the PG DD.

   ```
   C:\> cd %ORACLE_HOME%\dg4appc\admin
   SQL> @pgddupgr.sql
   ```

Configuring the Oracle Database for Gateways to Coexist

You may have an older version of the gateway already installed. Be aware that although a version 10 gateway can communicate with a version 9 data dictionary (PG DD), a version 9 gateway cannot communicate with a version 10 data dictionary. So, if you upgrade your data dictionary to version 10, no gateways which were configured with a version 9 data dictionary will be able to communicate with it.

Optional Configuration Steps to Permit Multiple Users

The following configuration steps are optional. Perform these steps if you want to permit users other than PGAADMIN to perform PG DD operations using PGAU.

1. Create public synonyms for the PG DD to allow other users to access the tables:

   a. Use SQL*Plus to connect to the Oracle database as user SYSTEM.

   b. From SQL*Plus, run the pgddcr8s.sql script in the %ORACLE_HOME%\dg4appc\admin directory. This script creates public synonyms for the PG DD.

   ```
   C:\> cd %ORACLE_HOME%\dg4appc\admin
   SQL> @pgddcr8s.sql
   ```

2. Create roles for accessing the PG DD, performing definitions of transactions, and generating TIP specifications. The PGAADMIN user can grant these roles to other users as necessary.

   a. Use SQL*Plus to connect to the Oracle database as user PGAADMIN.

   b. From SQL*Plus, run the pgddcr8r.sql script in the %ORACLE_HOME%\dg4appc\admin directory. This script creates two roles, PGDDDEF and PGDDGEN. The PGDDDEF role provides SELECT, INSERT, UPDATE, and DELETE privileges against some of the PG DD tables, and select privileges against others, and allows execution of the PGAU DEFINE, GENERATE, REDEFINE, REPORT, and UNDEFINE statements. The PGDDGEN role provides select privileges against the PG DD tables, and allows execution of the PGAU GENERATE and REPORT statements only.

   ```
   C:\> cd %ORACLE_HOME%\dg4appc\admin
   SQL> @pgddcr8r.sql
   ```

3. Grant access to PGA required packages.

   TIP developers require access to the following PL/SQL packages, which are shipped with the Oracle database :

   - DBMS_PIPE in the %ORACLE_HOME%\rdbms\admin directory

   - UTL_RAW in the %ORACLE_HOME%\rdbms\admin directory

   - UTL_PG in the %ORACLE_HOME%\rdbms\admin directory

   Explicit grants to execute these packages must be made to TIP developers.

   These grants can be private, as in the following example:

   ```
   C:\> sqlplus SYS\pw@database_specification_string
   SQL> GRANT EXECUTE ON UTL_RAW TO tip_developer;
   SQL> GRANT EXECUTE ON UTL_PG TO tip_developer;
   SQL> GRANT EXECUTE ON DBMS_PIPE TO tip_developer;
   SQL> CONNECT PGAADMIN\pw@database_specification_string
   SQL> GRANT EXECUTE ON PGAADMIN.PURGE_TRACE TO tip_developer;
   SQL> GRANT EXECUTE ON PGAADMIN.READ_TRACE TO tip_developer;
   SQL> GRANT EXECUTE ON PGAADMIN.GPGLOCAL TO tip_developer;
   SQL> exit
   ```

   Alternatively, these grants can be public, as in the following example:

   ```
   C:\> sqlplus SYS\pw@database_specification_string
   SQL> GRANT EXECUTE ON UTL_RAW TO PUBLIC;
   SQL> GRANT EXECUTE ON UTL_PG TO PUBLIC;
   SQL> GRANT EXECUTE ON DBMS_PIPE to PUBLIC;
   ```

```
SQL> CONNECT PGAADMIN\pw@database_specification_string
SQL> GRANT EXECUTE ON PGAADMIN.PURGE_TRACE TO PUBLIC;
SQL> GRANT EXECUTE ON PGAADMIN.READ_TRACE TO PUBLIC;
SQL> GRANT EXECUTE ON PGAADMIN.GPGLOCAL TO PUBLIC;
SQL> EXIT
```

You can use either private or public grants. Both grants are sufficient for using PGA. Public grants are easier and can be performed now. If you use private grants, then they must be issued each time a new TIP developer user ID is created.

SQL scripts for performing these grants are provided in the `%ORACLE_HOME%\dg4appc\admin` directory. The `pgddapub.sql` script performs these grants for public access to the packages. The `pgddadev.sql` script performs the grants for private access to the packages by a single TIP developer. If you are going to use private grants, then you must run the `pgddadev.sql` script once for each TIP developer's user ID:

a. Use SQL*Plus to connect to the Oracle database as user `PGAADMIN`.

b. From SQL*Plus, run the suitable script (`pgddapub.sql` or `pgddadev.sql`) from the `%ORACLE_HOME%\dg4appc\admin` directory. The script performs the necessary grants as described earlier. You are prompted for the required user IDs, passwords, and database specification strings.

If you are using private grants, then repeat this step for each user ID requiring access to the packages.

```
C:\> cd %ORACLE_HOME%\dg4appc\admin
SQL> @pgddapub.sql
```

or

```
SQL> @pgddadev.sql
```

4. If you are upgrading from a previous release of the gateway and if you want to upgrade your existing TIPs with new function and maintenance, then regenerate existing TIP specifications using the PGAU `GENERATE` statement.

> **Note:** The Procedural Gateway Administrative Utility (PGAU) has been enhanced to automatically upgrade existing PG DD entries with a new attribute when a PGAU `GENERATE` command is executed. To support this enhancement, add a new privilege to the `PGDDGEN` role. To do this, as the `PGAADMIN` user, use SQL*Plus to connect to the Oracle database where the PG DD is stored. Then issue the following SQL command:
>
> ```
> SQL> GRANT INSERT ON PGA_DATA_VALUES TO PGDDGEN
> ```

a. Call PGAU in the directory path where the PGAU control files are generated and where TIPs are stored:

```
C:\> pgau
PGAU> CONNECT PGAADMIN\pgaadmin@database_specification_string
PGAU> GENERATE tranname
PGAU> EXIT
```

For more information about the `GENERATE` command, refer to the PGAU `GENERATE` command section in Chapter 2 of the *Oracle Database Gateway for APPC User's Guide*.

Note that it is not necessary to define the PG DD entries again.

5. Call SQL*Plus in the same directory path where the newly-generated TIP specifications are stored.

```
C:\> sqlplus tip_owner\pw@database_specification_string
SQL> @tipname.pkh
SQL> @tipname.pkb
SQL> exit
```

PGAU GENERATE produces the TIP in two output file, a specification and a body. You must compile both, first the specification and then the body.

For more information about the GENERATE command, refer to the PGAU GENERATE command section in Chapter 2 of the *Oracle Database Gateway for APPC User's Guide*.

Configuring the Gateway

To configure the gateway, perform the following steps:

1. Tailor the Oracle Database Gateway for APPC parameters.

Parameters specific to the gateway are supplied in the gateway parameter file, init*sid*.ora, which is in the %ORACLE_HOME%\dg4appc\admin directory. A sample gateway parameter file, initPGA.ora, is provided in this subdirectory.

Note: In the init*sid*.ora file, substitute your dg4appc SID name for "*sid*" in this file name.

The init*sid*.ora file contains both APPC and TCP/IP parameters, separated by a description. You must modify the init*sid*.ora file by deleting the TCP/IP parameters. Refer to Appendix A, "Gateway Initialization Parameters for SNA Protocol" for the valid APPC parameters.

The parameters fall into two categories:

- gateway initialization parameters

  These parameters control the general operation of the gateway in the Oracle environment.

  Important: Before performing the following step, refer to Appendix A, "Gateway Initialization Parameters for SNA Protocol" for information about tailoring gateway initialization and PGA parameters. Pay special attention to the information about using the PGA_CAPABILITY parameter.

- PGA parameters

  PGA parameters control the APPC interface portion of the gateway. Use the SET gateway initialization parameter to specify PGA parameters. Oracle recommends that you group all SET commands for PGA parameters at the end of the init*sid*.ora file.

> **Note:** Misspelled parameters are ignored. However, if the
> `%ORACLE_HOME%\dg4appc\admin\initsid.ora` file is missing,
> then all calls to the gateway fail and return a `PGA-20928` error.

If you do not plan to configure commit-confirm, then proceed to "Verifying the
Gateway Installation and OLTP Configuration" on page 8-13.

Configuring Commit-Confirm

> **Note:** If you are planning to implement commit-confirm, then
> read the detailed explanation of commit-confirm capabilities in
> Chapter 5 of the *Oracle Database Gateway for APPC User's Guide*,
> "Implementing Commit-Confirm (SNA Only) before proceeding.

Follow these steps to configure the commit-confirm components.

The steps for configuring commit-confirm include:

- configuring the Oracle database where the gateway server will store its
  transaction log information
- configuring the gateway initialization parameters, and
- configuring the OLTP.

All these steps must be performed before attempting to use any application that uses
commit-confirm.

Configuring the Oracle Database for Commit-Confirm

The Oracle database where the gateway server will store its transaction log
information should ideally be on the same system where the gateway runs. The
configuration of the server consists of creating the gateway DBA user, creating the
commit-confirm log tables and creating the PL/SQL stored procedure used by the
gateway server for logging transactions.

The `pgaccau.sql` script from the `%ORACLE_HOME%\dg4appc\admin` directory
creates the gateway DBA user ID. The default user ID is `PGADBA` with the initial
password set to `PGADBA`. If you want to change the user ID or initial password, you
must modify the script.

1. Use SQL*Plus to connect to the Oracle database as user `SYSTEM`.

2. From SQL*Plus, run the `pgaccau.sql` script from the
   `%ORACLE_HOME%\dg4appc\admin` directory. This script creates the gateway
   DBA user ID. If you want to change the password at any time after running this
   script, then you can use the `ALTER USER` command to change the password. For
   further information, refer to the *Oracle Database SQL Language Reference*.

3. Use SQL*Plus to connect to the Oracle database as user `PGADBA`.

4. From SQL*Plus, run the `pgaccpnd.sql` script from the
   `%ORACLE_HOME%\dg4appc\admin` directory. This script creates the
   `PGA_CC_PENDING` table used by the gateway server for its commit-confirm
   transaction log.

5. From SQL*Plus, run the `pgacclog.sql` script from the `%ORACLE_HOME%\dg4appc\admin` directory. This script creates the `PGA_CC_LOG` PL/SQL stored procedure used by the gateway server for updating the `PGA_CC_PENDING` table.

6. Disconnect from the Oracle database .

Configuring Gateway Initialization Parameters for Commit-Confirm

The gateway initialization parameters are described in Appendix A, "Gateway Initialization Parameters for SNA Protocol". The parameters necessary for commit-confirm support in the gateway are:

- `PGA_CAPABILITY`
- `PGA_LOG_DB`
- `PGA_LOG_USER`
- `PGA_LOG_PASS`
- `PGA_RECOVERY_USER`
- `PGA_RECOVERY_PASS`
- `PGA_RECOVERY_TPNAME`

These parameters should be added to your `initsid`.ora file, where *sid* is the gateway SID for the commit-confirm gateway.

Refer to Appendix A, "Gateway Initialization Parameters for SNA Protocol" for full information about the gateway initialization parameters supported by the Oracle Database Gateway for APPC.

Configuring the OLTP for Commit-Confirm

Configuration of the OLTP includes the following:

- defining and installing the commit-confirm transaction log database
- defining and installing the commit-confirm forget/recovery transaction
- defining and installing the sample commit-confirm applications provided with the gateway.

> **Note:** A restart of the OLTP may be necessary to implement the changes required for commit-confirm support. You should plan for this with your OLTP system administrator.

Detailed instructions for configuring the Transaction Server for z/OS and IMS/TM are provided in the `%ORACLE_HOME%\dg4appc\demo\CICS\README.doc` and `%ORACLE_HOME%\dg4appc\demo\IMS\README.doc` files, respectively.

Refer to Chapter 5, "Implementing Commit-Confirm (SNA Only)" in the *Oracle Database Gateway for APPC User's Guide* for detailed information about commit-confirm.

- You will take steps to verify configuration of commit-confirm later in this chapter, in Section , "Verifying OLTP Configuration for Commit-Confirm" on page 8-16.

However, first proceed to Section to verify the gateway installation.

Verifying the Gateway Installation and OLTP Configuration

To verify the gateway installation and the OLTP configuration, perform the following procedures after installing Oracle Database Gateway for APPC. In addition, if you chose to configure commit-confirm, then follow the steps to verify the OLTP configuration for commit-confirm.

> **Note:** If your database link name is not PGA, then modify the demonstration .sql files to give them the particular database link name that you created in Step 5 of "Integrating Server Configuration: First-Time Gateway Installations" on page 8-3. You must modify the following .sql files:
>
> - `pgavsn.sql`
> - `pgaecho.sql`
> - `pgacics.sql`
> - `pgaidms.sql`
> - `pgaims.sql`
> - `pgamvs.sql`

Verifying the Gateway Installation

To verify the gateway software installation using the database link PGA previously created, perform the following steps:

1. Using SQL*Plus, connect to the Oracle database as user `PGAADMIN`.

2. Run `%ORACLE_HOME%\dg4appc\demo\pgavsn.sql`.

   ```
   C:\> cd %ORACLE_HOME%\dg4appc\demo
   SQL> @pgavsn.sql
   ```

   The server version number banner appears at your terminal. You will receive the following output:

   ```
   Oracle Database Gateway for APPC.
   Version 11.2.0.0.2 Thu Jun 11 14:39:15
   2009

   Copyright (c) Oracle Corporation 1979, 2009.  All rights reserved.


   PL/SQL procedure successfully completed.
   ```

3. Run `%ORACLE_HOME%\dg4appc\demo\pgaecho.sql`.

   ```
   C:\> cd %ORACLE_HOME%\dg4appc\demo
   SQL> @pgaecho.sql
   ```

   You will receive the following output:

   ```
   ==> Congratulations, your installation was successful. <==
   ```

Verifying the OLTP Configuration

The procedure for verifying the OLTP configuration varies, depending on which OLTP you are using and depending on the platform the OLTP is running on. CICS

Transaction Server for z/OS, IMS/TM, APPC/MVS and z/OS are the currently supported OLTPs. Follow the instructions for verifying the installation.

> **Attention:**   If you have not completed the file transfers detailed in "Preparing to Configure a Gateway Installation/Upgrade" on page 8-2, then complete them now, before proceeding to the next step.

CICS Verification

If your OLTP is CICS Transaction Server for z/OS, then perform the following steps to verify the CICS configuration.

1. To verify that the `FLIP` transaction is installed correctly, log on to your CICS Transaction Server for z/OS and enter the following transaction, replacing `FLIP` with the transaction ID you chose for `FLIP` when you configured your CICS Transaction Server for z/OS for the gateway:

   ```
   FLIP THIS MESSAGE
   ```

   The following output appears at your terminal:

   ```
   EGASSEM SIHT PILF
   ```

2. From the Windows platform, modify the pgacics.sql file which resides at `%ORACLE_HOME%\dg4appc\demo\CICS\pgacics.sql`.

   Customize the following three items used for accessing the gateway and the CICS Transaction Server for z/OS, as described in the comments at the beginning of the file:

   - the CICS transaction ID
   - the side profile name
   - the logmode entry name

3. Ensure that the SNA Server on the system has been started.

4. Log on to the CICS Transaction Server for z/OS and enter this transaction, where *name* is the name of the `CONNECTION` definition installed by the `DFHCSDUP` job you ran in the CICS configuration steps:

   ```
   CEMT SET CONNECTION(name) ACQUIRED
   ```

   This transaction activates the CICS connection to Windows.

5. Use SQL*Plus to connect to the Oracle database as `PGAADMIN`.

6. Run `pgacics.sql`.

   ```
   C:\> cd %ORACLE_HOME%\dg4appc\demo\CICS
   SQL> @pgacics.sql
   ```

   The following message appears:

   ```
   ==> Congratulations, your gateway is communicating with CICS <==
   ```

The CICS Transaction Server for z/OS installation verification is complete.

IMS/TM Verification

If the OLTP is IMS/TM, then perform the following steps to verify the IMS/TM configuration:

1. To verify that the `FLIP` transaction is installed correctly, log on to the IMS/TM system and enter the following transaction, replacing `FLIP` with the transaction ID you chose for `FLIP` when you configured the IMS/TM system for the gateway:

    ```
    FLIP THIS MESSAGE
    ```

    The following output appears on the terminal:

    ```
    EGASSEM SIHT PILF
    ```

2. From a Windows platform, modify the `pgaims.sql` file, which resides at

    `%ORACLE_HOME%\dg4appc\demo\IMS\pgaims.sql`. Customize the following three items used for accessing the gateway and the IMS/TM system, as described in the comments at the beginning of the file:

    - the IMS/TM transaction ID

    - the side profile name

    - the logmode entry name

3. Ensure that the SNA Server on the system has been started.

4. Use SQL*Plus to connect to the Oracle database as `PGAADMIN`.

5. Run `pgaims.sql`.

    ```
    C:\> cd %ORACLE_HOME%\dg4appc\demo\IMS
    SQL> @pgaims.sql
    ```

The following message appears:

```
==> Congratulations, your gateway is communicating with IMS/TM <==
```

The IMS/TM installation verification is now complete.

APPC/MVS Verification

If the OLTP is APPC/MVS, then perform the following steps to verify the APPC/MVS configuration:

1. Verify that the APPC/MVS subsystem is active.

2. From the Windows platform, modify the `pgamvs.sql` file, which resides at `%ORACLE_HOME%\dg4appc\demo\MVS\pgamvs.sql`. Customize the following three items used for accessing the gateway and the APPC/MVS system, as described in the comments at the beginning of the file:

    - the APPC/MVS transaction ID

    - the side profile name

    - the logmode entry name

3. Ensure that the SNA Server on the system has been started.

4. Use SQL*Plus to connect to the Oracle database as `PGAADMIN`.

5. Run `pgamvs.sql`.

    ```
    C:\> cd %ORACLE_HOME%\dg4appc\demo\MVS
    SQL> @pgamvs.sql
    ```

The following message appears:

```
=> Congratulations, your gateway is communicating with APPC/MVS <=
```

The APPC/MVS installation verification is now complete.

Verifying OLTP Configuration for Commit-Confirm

If you chose to configure commit-confirm in Section , "Configuring Commit-Confirm", then the following section will assist you in verifying the configuration.

> **Note:** Refer to Chapter 5, "Implementing Commit-Confirm" in the *Oracle Database Gateway for APPC User's Guide* for background information about the components and capabilities of commit-confirm.

Samples are provided with the gateway for Transaction Server for z/OS and IMS/TM for implementing commit-confirm support. They are in the following directories, respectively: `%ORACLE_HOME%\dg4appc\demo\CICS` and `%ORACLE_HOME%\dg4appc\demo\IMS.` A `README.doc` file in each directory provides detailed information about installing and using the samples. JCL files for compiling and linking the sample programs are provided as well. The samples included with the gateway assist you with the following:

- Creating and initializing the commit-confirm transaction log databases and defining those databases to the OLTP. For Transaction Server for z/OS, the sample uses a VSAM file for the log database. For IMS/TM, a SHISAM/VSAM database is used.

- Using subroutines for receiving the Oracle Global Transaction ID from the gateway and logging it into the commit-confirm transaction log database. These subroutines are provided in the `pgacclg.asm` files. They can be used in the applications to reduce the complexity of the code changes to the programs.

  For Transaction Server for z/OS, the subroutine provided is called using the EXEC CICS LINK interface. For IMS/TM, the subroutine provided is called using the standard `CALL` statement or its equivalent in the application's programming language. Both these subroutines are written in 370 assembler to eliminate any interlanguage interface complexities and compiler dependencies.

- Installing forget/recovery transactions in the OLTP. These are provided in the `pgareco.asm` files. Forget/recovery transactions must be accessible through APPC so that the gateway can call them to forget a transaction once it has been successfully committed, and to query a transaction's state during recovery processing. These transactions delete the entry for a particular Oracle Global Transaction ID from the OLTP commit-confirm transaction log database during forget processing, and query the entry for a particular Oracle Global Transaction ID from the OLTP commit-confirm transaction log database during recovery processing. For both Transaction Server for z/OS and IMS/TM, these transactions are written in 370 assembler.

- Using the sample commit-confirm transaction log databases and subroutines. For Transaction Server for z/OS, a sample DB2 update transaction, DB2C, is provided in the `pgadb2c.cob` file. This is a COBOL example that updates the DB2 sample `EMP` table. For IMS/TM, a sample DLI update transaction, PGAIMSU, is

provided in the `pgaimsu.cob` file. This is a COBOL example that updates the DLI sample `PARTS` database.

Performing Postinstallation Procedures

The following are optional steps that you can perform as necessary. Installation of the sample applications for your OLTP is recommended to help you to fully understand how the gateway works and how it interfaces with the OLTP.

Installing Sample Applications

The Oracle Database Gateway for APPC package contains sample PL/SQL procedures and OLTP transaction programs that demonstrate the capabilities of the gateway. Samples are provided for the following:

APPC/MVS

- z/OS dataset information

CICS Transaction Server for z/OS

- DB2 inquiry
- DB2 multirow inquiry
- DB2 update
- VSAM inquiry
- VSAM update
- DLI inquiry
- FEPI DB2 inquiry
- FEPI VSAM inquiry

IMS/TM

- IMS inquiry using IVTNO and IVTNV sample transactions
- IMS PARTS inquiry (CPI-C)
- IMS PARTS update (CPI-C)

Additional samples are added to the distribution media in later releases of the product. Wherever possible, the sample applications use the sample databases provided with the database products.

For this release, full documentation on installing and using the sample applications is available in the `README.doc` files in the following directories:

- `%ORACLE_HOME%\dg4appc\CICS\sample_CICS_applications.txt`
- `%ORACLE_HOME%\dg4appc\IMS\sample_IMS_applications.txt`
- `%ORACLE_HOME%\dg4appc\MVS\sample_MVS_applications.txt`

9

Gateway Configuration Using TCP/IP Communication Protocol

This chapter outlines the steps needed to configure the Oracle database for a gateway using TCP/IP for IMS Connect on your Microsoft Windows platform. It also provides the steps necessary to verify installation and configuration of the gateway and OLTP components. It contains the following sections:

- Before You Begin
- Preparing to Configure a Gateway Installation/Upgrade
- Configuring the Oracle Database : First Time Installation
- Upgrading or Migrating the Oracle Database from Previous Gateways
- Optional Configuration Steps to Permit Multiple Users
- Configuring TCP/IP for the Gateway
- Configuring the Gateway
- Loading the PGA_TCP_IMSC Table
- Verifying the Gateway Installation and OLTP Configuration
- Performing Postinstallation Procedures

Configuring the Oracle Database Gateway for APPC using TCP/IP support for IMS Connect involves working with the following components:

- the Oracle database
- your Windows system
- your network
- the OLTP

Before You Begin

This chapter requires you to input parameters unique to your system in order to properly configure the gateway and TCP/IP communications interface.

Refer to Appendix D, "Configuration Worksheet" for a worksheet listing the installation parameters you will need to know before you can complete the configuration process. Ask your network administrator to provide you with these unique parameter names before you begin.

Preparing to Configure a Gateway Installation/Upgrade

There are three ways to establish the gateway-Oracle database relationship when you are installing or upgrading/migrating the gateway:

- When the Oracle Database and the gateway are installed in the same ORACLE_HOME on the same machine;

- When the Oracle Database and the gateway are installed on separate machine; or

- When the Oracle Database and the gateway are on the same machine but in different directories

Depending on the location of your gateway and your Oracle database , you may need to transfer some of the gateway administrative files to the location where the Oracle database is installed.

Follow the instructions suitable to your combination of the gateway-Oracle database locations listed below.

When the Oracle Database and the gateway are installed in the same ORACLE_HOME on the same machine

You do not need to transfer files. Proceed to "Configuring the Oracle Database : First Time Installation" on page 9-3.

When the Oracle Database and the gateway are installed on separate machine

1. Locate the gateway administrative files in the gateway `%ORACLE_HOME%\dg4appc\admin` directory. All files in this directory that have the suffix `.sql`, `.pkh`, and `.pkb` must be copied into a similarly-named directory in the Oracle database 's Oracle home directory.

2. Now locate the gateway demo files and subdirectories in the gateway `%ORACLE_HOME%\dg4appc\demo` directory. Copy the `pgavsn.sql` and `pgaecho.sql` files into a similarly named directory in the Oracle database .

3. Copy the `pgaims.sql` file from the gateway Oracle home `%ORACLE_HOME%\dg4appc\demo\IMS` directory to your Oracle database 's Oracle home `%ORACLE_HOME%\dg4appc\demo\IMS` directory.

4. Optional Steps: If you want to run `IVTNV` and `IVTNO`, then you will need to copy the `ivtno.ctl`, `ivtnod.sql`, `ivtnv.ctl`, and `ivtnvd.sql` files into the Oracle database 's Oracle home `%ORACLE_HOME%\dg4appc\demo\IMS` directory as well. Make sure to generate the required TIPs and transfer them as well.

When the Oracle Database and the gateway are on the same machine but in different directories

You must change your gateway's Oracle home to the Oracle database 's Oracle home directory.

1. For example, if your gateway's Oracle home is set as follows:

```
C:\> echo %ORACLE_HOME%
C:\oracle\pga\11.2
```

and your server's Oracle home is located in the '\oracle\pga\11.2' directory, then you need to do the following:

```
C:\> SET ORACLE_HOME=C:\oracle\pga\11.2
```

2. Now, create the directories with the following commands:

```
C:\> cd %ORACLE_HOME%
C:\> mkdir dg4appc
C:\> mkdir dg4appc\admin
C:\> mkdir dg4appc\demo
C:\> mkdir dg4appc\demo\IMS
```

3. Use whatever file transfer mechanism is available on your system to copy all the `.sql`, `.pkh`, and `.pkb` files from the gateway's Oracle home `%ORACLE_HOME%\dg4appc\admin` directory to the Oracle database 's Oracle home `%ORACLE_HOME%\dg4appc\admin` directory.

4. You may also transfer the demo files from the gateway directory to the Oracle database directory. Copy the pgavsn.sql and `pgaecho.sql` files and directory recursively from the gateway Oracle home's `%ORACLE_HOME%\dg4appc\demo` directory to the Oracle database 's `%ORACLE_HOME%\dg4appc\demo` directory.

5. You may also copy the `pgaims.sql` file from the gateway Oracle home `%ORACLE_HOME%\dg4appc\demo\IMS` directory to the Oracle database Oracle home `%ORACLE_HOME%\dg4appc\demo\IMS` directory.

   Optional Steps: If you want to run `IVTNV` and `IVTNO`, then you will need to copy the `ivtno.ctl, ivtnod.sql, ivtnv.ctl,` and `ivtnvd.sql` files into the Oracle database 's Oracle home `%ORACLE_HOME%\dg4appc\demo\IMS` directory as well. Make sure to generate the required TIPs and transfer them as well.

Proceed to Following those steps, you may want to perform the

Configuring the Oracle Database : First Time Installation

Follow these steps to configure the Oracle database after installing the Oracle Database Gateway for APPC.

1. Ensure that the `UTL_RAW` PL/SQL package has been installed on the Oracle database . All PGAU-generated TIP specifications use `UTL_RAW`, which provides routines for manipulating raw data.

   a. Use SQL*Plus to connect to the Oracle database as user `SYS`.

   b. From SQL*Plus, enter the following command:

   ```
   SQL> DESCRIBE UTL_RAW
   ```

   The `DESCRIBE` statement produces output on the screen. If you browse through the output, then you should see some functions, including a compare function. If you do not see this output, then continue the `UTL_RAW` installation by performing Step d.

   If the `DESCRIBE` statement indicates success, then your Oracle database has `UTL_RAW` installed and you can proceed to Step 2.

   c. Use SQL*Plus to connect to the Oracle database as `SYS`.

   d. From SQL*Plus, run the utlraw.sql and `prvtrawb.plb` scripts in the Oracle database `%ORACLE_HOME%\rdbms\admin` directory, in the following order:

   ```
   C:\> cd %ORACLE_HOME%\rdbms\admin
   SQL> @utlraw.sql
   SQL> @prvtrawb.sql
   ```

2. Ensure that the DBMS_OUTPUT standard PL/SQL package is enabled on your Oracle database . The sample programs and installation verification programs on the distribution media use this standard package.

    a. If necessary, then use SQL*Plus to connect to the Oracle database as user SYS.

    b. Enter the following command:

```
SQL> DESCRIBE DBMS_OUTPUT
```

    The DESCRIBE statement produces output on your screen. If you browse through that output, then you should see some functions, including a put_line function.

    If you do not see this output, then you must create the DBMS_OUTPUT package. Refer to the *Oracle Database Rules Manager and Expression Filter Developer's Guide* for more information about creating the DBMS_OUTPUT package. After successful installation of the DBMS_OUTPUT package, issue the DESCRIBE statement.

    If the DESCRIBE statement indicates success, then your Oracle database has DBMS_OUTPUT created, and you can proceed to Step 3.

3. Install the UTL_PG PL/SQL package. All PGAU-generated TIP specifications use UTL_PG, which provides routines for performing numeric conversions to and from raw data.

    a. If necessary, then use SQL*Plus to connect to the Oracle database as user SYS.

    b. From SQL*Plus, run the utlpg.sql and prvtpgb.plb scripts in the Oracle database 's %ORACLE_HOME%\rdbms\admin directory, in the following order:

```
C:\> cd %ORACLE_HOME%\rdbms\admin
SQL> @utlpg.sql
SQL> @prvtpgb.plb
```

4. Install the Heterogeneous Services (HS) catalogs.

    a. If necessary, then use SQL*Plus to connect to the Oracle database as user SYS.

    b. Enter the following command:

```
SQL> DESCRIBE HS_FDS_CLASS
```

    The DESCRIBE statement produces output on the screen. If the DESCRIBE statement indicates success, then Heterogeneous Services catalogs have been created on your Oracle database and you can proceed to Step 5.

    If the DESCRIBE statement does not indicate success, then you must create Heterogeneous Services catalogs and you must perform Step c below:

    c. If it is necessary to create the Heterogeneous Services catalog, then enter the following command:

```
C:\> cd %ORACLE_HOME%\rdbms\admin
SQL> @caths.sql
```

5. Create a public database link to access the Oracle Database Gateway for APPC:

    Use SQL*Plus to connect to the Oracle database as user SYSTEM. You can use the following SQL*Plus sample whether the Oracle database and the gateway are on the same system or on different systems. In the following sample, pgasrv is the

tns_name_entry that will be assigned to the gateway when you modify the `tnsnames.ora` file later in this chapter.

```
SQL> CREATE PUBLIC DATABASE LINK PGA USING 'PGASRV'
```

6. Create the gateway administrator user `PGAADMIN` and install the PG DD.

    a. Use SQL*Plus to connect to the Oracle database as user `SYSTEM`.

    b. From SQL*Plus, run the `pgacr8au.sql` script in the `%ORACLE_HOME%\dg4appc\admin` directory. This script creates the `PGAADMIN` user ID.

        The initial password defined for `PGAADMIN` is `PGAADMIN`. Use the `ALTER USER` command to change the password. For further information about password issues, refer to the *Oracle Database SQL Language Reference*.

```
C:\> cd %ORACLE_HOME%\dg4appc\admin
SQL> @pgacr8au.sql
```

    c. Use SQL*Plus to connect to the Oracle database as user `PGAADMIN`.

    d. From SQL*Plus, run the `pgddcr8.sql` script in the `%ORACLE_HOME%\dg4appc\admin` directory. This script installs the PG DD.

```
C:\> cd %ORACLE_HOME%\dg4appc\admin
SQL> @pgddcr8.sql
```

    e. From SQL*Plus, connect to the Oracle database as user `SYS`.

    f. Grant execution privileges on `DBMS_PIPE` to `PGAADMIN`:

```
SQL> GRANT EXECUTE ON DBMS_PIPE TO PGAADMIN
```

7. Ensure that the `pg4tcpmap` package has been installed on your Oracle database . Follow Steps a through c to test for proper installation of `pg4tcpmap`.

Refer to "Output for the pg4tcpmap Tool" in Appendix B for a sample of the output from the `pg4tcpmap` tool, and refer to Chapter 6 of the *Oracle Database Gateway for APPC User's Guide* for details about the commands needed to run the tool.

    a. Use SQL*Plus to connect to the Oracle database as user `SYSTEM`.

    b. Enter the following command:

```
SQL> select table_name, owner
from dba_tables
where table_name = 'PGA_TCP_IMSC',
and owner = 'PGAADMIN';
SQL> column owner format a 10
SQL> column index_name format a 18
SQL> column table_name format a 14
SQL> select owner, index_name, table_name, uniqueness
from dba_indexes
where index_name = 'PGA_TCP_IMSC_IND';
```

        Both `SELECT` statements must produce one row each. Following is the result for the first select statement:

```
TABLE_NAME                      OWNER
----------------------------    -----------------------------
PGA_TCP_IMSC                    PGAADMIN
```

Following is the result of the second select statement:

```
OWNER          INDEX_NAME           TABLE_NAME       UNIQUENESS
----------     ------------------   --------------   ---------
PGAADMIN       PGA_TCP_IMSC_IND     PGA_TCP_IMSC     UNIQUE
```

If the select statements produce the preceeding output on the screen, then you can skip Step C. If the select statement produces no output or more than one row, then the result is not the same as the output described above, and it is necessary for you to perform Step c.

c. From SQL*Plus, run the `pgaimsc.sql` script in the Oracle database 's `%ORACLE_HOME%\dg4appc\admin` directory:

```
C:\> cd %ORACLE_HOME%\dg4appc\admin
SQL> @pgaimsc.sql
```

8. Install the TIP trace access PL/SQL routines. These routines require that the `DBMS_PIPES` standard PL/SQL package is installed and that `PGAADMIN` has execute privileges on it. For more information about `DBMS_PIPES`, refer to the *Oracle Database Rules Manager and Expression Filter Developer's Guide*.

a. If necessary, then use SQL*Plus to connect to the Oracle database as user `PGAADMIN`.

b. From SQL*Plus, run the `pgatiptr.sql` script in the `%ORACLE_HOME%\dg4appc\admin` directory. This script creates PL/SQL routines that can be called to read and purge trace information created by PGAU-generated TIP specifications. It also creates public synonyms for these routines. The script prompts you for the necessary user IDs and passwords.

```
C:\> cd %ORACLE_HOME%\dg4appc\admin
SQL> @pgatiptr.sql
```

9. Install the `GPGLOCAL` package. This package is required for compilation and execution of all PGAU-generated TIP specifications. TIP developers should be granted execute privileges on `GPGLOCAL` (refer to "Optional Configuration Steps to Permit Multiple Users" on page 9-8).

a. Use SQL*Plus to connect to the Oracle database as user `PGAADMIN`.

b. From SQL*Plus, run the `gpglocal.pkh` script in the `%ORACLE_HOME%\dg4appc\admin` directory. This script compiles the `GPGLOCAL` package specification.

```
C:\> cd %ORACLE_HOME%\dg4appc\admin
SQL> @gpglocal.pkh
```

c. From SQL*Plus, run the `gpglocal.pkb` script in the `%ORACLE_HOME%\dg4appc\admin` directory. This script compiles the `GPGLOCAL` package body.

```
C:\> cd %ORACLE_HOME%\dg4appc\admin
SQL> @gpglocal.pkb
```

> **Note:** **Recompile TIPs when changing communication protocol from SNA to TCP/IP:**
>
> If you have existing TIPs that were generated previously on a gateway using the SNA protocol and you want to use the new TCP/IP feature, then the TIPs will have to be regenerated by PGAU with mandatory `NLS_LANGUAGE` and Side Profile Settings. Specify the suitable ASCII character set in the `DEFINE TRANSACTION` command.
>
> This is due to the fact that the gateway assumes that the appropriate user exit in IMS Connect is being used, which would translate between the suitable ASCII and EBCDIC character sets.

Upgrading or Migrating the Oracle Database from Previous Gateways

Follow these instructions only if you have a previous version of the Oracle Database Gateway for APPC installed on your system and need to configure it for release 11.2 of the gateway.

1. Upgrade your Oracle Database Gateway for APPC to current version levels as follows:

    a. Use SQL*Plus to connect to the Oracle database as user `SYS`.

    b. Install the `UTL_RAW` package body. From SQL*Plus, run the `prvtrawb.plb` script from the `%ORACLE_HOME%\rdbms\admin` directory. This script upgrades the `UTL_RAW` package body.

    ```
    C:\> cd %ORACLE_HOME%\rdbms\admin
    SQL> @prvtrawb.plb
    ```

    c. Install the `UTL_PG` package body. From SQL*Plus, run the `prvtpgb.plb` script from the `%ORACLE_HOME%\rdbms\admin` directory. This script upgrades the `UTL_PG` package body.

    ```
    C:\> cd %ORACLE_HOME%\rdbms\admin
    SQL> @prvtpgb.plb
    ```

The `prvtrawb.plb` and `prvtpgb.plb` scripts should complete successfully. If they fail because specifications do not exist or were invalidated, then consider reinstalling the package specifications as directed in Section , "If You Must Reinstall Package Specifications".

If You Must Reinstall Package Specifications

If the `UTL_RAW` or `UTL_PG` package has been invalidated or removed, then the `prvtrawb.plb` and `prvtpgb.plb` scripts may not complete successfully, then, you might have to reinstall the package specifications.

If you do reinstall the package specifications, then any dependent objects (such as existing user TIPs and client applications) are invalidated and will subsequently need to be recompiled. The impact of this is a one-time performance delay while recompilation of the TIPs and dependent client applications proceeds.

> **Note:** Before proceeding with reinstallation of the package scripts, make sure that you are in the `%ORACLE_HOME%\dg4appc\admin` directory

TIPs were split into separate specification and body files in release 3.3 to avoid cascaded recompilations in later releases.

- Step 1: Run the following scripts before proceeding with the PGAU upgrade:

    a. If necessary, then use SQL*Plus to connect to the Oracle database as user `SYS`.

    b. From SQL*Plus, run the `utlraw.sql` and `utlpg.sql` scripts in the Oracle database `%ORACLE_HOME%\rdbms\admin` directory, in the following order, to upgrade their respective package specifications:

    ```
    C:\> cd %ORACLE_HOME%\rdbms\admin
    SQL> @utlraw.sql
    SQL> @utlpg.sql
    ```

- Step 2: Repeat installation of `UTL_RAW` and `UTL_PG` package body

After the scripts have run, repeat Steps 1a and 1b in Section , "Upgrading or Migrating the Oracle Database from Previous Gateways" and then, proceed to Section , "Upgrading PGAU from Previous Gateway Releases".

> **Note:** TIPs and dependent client applications must be recompiled after reinstallation of the package specifications. Refer to *Oracle Database Gateway for APPC User's Guide* for more information about compiling TIPs

Upgrading PGAU from Previous Gateway Releases

Upgrade the PG DD as follows before execution the new PGAU:

1. If necessary, then use SQL*Plus to connect to the Oracle database as user `PGAADMIN`.

2. From SQL*Plus, run the `pgddupgr.sql` script in the `%ORACLE_HOME%\dg4appc\admin` directory. This script upgrades the PG DD.

    ```
    C:\> cd %ORACLE_HOME%\dg4appc\admin
    SQL> @pgddupgr.sql
    ```

Optional Configuration Steps to Permit Multiple Users

The following configuration steps are optional. Perform these steps if you want to allow users other than `PGAADMIN` to perform PG DD operations using PGAU.

1. Create public synonyms for the PG DD to allow other users to access the tables:

    a. Use SQL*Plus to connect to the Oracle database as user `SYSTEM`.

    b. From SQL*Plus, run the `pgddcr8s.sql` script in the `%ORACLE_HOME%\dg4appc\admin` directory. This script creates public synonyms for the PG DD.

    ```
    C:\> cd %ORACLE_HOME%\dg4appc\admin
    SQL> @pgddcr8.sql
    ```

2. Create roles for accessing the PG DD, performing definitions of transactions, and generating TIP specifications. The `PGAADMIN` user can grant these roles to other users as necessary.

   a. Use SQL*Plus to connect to the Oracle database as user `PGAADMIN`.

   b. From SQL*Plus, run the `pgddcr8r.sql` script in the `%ORACLE_HOME%\dg4appc\admin` directory. This script creates two roles, `PGDDDEF` and `PGDDGEN`. The `PGDDDEF` role provides `SELECT`, `INSERT`, `UPDATE`, and `DELETE` privileges against some of the PG DD tables, and select privileges against others, and allows execution of the `PGAU DEFINE`, `GENERATE`, `REDEFINE`, `REPORT`, and `UNDEFINE` statements. The `PGDDGEN` role provides select privileges against the PG DD tables, and allows execution of the `PGAU GENERATE` and `REPORT` statements only.

   ```
   C:\> cd %ORACLE_HOME%\dg4appc\admin
   SQL> @pgddcr8r.sql
   ```

3. Grant access to PGA required packages.

   TIP developers require access to the following PL/SQL packages, which are shipped with the Oracle database :

   - `DBMS_PIPE` in the `%ORACLE_HOME%\rdbms\admin` directory

   - `UTL_RAW` in the `%ORACLE_HOME%\rdbms\admin` directory

   - `UTL_PG` in the `%ORACLE_HOME%\rdbms\admin` directory

   Explicit grants to execute these packages must be made to TIP developers.

   These grants can be private, as in the following example:

   ```
   C:\> sqlplus SYS\pw@database_specification_string
   SQL> GRANT EXECUTE ON UTL_RAW TO tip_developer;
   SQL> GRANT EXECUTE ON UTL_PG TO tip_developer;
   SQL> GRANT EXECUTE ON DBMS_PIPE TO tip_developer;
   SQL> CONNECT PGAADMIN\pw@database_specification_string
   SQL> GRANT EXECUTE ON PGAADMIN.PURGE_TRACE TO tip_developer;
   SQL> GRANT EXECUTE ON PGAADMIN.READ_TRACE TO tip_developer;
   SQL> GRANT EXECUTE ON PGAADMIN.GPGLOCAL TO tip_developer;
   SQL> exit
   ```

   Alternatively, these grants can be public, as in the following example:

   ```
   C:\> sqlplus SYS\pw@database_specification_string
   SQL> GRANT EXECUTE ON UTL_RAW TO PUBLIC;
   SQL> GRANT EXECUTE ON UTL_PG TO PUBLIC;
   SQL> GRANT EXECUTE ON DBMS_PIPE to PUBLIC;
   SQL> CONNECT PGAADMIN\pw@database_specification_string
   SQL> GRANT EXECUTE ON PGAADMIN.PURGE_TRACE TO PUBLIC;
   SQL> GRANT EXECUTE ON PGAADMIN.READ_TRACE TO PUBLIC;
   SQL> GRANT EXECUTE ON PGAADMIN.GPGLOCAL TO PUBLIC;
   SQL> EXIT
   ```

   You can use either private or public grants. Both are sufficient for using PGA. Public grants are easier and can be performed now. If you use private grants, then they must be issued each time a new TIP developer user ID is created.

   SQL scripts for performing these grants are provided in the `%ORACLE_HOME%\dg4appc\admin` directory. The `pgddapub.sql` script performs these grants for public access to the packages. The `pgddadev.sql`

script performs the grants for private access to the packages by a single TIP developer. If you are going to use private grants, then you must run the `pgddadev.sql` script once for each TIP developer's user ID:

a. Use SQL*Plus to connect to the Oracle database as user `PGAADMIN`.

b. From SQL*Plus, run the suitable script (`pgddapub.sql` or `pgddadev.sql`) from the `%ORACLE_HOME%\dg4appc\admin` directory. The script performs the necessary grants as described earlier. You are prompted for the required user IDs, passwords, and database specification strings. If you are using private grants, then repeat this step for each user ID requiring access to the packages.

```
C:\> cd %ORACLE_HOME%\dg4appc\admin
SQL> @pgddapub.sql
```

or

```
SQL> @pgddadev.sql
```

4. If you are upgrading from a previous release of the gateway when the communication protocol was SNA, to the current gateway using TCP/IP, and if you want to upgrade your existing TIPs with new function and maintenance, then regenerate existing TIP specifications using the PGAU `GENERATE` statement.

> **Note:** The Procedural Gateway Administrative Utility (PGAU) has been enhanced to automatically upgrade existing PG DD entries with a new attribute when a PGAU `GENERATE` command is executed. To support this enhancement, add a new privilege to the `PGDDGEN` role. To do this, as the `PGAADMIN` user, use SQL*Plus to connect to the Oracle database where the PG DD is stored. Then, issue the following SQL command:
>
> ```
> SQL> GRANT INSERT ON PGA_DATA_VALUES TO PGDDGEN
> ```

a. Call PGAU in the directory path where the PGAU control files are generated and where TIPs are stored:

```
C:\> pgau
PGAU> CONNECT PGAADMIN\pgaadmin@database_specification_string
PGAU> GENERATE tranname
PGAU> EXIT
```

For more information about the `GENERATE` command, refer to the PGAU GENERATE section in Chapter 2 of the *Oracle Database Gateway for APPC User's Guide*.

Note that it is not necessary to define the PG DD entries again.

5. Call SQL*Plus in the same directory path where the newly-generated TIP specifications are stored.

```
C:\> sqlplus tip_owner\pw@database_specification_string
SQL> @tipname.pkh
SQL> @tipname.pkb
SQL> exit
```

PGAU `GENERATE` produces the TIP in two output files, a specification and a body. You must compile both, first the specification and then the body.

For more information about the `GENERATE` command, refer to the PGAU GENERATE command section in Chapter 2 of the *Oracle Database Gateway for APPC User's Guide*.

Configuring TCP/IP for the Gateway

You must now configure the TCP/IP for IMS Connect communication package profiles for TCP/IP connections.

Configure the profiles to define the TCP/IP conversations with the OLTP.

When you have finished configuring your communications package, return to the following section, Section , "Configuring the Gateway" on page 9-11.

Configuring the Gateway

To configure the gateway, follow the instructions given subsequently:

1. Tailor the gateway parameters.

   There are a number of parameters specific to the Oracle Database Gateway for APPC when it is using TCP/IP for IMS Connect. These are supplied in the gateway parameter file, init*sid*.ora, which is in the `%ORACLE_HOME%\dg4appc\admin` directory. A sample gateway parameter file, `initPGA.ora`, is provided in this subdirectory.

   Note: In the init*sid*.ora file, substitute your gateway SID name for "*sid*" in this file name.

   The init*sid*.ora file contains both APPC and TCP/IP parameters, separated by a description. You must modify the init*sid*.ora file by deleting the APPC parameters. Refer to Appendix B, "Gateway Initialization Parameters for TCP/IP Communication Protocol" for the valid TCP/IP parameters.

   The parameters fall into two categories:

   - gateway initialization parameters

     These parameters control the general operation of the gateway in the Oracle environment.

     Important: Before performing the following step, refer to Appendix B, "Gateway Initialization Parameters for TCP/IP Communication Protocol" for information about tailoring gateway initialization and PGA parameters. Pay special attention to the information about using the `PGA_CAPABILITY` parameter.

   - PGA parameters

     PGA parameters control the TCP/IP interface portion of the gateway. Use the `SET` gateway initialization parameter to specify PGA parameters. Oracle recommends that you group all `SET` commands for PGA parameters at the end of the init*sid*.ora file.

> **Note:** Misspelled parameters are ignored. However, if the
> `%ORACLE_HOME%\dg4appc\admin\initsid.ora` file is missing,
> then all calls to the gateway fail and return a `PGA-20928` error.

Loading the PGA_TCP_IMSC Table

Gateway users who wish to employ the TCP/IP protocol do so by using the
`pg4tcpmap` tool.

The `pg4tcpmap` tool resides on the gateway. Its function is to map the side profile
name to TCP/IP and IMS Connect attributes. You must run this tool before executing
the PL/SQL gateway statements
(such as `%ORACLE_HOME%\dg4appc\demo\IMS\pgaims.sql`).

In PGAINIT TIP, for example, the user must specify a side profile name. The SNA
protocol recognizes and uses the parameter. In this release of the gateway, the
`pg4tcpmap` tool uses the original PGAINIT TIP format to map the relevant SNA
parameters to TCP/IP. The `pg4tcpmap` tool inserts the values of these parameters into
a table called `PGA_TCP_IMSC`.

Before executing `pg4tcpmap`, you must specify the `ORACLE_HOME`, Oracle SID and
modify init*sid*.ora. Refer to Appendix B, "Gateway Initialization Parameters for
TCP/IP Communication Protocol" in this guide and Chapter 6 in the *Oracle Database
Gateway for APPC User's Guide* for complete information about the `pg4tcpmap`
commands.

Chapter 6 of the *Oracle Database Gateway for APPC User's Guide* contains a list of
`pg4tcpmap` commands and instructions for using them, as well as an example of the
table. Refer to "Troubleshooting" in the *Oracle Database Gateway for APPC User's Guide*
for information about the trace file for the executed `pg4tcpmap` tool.

To operate this tool, execute the following command:

```
C:\> cd %ORACLE_HOME%\bin
C:\> pg4tcpmap
```

Refer to "Output for the pg4tcpmap Tool" on page B-3 for a sample of the `pg4tcpmap`
output.

Verifying the Gateway Installation and OLTP Configuration

To verify the gateway installation and the OLTP configuration, perform the following
procedures after installing the gateway.

> **Note:** If your database link name is not "PGA," modify the
> demonstration `.sql` files to give them the particular database link
> name that you created in Step 5 of "Configuring the Oracle
> Database : First Time Installation" on page 9-3. You must modify
> the following .sql files:
>
> - `pgavsn.sql`
>
> - `pgaecho.sql`
>
> - `pgaims.sql`

Verifying the Gateway Installation

To verify the gateway software installation using the database link PGA previously created, perform the following steps:

1. Using SQL*Plus, connect to your Oracle database as user `PGAADMIN`.

2. Run `%ORACLE_HOME%\dg4appc\demo\pgavsn.sql`.

    ```
    C:\> cd %ORACLE_HOME%\dg4appc\demo
    SQL> @pgavsn.sql
    ```

    The server version number banner appears at your terminal. You will receive the following output:

    ```
    Oracle Database Gateway for APPC (extension TCP/IP for IMS connect).
    Version
    11.2.0.0.2 Thu Jun 11 14:52:36 2009

    Copyright (c) Oracle Corporation 1979,
    2009.  All rights reserved.


    PL/SQL procedure successfully completed.
    ```

3. Run `%ORACLE_HOME%\dg4appc\demo\pgaecho.sql`.

    ```
    C:\> cd %ORACLE_HOME%\dg4appc\demo
    SQL> @pgaecho.sql
    ```

    You will receive the following output:

    ```
    ==> Congratulations, your installation was successful. <==
    ```

Verifying the OLTP Configuration

Use the following procedure to verify the OLTP configuration.

> **Attention:**   If you have not completed the file transfers detailed in "Preparing to Configure a Gateway Installation/Upgrade" on page 9-2,then complete them now, before proceeding to the next step.

IMS/TM Verification

Perform the following steps to verify the IMS/TM configuration. Be certain that you have installed and configured the IMS Connect and that it is up and running before you begin this procedure. Refer to the IBM *IMS Connect Guide and Reference* for information about how to perform the installation and configuration tasks.

> **Note:** TIPs must be recompiled when changing the communication protocol to TCP/IP.
>
> TCP/IP only: If you have existing TIPs that were generated previously on a gateway using the SNA protocol and you want to use the new TCP/IP feature, then the TIPs will have to be regenerated by PGAU with mandatory `NLS_LANGUAGE` and Side Profile Settings. Specify the suitable ASCII character set in the `DEFINE TRANSACTION` command.
>
> This is because the gateway assumes that the appropriate user exit in IMS Connect is being used, which would translate between the suitable ASCII and EBCDIC character sets.

1. To verify that the `FLIP` transaction is installed correctly, log on to your IMS/TM system and enter the following transaction (replacing `FLIP` with the transaction ID you chose for `FLIP` when you configured the IMS/TM system for the gateway):

   ```
   FLIP THIS MESSAGE
   ```

   The following output appears on your terminal:

   ```
   EGASSEM SIHT PILF
   ```

   > **Attention:** If you have not completed the file transfers detailed in Section "Preparing to Configure a Gateway Installation/Upgrade" on page 9-2, then complete them now, before proceeding to the next step.

2. From your Windows platform, modify the `pgaims.sql` file, which resides at `%ORACLE_HOME%\dg4appc\demo\IMS\pgaims.sql.` Customize the following three items used for accessing the gateway and the IMS/TM system, as described in the comments at the beginning of the file:

   - the IMS/TM transaction ID
   - the side profile name
   - the logmode entry name

3. Ensure that the TCP/IP communication protocol on your system has been started.

4. Using SQL*Plus, connect to your Oracle database from `PGAADMIN`.

5. Run `pgaims.sql`.

   ```
   C:\> cd %ORACLE_HOME%\dg4appc\demo\IMS
   SQL> @pgaims.sql
   ```

   The following message appears:

   ```
   ==> Congratulations, your gateway is communicating with IMS/TM <==
   ```

   Your IMS/TM installation verification is now complete.

Performing Postinstallation Procedures

The following are optional steps that you can perform as necessary. Installation of the sample applications for the OLTP is recommended to help you to fully understand how the gateway works and how it interfaces with the OLTP.

Installing Sample Applications

Your Oracle Database Gateway for APPC featuring TCP/IP for IMS Connect contains sample PL/SQL procedures and OLTP transaction programs that demonstrate the gateway's capabilities.

> **Note:** When you are calling a gateway that is using TCP/IP as the communication protocol, and you are using EBCDIC as the language in the control files, you must change the language from EBCDIC to ASCII. Some examples of control files that may be in EBCDIC language are `ivtno.ctl` and `ivtnv.ctl`.
>
> For more information, refer to the `%ORACLE_HOME%\dg4appc\demo\IMS\ivtno.ctl` and
>
> `%ORACLE_HOME%\dg4appc\demo\IMS\ivtnv.ctl` files.

Samples are provided for IMS/TM:

- IMS inquiry using `IVTNO` and `IVTNV` sample transactions

Additional samples are added to the distribution media in later releases of the product. Wherever possible, the sample applications use the sample databases provided with the database products.

For this release, full documentation on installing and using the sample applications is available in the `README.doc` files in the following directory:

`%ORACLE_HOME%\pg4appc\demo\IMS\sample_IMS_applications.txt`

10

Security Requirements

The gateway architecture involves multiple systems, database servers, and communications facilities, each having distinct security capabilities and limitations. To effectively plan and implement your security scheme, you must understand these capabilities and limitations, in addition to knowing your installation's security requirements.

Read this chapter to learn about the capabilities and limitations of the Oracle Database Gateway for APPC. This chapter contains the following sections:

- Overview of Security Requirements
- Authenticating Application Logons
- Defining and Controlling Database Links
- Using SNA Security Validation
- TCP/IP Security
- Passwords in the Gateway Initialization File

Overview of Security Requirements

Before implementing the security scheme, you must understand the existing security requirements and expectations in the environment. Because you are enabling application access to different databases on different systems, you must merge multiple security cultures. When developing your security scheme, the most stringent security requirements prevail. When you connect different systems into an operating whole, the system with the strictest security requirements generally dictates what the other systems can and cannot do.

Gateway security includes two main concerns:

- users and applications that are permitted access to a particular gateway instance and OLTP
- OLTP transactions that users and applications are able to execute

You can control access at several points in the gateway architecture. The primary options are discussed in the following sections. Control over remote transaction program (RTP) access is provided by each OLTP with native authorization mechanisms based on user ID. These facilities are described in the product documentation for your OLTP. Information in this chapter includes how the gateway facilities determine the user ID that is in effect for a particular OLTP connection.

When the gateway is involved in an RPC request, security mechanisms are in effect for each system component encountered by the gateway. The first system component that

is encountered is the application tool or 3GL program. The last system component that is encountered is the OLTP.

Each of the following sections identifies the component and the type of security processing that is available in that component. Each section offers a summary of key features and parameters. Refer to product-specific documentation for detailed information about the non-gateway components for Oracle and non-Oracle products.

Authenticating Application Logons

An application must connect to an Oracle database before using that gateway. The type of logon authentication that you use determines the resulting Oracle user ID and can affect gateway operation.

Two basic types of authentication are available:

1. Oracle authentication.

   With Oracle authentication, each Oracle user ID has an associated password that is known to Oracle. When an application connects to the server, it supplies a user ID and password. Oracle confirms that the user ID exists and that the password matches the one stored in the database.

2. Operating system authentication

   With operating system authentication, the server's underlying operating system is responsible for authentication. An Oracle user ID that is created with the IDENTIFIED EXTERNALLY attribute (instead of a password) is accessed with operating system authentication. To log on to such a user ID, the application supplies a slash (/) for a user ID and does not supply a password.

   To perform operating system authentication, the server determines the requester's operating system user ID, optionally adds a fixed prefix to it, and uses the result as the Oracle user ID. The server confirms that the user ID exists and is identified externally, but no password checking is done. The underlying assumption is that users were authenticated when they logged on to the operating system.

   Operating system authentication is not available on all platforms and is not available in some Oracle Net (client-server) and multi-threaded server configurations. Refer to your Windows Oracle database documentation and *Oracle Database Net Services Administrator's Guide* to determine the availability of this feature in your configuration.

For more information about authenticating application logons, refer to the *Oracle Database Administrator's Guide*.

Defining and Controlling Database Links

The following sections discuss database links for users of the gateway employing either TCP/IP or SNA communications protocols.

Link Accessibility

The first point of control for a database link is whether it is accessible to a given user. A public database link can be used by any user ID. A private database link can be used only by the user who created it. Database link usability is determined by its ability to open a session to the gateway. The Oracle database makes no distinction as to the type of use (such as read-only and update or write) or which remote objects can be accessed. These distinctions are the responsibility of the OLTP that is accessed.

Links and CONNECT Clauses

The CONNECT clause is another security-related attribute of a database link. You can use the CONNECT clause to specify an explicit user ID and password, which can differ from the user's Oracle user ID and password. This CONNECT user ID and password combination is sent to the gateway when the database link connection is first opened. Depending on gateway-specific options, the gateway might send that user ID and password to the OLTP to be validated.

If a database link is created without a CONNECT clause using Oracle authentication, then the user's Oracle user ID and password are sent to the gateway when the connection is opened. If the user logs on to the Oracle database with operating system authentication, then the gateway receives no user ID or password from the Oracle database . It is impossible for operating system-authenticated Oracle users to use a gateway database link defined without a CONNECT clause. However, if your OLTP provides user ID mapping facilities based on the gateway LU name from which the user is connecting, then such a connection is possible if all users on the same gateway instance can use the same OLTP user ID.

For more information about database links, refer to the *Oracle Database Administrator's Guide*.

Using SNA Security Validation

The information in this section applies only to the security needs of gateway users employing the SNA communications protocol. When an RPC request to start a remote transaction program is received by the gateway, the gateway attempts to start an APPC conversation with the OLTP. Before the conversation can begin, a session must start between Windows' Logical Unit (LU) and the OLTP LU.

APPC support for the Windows platform is provided by the SNA Server (Microsoft Host Integration Server or IBM Communication Server v6.1.1 or higher).

SNA and its various access method implementations, including VTAM and the SNA Server, provide security validation at session initiation time, allowing each LU to authenticate its partner. This validation is carried out entirely by network software before the gateway and OLTP application programs begin their conversation and process conversation-level security data. If session-level security is used, then correct password information must be established in Microsoft Windows' SNA profiles and in similar parameter structures in the OLTP to be accessed. Refer to the suitable communications software product documentation for detailed information about this subject.

Specifying SNA Conversation Security

The PGA_SECURITY_TYPE parameter of the gateway initialization file allows you to specify one of three options that determine the security conduct of the LU6.2 conversation that is allocated with the OLTP. These options are part of the SNA LU6.2 architecture, but their precise behavior might vary depending on the particular OLTP system.

SNA Security Option SECURITY=NONE

If you specify PGA_SECURITY_TYPE=NONE, then the gateway performs no processing of the client user ID and password. The conversation is allocated with SNA option SECURITY=NONE.

SNA Security Option SECURITY=PROGRAM

If you specify `PGA_SECURITY_TYPE=PROGRAM`, then the gateway allocates the conversation with SNA option `SECURITY=PROGRAM`, and the following information is sent to the OLTP:

- If the TIP user ID and password overrides are used, then the specified user ID and password are sent regardless of the database link specification.

- If the database link has explicit `CONNECT` information, then the specified user ID and password are sent.

- If the database link has no `CONNECT` clause, and if the application logged on to Oracle with an explicit user ID and password, then the Oracle user ID and password are sent.

- If the application logs on to Oracle with operating system authentication, and if the database link lacks explicit `CONNECT` information, then no user ID and password are sent. If no user ID and password are sent, and if the OLTP is not configured to assign a default user ID, then the connection fails.

In general, SNA option `SECURITY=PROGRAM` tells the OLTP to authenticate the user ID/password combination using whatever authentication mechanisms are available. For example, if CICS Transaction Server for z/OS is the OLTP, then RACF can be used. This is not always the case, however, because each OLTP can be configured to process inbound user IDs in other ways.

TCP/IP Security

The security information in this section applies only to users of the Oracle Database Gateway for APPC using the TCP/IP for IMS Connect feature. When an RPC request to start a RTP is received by the gateway, the gateway attempts to start the TCP/IP conversation with IMS Connect. IMS Connect would contact the OLTP (IMS) through OTMA and XCF. Refer to the IBM *IMS Connect Guide and Reference* for more information. The conversation between the gateway and IMS Connect occurs when the network uses the TCP/IP address or host name and port number to connect from the gateway to IMS Connect.

> **Note:** Because the gateway is using PGAU to generate TIPs, the TIPs contain SNA information. When using the Oracle Database Gateway for APPC with TCP/IP support for IMS Connect, you need to map the SNA names to the TCP/IP host name and port number in order for the gateway to communicate with IMS Connect. Use the `pg4tcpmap` tool to map the information from SNA to TCP/IP. Refer to Chapter 6, "pg4tcpmap Commands," of the *Oracle Database Gateway for APPC User's Guide* for more information.

IMS Connect provides validation at session initiation time, allowing each connection to authenticate its partner. This validation is carried out entirely by network software before the gateway and OLTP application programs at IMS begin their conversation and process conversation-level security data. If session-level security is used, then correct password information must be established in Microsoft Windows and in similar parameter structures in the OLTP to be accessed.

Specifying TCP/IP Conversation Security

The `PGA_SECURITY_TYPE` parameter of the gateway initialization file allows you to specify the security conduct for the conversation that is allocated by the gateway for OLTP. Refer to Appendix B, "Gateway Initialization Parameters for TCP/IP Communication Protocol".

TCP/IP Security Option SECURITY=NONE

If you specify `PGA_SECURITY_TYPE=NONE`, then the gateway performs no processing of the client user ID and password.

TCP/IP Security Option SECURITY=PROGRAM

If you specify `PGA_SECURITY_TYPE=PROGRAM`, then the following information is sent to the OLTP:

- If the TIP user ID and password overrides are used, then the specified user ID and password are sent regardless of the database link specification.

- If the database link has explicit `CONNECT` information, then the specified user ID and password are sent.

- If the database link has no `CONNECT` clause, and if the application logged on to Oracle with an explicit user ID and password, then the Oracle user ID and password are sent.

- If the application logs on to Oracle with operating system authentication, and if the database link lacks explicit `CONNECT` information, then no user ID and password are sent. If no user ID and password are sent, and if the OLTP is not configured to assign a default user ID, then the connection fails.

RACF is the only authentication mechanism available when the Oracle Database Gateway for APPC using TCP/IP for IMS Connect communicates with IMS Connect.

> **Important:** You must specify your RACF group name through the `pg4tcpmap` tool if you have set your PGA security option to `SECURITY=PROGRAM`. For more information about this issue, refer to the *Oracle Database Gateway for APPC User's Guide*.

Passwords in the Gateway Initialization File

Initialization parameters may contain sensitive information, such as user IDs or passwords. Initialization parameters are stored in plain text files and may be deemed insecure. An encryption feature has been added to Heterogenous Services making it possible to encrypt parameters values. This is done through the `dg4pwd` utility.

> **See Also:** Refer to *Oracle Database Heterogeneous Connectivity User's Guide* for more information about this utility

11

Migration from Existing Gateways

Migrating to new instances of the Oracle Database Gateway for APPC from an existing installation is straightforward, provided you follow some guidelines. This chapter provides information to make these new installations as easy as possible. It also provides the parameters you will need if you are using the TCP/IP for IMS Connect communication protocol on your gateway.

This chapter provides information that is specific to this release of the Oracle Database Gateway for APPC for Microsoft Windows. It contains the following sections:

- Migrating an Existing Gateway Instance to New Release Using SNA Protocol
- Migrating from an Existing Gateway Using SNA to TCP/IP

Migrating an Existing Gateway Instance to New Release Using SNA Protocol

Follow these steps to migrate an existing gateway to Release 11.2 of the gateway using the SNA communication protocol.

Note that if you are using the gateway's TCP/IP support for IMS Connect, you will not be migrating an existing release to the current release of the gateway. However, you will need to place valid Heterogeneous Services parameters into your init*sid*.ora file. Proceed to "Parameter Changes: Version 4 to Release 11.2 of the Gateway" on page 11-2.

Step 1: Install the New Release

Install the new release of the gateway in a separate directory as outlined in Chapter 4, "Installing the Gateway".

> **Caution:** Do not install the gateway over an existing gateway installation. Doing so will corrupt the existing installation.

Step 2: Transfer init*sid*.ora Gateway Initialization File Parameters

Copy the init*sid*.ora file from the old gateway instance to the new instance.

Note that if you are migrating from Release 9.0.1 or earlier of the gateway, then `PGA_TRACE` is not supported, you will need to modify the parameter to `TRACE_LEVEL`, instead.

> **Note:** If you are using `TRACE_LEVEL`, then you must set the path for the `LOG_DESTINATION` parameter.

Backout Considerations When Migrating to New Releases

Oracle recommends that you keep the old gateway Oracle home directory and instance configurations intact and operational when you are installing a new release of the gateway and upgrading existing instances, in case there are problems with the upgrade. This will help ensure minimal downtime between changes to different gateway instances.

Oracle Net Considerations

The Oracle Database Gateway for APPC uses the Heterogeneous Services (HS) facilities of Oracle and Oracle Net. If you are upgrading from a version 4 gateway, then you need to slightly modify the gateway service name entries in the `tnsnames.ora` file. Add an `(HS=)` clause to communicate to Oracle Net that the gateway uses HS facilities. For more information, refer to "Configuring the Oracle Network for the Gateway" on page 5-1.

Parameter Changes: Version 4 to Release 11.2 of the Gateway

This release of the Oracle Database Gateway for APPC introduces new and changed initialization parameters if you are migrating from version 4 gateway to release 11.2 of the gateway.

> **Note:** This section does not apply to you if you are migrating to release 11.2 from version 8 of the Oracle Database Gateway for APPC.

If you are using the gateway's TCP/IP support for IMS Connect, then you will not be migrating from version 4 to the current release of the gateway. However, you will need to place valid Heterogeneous Services parameters into your `init`*sid*.ora file.

This chapter contains references to the particular HS parameters you need to run the gateway.

> **Note:** Refer to the *Oracle Database Heterogeneous Connectivity User's Guide* for a complete list and descriptions of all HS parameters used in Oracle products.

New Gateway Initialization Parameters

The following parameters are in the gateway initialization file (`init`*sid*.ora):

- `FDS_CLASS`
- `FDS_INSTANCE`
- `HS_FDS_FETCH_ROWS`
- `LOG_DESTINATION`
- `TRACE_LEVEL`
- `PGA_TCP_DB` (TCP/IP only)

- `PGA_TCP_USER` (TCP/IP only)

- `PGA_TCP_PASS` (TCP/IP only)

> **Note:** The `HS_` parameters are specific to Oracle Heterogeneous Services. For details on HS parameters, refer to the *Oracle Database Administrator's Guide*.

Renamed Gateway Initialization File Parameters

Following is a list of the gateway initialization file (init*sid*.ora) parameters that have been renamed in this release of the gateway, the parameters' former names are shown in parentheses.

- `HS_COMMIT_STRENGTH_POINT` (`COMMIT_STRENGTH_POINT`)

- `HS_DB_DOMAIN` (`DB_DOMAIN`)

- `HS_DB_INTERNAL_NAME` (`DB_INTERNAL_NAME`)

- `HS_DB_NAME` (`DB_NAME`)

- `HS_DESCRIBE_CACHE_HWM` (`DESCRIBE_CACHE_HWM`)

- `HS_LANGUAGE` (`LANGUAGE`)

- `HS_NLS_DATE_FORMAT` (`NLS_DATE_FORMAT`)

- `HS_NLS_DATE_LANGUAGE` (`NLS_DATE_LANGUAGE`)

- `HS_OPEN_CURSORS` (`OPEN_CURSORS`)

- `HS_ROWID_CACHE_SIZE` (`ROWID_CACHE_SIZE`)

Obsolete Parameters

The following parameters are now obsolete. Please remove them from your configuration files:

- `MODE`

- `SERVER_PATH`

- `ERROR_LOGGING`

- `ERROR_REPORTING`

- `ERRORTAG`

- `GATEWAY_SID`

- `GROUP_BY_OFF`

- `GTWDEBUG`

- `INCREMENT_CURSORS`

- `INIT_CURSORS`

- `LIST`

- `MAX_LOG_SIZE`

- `OPTIMIZE_FILE_OPEN`

- `ORDER_BY_OFF`

- `RESOLVE_BINDS`

- `RETRY`

- `SET`

- `SNMP_SUPPORT`

- `SQL_TRACE`

- `TRIM_CURSORS`

- `D_OPEN_CURSORS`

- `D_INIT_CURSORS`

- `D_INCREMENT_CURSORS`

- `D_TRIM_CURSORS`

- `PGA_TRACE`

Parameter Changes: Version 8 or Earlier to Gateway Release 11.2

The following startup shell script parameter must be added to init*sid*.ora if you are migrating from version 4 or version 8 gateway to the current release of the Oracle Database Gateway for APPC:

- `FDS_CLASS_VERSION`

Migrating from Gateway Release 9.0.1 or 9.2.0 to Gateway Release 11.2

No new parameters were added between release 9.0.1 and this release of the gateway.

Migrating from an Existing Gateway Using SNA to TCP/IP

The following sections are for users who have an existing release of the gateway using the SNA protocol but who wish to switch to using TCP/IP support for IMS Connect.

The TCP/IP support for IMS Connect feature in this release of the gateway allows you to continue to use existing TIPs.

To Use Existing TIPs with Existing Side Profile Definitions

Follow these instructions:

1. Make sure you have used the `pg4tcpmap` tool to insert valid parameter values into the `PGA_TCP_IMSC` table.

   Refer to Chapter 9, "Gateway Configuration Using TCP/IP Communication Protocol" for instructions on loading the `PGA_TCP_IMSC` table.

2. Make sure that the `LANGUAGE` parameter in your TIPs is set to `american_america_us7ascii`.

3. Use PGAU to regenerate the IMS TIPs.

4. Add the following new TCP/IP parameters to the init*sid*.ora file:

   - `PGA_TCP_DB`

   - `PGA_TCP_USER`

   - `PGA_TCP_PASS`

   You will find descriptions and information about adding these parameters in Section , "Parameter Changes: Version 4 to Release 11.2 of the Gateway" on

page 11-2. You will also find descriptions of the parameters in Appendix B, "Gateway Initialization Parameters for TCP/IP Communication Protocol".

> **Note:** If your TIPs from a previous version of the gateway were already defined using a side profile name and the `NLS_LANGUAGE` parameter has been set to a value of `american_america_us7ascii`, then you will not need to recompile these TIPs. You will still need to map your parameter values using the pg4tcpmap tool.

> **Note:** TIPS must be recompiled when changing the communication protocol from SNA to TCP/IP.
>
> If you have existing TIPs that were generated previously on a gateway using the SNA protocol and you want to use the new TCP/IP feature, then the TIPs will have to be regenerated by PGAU with mandatory `NLS_LANGUAGE` and Side Profile Settings. Specify the suitable ASCII character set in the `DEFINE TRANSACTION` command.
>
> This is due to the fact that the gateway assumes that the suitable user exit in IMS Connect is being used, which would translate between the suitable ASCII and EBCDIC character sets.

A

Gateway Initialization Parameters for SNA Protocol

This appendix describes the gateway initialization file location and lists the gateway initialization parameters supported by the Oracle Database Gateway for APPC specifically for the SNA protocol. These parameters are fully documented in Section , "Migrating an Existing Gateway Instance to New Release Using SNA Protocol" on page 11-1. In addition, this appendix contains sample listener.ora and tnsnames files for a gateway using SNA.

This appendix contains the following section:

- Gateway Initialization Parameter File Using SNA Protocol

Gateway Initialization Parameter File Using SNA Protocol

The parameter file for the gateway is located in the `%ORACLE_HOME%\dg4appc\admin` directory and is called init*sid*.ora.

> **Note:** The init*sid*.ora file contains both SNA and TCP/IP parameters. You must modify these files with the suitable parameters.

PGA Parameters

The PGA parameters control the APPC interface portion of the gateway.

PGA parameters are specified using the `SET` gateway initialization parameter. For example:

```
SET pga_parm=value
```

where:

- `pga_parm` is one of the PGA parameter names in the list that follows; and
- `value` is a character string with contents that depend on `pga_parm`

Table A–1 provides a list of PGA parameters and their descriptions.

Table A–1   PGA Parameters for Gateway Using SNA

Parameter	Description
LOG_ DESTINATION=*logpath*	*logpath* specifies the destination at which STDERR is reopened. LOG_DESTINATION specifies a directory only and STDERR is reopened to logpath*sid_pid*.log
	where:
	■ *sid* is the sid name
	■ *pid* is the process ID assigned to the gateway
PGA_CAPABILITY	PGA transaction capability. This controls whether updates are allowed through the gateway. The following are valid values:
	READ_ONLY or RO, read-only capabilities.
	SINGLE_SITE or SS, single-site update only. This indicates that in a distributed environment, only the gateway can perform updates. No other database updates can occur within the Oracle transaction.
	COMMIT_CONFIRM or CC, commit-confirm. This indicates that in a distributed environment, updates can be performed by both the gateway and other participants within the Oracle transaction. The gateway is always committed first in this mode, and no other commit-confirm sites are allowed to participate in the Oracle transaction.
	The default is SINGLE_SITE.
PGA_CONFIRM	Incoming APPC CONFIRM request handling option. This controls what the gateway does when an APPC CONFIRM request is received from the remote transaction program (RTP). This parameter has meaning only when the conversation is running with SYNCLEVEL > 0. The following are valid values:
	ACCEPT - respond to incoming APPC CONFIRM requests with APPC CONFIRMED responses.
	REJECT - treat incoming APPC CONFIRM requests as errors causing the conversation to be deallocated and an error message to be issued.
	The default is REJECT.
PGA_LOG_DB	The Oracle Net service name for the Oracle database in which the gateway maintains its transaction log. This parameter can be from 1 to 255 characters long. This parameter is required only when PGA_CAPABILITY parameter is set to COMMIT_CONFIRM.
	There is no default value.
PGA_LOG_PASS	The Oracle password to be used by the gateway when connecting to the Oracle database specified by the PGA_LOG_DB parameter. The password can be from 1 to 30 characters long. This parameter is required only when PGA_CAPABILITY parameter is set to COMMIT_CONFIRM. The password can be encrypted. For more information about encrypting the password, refer to Section , "Passwords in the Gateway Initialization File".
	There is no default value.

Table A–1   (Cont.) PGA Parameters for Gateway Using SNA

Parameter	Description
PGA_LOG_USER	The Oracle user ID to be used by the gateway when connecting to the Oracle database specified by the PGA_LOG_DB parameter. The user ID can be from 1 to 30 characters long. This parameter is required only when PGA_CAPABILITY parameter is set to COMMIT_CONFIRM.
	There is no default value.
PGA_RECOVERY_PASS	The password to be used by the gateway when allocating an APPC conversation with the transaction specified by the PGA_RECOVERY_TPNAME parameter. The password can be from 1 to 8 characters long. This parameter is required only when PGA_CAPABILITY parameter is set to COMMIT_CONFIRM and PGA_SECURITY_TYPE parameter is set to PROGRAM. The password can be encrypted. For more information about encrypting the password, refer to Section , "Passwords in the Gateway Initialization File".
	There is no default value.
PGA_RECOVERY_TPNAME	The TP name of the transaction installed in the online transaction processor (OLTP) for commit-confirm FORGET and RECOVERY processing. The TP name can be from 1 to 64 characters long. For CICS Transaction Server for z/OS, the TP name is limited to four characters. For IMS/TM, the TP name is limited to eight characters. Other OLTPs might have other limits on the length of the TP name. This parameter is required only when PGA_CAPABILITY parameter is set to COMMIT_CONFIRM.
	The default value is RECO.
PGA_RECOVERY_USER	The user ID to be used by the gateway when allocating an APPC conversation with the transaction specified by the PGA_RECOVERY_TPNAME parameter. The user ID can be from 1 to 8 characters long. This parameter is required only when PGA_CAPABILITY parameter is set to COMMIT_CONFIRM and PGA_SECURITY_TYPE parameter is set to PROGRAM or SAME.
	There is no default value.
PGA_SECURITY_TYPE	APPC conversation security option. This controls what security parameters are sent to the OLTP in the FMH-5 at conversation allocation. The following are valid values:
	NONE, which sends no security parameters
	SAME, which sends only a user ID
	PROGRAM, which sends a user ID and password
	The default is NONE.
	For further information about these options, refer to Chapter 10, "Security Requirements".
TRACE_LEVEL	PGA trace level. This controls tracing output written to STDERR (the target of the LOG_DESTINATION parameter. The value must be an integer from 0 to 255.
	The default is 0, indicating no tracing.
	Any value between 1 and 255 will turn on tracing.

PGA_CAPABILITY Parameter Considerations

When choosing a setting for the PGA_CAPABILITY parameter, take care to ensure that the correct setting is used based on what the RTPs will be doing.

The `READ_ONLY` setting should always be used when the RTPs are read-only, that is, when the RTPs perform no database updates. `READ_ONLY` should never be used when the RTPs perform database updates. For example, if the `READ_ONLY` setting is chosen, and if a RTP called by the gateway performs updates to a foreign database, then the Oracle database does *not* provide any integrity protection for those updates. Further, `READ_ONLY` mode allows a gateway transaction to be part of a distributed transaction that might update several other databases. If the gateway calls a RTP that performs updates in this situation and if a failure occurs, then the database updated by the RTP is out of sync with the other databases.

In cases where the RTPs perform updates to foreign databases, there are two options for `PGA_CAPABILITY`:

- `SINGLE_SITE`

- `COMMIT_CONFIRM`

Each of these options provides protection against data integrity problems by permitting `COMMIT` and `ROLLBACK` requests to be forwarded to the RTP and by informing the Oracle database about the distributed update and recovery capabilities of the gateway. The particular option chosen depends on the design of the RTPs and upon the capabilities of the OLTP where they execute.

If the OLTP has LU6.2 `SYNCLEVEL 1` or 2 support, then the `COMMIT_CONFIRM` capability provides limited two-phase commit between the Oracle database and the OLTP, with the restriction that no other commit-confirm site (gateway or Oracle) can be part of the distributed transaction. If it is not possible to use `COMMIT_CONFIRM`, then the `SINGLE_SITE` capability provides update capability between the Oracle database and the OLTP, with the restriction that only the OLTP can perform updates, and no updates can occur on the Oracle side.

Each of the `PGA_CAPABILITY` options for update control imposes specific requirements on the RTP and on the OLTP. For `COMMIT_CONFIRM` capability, these requirements are discussed in detail in Chapter 5, "Implementing Commit-Confirm," of the *Oracle Database Gateway for APPC User's Guide*. Also refer to Section , "Configuring the OLTP for Commit-Confirm" on page 8-12 in this guide. For `SINGLE_SITE` capability, the RTP is responsible for performing the suitable tasks in response to `COMMIT` and `ROLLBACK` requests received from the gateway on behalf of the Oracle database . The gateway uses the `APPC CONFIRM` and `SEND_ERR` requests to implement `COMMIT` and `ROLLBACK`, respectively. On receipt of a `CONFIRM` command, the RTP must perform `COMMIT` processing and then respond to the gateway with an `APPC CONFIRMED` response. On receipt of a `SEND_ERR` command, the RTP must perform `ROLLBACK` processing.

Because the distributed transaction capability of the Oracle database is affected by the `PGA_CAPABILITY` option used by the gateway, it is desirable to separate inquiry and update applications by using different gateway instances for each. One gateway can be defined with `PGA_CAPABILITY` set to `READ_ONLY` and others with `PGA_CAPABILITY` set to `SINGLE_SITE` or `COMMIT_CONFIRM`.

This allows read-only transaction programs to participate in distributed transactions under the control of the Oracle database . For example, data from DB2 can be retrieved through the `READ_ONLY` gateway by an inquiry-only RTP, and can then be used as input to database updates on the Oracle database , all in one Oracle transaction. A `SINGLE_SITE` gateway can be used only for accessing RTPs which perform updates to foreign databases outside the scope of control of the Oracle database . Data can be read from any databases accessible to the Oracle database , and that data can be used to perform updates through the gateway.

When it is necessary to update resources on both the Oracle side and the OLTP side, a COMMIT_CONFIRM gateway can be used, provided that the OLTP and the RTPs are set up to implement COMMIT_CONFIRM.

All that is necessary to set up multiple gateway instances is to set up the following for each instance:

- an entry in the listener.ora file defining the sid of the gateway instance
- an entry in the tnsnames.ora file defining an alias to be used to connect to the gateway instance defined in listener.ora
- a database link in the Oracle database that specifies the alias defined in the tnsnames.ora file in its USING parameter.

Note that the gateway instances can share one common directory structure and use the same executables.

For example, to set up two gateways, PGAI and PGAU (for inquiry and update use, respectively), the following steps are required:

1. Define entries in listener.ora for two *sid*s, PGAI and PGAU.

2. Define two aliases in tnsnames.ora that connect to the two new *sid*s, PGAI and PGAU.

3. Define two database links in the Oracle database , one connecting to PGAI and the other connecting to PGAU.

4. Finally, create the initialization files initPGAI.ora and initPGAU.ora.

   In initPGAI.ora, set PGA_CAPABILITY to READ_ONLY, and in initPGAU.ora, set PGA_CAPABILITY to SINGLE_SITE or COMMIT_CONFIRM. Then, use the PGAI gateway for inquiry-only transactions, and use the PGAU gateway for update transactions.

   The same steps can be used to set up additional gateway instances.

PGA_CONFIRM Parameter Considerations

When deciding upon the setting for the PGA_CONFIRM parameter, it is important to understand the effects of each setting. First, keep in mind that this parameter affects only those conversations running at SYNCLEVEL 1. The default setting, PGA_CONFIRM=REJECT, is suitable for most applications. With this setting, the gateway generates an error if a CONFIRM request is received from the remote transaction program. If you have a remote transaction that uses CONFIRM to verify that data was received by the gateway, then you must use PGA_CONFIRM=ACCEPT to allow the gateway to respond to those incoming CONFIRM requests with CONFIRMED responses. You must be aware that the gateway sends CONFIRM requests to the remote transaction when the Oracle application has issued a COMMIT request. In order for the COMMIT processing to work correctly, the remote transaction must be written to perform its local commit processing whenever a CONFIRM request is received from the gateway, and respond to the gateway with CONFIRMED after the commit processing has successfully completed. If an error occurs during commit processing, then the remote transaction must respond to the gateway with SEND_ERR to indicate that the commit failed.

One special case for the use of PGA_CONFIRM=ACCEPT is with IMS/TM version 7. When using the implied APPC support that is provided by IMS/TM version 7, conversations that run at SYNCLEVEL 1 are handled differently than conversations that run at SYNCLEVEL 0. IMS/TM automatically generates CONFIRM requests after each APPC SEND when the conversation is at SYNCLEVEL 1. On the gateway side, if

PGA_CONFIRM=ACCEPT is not specified, then the CONFIRM requests sent by IMS/TM result in errors generated by the gateway. Using PGA_CONFIRM=ACCEPT alleviates this problem, allowing the gateway to respond to incoming CONFIRM requests with CONFIRMED responses. The only limitation with running this way is that the implied APPC support provided by IMS does not notify the application when a CONFIRM request is received from the gateway. This means that the gateway cannot use CONFIRM to implement COMMIT, thereby disabling the use of COMMIT/ROLLBACK to control updates on the IMS side of the conversation.

Sample listener.ora File for a Gateway Using SNA

```
LISTENER =
  (ADDRESS_LIST =
        (ADDRESS=
          (COMMUNITY= TCP.world)
          (Host = bay)
          (PROTOCOL= TCP)
          (Port= 2621)
        )
        (ADDRESS=
          (COMMUNITY= TCP.world)
          (Host = bay)
          (PROTOCOL= TCP)
          (Port= 2623)
        )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PGA)
      (ORACLE_HOME = C:\oracle\pga\11.2)
      (PROGRAM = pg4asrv)
    )
  )
```

Sample tnsnames.ora File for a Gateway Using SNA

```
ORA920 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = bay.us.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ORA920.bay)
    )
  )

PGA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = bay)(PORT = 2621))
    )
    (CONNECT_DATA =
      (SID = PGA)
    )
    (HS = OK)
  )
```

B

Gateway Initialization Parameters for TCP/IP Communication Protocol

This appendix lists and describes the parameters needed specifically for a gateway featuring the TCP/IP for IMS Connect communication protocol. It also provides a sample output of the `pg4tcpmap` tool. In addition, this appendix contains sample `listener.ora` and `tnsnames.ora` files for a gateway using TCP/IP. It contains the following sections:

- Gateway Initialization Parameter File Using TCP/IP
- Output for the pg4tcpmap Tool

Gateway Initialization Parameter File Using TCP/IP

The parameter file for the Oracle Database Gateway for APPC using TCP/IP for IMS Connect is located in the `%ORACLE_HOME%\dg4appc\admin` directory and is called `init`*sid*`.ora`.

> **Note:** The `init`*sid*`.ora` file contains both SNA and TCP/IP parameters. You will need to modify these files with the appropriate parameters.

PGA Parameters

The Procedural Gateway Administration (PGA) parameters control the TCP/IP interface portion of the gateway.

PGA parameters are specified using the `SET` gateway initialization parameter. For example:

```
SET pga_parm=value
```

where:

- `pga_parm` is one of the PGA parameter names in the list that follows
- `value` is a character string with contents that depend on `pga_parm`

Table B–1 provides a list of PGA parameters and their descriptions.

Table B–1   PGA Parameters on Gateway Using TCP/IP for IMS Connect

Parameter	Description
LOG_ DESTINATION=*logpath*	*logpath* specifies the destination at which STDERR is reopened. LOG_DESTINATION specifies a directory only, and STDERR is reopened to logpath*sid_pid*.log
	where:
	■ *sid* is the sid name
	■ *pid* is the process ID assigned to the gateway
	NOTE: This parameter will be used for the pg4tcpmap tool when you set your Trace Level to 255. The log file for this tool will reside in the same place as the gateway log file, in pg4tcpmap_*sid*.log.
PGA_CAPABILITY	PGA transaction capability. The following are valid values:
	READ_ONLY or RO: read-only capabilities.
	SINGLE_SITE or SS: single-site update only. This indicates that in a distributed environment, only the gateway can perform updates. No other database updates can occur within the Oracle transaction.
	The default is SINGLE_SITE.
PGA_SECURITY_TYPE	TCP/IP conversation security option. This controls what security parameters are sent to the OLTP. The following are valid values:
	NONE, which sends no security parameters
	PROGRAM, which sends a user ID and password
	The default is NONE.
	For further information on these options, refer to Chapter 10, "Security Requirements".
	Important: You must specify your RACF group name through the pg4tcpmap tool if you have set your PGA security option to SECURITY=PROGRAM. For more information about this issue, refer to the *Oracle Database Gateway for APPC User's Guide*.
	If you have already loaded the table PGA_TCP_IMSC and you did not first specify the RACF group name, then delete the row and reinsert it with the value for the RACF group name.
PGA_TCP_DB	The Oracle Net service name for the Oracle database in which the gateway receives its TCP/IP for IMS Connect information, such as host name and port number. This parameter can be from 1 to 255 characters long. This parameter is required.
	There is no default value.
PGA_TCP_PASS	The Oracle password to be used by the gateway when connecting to the Oracle database specified by the PGA_TCP_DB parameter. The password can be from 1 to 30 characters long. This parameter is required. The password can be encrypted. For more information about encrypting the password, refer to Section , "Passwords in the Gateway Initialization File".
	There is no default value.

Table B–1   (Cont.) PGA Parameters on Gateway Using TCP/IP for IMS Connect

Parameter	Description
PGA_TCP_USER	The Oracle user ID to be used by the gateway when connecting to the Oracle database specified by the PGA_TCP_DB parameter. The user ID can be from 1 to 30 characters long. This parameter is required.
	There is no default value.
TRACE_LEVEL	PGA trace level. This controls tracing output written to STDERR (the target of the LOG_DESTINATION parameter.) The value must be an integer from 0 to 255.
	The default is 0, indicating no tracing.
	NOTE: This parameter will be used in the pg4tcpmap tool as well as the gateway.

Output for the pg4tcpmap Tool

The following output illustrates the results from executing the pg4tcpmap tool when running TCP/IP for IMS Connect on the gateway. Refer to Section , "Loading the PGA_TCP_IMSC Table" on page 9-12 of this guide and to Chapter 6 of the *Oracle Database Gateway for APPC User's Guide* for detailed information about the function and parameters of the pg4tcpmap tool.

Note that input in this sample is shown within brackets angle brackets (<>).

```
C:\oracle\bin\<pg4tcpmap>
PG4TCPMAP: Release 11.2.0.1.0 - Production on Thu Jun 11 15:09:00 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All rights reserved.


This tool takes the IMS Connect TCP/IP information, such as host name
and port number and maps them to your TIPs.

You may use this tool to insert or delete IMS Connect TCP/IP information.
If you want to insert a row, type I
If you want to delete a row, type D
i
Enter the Unique Side Profile.
IMSPGA
Enter either the remote hostname or its TCP/IP address.
mvs09
Enter the IMS CONNECT port number.
9900
Do you want to select a CONVERSATIONAL PROTOCOL?(Y|N)
The default is NO, 'no request for acknowledgment or deallocation'
n
Enter one of the following letters for Timer.
For .25 second, enter 'D'.
For .01 to .25 second, enter 'S'.
For 'does not set the timer, no wait occurs', enter 'N'.
For Receive waits indefinitely, enter 'I'.
The default is 'D'.
D
Enter one of the following letters for 'socket connection type'.
For transaction socket, enter 'T'.
For persistent socket, enter 'P'.
For non-persistent socket, enter 'N'.
The default is 'T'.
```

```
                   T
                   Do you want to enter the CLIENT ID name? (Y|N)
                   If NO, IMS CONNECT (USER EXIT) will generate it.
                   n
                   Enter one of the following letters for 'COMMIT MODE'.
                   For Commit Mode set to 0, enter '0'.
                   For Commit Mode set to 1, enter '1'.
                   The default is '1'.
                   1
                   Enter the DATASTORE name (IMS DESTINATION ID).
                   The maximum string length is 8 and the Datastore name must be specified.
                   IMSE
                   Do you want to enter the LTERM? (Y|N)
                   If NO, the default is blank.
                   n
                   Do you want to enter the RACF GROUP name? (Y|N)
                   If NO, the default is blank.
                   n
                   Do you want to enter the IRM_ID? (Y|N)
                   If NO, the default is *IRMREQ*.
                   n
                   Does your exit return the LLLL prefix field? (Y|N)
                   The default is 'N'.
                   n
                   Requested to INSERT a row.
                   'Side Profile name' is 'IMSPGA'
                   'remote host name' is 'MVS09'
                   'IMS Connect port number' is '9900'
                   'conversational protocol' is ' '
                   'Timer' is 'D'
                   'socket connection type' is 'T'
                   'client ID' is '        '
                   'commit mode' is '1'
                   'Datastore name (IMS destination ID)' is 'IMSE    '
                   'IMS LTERM override' is '        '
                   'RACF group name' is '        '
                   'IRM ID' is '*IRMREQ*'
                   'LLLL prefix present' is 'N'
                   PG4TCPMAP is complete.
```

Sample listener.ora File for a Gateway Using TCP/IP

```
LISTENER =
  (ADDRESS_LIST =
        (ADDRESS=
          (COMMUNITY= TCP.world)
          (Host = bay)
          (PROTOCOL= TCP)
          (Port= 2621)
        )
        (ADDRESS=
          (COMMUNITY= TCP.world)
          (Host = bay)
          (PROTOCOL= TCP)
          (Port= 2623)
        )
  )
```

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PGA)
      (ORACLE_HOME = C:\oracle\pga\11.2)
      (PROGRAM = pg4t4ic)
    )
  )
```

Sample tnsnames.ora File for a Gateway Using TCP/IP

```
ORA920 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = bay.us.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ORA920.bay)
    )
  )

PGA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = bay)(PORT = 2623))
    )
    (CONNECT_DATA =
      (SID = PGA)
    )
    (HS = OK)
  )
```

C

Gateway Terminology

This appendix contains a list of terms and definitions pertaining to the gateway and its components and function.

For a list of other terms and definitions associated with the gateway, refer to Chapter 1 of the *Oracle Database Gateway for APPC User's Guide*.

This appendix contains the following section:

- Gateway Term Definitions

Gateway Term Definitions

Following is a list of some of the terms and definitions used in this edition of the *Oracle Database Gateway for APPC Installation and Configuration Guide*.

Gateway Initialization File

This file is known as init*sid*.ora and it contains parameters that govern the operation of the gateway. If you are using the SNA protocol, then refer to Appendix A, "Gateway Initialization Parameters for SNA Protocol" for more information. Refer to Appendix B, "Gateway Initialization Parameters for TCP/IP Communication Protocol" if your protocol is TCP/IP.

Gateway Remote Procedure

The Oracle Database Gateway for APPC provides prebuilt remote procedures. In general, the following three remote procedures are used:

- PGAINIT, which initializes transactions

- PGAXFER, which transfers data

- PGATERM, which terminates transactions

Refer to "Remote Procedural Call Functions" on page 1-7 in this guide and to Appendix B, "Gateway RPC Interface" in the *Oracle Database Gateway for APPC User's Guide* for more information about gateway remote procedures.

dg4pwd

dg4pwd is a utility which encrypts passwords that are normally stored in the gateway initialization file. Passwords are stored in an encrypted form in the password file, making the information more secure. Refer to "Passwords in the Gateway Initialization File" on page 10-5 for detailed information about how the dg4pwd utility works.

pg4tcpmap tool

This gateway mapping tool is applicable only when the gateway is using TCP/IP support for IMS Connect. Its function is to map the side profile name to TCP/IP and IMS Connect attributes into the `PGA_TCP_IMSC` table.

PGA

Procedural Gateway Administration (PGA) is a general reference within this guide to all or most components comprising the Oracle Database Gateway for APPC. This term is used when references to a specific product or component are too narrow.

PGDL

Procedural Gateway Definition Language (PGDL) is the collection of statements used to define transactions and data to the PGAU.

PL/SQL Stored Procedure Specification (PL/SQL package)

This is a precompiled PL/SQL procedure that is stored in Oracle database .

UTL_RAW PL/SQL Package (the UTL_RAW Functions)

This component of the gateway represents a series of data conversion functions for PL/SQL RAW variables and remote host data. The types of conversions performed depend on the language of the remote host data. Refer to "UTL_RAW Functions" in Appendix D of the *Oracle Database Gateway for APPC User's Guide* for more information.

UTL_PG PL/SQL Package (the UTL_PG Functions)

This component of the gateway represents a series of COBOL numeric data conversion functions. Refer to "NUMBER_TO_RAW and RAW_TO_NUMBER Argument Values" in Appendix D of the *Oracle Database Gateway for APPC User's Guide* for supported numeric datatype conversions.

D

Configuration Worksheet

The following worksheet lists the parameter names and the reasons you will need them to configure the gateway and the communications interface you have chosen (either SNA or TCP/IP). Use the worksheet to gather the specific information you need before you begin the configuration process.

Ask your systems administrator to provide you with any parameter names you do not know.

Table D–1   Parameters for Configuring Gateway and Communication Protocols

Name of Parameter Needed	Purpose	Your Specific Parameters Here
`ORACLE_HOME`	for: gateway's Oracle Home	_____
`ORACLE_SID`	for: gateway's system ID	_____
Any Security Options Needed	for: SNA remote LU properties options	_____
Suitable Name for Each Side Information Profile	for: SNA creating CPI-C symbolic destination names (side information profiles), general information	_____
Suitable Mode	for: SNA	_____
TP Name	for: SNA partner information in CPI-C name properties	_____
Partner LU Name Alias	for: SNA partner information in CPI-C name properties	_____
Unique Side Profile Name	for: Configuring TCP/IP support for IMS Connect	_____
Remote Host name or TCP/IP Address	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____
IP Address	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____

Table D–1   (Cont.) Parameters for Configuring Gateway and Communication Protocols

Name of Parameter Needed	Purpose	Your Specific Parameters Here
IMS Connect Port Number	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____
Conversational Protocol (Y/N)	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____
Timer choice: a) .25 b) .01 to .25 c) Does not exist d) Receives wait	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____
Socket Connection Type Choice: a) transaction b) persistent c) nonpersistent	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____
IMS Client ID Name	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____
IMS Commit Mode Choice: a) 0 b) 1	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____
IMS Destination ID, datastore name	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____
`LTERM`	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____
RACF Group Name	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____
`IRM_ID`	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____
`LLLL (Y/N)`	for: Configuring TCP/IP support for IMS Connect, `pg4tcpmap` tool	_____

Index