

Oracle® Database Gateway for APPC

Installation and Configuration Guide



19c for IBM AIX on POWER Systems (64-Bit), Linux x86-64, Oracle Solaris on SPARC (64-Bit), and HP-UX Itanium
F18241-01
April 2019



Oracle Database Gateway for APPC Installation and Configuration Guide, 19c for IBM AIX on POWER Systems (64-Bit), Linux x86-64, Oracle Solaris on SPARC (64-Bit), and HP-UX Itanium

F18241-01

Copyright © 2002, 2019, Oracle and/or its affiliates. All rights reserved.

Primary Author: Rhonda Day

Contributing Authors: Vira Goorah, Govind Lakkju, Peter Wong, Juan Pablo Ahues-Vasquez, Peter Castro, Charles Benet

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Intended Audience	xi
Documentation Accessibility	xi
Related Documents	xii
Conventions	xii
Command Syntax	xiii

1 Introduction to Oracle Database Gateway for APPC

Overview of the Gateway	1-1
Features of the Gateway	1-2
Terms	1-3
Architecture of the Gateway	1-5
Implementation of the Gateway	1-6
Communication With the Gateway	1-7
RPC Functions	1-7
Description of RPC Functions	1-7
Remote Transaction Initiation	1-8
Data Exchange	1-8
Remote Transaction Termination	1-8
Transaction Types for Gateway Using SNA	1-8
Transaction Types for Gateway Using TCP/IP	1-9

2 Release Information

Product Set	2-1
Changes and Enhancements	2-1
Gateway Password Encryption Tool	2-1
Partial IPv6 Support	2-1
Known Restrictions	2-1
Known Restrictions for the Gateway	2-2
Known Restrictions for PGAU	2-2

3 System Requirements

Hardware Requirements	3-1
Network Attachment Requirements	3-2
Software Requirements	3-2
Operating System Requirements	3-2
Communication Protocol Requirements	3-2
Oracle Database Requirements	3-3
Oracle Networking Product Requirements	3-3
IBM Mainframe Requirements	3-3

4 Installing the Gateway

Before You Begin	4-1
Planning to Upgrade or Migrate the Gateway	4-2
Preupgrade Procedures	4-2
Upgrade and Migration Considerations	4-2
Restoration	4-3
Preinstallation Steps	4-3
Gateway Installation Methods	4-4
Installing the Gateway Software	4-4
Installation Steps	4-4
Step through the Oracle Universal Installer	4-5
Oracle Universal Installer on UNIX platforms	4-5
Deinstalling Oracle Database Gateway for APPC	4-6
About the Deinstallation Tool	4-7
Removing Oracle Software	4-8

5 Configuring Your Oracle Network

6 Configuring the SNA Communication Package on Linux

Using SNA Security Validation	6-1
Processing Inbound Connections	6-2
Independent Versus Dependent LUs	6-2
Definition Types	6-3
Creating IBM Communications Server Definitions for the Gateway	6-3
Creating the Configuration	6-4
Creating the Node	6-4
Creating Devices	6-4
Choosing the Device Type	6-4

Configuring a LAN Device	6-4
Creating Peer Connections	6-4
Defining the Link Station	6-5
Defining the Adjacent Node	6-5
Creating Local LUs	6-5
Defining Local LUs	6-5
Creating Partner LUs	6-5
Defining Partner LUs	6-5
Creating the CPI-C Side Information Profile	6-6
Testing the Connection	6-6
Resume Configuration of the Gateway	6-7

7 Configuring the SNA Communication Package on AIX-Based Systems

Processing Inbound Connections	7-1
Independent Versus Dependent LUs	7-2
Creating SNA Profiles for the Gateway	7-3
Profile Types	7-3
SNA Server Profiles	7-3
SNA Node Profile	7-3
Link Station Profile	7-4
Mode Profile	7-4
Local LU Profile	7-4
Partner LU Profile	7-5
Partner LU Location Profile	7-5
Side Information Profile	7-6
Activating Profiles	7-7
Resume Configuration of the Gateway	7-8

8 Configuring the SNA Communication Package on Solaris

Processing Inbound Connections	8-1
Configuring SNAP-IX Version 6	8-1
Before You Begin	8-2
SNAP-IX Configuration Tool	8-2
Creating SNAP-IX Profiles for the Gateway	8-2
Independent Versus Dependent LUs	8-2
Creating SNA Definitions for the Gateway	8-3
Sample SNAP-IX Definitions	8-3
Configuring SNAP-IX	8-3

Starting xsnaadmin	8-3
Configuring the SNA node	8-3
Adding a Port	8-4
Create a Link Station	8-4
Creating Local LUs	8-4
Creating Partner LUs	8-5
Creating Mode and CPI-C Profiles	8-5
Testing the Connection	8-5
Resuming Gateway Configuration	8-6

9 Configuring the Gateway Using SNA Communication Protocol

Before You Begin	9-1
Preparing to Configure a Gateway Installation/Upgrade	9-2
Oracle Database Configuration: First-Time Gateway Installations	9-4
Upgrading or Migrating the Oracle Database from Previous Gateways	9-6
If You Must Reinstall Package Specifications	9-7
Upgrading PGAU From Previous Gateway Releases	9-7
Configuring the Oracle Database for Gateways to Coexist	9-8
Optional Configuration Steps to Permit Multiple Users	9-8
Configuring the Gateway	9-10
Configuring Commit-Confirm	9-11
Configuring the Oracle Database for Commit-Confirm	9-11
Configuring Gateway Initialization Parameters for Commit-Confirm	9-12
Configuring the OLTP for Commit-Confirm	9-12
Verifying the Gateway Installation and OLTP Configuration	9-13
Verifying the Gateway Installation	9-13
Verifying the OLTP Configuration	9-14
CICS Verification	9-14
IMS/TM Verification	9-15
APP/C/MVS Verification	9-16
Verifying OLTP Configuration for Commit-Confirm	9-16
Performing Postinstallation Procedures	9-17
Installing Sample Applications	9-17

10 Configuring the OLTP

Configuring the OLTP for an SNA Environment	10-1
Configuring CICS Transaction Server for z/OS	10-1
Configuring IMS/TM	10-2
Configuring APPC/MVS	10-3

11 Configuring the Gateway Using TCP/IP Communication Protocol

Before You Begin	11-1
Preparing to Configure a Gateway Installation/Upgrade	11-1
Configuring Oracle Database: First-Time Installation	11-3
Upgrading or Migrating the Oracle Database from Previous Gateways	11-7
If You Must Reinstall Package Specifications	11-7
Upgrading PGAU from Previous Gateway Releases	11-8
Optional Configuration Steps to Permit Multiple Users	11-8
Configuring TCP/IP for the Gateway	11-10
Configuring the Gateway	11-11
Loading the PGA_TCP_IMSC Table	11-12
Verifying the Gateway Installation and OLTP Configuration	11-12
Verifying the Gateway Installation	11-13
Verifying the OLTP Configuration	11-13
IMS/TM Verification	11-14
Performing Postinstallation Procedures	11-15
Installing Sample Applications	11-15

12 Security Requirements

Overview of Security Requirements	12-1
Authenticating Application Logons	12-2
Defining and Controlling Database Links	12-2
Link Accessibility	12-2
Links and CONNECT Clauses	12-3
Using SNA Security Validation	12-3
Specifying SNA Conversation Security	12-3
SNA Security Option SECURITY=NONE	12-4
SNA Security Option SECURITY=PROGRAM	12-4
SNA Security Option SECURITY=SAME	12-4
TCP/IP Security	12-5
Specifying TCP/IP Conversation Security	12-5
TCP/IP Security Option SECURITY=NONE	12-5
TPC/IP Security Option SECURITY=PROGRAM	12-5
Passwords in the Gateway Initialization File	12-6

13 Migrating From Existing Gateways

Migrating An Existing Gateway Instance to a New Release Using SNA Protocol	13-1
Step 1: Install the New Release	13-1
Step 2: Transferring the initSID.ora Gateway Initialization File Parameters	13-1
Backout Considerations When Migrating to New Releases	13-2
Oracle Net Considerations	13-2
Parameter Changes: Version 4 to 12c Release 2 (12.2) of the Gateway	13-2
Parameter Changes: Version 8 or Earlier to Gateway 12c Release 2 (12.2)	13-4
Migrating from Gateway Release 9.0.1 or 9.2.0 or 10.1.0 to Gateway 12c Release 2 (12.2)	13-5
Migrating from an Existing Gateway to TCP/IP Using SNA	13-5
Using Existing TIPs with Existing Side Profile Definitions	13-5

A Gateway Initialization Parameters for SNA Protocol

PGA Parameters	A-1
PGA_CAPABILITY Parameter Considerations	A-4
PGA_CONFIRM Parameter Considerations	A-6
Sample listener.ora file for a Gateway Using SNA	A-6
Sample tnsnames.ora file for a Gateway Using SNA	A-7

B Gateway Initialization Parameters for TCP/IP Communication Protocol

Gateway Initialization Parameter File Using TCP/IP	B-1
PGA Parameters	B-1
Output for the pg4tcpmap Tool	B-3
Sample listener.ora File for a Gateway Using TCP/IP	B-4
Sample tnsnames.ora File for a Gateway Using TCP/IP	B-5

C Gateway Terminology

D Configuration Worksheet

Index

List of Figures

1-1	Relationship of Gateway and the Oracle Database	1-5
1-2	Gateway Architecture	1-6
6-1	SNA server definitions and VTAM	6-7
7-1	Relationship Between SNA Profiles and Host VTAM Definitions	7-7
8-1	Relationship Between SNAP-IX Definitions and Host VTAM Definitions	8-6

List of Tables

1-1	RPC Functions and Commands in the Gateway and Remote Host	1-7
3-1	Hardware requirements for Oracle Database Gateway for APPC	3-1
4-1	The Oracle Universal Installer: Steps for Installing Oracle Database Gateway for APPC	4-5
A-1	PGA Parameters for Oracle Database Gateway for APPC Using SNA	A-2
B-1	PGA Parameters for Oracle Database Gateway for APPC Using TCP/IP for IMS Connect	B-2
D-1	Parameters for Configuring Gateway and Communication Protocols	D-1

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.

The UNIX platforms supported by this gateway release are:

- IBM AIX on POWER Systems (64-Bit)
- Linux x86-64
- Oracle Solaris on SPARC (64-Bit)

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

<https://support.oracle.com>

Intended 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 operating system for your platform and Oracle Database Gateways.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/>

[lookup?ctx=acc&id=info](http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info) or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

The *Oracle Database Gateway for APPC Installation and Configuration Guide for IBM AIX on POWER Systems (64-Bit), Linux x86-64, Oracle Solaris on SPARC (64-Bit), and HP-UX Itanium* is included as 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 might be referenced in this guide:

- *Oracle Database Installation Guide*
- *Oracle Database Administrator's Guide*
- *Oracle Database Development Guide*
- *Oracle Database Concepts*
- *Oracle Database Error Messages*
- *Oracle Database Net Services Administrator's Guide*
- *Oracle Database PL/SQL Language Reference*
- *Oracle Database Net Services Administrator's Guide*
- *Oracle Database Heterogeneous Connectivity User's Guide*
- *Oracle Call Interface Programmer's Guide*

In addition to your Oracle documentation, ensure that you have appropriate documentation for your platform, for your operating system and for your communications packages. You may find IMS Connect Guide and Reference IBM documentation useful.

 **Note:**

For other Operating System and SNA communication package and TCP/IP package references, refer to the appropriate vendor documentation for your system

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Command Syntax

UNIX command syntax appears in monospace font. The dollar character (\$), number sign (#), or percent character (%) are UNIX command prompts. Do not enter them as part of the command. The following command syntax conventions are used in this guide:

Convention	Description
backslash \	A backslash is the UNIX command continuation character. It is used in command examples that are too long to fit on a single line. Enter the command as displayed (with a backslash) or enter it on a single line without a backslash: <code>dd if=/dev/rdsk/c0t1d0s6 of=/dev/rst0 bs=10b \ count=10000</code>
braces { }	Braces indicate required items: <code>.DEFINE {macro1}</code>
brackets []	Brackets indicate optional items: <code>cvt crt termname [outfile]</code>
ellipses ...	Ellipses indicate an arbitrary number of similar items: <code>CHKVAL fieldname value1 value2 ... valueN</code>
<i>italics</i>	Italic type indicates a variable. Substitute a value for the variable: <code>library_name</code>
vertical line	A vertical line indicates a choice within braces or brackets: <code>FILE filesize [K M]</code>

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 a 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.

The following topics discuss the architecture, uses, and features of the gateway:

- [Overview of the Gateway](#)
- [Features of the Gateway](#)
- [Terms](#)
- [Architecture of the Gateway](#)
- [Implementation of the Gateway](#)
- [Communication With the Gateway](#)
- [RPC 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 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. As a result, the gateway uses existing transactions with little or no programming effort on the remote host.

 **Note:**

For gateways using SNA only the use of a generic and standard protocol, APPC, enables the gateway to access a multitude of systems. The gateway can communicate with virtually any APPC-enabled system, including IBM Corporation CICS on any platform, IBM Corporation 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, providing users a choice between the SNA or TCP/IP communication protocol. IMS Connect is an IBM product which enables 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 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 [Gateway Configuration Using TCP/IP Communication Protocol](#) in this guide.

- Fast interface

The gateway is optimized for minimum network traffic when you execute programs remotely. The interface to the gateway is an optimized PL/SQL stored procedure specification called the transaction interface package (TIP). This specification is precompiled in the Oracle database. Because there are no additional software layers on the remote host system, overhead occurs only when your program executes.

- Platform independence

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 as an interface with an existing procedural logic or to integrate new procedural logic into an Oracle database environment.

- Oracle database integration

Integration of Oracle database with the gateway enables you 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 transaction 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 perform tasks such as authenticate users. It also provides role-based security compatible with any security package running on your mainframe computer.

- Automatic conversion

Through TIP, the following conversions are performed:

- ASCII to and from EBCDIC
- Remote transaction program data types to and from PL/SQL data types
- National language support for many languages

Terms

The following terms and definitions are used throughout this guide. Refer to [Gateway Terminology](#) for the complete 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 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](#) for a view of the gateway architecture.

Online Transaction Processor (OLTP)

OLTP is an online transaction processor available from other vendors, including CICS Transaction Server for z/OS, IMS/TM, and z/OS.

Procedural Gateway Administration Utility (PGAU)

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.

Data Dictionary (PG DD)

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

RPC

RPC is a programming call that executes program logic on one system in response to a request from another system. Refer to [Gateway Terminology](#) for more information, and refer to Appendix B, Gateway RPC Interface in the *Oracle Database Gateway for APPC User's Guide* as well.

RTP

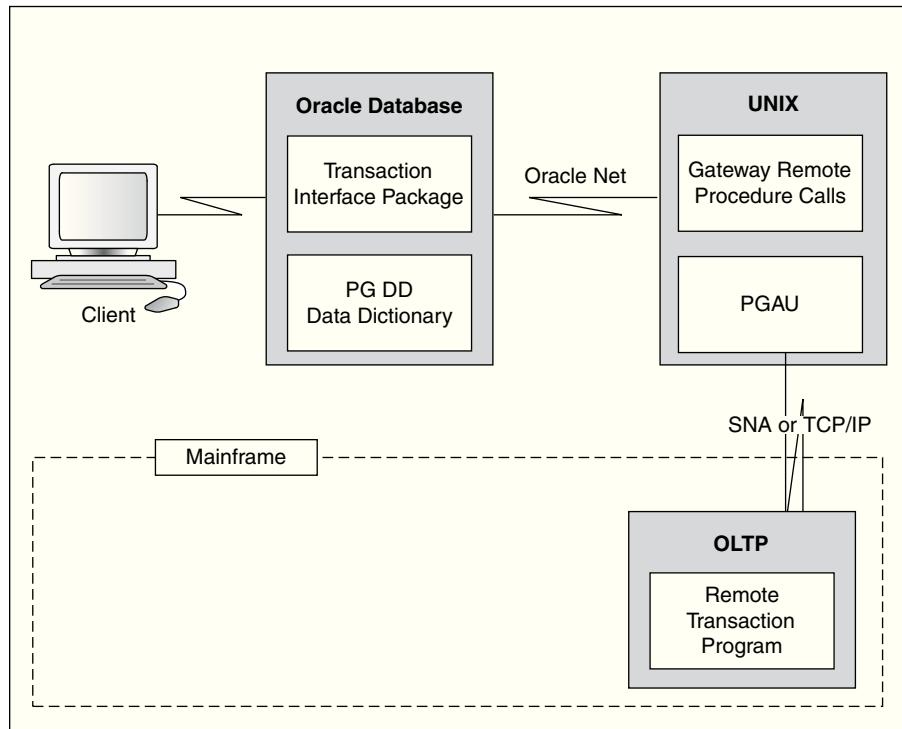
A remote transaction program is a customer-written transaction, running under the control of an OLTP, which the user invokes remotely using a PL/SQL procedure. To execute a remote transaction program through the gateway, you must use RPC to execute a PL/SQL program to call the gateway functions.

TIP (Transaction Interface Package)

A 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/UTL_PG functions.

[Figure 1-1](#) illustrates the Relationship of the gateway and the Oracle database. The terminology discussed in the preceding sections has been used in the architecture of the gateway.

Figure 1-1 Relationship of Gateway and the Oracle Database



Architecture of the Gateway

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

- Oracle database
- The gateway

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

- An OLTP (online transaction processor)

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

Note:

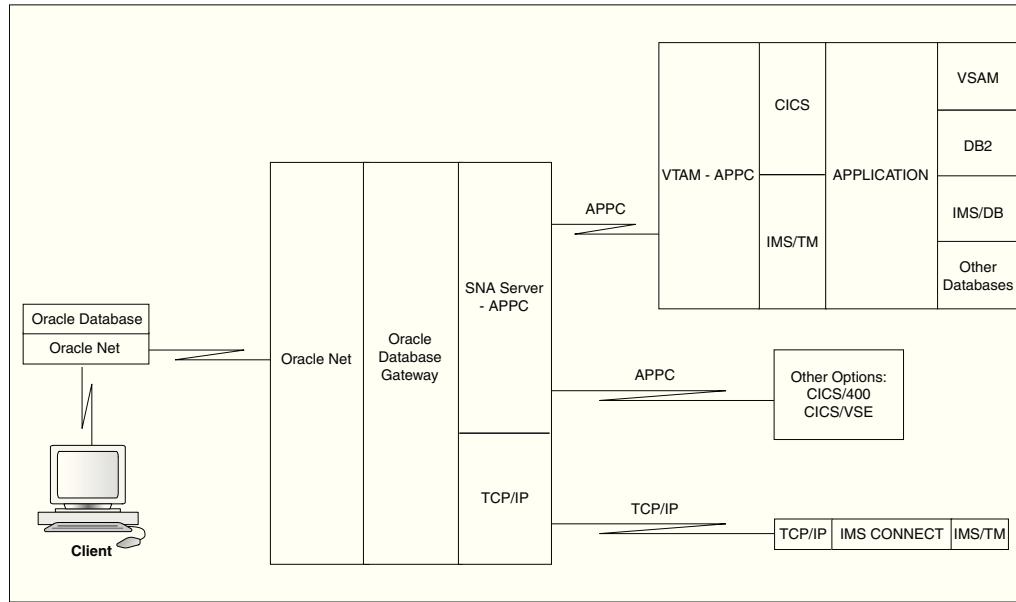
For a gateway using TCP/IP support for IMS Connect the only OLTP that is supported through TCP/IP is IMS through IMS Connect

The OLTP must be accessible to the system using the TCP/IP protocol. Multiple Oracle databases 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, 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 UNIX. It has the following components:

- A home directory, similar to the one associated with an Oracle instance `ORACLE_HOME` environment variable
- A system identifier, identified as `sid` or `ORACLE_SID`
- An initialization parameter file, similar to the Oracle database `initsid.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. Each Oracle database user session that accesses a particular gateway creates an independent process on UNIX 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 communication between the user or client program and the gateway is handled through a transaction interface package (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 remote transaction program
- Calls the gateway packages that handle the communications for starting the conversation, exchanging data and terminating the conversation
- Handles all data type conversions between PL/SQL data types and the target program data types

The Procedural Gateway Administration Utility (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

RPC Functions

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

Table 1-1 lists the remote procedural call 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](#)
- [Data Exchange](#)

- [Remote Transaction Termination](#)

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. Any future calls to the target program made by PGAXFER and PGATERM use the socket file descriptor as an input parameter.

Refer to [Gateway Initialization Parameters for TCP/IP Communication Protocol](#), and 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 PL/SQL data types to RAW before sending the buffer to the gateway. It is also responsible for converting RAW to the PL/SQL data types 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 made 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 remote transaction program.

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 on 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 system.
- 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 pg4tcpmap Tool Commands in Chapter 6 of the *Oracle Database Gateway for APPC User's Guide* and to [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

The following topics provide information about this release of the Oracle Database Gateway for APPC:

- [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, 12c Release 2 (12.2)
- Oracle Net, 12c Release 2 (12.2)

Changes and Enhancements

The following sections describe the changes and enhancements unique to the gateway:

- [Gateway Password Encryption Tool](#)
- [Partial IPv6 Support](#)

Gateway Password Encryption Tool

The Gateway Password Encryption tool (`tg4pwd`) 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 IBM AIX on POWER Systems (64-Bit), Linux x86-64, Oracle Solaris on SPARC (64-Bit), and Oracle Solaris on x86-64 (64-Bit) or 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](#)
- [Known Restrictions for PGAU](#)

Known Restrictions for the Gateway

Oracle Database Gateway for APPC has the following restrictions.

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 data types from version 3.4 onwards. However, the non-numeric character data (such as \$, (,), +, -, ,) that is allowed in DISPLAY data types 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 Datatype Conversion is Not Supported

Oracle Database Gateway for APPC does not support floating point data type conversion.

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

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

3

System Requirements

The following topics describe the system requirements of the gateway.

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

Refer to the *Oracle Database Installation Guide* 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:

<https://support.oracle.com>

Hardware Requirements

The hardware requirements for this release of the gateway on your platform are described in the following sections:

[Table 3-1](#) lists the minimum hardware requirements for Oracle Database Gateway for APPC.

Table 3-1 Hardware requirements for Oracle Database Gateway for APPC

Hardware Items	Required for IBM AIX on POWER Systems (64-Bit)	Required for Linux x86 64 bit	Required for Oracle Solaris on SPARC (64-Bit)	Required for HP-UX Itanium
Temporary Disk Space	400 MB	400 MB	400 MB	400 MB
Disk Space	1.5 GB	750 MB	750 MB	1.5 GB
Physical Memory*	512 MB	512 MB	512 MB	512 MB
Swap Space	1 GB	1 GB	1 GB	1 GB
Processor	IBM RS/6000 AIX-Based System Processor	x86_64	Sun Solaris Operating System (SPARC) Processor	HP Itanium processor for hp-ux 11

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

- Number of concurrent gateway connections open by each user
- Number of data items being transferred between the gateway and the remote transaction program
- The Oracle Net protocol adapters that were included during the gateway installation
- Additional factors, such as configured network buffer size

Network Attachment Requirements

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

However, if you are using only the new TCP/IP support for IMS Connect feature, you will not need an SNA package. Your operating system comes with TCP/IP installed.

Software Requirements

The system software configuration described in this section 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.

Topics:

- [Operating System Requirements](#)
- [Communication Protocol Requirements](#)
- [Oracle Database Requirements](#)
- [Oracle Networking Product Requirements](#)
- [IBM Mainframe Requirements](#)

Operating System Requirements

Refer to the *Oracle Database Installation Guide* 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 Linux 64-bit systems, AIX-Based, or Solaris system. The My Oracle Support Web site is available at

<https://support.oracle.com>

Communication Protocol Requirements

Each operating system uses specific communications servers, as described in this section.

Note:

If you choose to use TCP/IP support for IMS Connect as communication protocol, then you do not need to use an SNA communication package from the following list. Each operating system comes with a TCP/IP protocol automatically installed. If you choose to use the TCP/IP protocol, then you need to configure it to work properly with the gateway. Refer to [Gateway Configuration Using TCP/IP Communication Protocol](#).

Communications Protocols for Linux for Intel Pentium-based 32-bit Systems

Communications protocols for Linux for Intel Pentium-based 32-bit System IBM Communications Server V6.0.1.1 for Linux or higher, or the TCP/IP communication software that comes with your operating system.

Communications Protocols for AIX-based Systems

SNA Server for AIX version 6.01 or higher, or the TCP/IP communications software that comes with your operating system.

Communications Protocols for Solaris

SNAP-IX version 6 or higher, or the TCP/IP communications software that comes with your operating system.

Oracle Database Requirements

All UNIX platforms require that the Oracle database, that is to act as the Oracle database be up to date with the latest patchset for supported Oracle database releases.

Oracle Networking Product Requirements

Oracle Net is automatically installed on the system where Oracle database is installed and on the system where the gateway is installed. Refer to [Configuring Your Oracle Network](#) in this guide for detailed configuration information. Additionally, you might refer to the *Oracle Database Net Services Administrator's Guide*.

IBM Mainframe Requirements

In addition to the other software requirements of the gateway and the platform being used, 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, IDMS and z/OS.

 **Note:**

For a list of known restrictions, read the "[Known Restrictions](#)" section before proceeding with the 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 describe how to install and configure the Oracle Database Gateway for APPC.

- [Before You Begin](#)
- [Planning to Upgrade or Migrate the Gateway](#)
- [Preinstallation Steps](#)
- [Installing the Gateway Software](#)
- [Installation Steps](#)
- [Deinstalling 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 Installing the Gateway 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 configuration actions on the OLTP might require you to restart the OLTP. Ensure that your host system programmer or DBA review the instructions for your OLTP before you restart it.

To install and configure the gateway with a single Oracle database and a single OLTP, perform the procedures described in [Installing the Gateway Software](#).

Note:

If your gateway uses the SNA communication protocol, then follow the instructions for installation and configuration in this chapter, in [Configuring Your Oracle Network](#), and in [Gateway Configuration Using SNA Communication Protocol](#).

If your gateway uses the TCP/IP communication protocol, then follow the instructions for installation and configuration in this chapter, in [Configuring Your Oracle Network](#), and in [Gateway Configuration Using TCP/IP Communication Protocol](#).

Planning to Upgrade or Migrate the Gateway

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 need to perform the tasks mentioned in the following topics before you can install 12c Release 2 (12.2) of the Oracle Database Gateway for APPC.

 **Note:**

After reading this section, you must read [Migration From Existing Gateways](#) to determine the specific actions you must perform to prepare for upgrade or migration of your gateway. If you are migrating to Oracle Database Gateway for APPC 12c Release 2 (12.2) from version 4.01 or earlier, then you will find content related to migrating the gateway in [Migration From Existing Gateways](#).

If you are installing Oracle Database Gateway for APPC for the first time, then begin with "[Preinstallation Steps](#)".

This section includes the following topics:

- [Preupgrade Procedures](#)
- [Upgrade and Migration Considerations](#)
- [Restoration](#)

Preupgrade Procedures

Perform the following steps to prepare for upgrading the previous versions of 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 Data Dictionary (PG DD) contains all of its earlier entries without modification. New PGAU control information is 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 because of the changes in the following:
 - PL/SQL compatibility
 - Gateway server RPC interface

- UTL_PG interface
- If you have existing TIPs that were generated previously on a gateway using the SNA communication protocol and you want to utilize the new TCP/IP feature, then 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 because the gateway assumes that the user exit in IMS Connect is being used, which would translate between the 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

Preinstallation Steps

Before you install the gateway, perform the following pre-installation procedures:

- Ensure that your system meets all of the hardware and software requirements specified in [System Requirements](#).
- Ensure that your security requirements are met.
Refer to [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 [Configuration Worksheet](#).
- Decide on a SID (system identifier) for your gateway. This SID is used in [Configuring the Gateway](#).

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 communication package appropriate for your platform.

For more information about setting up and configuring the SNA communication package your platform needs to run the Oracle Database Gateway for APPC, refer to the appropriate chapter in this guide from the following list:

- For Linux for Intel Pentium-based 32-bit systems, refer to [Configuring the SNA Communication Package on Linux](#) .
 - For AIX-Based Systems, refer to [Configuring the SNA Communication Package on AIX-Based Systems](#).
 - For Solaris Operating System (SPARC 64-bit), refer to [Configuring the SNA Communication Package on Solaris](#) .
- 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*.

Gateway Installation Methods

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

- 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.
- On a system different from a local Oracle database.
- 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* and perform all necessary tasks there first.

If your server release is different than your gateway release, *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 want to install the gateway in the same Oracle home as the Oracle database, then the release number of both products must be the same. This section provides the steps for installing the gateway. It contains the following topics:

- [Step through the Oracle Universal Installer](#)
- [Oracle Universal Installer on UNIX platforms](#)

Step through the Oracle Universal Installer

Note:

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 UNIX 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 UNIX platform.

Table 4-1 The Oracle Universal Installer: Steps for Installing Oracle Database Gateway for APPC

Prompt	Response
Oracle Universal Installer: Welcome	Click Next .
Oracle Universal Installer: Specify Inventory Directory and credentials	<ol style="list-style-type: none">Specify the full path of the inventory directory.Specify the operating system group name.Click Next.
Oracle Universal Installer: Specify Home Details	<ol style="list-style-type: none">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	<ol style="list-style-type: none">Deselect the checked products.Select "Oracle Database Gateway 12.2", open up this row.Select "Oracle Database Gateway for APPC 12.2".Click Next.
Oracle Universal Installer: Network Software	Specify your network package and click Next .
Oracle Universal Installer: Summary	Click Install .
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 .

Table 4-1 (Cont.) The Oracle Universal Installer: Steps for Installing Oracle Database Gateway for APPC

Prompt	Response
Oracle Net Configuration Assistance: Listener Configuration, 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 Finish .
Execute Configuration scripts	You must run the <code>root.sh</code> configuration script from the <code>\$ORACLE_HOME</code> directory at this point. Leave the installation open, run the script as the <code>root</code> user from another window, then come back to the installation screen and click OK to continue.
Oracle Universal Installer: End of Installation	Click Exit .

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 `$ORACLE_HOME/install` directory. The default file name is `make.log`.

 **Note:**

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 configuration of the gateway.

Deinstalling Oracle Database Gateway for APPC

This section 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

The Deinstallation Tool (`deinstall`) is available in the installation media before installation, and is available in Oracle home directories after installation. It is located in the path `$ORACLE_HOME/deinstall`.

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

The command 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 silent or response file mode. If you use the `-silent` flag, then you must use the `-paramfile` flag, and provide a parameter file that contains the configuration values for the Oracle home that you want to deinstall or deconfigure.

You can generate a parameter file to use or modify by running `deinstall` with the `-checkonly` flag. The `deinstall` command then discovers information from the Oracle home that you want to deinstall and deconfigure. It generates the properties file, which you can then use with the `-silent` option.

You can also modify the template file `deinstall.rsp.tmpl`, located in the response folder.

- `-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. The `-checkonly` flag generates a parameter file that you can use with the `deinstall` command.

- `-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/inventory/response`
- From an unzipped archive file from Oracle Technology Network: /
`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 response files`

Use this flag to provide a path other than the default location where the properties file (deinstall.rsp.tmpl) is saved.

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 Oracle Technology Network: /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 command option flags.

Removing Oracle Software

Complete the following procedure to remove Oracle software:

1. Log in as the installation owner.
2. Run the `deinstall` command, providing information about your servers as prompted.

5

Configuring Your Oracle Network

The instructions in this section describe how to configure the network if the gateway is utilizing 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=/oracle/pga/12.2)
  (PROGRAM=pg4asrv)
)
```

where: `/oracle/pga/12.2` 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=/oracle/pga/12.2)
  (PROGRAM=pg4t4ic)
)
```

where: `/oracle/pga/12.2` is the 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`:

```
pgaiipc=
  (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:

For the following cases:

- If your gateway and Oracle database are not on the same system,
- If the gateway and the Oracle database are on the same system but the Oracle database Listener is different than the gateway listener

You must define the Oracle database to PGAU by adding a service name to `tnsnames.ora` on the system where your gateway is located.

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 is located; and
- *ora_server* is the SID of the Oracle database.

Ensure that you start the defined listener(s).

See Also:

Oracle Database Net Services Administrator's Guide for more information about configuring the network

Example 5-1 Proceed with Configuring the Communication Package for the Gateway

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

Configure the profiles to define LU6.2 conversations with the OLTP. Refer to the appropriate chapter from the following list to read about the SNA communication package or the TCP/IP package for your platform.

- For Linux for Intel Pentium-based 32-bit systems, refer to [Configuring the SNA Communication Package on Linux](#) .
- For AIX-Based Systems, refer to [Configuring the SNA Communication Package on AIX-Based Systems](#).
- For Solaris Operating System (SPARC 64-bit), refer to [Configuring the SNA Communication Package on Solaris](#) .

In addition, if your communication protocol is SNA, refer to "[Sample listener.ora file for a Gateway Using SNA](#)" and "[Sample tnsnames.ora file for a Gateway Using SNA](#)" in [Gateway Initialization Parameters for SNA Protocol](#).

If your communication protocol is TCP/IP, refer to "[Sample listener.ora File for a Gateway Using TCP/IP](#)" and "[Sample tnsnames.ora File for a Gateway Using TCP/IP](#)" in [Gateway Initialization Parameters for TCP/IP Communication Protocol](#) .

6

Configuring the SNA Communication Package on Linux

The Oracle Database Gateway for APPC uses the SNA Advanced Program to Program Communication (APPC/LU6.2) protocol to communicate with an OLTP. Linux for Intel Pentium-based 32-bit system support for APPC is provided by IBM Communications Server for Linux.

The following topics describe how to configure your SNA Server on a Linux for Intel Pentium-based 32-bit system to run the Oracle Database Gateway for APPC, using IBM Communications Server for Linux.

- [Using SNA Security Validation](#)
- [Processing Inbound Connections](#)
- [Independent Versus Dependent LUs](#)
- [Creating IBM Communications Server Definitions for the Gateway](#)
- [Testing the Connection](#)
- [Resume Configuration of the Gateway](#)

 **Note:**

The term SNA Server is used in this guide to generally refer to the IBM Communications Server.

Using SNA Security Validation

When the gateway receives an RPC request to start a remote transaction program, the gateway attempts to start an APPC conversation with online transaction processing (OLTP). Before the conversation can begin, a session must start between the Linux Logical Unit (LU) and the OLTP LU.

SNA and its various access method implementations (including SNA Server and VTAM) provide security validation at session initiation time, enabling 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 gateway SNA Server definitions and in similar parameter structures in the OLTP to be accessed. Refer to the appropriate 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 the appropriate OLTP documentation for detailed information about this topic.

 **Note:**

For CICS, one security option is not supported by the gateway: ATTACHSEC=PERSISTENT, specified in the CICS CONNECTION definition, requires capability that is not yet available in the gateway.
However, the ATTACHSEC=LOCAL, ATTACHSEC=IDENTIFY, ATTACHSEC=VERIFY, and ATTACHSEC=MIXIDPE security options are fully supported by the gateway.

Independent Versus Dependent LUs

Oracle recommends that you use 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 using the same OLTP through the independent LU.

Dependent LUs support only a single active session. The CP (Control Point for the node) 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 you do not need to alter the configuration for the Oracle Database Gateway for APPC and the gateway server.

The operational impact of dependent LUs is that the first client application can initiate a conversation through the Oracle 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 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 conversation requires that you define additional dependent LUs on the remote host, plus additional IBM Communication Server configuration entries, which define the additional dependent LUs on the gateway 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.

 **See Also:**

Refer to PGAU DEFINE TRANSACTION SIDEPROFILE and LUNAME parameters in Chapter 2, "Procedural Gateway Administration Utility," in *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 in [Creating IBM Communications Server Definitions for the Gateway](#) in hierarchical order. This section provides an overview of IBM Communications Server definitions in relation to the Oracle Database Gateway for APPC. 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.

Creating IBM Communications Server Definitions for the Gateway

IBM Communications Server definitions are created using the SNA Node Configuration tool, while the 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 SNA definitions for IBM Communications Server using the SNA Node Configuration tool. All of the tasks described in this section are performed within SNA Node Configuration.

The section includes the following topics:

- [Creating the Configuration](#)
- [Creating the Node](#)
- [Creating Devices](#)
- [Choosing the Device Type](#)
- [Configuring a LAN Device](#)
- [Creating Peer Connections](#)
- [Defining the Link Station](#)
- [Defining the Adjacent Node](#)
- [Creating Local LUs](#)
- [Defining Local LUs](#)
- [Creating Partner LUs](#)

- Defining Partner LUs
- Creating the CPI-C Side Information Profile

Creating the Configuration

The SNA Node Configuration screen prompts you to specify 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 next prompts 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**.

In the Define the Node dialog box:

- a. Select the **Basic** tab.
- b. Enter information in the Control Point, Local Node ID, and Node Type boxes.

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

3. Click **OK**.

Creating Devices

To configure communication devices:

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

To configure a LAN device:

1. Select the **Basic** tab.
2. Choose the Adapter to use and the Local SAP. The other tabs provide options for network tuning parameters.
3. Click **OK**.

Creating Peer Connections

To create peer connections:

1. Click **Peer Connections**.
2. Click **Create**.

Defining the Link Station

To define the link station:

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

To define the adjacent node:

1. Select the **Adjacent Node** tab.
2. Enter the Adjacent CP name of the remote system and select its CP type.

You might 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

To create local LUs for the node:

1. Click **Local LU 6.2 LUs**.
2. Click **Create**.

Defining Local LUs

To define local LUs:

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

To create partner LUs:

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

Defining Partner LUs

To define partner LUs:

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

To define the CPI-C profile that will be used to create the gateway:

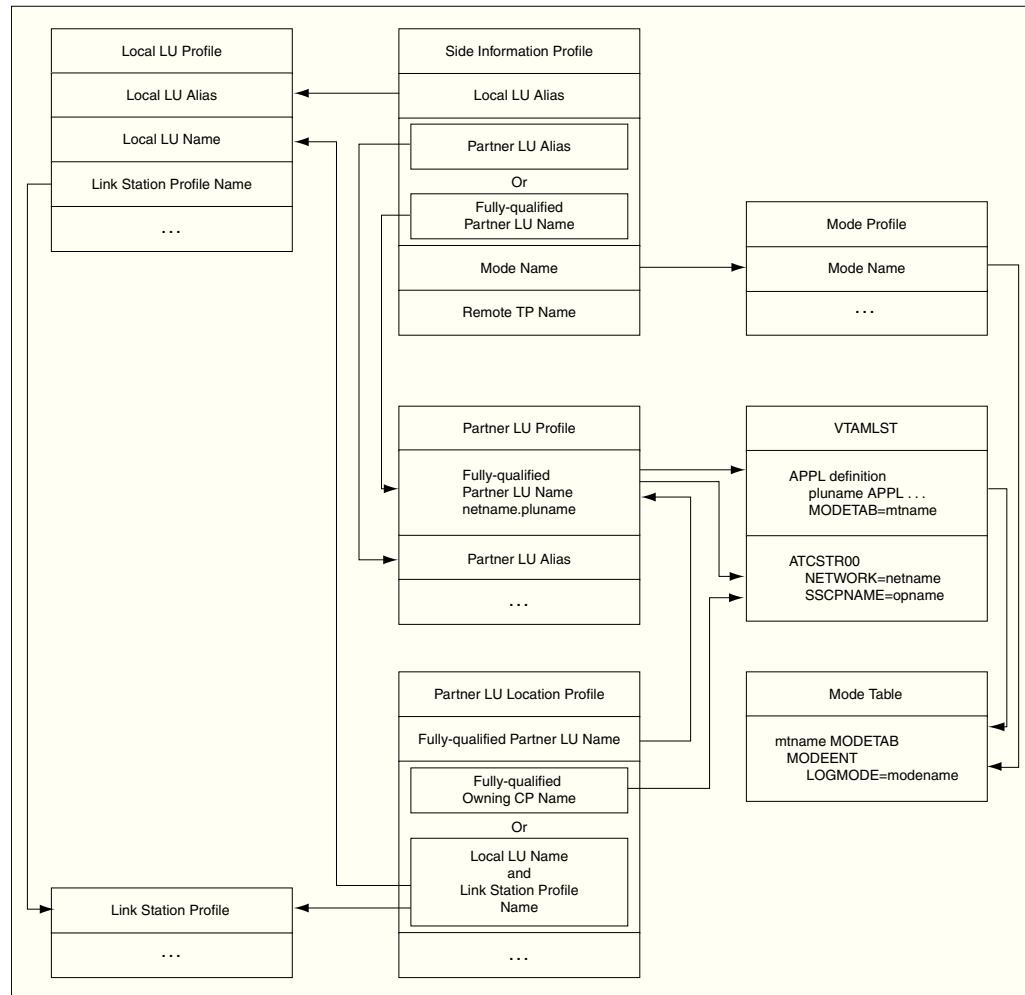
1. Click the **CPI-C Side Information Definitions**.
2. Click **Create**.

Testing the Connection

Before proceeding with the gateway configuration tasks, ensure that your connection is working.

[Figure 6-1](#) shows the relationship between SNA Server definitions and the VTAM definitions on the host.

Figure 6-1 SNA server definitions and VTAM



Resume Configuration of the Gateway

When you have finished configuring the SNA communication package for your Linux for Intel Pentium-based 32-bit System, proceed to [Configuring the OLTP](#) to continue configuring the network.

Configuring the SNA Communication Package on AIX-Based Systems

The Oracle Database Gateway for APPC uses the SNA Advanced Program to Program Communication (APPC/LU6.2) protocol to communicate with online transaction processing (OLTP). All AIX-based system support for APPC is provided by IBM SNA server product.

This product requires a stored set of definitions, called profiles, to support connections between the gateway and the applications using OLTP. Each profile consists of a profile name, a profile type, and a set of fields describing the profile. The fields in a profile type are generally a mixture of operating parameter values and names of other SNA profiles relevant to the profile.

Refer to the following topics if your gateway uses the SNA communication protocol. The topics describe how to create and activate SNA server profiles.

 **Note:**

When you finish following the instructions in this chapter to configure your communication protocol, refer to [Configuring Your Oracle Network](#) to continue your network configuration.

- [Processing Inbound Connections](#)
- [Independent Versus Dependent LUs](#)
- [Creating SNA Profiles for the Gateway](#)
- [Profile Types](#)
- [SNA Server Profiles](#)
- [Activating Profiles](#)
- [Resume Configuration of the Gateway](#)

Processing Inbound Connections

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

 **Note:**

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.

Independent Versus Dependent LUs

Oracle recommends independent LUs for 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 (SNA server for AIX, 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 you do not need to alter the Oracle Database Gateway for APPC configuration, 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 Oracle 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 a 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 profiles, which define the additional dependent LUs on the IBM pSeries workstation. The TIP that initiates the conversation must specify the different Partner LU through a different side information profile or by overriding the LU name.

 **See Also:**

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* and the SNA server-side information profile discussed in the section, [Side Information Profile](#)

Creating SNA Profiles for the Gateway

You can create and modify the SNA server profile definitions using menus in the AIX System Management Interface Tool (smit).

Maintenance of SNA server profiles is normally done by a user with `root` access. The following information is intended for the user who creates profiles for the gateway. You should have some knowledge of SNA before reading this section.

By using smit, you should be able to accept most of the defaults. The default values assigned to many of the fields in a new set of profiles are acceptable for the gateway.

The `$ORACLE_HOME/dg4appc/sna` subdirectory contains a sample set of profiles for the gateway in the `pgasna.export` file.

Before building the SNA server profiles, examine the appropriate sample export file to determine the profiles needed, their contents, and their interrelationships. The export file format is text-oriented, and each field of each profile is clearly labeled. You can print a copy of the export file to use while working with the profiles in a smit session.

Profile Types

Several types of SNA server profiles are relevant to gateway APPC/LU6.2 operation. Each profile can be created and edited using a corresponding smit menu that can be reached from the Communications Applications and Services primary menu choice.

The profiles are presented in hierarchical order. The profile types that are lowest in the hierarchy are discussed first. This matches the logical sequence in which to create the profiles. You can use the smit list menu to specify the profile names.

SNA Server Profiles

Refer to the appropriate vendor documentation for a complete discussion of SNA profiles. This section is an overview of SNA server profiles in relation to the Oracle Database Gateway for APPC. It includes the following topics:

- [SNA Node Profile](#)
- [Link Station Profile](#)
- [Mode Profile](#)
- [Local LU Profile](#)
- [Partner LU Profile](#)
- [Partner LU Location Profile](#)
- [Side Information Profile](#)

SNA Node Profile

The SNA node profile defines miscellaneous SNA system defaults. Set the "Maximum number of sessions" and "Maximum number of conversations" fields to values large enough to handle the maximum number of concurrent gateway conversations

anticipated, plus any non-gateway sessions and conversations that are in use on your system by other applications.

Set the "Recovery Resource Manager (RRM) enabled?" field to no, unless you already have other applications running on your AIX system that require this field to be set to yes. For example, CICS pSeries and Encina need this field set to yes.

Link Station Profile

The Link Station Profile and the related DLC profile describe and control the connection of the IBM pSeries to the network. Details on profile contents are not discussed here because Oracle Database Gateway for APPC does not impose special requirements on these profiles. The sample profile distributed in `pgasna.export` includes a profile created for a Token-Ring network connection. The Link Station Profile name is specified later in the Partner LU Location Profile, if one is necessary.

Mode Profile

The Mode Profile specifies parameters that determine:

- APPC/LU6.2 parallel session limits
- send and receive pacing values
- SNA RU size
- the mode name that is sent to OLTP at session initiation

The mode name that you specify must be defined to the OLTP communications software. Choose the mode name in addition to the other mode parameters after consulting the user who is responsible for configuring the OLTP communications software.

The parameters that are related to parallel session limits determine the maximum number of concurrent conversations allowed between a gateway instance and the OLTP. This equates to the maximum number of concurrently active remote transaction program invocations through the gateway instance.

The mode name, for example, `ORAPLU62`, is specified later in the Side Information Profile.

 **Note:**

Do not confuse the Mode Profile with the mode name.

Local LU Profile

The Local LU (Logical Unit) Profile describes the SNA LU through which the gateway communicates.

An LU name must be assigned to the gateway. The LU name assigned to the gateway might be required elsewhere in the SNA network. Contact the SNA network administrator to determine the correct LU name to specify in the profile.

To create a Local LU Profile, perform the following steps:

1. Set the "Local LU name" to the LU name assigned to the gateway.
2. An alias should be assigned to the LU using the "Local LU alias" field. This alias is used later in the side information profile.
3. Set the "Local LU is dependent" field to no.

The Local LU Profile name is specified later in the side information profile.

Partner LU Profile

The Partner LU Profile describes the SNA LU of the OLTP system with which the gateway communicates. The name of the OLTP LU and the name of your SNA network must be specified in this profile. Contact your SNA network administrator to determine the correct LU and network names.

To create a Partner LU Profile, perform the following steps:

1. Set the "Fully qualified partner LU name" field to the network name, followed by a period, followed by the OLTP LU name. For example, network.oltplu.
2. You can assign an alias to the partner LU name by setting the "Partner LU alias" to the value of your choice. This enables you to reference the partner LU without knowing the fully qualified partner LU name and minimizes the change if the partner LU name is changed.
3. Set the "Parallel sessions supported?" field to yes unless your OLTP does not support parallel sessions.
4. If you plan to use SNA session or conversation security, then set the "Session security supported?" and "Conversation security supported?" fields as required. These settings require the Session Security and Conversation Security Profiles.

See Also:

The vendor documentation for more information

Partner LU Location Profile

The Partner LU Location Profile is used when the remote host where the Partner LU is located is not an APPN-capable node. Many mainframe systems do not have APPN capability. For example, z/OS systems running VTAM versions before version 4 do not support APPN. Also, if your hardware connection is through a front-end processor running NCP versions before version 5, then APPN is not supported. In these cases, the Partner LU Location Profile can be used to specify the name of the System Services Control Point (SSCP) or Control Point (CP), which owns the network connection to the partner LU.

Set the "Fully qualified partner LU name" field to the network name, followed by a period, followed by the OLTP LU name. For example, network.oltplu.

Set the "Partner LU location method" and associated fields as required by your network configuration. If you use the `owning cp` option, then the "Fully qualified owning Control Point (CP) name" field should be set to the SSCP or CP name, which owns the network connection to the partner LU. For VTAM, the SSCP name is the value of the VTAM `NETID` start parameter, usually found in `VTAMLST` member `ATCSTR00`.

Side Information Profile

The side information profile is a required profile, which is used to identify target OLTP systems to be accessed through Oracle Database Gateway for APPC.

The side information profile identifies the following:

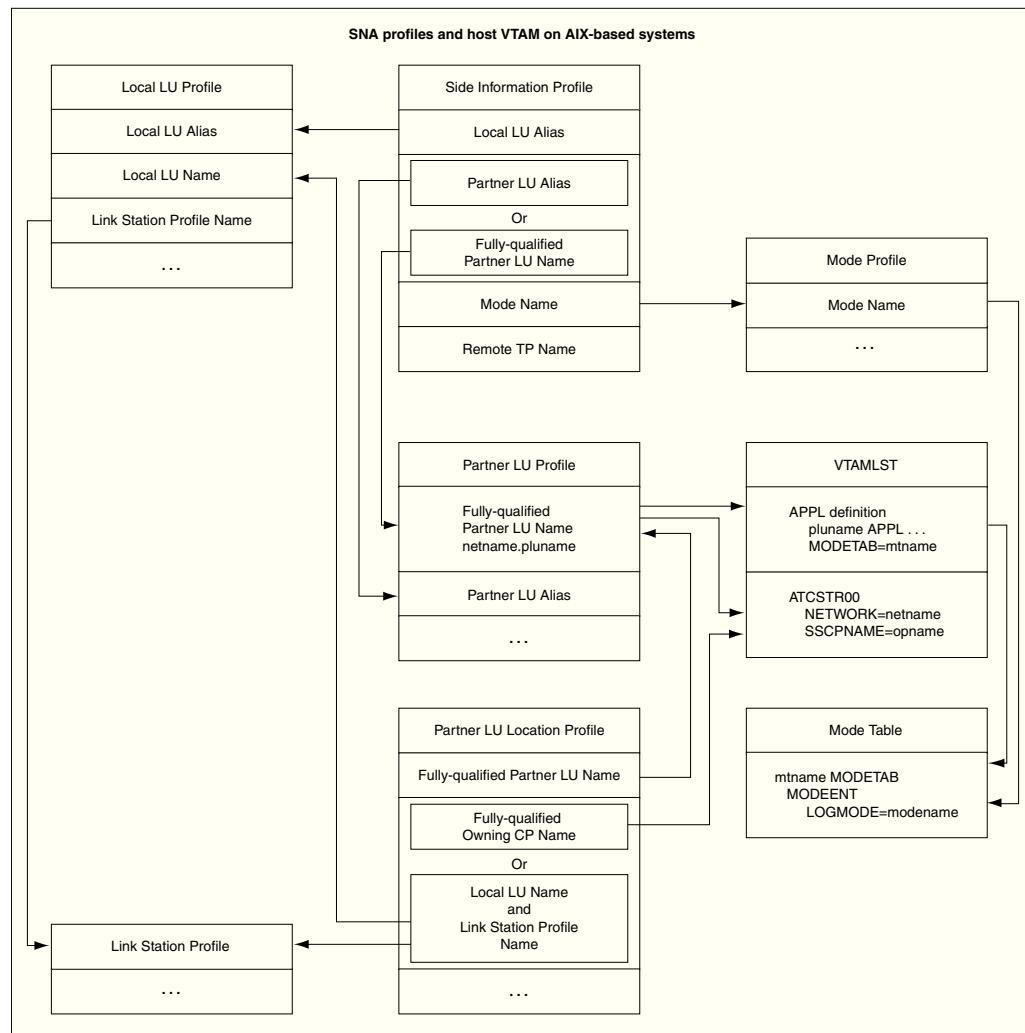
- the local LU alias
- the partner LU alias or fully-qualified name
- the remote transaction program name (optional)
- the mode name

Set the profile information as follows for each side information profile field:

- Set the Local LU or Control Point alias to the alias assigned to the local LU in the Local LU Profile.
- Set the Mode name to the actual mode name as specified in the Mode Profile.
- Set the Remote transaction program name (RTPN) using the actual remote TP name or a dummy name to be overridden at execution time.
- Set the "RTPN in hexadecimal?" field to yes, if the remote TP name is hexadecimal.
- If there is a field for "Partner LU alias", then add the alias assigned to the partner LU in the Partner LU Profile.
- If there is a field for "Fully qualified partner LU name", then add the fully qualified partner LU name of the partner LU.

[Figure 7-1](#) shows the relationship between SNA server profiles and the VTAM definitions on the host.

Figure 7-1 Relationship Between SNA Profiles and Host VTAM Definitions



Activating Profiles

After you have built all the necessary SNA server profiles for communicating with the remote host, you must verify the profiles. Use the "Verify Configuration Profiles" option under the "Advanced Configuration" option of the smit SNA server menu. Then use smit to start the link station profile.

See Also:

the vendor documentation for more information about using smit to start link stations

Resume Configuration of the Gateway

When you finish configuring the SNA communication package for your AIX-based system, proceed to [Configuring the OLTP](#) to continue configuring the network.

Configuring the SNA Communication Package on Solaris

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 Solaris Operating System (SPARC 64-bit) is provided by the SNAP-IX product.

The following topics describe how to configure SNAP-IX on a Solaris system to run Oracle Database Gateway for APPC.

 **Note:**

When you finish following the instructions in this chapter, refer to [Configuring Your Oracle Network](#) to continue network configuration.

- [Processing Inbound Connections](#)
- [Configuring SNAP-IX Version 6](#)
- [Resuming Gateway Configuration](#)

Processing Inbound Connections

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

Note that for CICS, the gateway provides the following support:

- ATTACHSEC=LOCAL, ATTACHSEC=IDENTIFY, ATTACHSEC=VERIFY, and ATTACHSEC=MIXIDPE are fully supported by the gateway.
- ATTACHSEC=PERSISTENT, specified in the CICS CONNECTION definition, requires capability that is not yet available in the gateway.

Configuring SNAP-IX Version 6

The following topics describe how to configure SNAP-IX version 6.

- [Before You Begin](#)
- [SNAP-IX Configuration Tool](#)
- [Creating SNAP-IX Profiles for the Gateway](#)
- [Independent Versus Dependent LUs](#)
- [Creating SNA Definitions for the Gateway](#)
- [Sample SNAP-IX Definitions](#)

- [Configuring SNAP-IX](#)
- [Starting xsnaadmin](#)
- [Testing the Connection](#)

Before You Begin

You must specify parameters that are unique to your system to configure SNAP-IX version 6 properly. Before you begin, request these parameters from your network administrator.

SNAP-IX Configuration Tool

All SNAP-IX product configuration is done using the `xsnaadmin` program. This tool is an X-Windows application that provides a graphical interface to view and modify the current SNAP-IX configuration and the current running state of the host SNA node.

Creating SNAP-IX Profiles for the Gateway

Oracle Database Gateway for APPC requires a stored set of definitions, called Side Information Profiles, to support connections between the gateway and gateway servers. Each profile consists of a profile name and a profile type, which is a set of fields describing the profile. The fields in a given profile type are generally a mix of operating parameter values and names of other SNA profiles relevant to the profile. Each functional part of APPC, such as the Mode, Remote Transaction Program name, and Logical Unit (LU), is described by a distinct profile type.

Independent Versus Dependent LUs

The gateway configuration can accommodate either independent or dependent LUs. If you choose to use dependent LUs, or are restricted to using dependent LUs, the gateway functions properly. If a dependent LU is correctly defined, then you do not need to make changes to the configuration of the gateway, nor should any changes be needed to the gateway server. However, Oracle recommends that you use independent LUs for the gateway because they support multiple parallel sessions or conversations. This means that multiple Oracle client applications can be active simultaneously with the same gateway server through the independent LU.

In contrast to independent LUs, dependent LUs support only a single active session. The CP (Control Point for the Node) queues each additional conversation request from the gateway behind an already active conversation. In other words, conversations are single-threaded for dependent LUs.

The operational impact of dependent LUs is that the first client application can initiate a conversation through the gateway with the gateway server, but while that session is active (which could be for seconds, minutes, or 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 not be a problem. However, at some point, you might require additional concurrent conversations for testing or for other application development. Having more than one conversation requires that additional dependent LUs be defined on the

remote host. Additional configuration entries must be added to SNAP-IX. Additional Side Information Profiles should be defined to use the new dependent LUs. Gateway instances should be created and configured to use these new Side Information Profiles.

Creating SNA Definitions for the Gateway

SNAP-IX definitions are stored in the following files, which are located in the /etc/opt/sna directory:

- SNA node definitions: sna_node.cfg
- SNA domain definitions: sna_domain.cfg

These files are created and maintained with the xsnaadmin tool. Maintenance of SNA definitions is usually done by a user with administrative authority. The following information is intended for a user creating SNA definitions for the gateway. You must have some knowledge of SNA before reading this section.

Sample SNAP-IX Definitions

The \$ORACLE_HOME/dg4appc/sna subdirectory contains a set of sample SNAP-IX definition files for the gateway, which are created with the xsnaadmin. These sample files are sna_domain.cfg and sna_node.cfg. SNA definitions are very specific to the host and SNA network. As such, the sample definitions provided will not work without being tailored for the local host and SNA network.

Configuring SNAP-IX

This section describes the process of creating SNA definitions for SNAP-IX, using xsnaadmin. All configuration is done using the various dropdown menus and panels in xsnaadmin. The following configuration descriptions follow the samples provided. Please tailor the various SNA values for your local host and SNA network.

Starting xsnaadmin

Use the following commands to invoke xsnaadmin. The DISPLAY environment variable must be set correctly. If you are running xsnaadmin from the local console, then DISPLAY should already be set. If you are running xsnaadmin from a remote X display, then set DISPLAY to the host name or IP address of that display.

```
$ DISPLAY=<your_display>:0
$ export DISPLAY
$ xsnaadmin &
```

On startup of xsnaadmin, the main screen opens and displays the current configuration of the local SNA node.

Configuring the SNA node

To configure the SNA node, you need to do the following:

1. From the Services menu, select **Configure Node Parameters**.

2. In the Node Parameters dialog box, enter the APPN support type, Control Point Name, Control Point Alias, and Node ID as needed. The Control Point Name is composed of the SNA Network Name and the CP name of the local host.
3. Click **OK**.

Adding a Port

To add a new port, from the Services menu, select **Connectivity and New Port**.

1. In the Add to *Nodename* dialog box, select the Port type and click **OK**.
2. In the SAP dialog box, enter a Port name and network card number. The Port name is used to logically name the physical network card that you are using and is used to bind a Service Access Port to the card for SNA protocols. Usually, you can accept the values provided in the dialog box. If a different network card is needed, however, enter the card number as reported by the `dmesg` command.
3. Click **OK**.

Create a Link Station

When the Port has been defined, you must create a Link Station. The Link Station represents the SNA node of the remote host of the gateway server. But before you create the Link Station, you must create a Remote Node definition as described in the following procedure:

1. From the Services menu, select **APPN and Add Remote Node**.
2. In the dialog box, enter the SNA CPNAME of the remote node and click **OK**.

Now you can create a Link Station as follows:

1. From the Services menu, select **Connectivity and New Link Station**. In the dialog box, select the Port previously defined and click **OK**.
2. In the Link Station dialog box, enter a name for the Link Station, choose the SNA Port name, the type of link activation, and the LU Traffic type. For maximum flexibility, select the **Any** option.
3. For Independent LU traffic, specify the Remote Node name. Click **Remote Node** and select the node you previously created, and then click **OK**. Choose the type of the Remote node, typically a Network node.
4. For Dependent LU traffic, specify the role of the Remote node, typically 'host', the Local Node ID, and optionally, Remote Node ID.
5. Specify the Contact Information. Contact information contains the MAC address of the remote host as well as the SAP number.
6. Click **Advanced** for additional parameters of the Link Station. The Token Ring Parameters dialog box shows additional parameters of the Link Station. These parameters affect initial XID contact and retransmission times and limits. You usually do not need to change the default values.
7. Click **OK**.

Creating Local LUs

When the Remote Node definitions have been made, create the Local LU names for the local host as follows:

1. From the Services menu, select **APPN and New Local LU**. In the local LU dialog box, enter the name of the local LU and an alias. This name must correspond to the VTAM definitions on the remote gateway server host for the UNIX host.
2. Click **OK**.

Creating Partner LUs

Now define a Partner LU that represents the LU that the gateway server is using to communicate.

1. From the Services menu, select **APPN and New Partner LUs** and **Partner LU on Remote Node**.
2. In the Partner LU dialog box, enter the Partner LU name and characteristics. The Partner LU name contains the SNA Network Name as well as the LU name of the remote LU. Enable parallel session support. The location is the name as the Remote Node name. You can click **Location** for a list.
3. Click **OK**.

Creating Mode and CPI-C Profiles

When the local and remote LU definitions have been created, create the necessary Mode and CPI-C definitions.

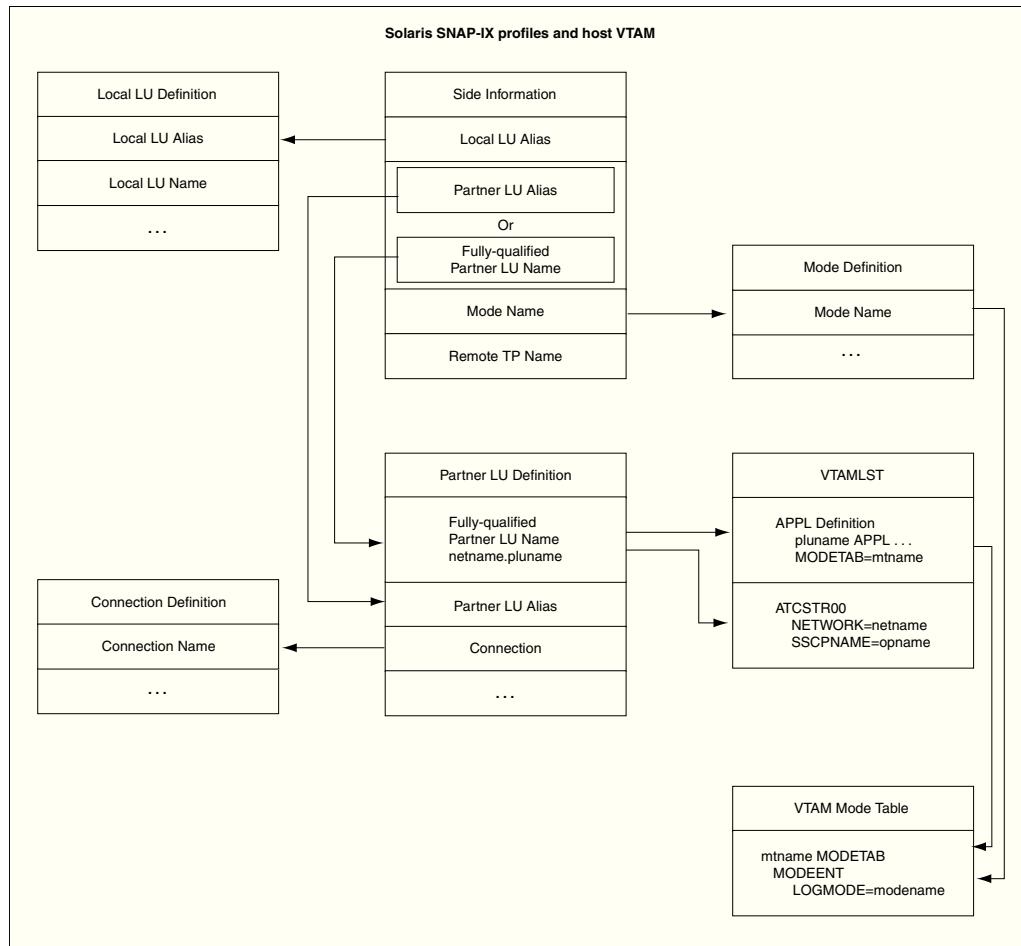
1. From the Services menu, select **APPN and Modes**. In the Modes dialog box, click **New** to add a new mode.
2. In the Mode dialog box, enter the Mode Name and other session parameters. The recommended name for a gateway mode is **CICSPGA**. Contact your Remote Host system administrator for appropriate mode parameters.
3. Click **OK**.
4. Now that the Mode has been defined, create the CPI-C Side Information Profile, which the gateway uses as a connection name. From the menu, select **APPN and CPI-C**.
5. In the CPI-C destination names dialog box, click **New** to add a new Profile.
6. In the CPI-C destination dialog box, enter the Profile name, Local LU name, Partner TP, Partner LU and mode, and Security option. The partner TP name is the name of the host transaction program or a dummy value to be overridden in the TIP.
7. For the Local LU, you may specify a specific LU or choose the default LU. For the Partner LU, enter either the full LU name or the alias created previously.
8. Enter **ORAPLU62** for the mode name. Choose the type of security for these sessions to use. This affects how session authorization is done.
9. Click **OK**.

Testing the Connection

Before proceeding with gateway configuration tasks, ensure that your connection is working. Perform this by starting the SNAP-IX Node and then starting the individual link stations.

[Figure 8-1](#) shows the relationship between SNAP-IX definitions and the VTAM definitions on the remote host.

Figure 8-1 Relationship Between SNAP-IX Definitions and Host VTAM Definitions



Resuming Gateway Configuration

When you have finished configuring the SNA communication package for Solaris, proceed to [Configuring the OLTP](#) to continue configuring the network.

Configuring the Gateway Using SNA Communication Protocol

The following topics describe how to configure the Oracle database for a gateway using the SNA protocol on your UNIX based platform. It also shows you how to configure commit-confirm, should you choose to implement it.

The topics provide the steps necessary to verify installation and configuration of the gateway components, including optional commit-confirm.

- [Before You Begin](#)
- [Preparing to Configure a Gateway Installation/Upgrade](#)
- [Oracle Database 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:

- Oracle database
- UNIX system
- Network
- OLTP

Before You Begin

Gateway configuration using SNA communication protocol requires you to input parameters unique to your system to properly configure the gateway and SNA communications interface.

Refer to [Configuration Worksheet](#) for a worksheet listing the installation parameters that you 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 or migrating the gateway:

- When Oracle Database and the Gateway Are Installed in the Same ORACLE_HOME
- When Oracle Database and the Gateway Are Installed on Separate Systems
- When Oracle Database and the Gateway Are on the Same System but in Different Directories

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

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

When Oracle Database and the Gateway Are Installed in the Same ORACLE_HOME

You do not need to transfer files. Proceed to [Oracle Database Configuration: First-Time Gateway Installations](#).

When Oracle Database and the Gateway Are Installed on Separate Systems

You need to perform the following tasks if Oracle database and the gateway are installed on separate systems:

1. Locate the gateway administrative files in the gateway \$ORACLE_HOME/dg4appc/admin directory. All files in this directory that have the .sql, .pkh, or .pkb suffixes must be copied into a similarly-named directory in the Oracle database Oracle home directory.
2. Now locate the gateway demo files and subdirectories in the \$ORACLE_HOME/dg4appc/demo directory of the gateway. Copy the pgavsn.sql and pgaecho.sql files into a similarly named directory in 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 \$ORACLE_HOME/dg4appc/demo/CICS gateway files into a similarly named directory in the Oracle database.

Note:

Before transferring the files from the \$ORACLE_HOME/dg4appc/demo directory, ensure 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 Procedural Gateway Administrative Utility (PGAU).

When Oracle Database and the Gateway Are on the Same System but in Different Directories

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

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

```
$ echo $ORACLE_HOME  
/oracle/pga/12.2
```

and your server Oracle home is located in the `/oracle/pga/12.2` directory, then you need to do the following:

```
$ ORACLE_HOME=/oracle/pga/12.2; export ORACLE_HOME
```

2. Now create the directories with the following commands:

```
$ cd $ORACLE_HOME  
$ mkdir dg4appc  
$ mkdir dg4appc/admin
```

3. Use whatever file transfer mechanism is available on your system to copy all of the `.sql`, `.pkh`, and `.pkb` files from the gateway Oracle home `$ORACLE_HOME/dg4appc/admin` directory to the Oracle database Oracle home `$ORACLE_HOME/dg4appc/admin` directory.
4. You might also transfer the demo files from the gateway directory to the Oracle database directory. Copy the files and directory recursively from the gateway Oracle home `$ORACLE_HOME/dg4appc/demo` directory to the Oracle database `$ORACLE_HOME/dg4appc/demo` directory.

For example:

```
$ cp -p -R /oracle/pga/12.2/dg4appc/demo $ORACLE_HOME/dg4appc
```

Note:

Before transferring the files from the `$ORACLE_HOME/dg4appc/demo` directory, ensure 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.

If this is a first-time installation, proceed with [Oracle Database Configuration: First-Time Gateway Installations](#).

If this is an upgrade, proceed with [Upgrading or Migrating the Oracle Database from Previous Gateways](#).

Following those steps, you might want to perform the [Optional Configuration Steps to Permit Multiple Users](#).

Oracle Database 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 the SYS user.

- 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, 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 1.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 the SYS user.

- 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:

```
SQL> @$ORACLE_HOME/rdbms/admin/utlraw.sql
SQL> @$ORACLE_HOME/rdbms/admin/prvtrawb.plb
```

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

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

- 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 PL/SQL Packages and Types Reference* for more information about 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, use SQL*Plus to connect to the Oracle database as the SYS user.

- b. From SQL*Plus, run the `ut1pg.sql` and `prvtpgb.plb` scripts in the Oracle database `$ORACLE_HOME/rdbms/admin` directory, in the following order:

```
SQL> @$ORACLE_HOME/rdbms/admin/ut1pg.sql  
SQL> @$ORACLE_HOME/rdbms/admin/prvtpgb.plb
```

4. Install the Heterogeneous Services (HS) catalogs.
- If necessary, use SQL*Plus to connect to the Oracle database as the `SYS` user.
 - 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. Step c creates the Heterogeneous Services catalog.

- If it is necessary to create the Heterogeneous Services catalog, enter the following command:
- ```
SQL> $ORACLE_HOME/rdbms/admin/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 the `SYSTEM` user. 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.

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

6. Create the gateway administrator user `PGAADMIN` and install the PG DD.
- Use SQL\*Plus to connect to the Oracle database as the `SYSTEM` user.
  - 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 more information about password issues, refer to the *Oracle Database SQL Language Reference*.

```
SQL> @$ORACLE_HOME/dg4appc/admin/pgacr8au.sql
```

- Use SQL\*Plus to connect to the Oracle database as user `PGAADMIN`.
  - From SQL\*Plus, run the `pgddcr8.sql` script in the `$ORACLE_HOME/dg4appc/admin` directory. This script installs the PG DD.
- ```
SQL> @$ORACLE_HOME/dg4appc/admin/pgddcr8.sql
```

- From SQL*Plus, connect to the Oracle database as the `SYS` user.
- 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 PL/SQL Packages and Types Reference*.

- a. If necessary, 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.

```
SQL> @$ORACLE_HOME/dg4appc/admin/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").
 - 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.

```
SQL> @$ORACLE_HOME/dg4appc/admin/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.

```
SQL> @$ORACLE_HOME/dg4appc/admin/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 12c Release 2 (12.2) of the gateway.

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

1. Use SQL*Plus to connect to the Oracle database as the `SYS` user.
2. 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.

```
SQL> @$ORACLE_HOME/rdbms/admin/prvtrawb.plb
```

3. Install the `UTL_PG` package body. From SQL*Plus, run the `prvtppgb.plb` script from the `$ORACLE_HOME/rdbms/admin` directory. This script upgrades the `UTL_PG` package body.

```
SQL> @$ORACLE_HOME/rdbms/admin/prvtppgb.plb
```

The `prvtrawb.plb` and `prvtppgb.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 the following section.

If You Must Reinstall Package Specifications

If the UTL_RAW or UTL_PG package has been invalidated or deinstalled, the prvtrawb.p1b and prvtppg.p1b scripts might not complete successfully and you might have to reinstall the package specifications.

If you do reinstall the package specifications, 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

From SQL*Plus, run the utlraw.sql Script:

1. If necessary, use SQL*Plus to connect to the Oracle database as the SYS user.
2. From SQL*Plus, run the utlraw.sql and utlpg.sql scripts in Oracle database \$ORACLE_HOME/rdbms/admin directory, in the following order, to upgrade their respective package specifications:

```
SQL> @$ORACLE_HOME/rdbms/admin/utlraw.sql  
SQL> @$ORACLE_HOME/rdbms/admin/utlpg.sql
```

Step 2 Repeat Installation of UTL_RAW and UTL_PG Package Body

After the scripts have run, repeat Steps 2 and 3 in [Upgrading or Migrating the Oracle Database from Previous Gateways](#). Then proceed to the section titled "[Upgrading PGAU From Previous Gateway Releases](#)".

 **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

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

1. If necessary, 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.

Configuring the Oracle Database for Gateways to Coexist

You might 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 (PGDD), a version 9 gateway cannot communicate with a version 10 data dictionary. Thus, if you upgrade your data dictionary to a 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 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 the SYSTEM user.
 - 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.

```
SQL> @$ORACLE_HOME/dg4appc/admin/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.

```
SQL> @$ORACLE_HOME/dg4appc/admin/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:

```
$ 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:

```
$ 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 user ID:

- a. Use SQL*Plus to connect to the Oracle database as user PGAADMIN.
- b. From SQL*Plus, run the appropriate 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.

```
SQL> @$ORACLE_HOME/dg4appc/admin/pgddapub.sql
SQL> @$ORACLE_HOME/dg4appc/admin/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 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. Invoke PGAU in the directory path where the PGAU control files are generated and where TIPs are stored:

```
$ 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. Invoke SQL*Plus in the same directory path where the newly-generated TIP specifications are stored.

```
$ 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 the Gateway

To configure the gateway, perform the following steps:

1. Customize the Oracle Database Gateway for APPC parameters.

Parameters specific to the gateway are supplied in the gateway parameter file, `initsid.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 `initsid.ora` file, substitute your `dg4appc` SID name for "sid" in this file name.

The `initsid.ora` file contains both APPC and TCP/IP parameters, separated by a description. You must modify the `initsid.ora` file by deleting the TCP/IP parameters. Refer to [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.

 **Note:**

Before performing the following step, refer to [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 `initsid.ora` file.

 **Note:**

Misspelled parameters are ignored. However, if the `$ORACLE_HOME/dg4appc/admin/initsid.ora` file is missing, all calls to the gateway fail and return a `PGA-20928` error.

Configuring Commit-Confirm

 **Note:**

If you are planning to implement commit-confirm, read the detailed explanation of commit-confirm's 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
- Configuring the OLTP

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

Configuring the Oracle Database for Commit-Confirm

The Oracle database installation 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 the SYSTEM user.
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, 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 Oracle database as the PGADBA user.
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 Oracle database.

Configuring Gateway Initialization Parameters for Commit-Confirm

The gateway initialization parameters are described in [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 your commit-confirm gateway.

Configuring the OLTP for Commit-Confirm

Configuration of the OLTP includes defining and installing the following:

- Commit-confirm transaction log database
- Commit-confirm forget or recovery transaction
- 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 "[Verifying OLTP Configuration for Commit-Confirm](#)".

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, follow the steps to verify the OLTP configuration for commit-confirm.

 **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 "[Oracle Database Configuration: First-Time Gateway Installations](#)". 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 your Oracle database as `PGADMIN`.
2. Run `$ORACLE_HOME/dg4appc/demo/pgavsn.sql`.

```
SQL> @$ORACLE_HOME/dg4appc/demo/pgavsn.sql
```

The server version number banner appears at your terminal. The following output appears:

```
Oracle Database Gateway for APPC.  
Version 12.2.0.1.0 Wed Aug 24 14:39:15  
2016
```

```
Copyright (c) Oracle Corporation 1979,2016. All rights reserved.
```

```
PL/SQL procedure successfully completed.
```

3. Run \$ORACLE_HOME/dg4appc/demo/pgaecho.sql.

```
SQL> @$ORACLE_HOME/dg4appc/demo/pgaecho.sql
```

```
You will receive the following output:
```

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

Verifying the OLTP Configuration

The procedure for verifying your OLTP configuration varies, depending on which OLTP you are using and depending upon which 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 in the following sections for verifying installation:

- [CICS Verification](#)
- [IMS/TM Verification](#)
- [APPC/MVS Verification](#)



Note:

If you have not completed the file transfers detailed in "[Preparing to Configure a Gateway Installation/Upgrade](#)", complete them now, before proceeding to the next step.

CICS Verification

If your OLTP is CICS Transaction Server for z/OS, 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 configure your CICS Transaction Server for z/OS for the gateway:

```
FLIP THIS MESSAGE
```

The following output appears:

```
EGASSEM SIHT PILF
```

2. Log on to UNIX.
3. Modify the pgacics.sql file, which is located 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:

- CICS transaction ID
 - Side Profile name
 - Logmode entry name
4. Ensure that the SNA communication package on your system has been started.
 5. Log on to your CICS Transaction Server for z/OS and run this transaction, where *name* is the name of the CONNECTION definition installed by the DFHCSDUP job run in the CICS configuration steps:

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

This transaction activates the CICS connection to UNIX.

6. Use SQL*Plus to connect to your Oracle database as PGAADMIN.
7. Run pgacics.sql.

```
SQL> @$ORACLE_HOME/dg4appc/demo/CICS/pgacics.sql
```

The following message appears:

```
=> Congratulations, your gateway is communicating with CICS <=
Your CICS Transaction Server for z/OS installation verification is complete.
```

IMS/TM Verification

If your 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 your IMS/TM system and enter the following transaction, replacing FLIP with the transaction ID you chose for FLIP when you configured your IMS/TM system for the gateway:

```
FLIP THIS MESSAGE
```

The following output appears on your terminal:

```
EGASSEM SIHT PILF
```

2. Log on to UNIX.
3. Modify the pgaims.sql file, which is located 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:
 - IMS/TM transaction ID
 - Side Profile Name
 - Logmode entry name
4. Ensure that the SNA communication package on your system has been started.
5. Use SQL*Plus to connect to Oracle database as PGAADMIN.
6. Run pgaims.sql.

```
SQL> @$ORACLE_HOME/dg4appc/demo/IMS/pgaims.sql
```

The following message appears:

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

Your IMS/TM installation verification is now complete.

APPC/MVS Verification

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

1. Verify that your APPC/MVS subsystem is active.
2. Log on to UNIX
3. Modify the `pgamvs.sql` file, which is located 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:
 - APPC/MVS transaction ID
 - Side Profile Name
 - Logmode entry name
4. Ensure that the SNA communication package on your system has been started.
5. Use SQL*Plus to connect to your Oracle database as `PGAADMIN`.
6. Run `pgamvs.sql`.

```
SQL> @$ORACLE_HOME/dg4appc/demo/MVS/pgamvs.sql
```

The following message appears:

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

Your APPC/MVS installation verification is now complete.

Verifying OLTP Configuration for Commit-Confirm

If you chose to configure commit-confirm in [Configuring Commit-Confirm](#), 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 on 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 your applications to reduce the complexity of the code changes to your 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 programming language. Both of these subroutines are written in 370 assembler to eliminate any interlanguage interface complexities and compiler dependencies.
- Forget and recovery transactions

These are provided in the `pgareco.asm` files. Forget and recovery transactions must be installed into your OLTP and accessible through APPC so that the gateway can invoke them to forget a transaction once it has been successfully committed, and to query transaction 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 your OLTP.

Installing Sample Applications

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

APPC/MVS

- z/OS data set information

CICS Transaction Server for z/OS

- DB2 inquiry
- DB2 multi-row 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, 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 following directories and files:

- \$ORACLE_HOME/dg4appc/CICS/sample_CICS_applications.txt
- \$ORACLE_HOME/dg4appc/IMS/sample_IMS_applications.txt
- \$ORACLE_HOME/dg4appc/MVS/sample_MVS_applications.txt

10

Configuring the OLTP

The following sections describe how to configure the online transaction processing OLTP.

- **If your communications protocol is SNA:** Proceed to [Configuring the OLTP for an SNA Environment](#).
- **If your communications protocol is TCP/IP:** Proceed to [Configuring the OLTP for a TCP/IP Environment](#).

 **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 you are using SNA support, you do not need to specify EDIT=ULC in the IMS TRANSACT macro.

Configuring the OLTP for an SNA Environment

The steps for configuring 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 corresponding to your OLTP from the following sections:

- [Configuring CICS Transaction Server for z/OS](#)
- [Configuring IMS/TM](#)
- [Configuring APPC/MVS](#)

 **Note:**

You need to perform the configuration steps for an OLTP only if this is the first time that you are configuring 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 communication with the gateway:

1. Configure MVS VTAM for the SNA communication package 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 a 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:
 - `dfhcstdup.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 `dfhcstdup.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 link edits 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 UNIX. 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 your IMS system for the APPC.
2. Configure MVS VTAM for the SNA APPC connection to UNIX. 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 link edits the pgaflip.asm file into a load module library that is accessible to your IMS/TM system and creates an 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 UNIX. 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 data set (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 appropriate for the device type on which it is located. Approximately two tracks of 3390 disk space are required with one directory block. Oracle suggests naming this partitioned data set (PDS) ORAPGA.APPCMVS.SAMPLIB.
4. Using a 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: 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: 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 data set. Ensure that you change the data set 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, refer to [Gateway Configuration Using SNA Communication Protocol](#). Refer to "Configuring Commit-Confirm" for more information on configuring commit-confirm.

Configuring the OLTP for a TCP/IP Environment

These are the steps for configuring 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 your IMS system.
2. Configure IMS Connect

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

3. Using a 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: Assembler source for IMS FLIP transaction
 - pgaflip.jcl: JCL to assemble and linkedit IMS FLIP transaction
 - imsgen.asm: IMS stage 1 gen definitions for the IMS FLIP transaction
4. 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.
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 link edits the pgaflip.asm file into a load module library that is accessible to your IMS/TM system and creates an 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 [Gateway Configuration Using TCP/IP Communication Protocol](#) to complete configuration of the gateway and its components.

11

Configuring the Gateway Using TCP/IP Communication Protocol

The following topics describe how to configure Oracle database for a gateway using TCP/IP for IMS Connect on your UNIX platform. It also provides the steps necessary to verify installation and configuration of the gateway and OLTP components.

- [Before You Begin](#)
- [Preparing to Configure a Gateway Installation/Upgrade](#)
- [Configuring Oracle Database: First-Time Installation](#)
- [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 Oracle Database Gateway for APPC using TCP/IP support for IMS Connect involves working with the following components:

- Oracle database
- UNIX system
- Network
- OLTP

Before You Begin

Gateway configuration using TCP/IP communication protocol requires you to input parameters unique to your system to correctly configure the gateway and TCP/IP communications interface.

Refer to [Configuration Worksheet](#) for a worksheet listing the installation parameters that 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 and Oracle database relationship when you are installing, upgrading, or migrating the gateway:

- [When Oracle Database and the Gateway Are Installed in the Same ORACLE_HOME](#)

- [When Oracle Database and the Gateway Are Installed on Separate Systems](#)
- [When Oracle Database and the Gateway Are on the Same System but in Different Directories](#)

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

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

When Oracle Database and the Gateway Are Installed in the Same ORACLE_HOME

You do not need to transfer files. Proceed to [Configuring Oracle Database: First-Time Installation](#).

When Oracle Database and the Gateway Are Installed on Separate Systems

When Oracle database and the gateway are installed on separate systems, you need to perform the following tasks:

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 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 on Oracle database.
3. 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 Oracle home at \$ORACLE_HOME/dg4appc/demo/IMS directory as well. Ensure that you generate the required TIPs and transfer them as well.

When Oracle Database and the Gateway Are on the Same System but in Different Directories

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

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

```
$ echo $ORACLE_HOME  
/oracle/pga/12.2
```

and the server Oracle home is located in the /oracle/pga/12.2 directory, then you need to do the following:

```
$ ORACLE_HOME=/oracle/pga/12.2; export ORACLE_HOME
```

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

```
$ cd $ORACLE_HOME  
$ mkdir dg4appc  
$ mkdir dg4appc/admin
```

```
$ mkdir dg4appc/demo
$ mkdir dg4appc/demo/IMS
```

3. Use whatever file transfer mechanism is available on your system to copy all of the .sql, .pkh, and .pkb files from the gateway Oracle home \$ORACLE_HOME/dg4appc/admin directory to the Oracle database Oracle home \$ORACLE_HOME/dg4appc/admin directory.
4. You might 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 \$ORACLE_HOME/dg4appc/demo directory to the Oracle database \$ORACLE_HOME/dg4appc/demo directory.
5. You might 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 Oracle home \$ORACLE_HOME/dg4appc/demo/IMS directory as well. Ensure that you generate the required TIPs and transfer them as well.

Proceed with [Configuring Oracle Database: First-Time Installation](#). Following those steps, you may want to perform the [Optional Configuration Steps to Permit Multiple Users](#).

Configuring Oracle Database: First-Time Installation

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

1. Ensure that the UTL_RAW PL/SQL package has been installed on Oracle database. All TIP specifications generated by Procedural Gateway Administrative Utility (PGAU) use UTL_RAW, which provides routines for manipulating raw data.
 - a. Use SQL*Plus to connect to Oracle database as the SYS user.
 - 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 executing Step 1.d below.

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

- c. Use SQL*Plus to connect to 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:

```
SQL> @$ORACLE_HOME/rdbms/admin/utlraw.sql
SQL> @$ORACLE_HOME/rdbms/admin/prvtrawb.sql
```

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

- a. If necessary, use SQL*Plus to connect to the Oracle database as the **SYS** user.
- b. Enter the following command:

```
SQL> DESCRIBE DBMS_OUTPUT
```

The `DESCRIBE` statement produces output on 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 PL/SQL Packages and Types Reference* for more information about the `DBMS_OUTPUT` package. After installing the `DBMS_OUTPUT` package successfully, run the `DESCRIBE` statement.

If the `DESCRIBE` statement indicates success, then 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, use SQL*Plus to connect to Oracle database as the **SYS** user.
- b. From SQL*Plus, run the `utlpg.sql` and `prvtpgb.plb` scripts in the Oracle database `$ORACLE_HOME/rdbms/admin` directory in the following order:

```
SQL> @$ORACLE_HOME/rdbms/admin/utlpg.sql
SQL> @$ORACLE_HOME/rdbms/admin/prvtpgb.plb
```

4. Install the Heterogeneous Services (HS) catalogs.

- a. If necessary, use SQL*Plus to connect to Oracle database as the **SYS** user.
- 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 Oracle database and you can proceed to Step 2.

If the `DESCRIBE` statement does not indicate success, then you must create Heterogeneous Services catalogs, and you must follow Step 4.c:

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

```
SQL> @$ORACLE_HOME/rdbms/admin/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 the **SYSTEM** user. 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.

```
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 Oracle database as the **SYSTEM** user.

- 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*.

```
SQL> @$ORACLE_HOME/dg4appc/admin/pgacr8au.sql
```

- c.** Use SQL*Plus to connect to the Oracle database as PGAADMIN.
- d.** From SQL*Plus, run the pgddcr8.sql script in the \$ORACLE_HOME/dg4appc/admin directory. This script installs the PG DD.

```
SQL> @$ORACLE_HOME/dg4appc/admin/pgddcr8.sql
```

- e.** From SQL*Plus, connect to the Oracle database as the SYS user.
- f.** Grant execute 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 Gateway Initialization Parameters for TCP/IP Communication Protocol](#) 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 Oracle database as the SYSTEM user.
- b.** Enter the following command:

```
SQL> select owner, table_name
  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
  from dba_indexes
 where index_name = 'PGA_TCP_IMSC_IND';
```

Each SELECT statement must produce one row. The following output is the result of the first SELECT statement:

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

The following output 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 preceding output on your screen, then you can skip to Step 8. 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 1.

- c. From SQL*Plus, run the `pgaimsc.sql` script in the Oracle database `$ORACLE_HOME/dg4appc/admin` directory:

```
SQL> @$ORACLE_HOME/dg4appc/admin/pgaimsc.sql
```

8. Install the TIP trace access PL/SQL routines. These routines require that the `DBMS_PIPE` standard PL/SQL package is installed and that `PGAADMIN` has execute privileges on it. For more information on `DBMS_PIPE`, refer to the *Oracle Database PL/SQL Packages and Types Reference*.

- a. If necessary, 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.

```
SQL> @$ORACLE_HOME/dg4appc/admin/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](#)).

- a. Use SQL*Plus to connect to Oracle database as `PGAADMIN`.
- b. From SQL*Plus, run the `gpglocal.pkh` script in the `$ORACLE_HOME/dg4appc/admin` directory. This script compiles the `GPGLOCAL` package specification.
- c. From SQL*Plus, run the `gpglocal.pkb` script in the `$ORACLE_HOME/dg4appc/admin` directory. This script compiles the `GPGLOCAL` package body.

```
SQL> @$ORACLE_HOME/dg4appc/admin/gpglocal.pkh
```

```
SQL> @$ORACLE_HOME/dg4appc/admin/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 communication package protocol and you want to utilize the new TCP/IP feature, then the TIPs have to be regenerated by PGAU with certain mandatory `NLS_LANGUAGE` and Side Profile Settings. Specify the correct ASCII character set in the `DEFINE TRANSACTION` command.

This is because the gateway assumes that the correct user exit in IMS Connect is being used, which would translate between the correct 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 12c Release 2 (12.2) of the gateway.

1. Upgrade your Oracle Database Gateway for APPC to current version levels as follows:
 - a. Use SQL*Plus to connect to Oracle database as the `SYS` user.
 - 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.

```
SQL> @$ORACLE_HOME/rdbms/admin/prvtrawb.plb
```
 - c. Install the `UTL_PG` package body. From SQL*Plus, run the `prvtpgb.plb` script from the `$ORACLE_HOME/rdbms/admin` directory. This script upgrades the `UTL_PG` package body.

```
SQL> @$ORACLE_HOME/rdbms/admin/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 the following section.

If You Must Reinstall Package Specifications

If the `UTL_RAW` or `UTL_PG` package has been invalidated or deinstalled, then the `prvtrawb.plb` and `prvtpgb.plb` scripts might not complete successfully. You may 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 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, ensure 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, use SQL*Plus to connect to the Oracle database as the `SYS` user.

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:

```
SQL> @$ORACLE_HOME/rdbms/admin/utlraw.sql
SQL> @$ORACLE_HOME/rdbms/admin/utlpg.sql
```

Step 2 Repeat Installation of UTL_RAW and UTL_PG Package Body

After the scripts have run, repeat Steps b and c in "[Upgrading or Migrating the Oracle Database from Previous Gateways](#)". Then, proceed to the section titled "[Upgrading PGAU from Previous Gateway Releases](#)".

 **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

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

1. If necessary, use SQL*Plus to connect to Oracle database as the `PGAADMIN` user.
2. From SQL*Plus, run the `pgddupgr.sql` script in the `$ORACLE_HOME/dg4appc/admin` directory. This script upgrades the PG DD.

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 Oracle database as the `SYSTEM` user.
 - 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.

```
SQL> @$ORACLE_HOME/dg4appc/admin/pgddcr8s.sql
```
2. Create roles for accessing the PG DD, performing definitions of transactions, and generating TIP specifications. The `PGAADMIN` user can grant these roles to other users as necessary.
 - a. Use SQL*Plus to connect to Oracle database as user `PGAADMIN`.
 - b. From SQL*Plus, run the `pgddcr8r.sql` script in the `$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.

```
SQL> @$ORACLE_HOME/dg4appc/admin/pgddcr8r.sql
```

3. Grant access to PGA required packages.

TIP developers require access to the following PL/SQL packages, which are shipped with 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 run these packages must be made to TIP developers.

These grants can be private as in the following example:

```
$ 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:

```
$ 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 given 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 user ID:

- a. Use SQL*Plus to connect to Oracle database as PGAADMIN.
- b. From SQL*Plus, run the appropriate 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.

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

```
SQL> @$ORACLE_HOME/dg4appc/admin/pgddapub.sql
```

or

```
SQL> @$ORACLE_HOME/dg4appc/admin/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 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 installation where the PG DD is stored. Then issue the following SQL command:

```
SQL> GRANT INSERT ON PGA_DATA_VALUES TO PGDDGEN
```

- a. Start PGAU from the directory path where the PGAU control files are generated and where TIPs are stored:

```
$ pgau
PGAU> CONNECT PGAADMIN/pgaadmin@database_specification_string
PGAU> GENERATE trancode
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. Start SQL*Plus in the same directory path where the newly-generated TIP specifications are stored.

```
$ 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 "[Configuring the Gateway](#)".

Configuring the Gateway

To configure the gateway, perform the following:

1. Tailor the gateway parameters.

There are a number of parameters specific to Oracle Database Gateway for APPC when it is using TCP/IP for IMS Connect. These are supplied in the gateway parameter file, `initsid.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 `initsid.ora` file, substitute your gateway SID name for "sid" in this file name.

The `initsid.ora` file contains both APPC and TCP/IP parameters separated by a description. You must modify the `initsid.ora` file by deleting the APPC parameters. Refer to "[Gateway Initialization Parameter File Using TCP/IP](#)" 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.

 **Note:**

Before performing the following step, refer to "[Gateway Initialization Parameter File Using TCP/IP](#)" 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 `initsid.ora` file.

 **Note:**

Misspelled parameters are ignored. However, if the \$ORACLE_HOME/dg4appc/admin/initsid.ora file is missing all calls to the gateway fail and return a PGA-20928 error.

Loading the PGA_TCP_IMSC Table

Gateway users who want to employ the TCP/IP protocol do so by using the pg4tcpmap tool.

The pg4tcpmap tool is located 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 utilizes 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 initsid.ora. Refer to [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 the pg4tcpmap commands and instructions for using them, as well as an example of the table. Refer to Chapter 8, "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:

```
$ $ORACLE_HOME/bin/pg4tcpmap
```

Refer to [Output for the pg4tcpmap Tool](#) 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 Oracle Database: First-Time Installation](#). 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 PGAADMIN.
2. Run \$ORACLE_HOME/dg4appc/demo/pgavsn.sql.

```
SQL> @$ORACLE_HOME/dg4appc/demo/pgavsn.sql
```

The server version number banner appears at your terminal. The following output appears:

```
Oracle Database Gateway for APPC (extension TCP/IP for IMS connect).  
Version  
12.2.0.1.0 Wed Aug 24 14:52:36 2016
```

```
Copyright (c) Oracle Corporation 1979,  
2016. All rights reserved.
```

```
PL/SQL procedure successfully completed.
```

3. Run \$ORACLE_HOME/dg4appc/demo/pgaecho.sql.

```
SQL> @$ORACLE_HOME/dg4appc/demo/pgaecho.sql
```

You will receive the following output:

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

Verifying the OLTP Configuration

Use the following procedure to verify your OLTP configuration.

 **Note:**

If you have not completed the file transfers detailed in [Preparing to Configure a Gateway Installation/Upgrade](#), 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 communication protocol to TCP/IP.

TCP/IP only: If you have existing TIPs that were generated previously on a gateway using the SNA communication package protocol and you want to utilize 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 because 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.

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 your IMS/TM system for the gateway):

`FLIP THIS MESSAGE`

The following output appears on your terminal:

`EGASSEM SIHT PILF`

2. Log on to the UNIX based system.

 **Note:**

If you have not completed the file transfers detailed in [Preparing to Configure a Gateway Installation/Upgrade](#), complete them now, before proceeding to the next step.

3. Modify the `pgaims.sql` file, which is located 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:
 - IMS/TM transaction ID
 - Side Profile Name
 - Logmode entry name
4. Ensure that the TCP/IP communication protocol on your system has been started.
5. Using SQL*Plus, connect to your Oracle database from `PGAADMIN`.

6. Run pgaims.sql.

```
SQL> @$ORACLE_HOME/dg4appc/demo/IMS/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 your OLTP is recommended to help you to fully understand how the gateway works and how it interfaces with your 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 call a gateway that uses TCP/IP as the communication protocol, and you use EBCDIC as the language in the control files, then you must change the language from EBCDIC to ASCII. Some examples of control files that might 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 newer releases of the product. Wherever possible, the sample applications use the sample databases provided with the database products.

For this release, the entire documentation about installing and using the sample applications is available in:

```
$ORACLE_HOME/dg4appc/demo/IMS/sample_IMS_applications.txt
```

12

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 security requirements.

Read the following topics to learn about the capabilities and limitations of the Oracle Database Gateway for APPC:

- [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 your security scheme, you must understand the existing security requirements and expectations in your environment. Because you are enabling application access to different databases on different systems, you may need to merge multiple security cultures. When you connect several 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 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 Security Requirements include 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 third-generation language (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 the gateway. The type of logon authentication that you use determines the resulting Oracle user ID and can affect gateway operation.

The following types of authentication are available:

- 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.

- Operating system authentication

With operating system authentication, the underlying server 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 with such a user ID, the application supplies a forward slash (/) for a user ID and does not supply a password.

To perform operating system authentication, the server determines the requester 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 are authenticated when they log 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 multithreaded server configurations. Refer to your platform-specific 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 simply if 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. Oracle database makes no distinction as to the type of use (such as read-only versus 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 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 Oracle user ID and password of the user are sent to the gateway when the connection is opened. If the user logs on to Oracle database with operating system authentication, then the gateway receives no user ID or password from 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 [Using SNA Security Validation](#) 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 the platform's Logical Unit (LU) and the OLTP LU.

APPC support for your platform is provided by a SNA communication package (SNAP-IX for Solaris Operating System (SPARC) and SNA Server for AIX-based systems).

SNA and its various access method implementations, including VTAM and the SNA communication package for your platform, 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 your platform's SNA profiles and in similar parameter structures in the OLTP to be accessed. Refer to the appropriate 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.

SNA Security Option SECURITY=SAME

If you specify `PGA_SECURITY_TYPE=SAME`, the gateway allocates the conversation with SNA option `SECURITY=SAME` and sends only a user ID, without a password, to the OLTP. In this case, your SNA communication package sends the owning user ID of the gateway server executable, `$ORACLE_HOME/bin/pg4asrv`, as the user ID. The user ID that is sent is not the Oracle user ID. This user ID can be viewed with the UNIX `ls` command and can be changed by an authorized user with the `chown` command. Because this user ID is the same for all users of a given gateway instance, this option is of limited use.

 **Note:**

The user ID sent is not translated to uppercase by your SNA communication package. If your OLTP is running on a system which does not allow lowercase user IDs (z/OS, for example), you must set up an uppercase user ID on your platform to be the owner of the gateway executable file.

`SECURITY=SAME` is similar to your platform operating system authentication. It tells the OLTP that the user has already been authenticated at the originating side of the

conversation. There might be configuration parameters or options on the server side that affect whether SECURITY=SAME conversations are accepted. When properly configured, the OLTP only confirms that the user ID itself is valid and then accepts the connection. As with SECURITY=PROGRAM, you can change this using configuration options in many OLTPs.

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 remote transaction program 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:**

As 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 talk to 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 your platform 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 enables you to specify the security conduct for the conversation that is allocated by the gateway for OLTP. Refer to [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.

TPC/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 talks to IMS Connect.

 **Note:**

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, which can be insecure. An encryption feature has been added to Heterogeneous Services making it possible to encrypt parameters values. This is done through the `dg4pwd` utility. For more information on this utility refer to *Oracle Database Heterogeneous Connectivity User's Guide*.

Migrating From Existing Gateways

Migrating to new instances of Oracle Database Gateway for APPC from an existing installation is straightforward, provided you follow some guidelines. The following topics provide information to make these new installations as easy as possible. They also provide the parameters you need if you are using the TCP/IP for IMS Connect communication protocol on your gateway.

The following sections provide information that is specific to this release of Oracle Database Gateway for APPC for IBM AIX on POWER Systems (64-Bit), Linux x86-64, Oracle Solaris on SPARC (64-Bit), and HP-UX Itanium:

- [Migrating An Existing Gateway Instance to a New Release Using SNA Protocol](#)
- [Migrating from an Existing Gateway to TCP/IP Using SNA](#)

Migrating An Existing Gateway Instance to a New Release Using SNA Protocol

Follow these steps to migrate an existing gateway to 12c Release 2 (12.2) of the gateway using the SNA communication protocol.

Note that if you are using the gateway 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 `initsid.ora` file. Proceed to "[Parameter Changes: Version 4 to 12c Release 2 \(12.2\) of the Gateway](#)".

Step 1: Install the New Release

Install the new release of the gateway in a separate directory as outlined in [Installing the Gateway](#).

Note:

Do not install the gateway over a previously existing gateway installation. Doing this corrupts the existing installation.

Step 2: Transferring the `initsid.ora` Gateway Initialization File Parameters

Copy the `initsid.ora` file from the old gateway instance to the new instance.

If you are migrating from Release 9.0.1 or earlier of the gateway, `PGA_TRACE` is not supported. You need to modify the parameter to `TRACE_LEVEL` instead.

 **Note:**

If you are using `TRACE_LEVEL`, you must set the path for the `LOG_DESTINATION` parameter.

If you have encrypted some of the parameters at file `initsid.ora` by using `dg4pwd`, copy the `$ORACLE_HOME/dg4appc/admin/initsid.pwd` file from the old gateway instance to the new instance.

 **Note:**

If you are migrating from Release 9.0.1 or earlier of the gateway, you do not need to copy this file into the new instance.

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 helps to ensure minimum down time between changes to different gateway instances.

Oracle Net Considerations

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 tell Oracle Net that the gateway uses HS facilities. For more information, refer to [Configuring Your Oracle Network](#) .

Parameter Changes: Version 4 to 12c Release 2 (12.2) of the Gateway

This release of Oracle Database Gateway for APPC introduces new and changed initialization parameters if you are migrating from a Version 4 gateway to 12c Release 2 (12.2) of the gateway.

 **Note:**

The "[Parameter Changes: Version 4 to 12c Release 2 \(12.2\) of the Gateway](#)" section does not apply to you if you are migrating to Release 10.2.0 from Version 8 of the Oracle Database Gateway for APPC.

If you are using the gateway TCP/IP support for IMS Connect, you will not be *migrating* from Version 4 to the current release of the gateway. However, you need to place valid Heterogeneous Services parameters into your `initsid.ora` file.

Migration from existing gateways contains references to the particular HS parameters that 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 (`initsid.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 Heterogeneous Connectivity User's Guide*.

Renamed Gateway Initialization File Parameters

Following is a list of the gateway initialization file (`initsid.ora`) parameters that have been renamed in this release of the gateway. The former names of the parameters are shown in parentheses.

- `HS_COMMIT_STRENGTH_POINT` (formerly `COMMIT_STRENGTH_POINT`)
- `HS_DB_DOMAIN` (formerly `DB_DOMAIN`)
- `HS_DB_INTERNAL_NAME` (formerly `DB_INTERNAL_NAME`)
- `HS_DB_NAME` (formerly `DB_NAME`)
- `HS_DESCRIBE_CACHE_HWM` (formerly `DESCRIBE_CACHE_HWM`)
- `HS_LANGUAGE` (formerly `LANGUAGE`)
- `HS_NLS_DATE_FORMAT` (formerly `NLS_DATE_FORMAT`)

- HS_NLS_DATE_LANGUAGE (formerly NLS_DATE_LANGUAGE)
- HS_OPEN_CURSORS (formerly OPEN_CURSORS)
- HS_ROWID_CACHE_SIZE (formerly ROWID_CACHE_SIZE)

Obsolete Parameters

The following parameters are obsolete. If necessary, 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 12c Release 2 (12.2)

The following startup shell script parameter must be added to `initsid.ora` if you are migrating from a Version 4 or Version 8 gateway to 12c Release 2 (12.2) of the Oracle Database Gateway for APPC:

- FDS_CLASS_VERSION

Migrating from Gateway Release 9.0.1 or 9.2.0 or 10.1.0 to Gateway 12c Release 2 (12.2)

No new parameters were added between release 9.0.1 and this release of the gateway.

Migrating from an Existing Gateway to TCP/IP Using SNA

The following sections are for those users who have an existing release of the gateway using the SNA protocol, but who want to switch to using TCP/IP support for IMS Connect.

The TCP/IP support for IMS Connect feature in this release of the gateway enables you to continue to use existing TIPs.

Using 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 [Gateway Configuration Using TCP/IP Communication Protocol](#) for instructions to load the `PGA_TCP_IMSC` table.
2. Make sure 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 `initsid.ora` file:
 - `PGA_TCP_DB`
 - `PGA_TCP_USER`
 - `PGA_TCP_PASS`

You will find descriptions and information about adding these parameters in [Parameter Changes: Version 4 to 12c Release 2 \(12.2\) of the Gateway](#). You will also find descriptions of the parameters in [Gateway Initialization Parameters for TCP/IP Communication Protocol](#).

Note:

If your TIPs from a previous version of the gateway were already defined using a `SideProfileName` 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 still need to map your parameter values using the `pg4tcpmap` tool.

 **Note:**

TIPs must be recompiled when changing communication protocol from SNA to TCP/IP.

If you have existing TIPs that were generated previously on a gateway using the SNA communication package protocol, and you want to use the new TCP/IP feature, then the TIPs 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 because 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.

A

Gateway Initialization Parameters for SNA Protocol

The following topics describe the gateway initialization file location and lists the gateway initialization parameters supported by Oracle Database Gateway for APPC, specifically for the SNA protocol. These parameters are fully documented in "[Migrating An Existing Gateway Instance to a New Release Using SNA Protocol](#)". In addition, the topics contain sample `listener.ora` and `tnsnames` files for a gateway using SNA.

The parameter file for the gateway is located in the `$ORACLE_HOME/dg4appc/admin` directory and is called `initsid.ora`.

- [PGA Parameters](#)
- [PGA_CAPABILITY Parameter Considerations](#)
- [PGA_CONFIRM Parameter Considerations](#)
- [Sample listener.ora file for a Gateway Using SNA](#)
- [Sample tnsnames.ora file for a Gateway Using SNA](#)

 **Note:**

The `initsid.ora` file contains both SNA and TCP/IP parameters. You must modify these files with 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
- `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 Oracle Database Gateway for APPC Using SNA

Parameter	Description
LOG_DESTINATION= <i>logpath</i> <i>h</i>	<i>logpath</i> specifies the destination at which STDERR is reopened. LOG_DESTINATION specifies a directory only and STDERR is reopened to <i>logpath/sid_pid.log</i> where: <ul style="list-style-type: none">• <i>sid</i> is the sid name• <i>pid</i> 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. This parameter has meaning only when the conversation is running with SYNCLEVEL set to a value higher than 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 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 is set to COMMIT_CONFIRM. The password can be encrypted. For more information about encrypting the password, refer to Passwords in the Gateway Initialization File . There is no default value.

Table A-1 (Cont.) PGA Parameters for Oracle Database Gateway for APPC 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 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 is set to COMMIT_CONFIRM and PGA_SECURITY_TYPE is set to PROGRAM. The password can be encrypted. For more information about encrypting the password, refer to Passwords in the Gateway Initialization File . There is no default value.
PGA_RECOVERY_TPNAME	The TP name of the transaction installed in the 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 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 is set to COMMIT_CONFIRM and PGA_SECURITY_TYPE 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: Sends no security parameters SAME: Sends only a user ID PROGRAM: Sends a user ID and password The default is NONE. For further information on these options, refer to Security Requirements .
PGA_SIGDANGER (AIX only parameter)	Action to take on receipt of a SIGDANGER signal from the system indicating a shortage of paging space. The following is a valid value: IGNORE: Ignores the signal The default is IGNORE.

Table A-1 (Cont.) PGA Parameters for Oracle Database Gateway for APPC Using SNA

Parameter	Description
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 between 0 and 255. The default is 0, indicating no tracing. Any value between 1 and 255 will turn tracing on.

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 remote transaction programs will be doing.

The `READ_ONLY` setting should always be used when the remote transaction programs are read-only, that is, when the remote transaction programs perform no database updates. `READ_ONLY` should never be used when the remote transaction programs perform database updates. For example, if the `READ_ONLY` setting is chosen and if a remote transaction program invoked by the gateway performs updates to a foreign database, then Oracle database does not provide any integrity protection for those updates. Furthermore, the `READ_ONLY` mode enables a gateway transaction to be part of a distributed transaction that might update several other databases. If the gateway invokes a remote transaction program that performs updates in this situation and if a failure occurs, then the database updated by the remote transaction program is not synchronized with the other databases.

In cases where the remote transaction programs perform updates to foreign databases, there are two options for the value of `PGA_CAPABILITY`:

- `SINGLE_SITE`
- `COMMIT_CONFIRM`

Each of these options provides protection against data integrity problems by allowing `COMMIT` and `ROLLBACK` requests to be forwarded to the remote transaction program and by informing Oracle database about the distributed update and recovery capabilities of the gateway. The particular option depends on the design of the remote transaction programs and on the capabilities of the OLTP (online transaction processor) 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 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 remote transaction program 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

Configuring the OLTP for Commit-Confirm in this guide. For SINGLE_SITE capability, the remote transaction program is responsible for performing the required 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, the remote transaction program must perform COMMIT processing and then respond to the gateway with an APPC CONFIRMED response. On receipt of SEND_ERR, the remote transaction program 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 remote transaction program 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 remote transaction programs which perform updates to foreign databases outside the scope of the Oracle database control. 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 remote transaction programs 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 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 sids, PGAI and PGAU.
2. Define two aliases in `tnsnames.ora` that connect to the two new sids, PGAI and PGAU.
3. Define two database links in 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 on 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 correct 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 sent a COMMIT request. 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 and 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 = /oracle/pga/12.2)
  (PROGRAM = pg4asrv)
)
)
```

Sample tnsnames.ora file for a Gateway Using SNA

```
ORA920 =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = bay.us.example.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

The following topics describe the parameters needed for a gateway featuring the TCP/IP for IMS Connect communication protocol. It also provides a sample output of the pg4tcpmap tool. In addition, the topics contain 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 `initsid.ora`.

 **Note:**

The `initsid.ora` file contains both SNA and TCP/IP parameters. You will need to modify these files with the suitable parameters.

PGA Parameters

The 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 for Oracle Database Gateway for APPC Using TCP/IP for IMS Connect

Parameter	Description
LOG_DESTINATION= <i>logpath</i>	<p><i>logpath</i> specifies the destination at which STDERR is reopened. LOG_DESTINATION specifies a directory only and STDERR is reopened to <i>logpath/sid_pid.log</i> where:</p> <ul style="list-style-type: none"> • <i>sid</i> is the sid name • <i>pid</i> is the process ID assigned to the gateway <p>NOTE: This parameter will be used for the pg4tcpmap tool when you set the 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.</p>
PGA_CAPABILITY	<p>PGA transaction capability. The following are valid values: READ_ONLY or RO: Read-only capabilities.</p> <p>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.</p> <p>The default is SINGLE_SITE.</p>
PGA_SECURITY_TYPE	<p>TCP/IP conversation security option. This controls what security parameters are sent to the OLTP. The following are valid values:</p> <p>NONE: Sends no security parameters</p> <p>PROGRAM: Sends a user ID and password</p> <p>The default is NONE.</p> <p>For further information on these options, refer to Security Requirements.</p> <p>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 <i>Oracle Database Gateway for APPC User's Guide</i>. If you have already loaded the table pga_tcp_imsc and you did not first specify the RACF group name, delete the row and reinsert it with the value for the RACF group name.</p>
PGA_TCP_DB	<p>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.</p> <p>There is no default value.</p>
PGA_TCP_PASS	<p>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 Passwords in the Gateway Initialization File.</p> <p>There is no default value.</p>

Table B-1 (Cont.) PGA Parameters for Oracle Database Gateway for APPC 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 is 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 [Loading the PGA_TCP_IMSC Table](#) 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 angle brackets (<>).

```
$ pg4tcpmap

PG4TCPMAP: Release 12.2.0.1.0 - Production on Wed Aug 24 15:09:00 2016

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*'
'LLL prefix present' is 'N'
PG4TCPMAP is complete.

```

Sample listener.ora File for a Gateway Using TCP/IP

The following is an example of a `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 = /oracle/pga/12.2)  
(PROGRAM = pg4t4ic)  
)  
)
```

Sample tnsnames.ora File for a Gateway Using TCP/IP

The following is an example of a `tnsnames.ora` file for a gateway using TCP/IP:

```
ORA920 =  
(DESCRIPTION =  
(ADDRESS_LIST =  
(ADDRESS = (PROTOCOL = TCP)(HOST = bay.us.example.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

The following topic 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*.

Gateway Initialization File

This file is known as `initsid.ora` and it contains parameters that govern the operation of the gateway. If you are using the SNA protocol, refer to [Gateway Initialization Parameters for SNA Protocol](#) for more information. Refer to [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 "[RPC Functions](#)" 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](#)" for detailed information about how the `dg4pwd` utility works.

tg4tcpmap 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 constituting 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 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 C of the *Oracle Database Gateway for APPC User's Guide* for supported numeric data type conversions.

D

Configuration Worksheet

Table D-1 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). You can use this table as a 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	_____
Appropriate Name for Each Side Information Profile	for: SNA creating CPI-C symbolic destination names (side information profiles), general information	_____
Appropriate 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	_____
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	_____

Table D-1 (Cont.) Parameters for Configuring Gateway and Communication Protocols

Name of Parameter Needed	Purpose	Your Specific Parameters Here
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, data store 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

Symbols

(HS=) (TNSNAMES parameter for Oracle Net), [5-1](#)
 with TCP/IP protocol, [5-2](#)
\$ cd \$ORACLE_HOME \$ mkdir dg4appc, [9-3](#), [11-2](#)
\$ORACLE_HOME, [9-10](#), [9-15](#), [10-3](#), [11-11](#)

A

action items
 for installing the gateway, [4-5](#)
activating and verifying SNA server profiles, [7-7](#)
AIX
 activating profiles, [7-7](#)
 configuring SNA, [5-1](#)
 creating mode profiles, [7-4](#)
 processing inbound connections, [7-1](#)
 SNA conversation security, [12-3](#)
 SNA security option SECURITY =SAME, [12-4](#)
 SNA security option SECURITY=NONE, [12-4](#)
 SNA security option
 SECURITY=PROGRAM, [12-4](#)
 SNA security validation, [12-3](#)
 System Management Interface Tool, [7-3](#)
AIX-based communications package
 SNA server, [7-1](#)
ALTER USER command, [9-5](#), [11-5](#)
american_america_us7ascii, [13-5](#)
APPc, [1-1](#)
 conversation security option, [A-3](#), [B-2](#)
APPC/MVS
 installation verification, [9-16](#)
 verification of configuration, [9-16](#)
architecture
 components of the gateway, [1-5](#)
ASCII
 automatic conversion, [1-3](#)
authentication
 for operating system, [12-2](#)
 for Oracle, [12-2](#)
 types, security, [12-2](#)

B

backout possibilities during migration, [13-2](#)

C

choosing the device type
 on Linux, [6-4](#)
CICS, [1-9](#), [3-3](#)
ATTACHSEC parameter
 on Solaris, [8-1](#)
installation verification
 on gateway using SNA, [9-14](#)
security options not supported by the
 gateway
 on Solaris, [8-1](#)
transaction ID, [9-15](#)
verifying configuration
 on gateway using SNA, [9-14](#)
CICS Transaction Server for z/OS
 authentication mechanism
 on all platforms, [12-4](#)
OLTP
 configuration verification, [9-14](#)
 TP name length, [A-3](#)
command
 SET, [9-11](#), [11-11](#)
 COMMIT, [A-4](#)
 COMMIT_CONFIRM, [A-2](#), [A-4](#), [B-2](#)
 and PGA_CAPABILITY parameter, [A-4](#)
 capability, [A-4](#)
commit-confirm, [A-4](#)
 configuring, [9-11](#), [9-13](#)
 gateway initialization parameters, [9-12](#)
 OLTP, [9-12](#)
 Oracle database, [9-11](#)
 sample applications, [9-16](#)
 transaction log, [9-17](#)
 COMMIT/ROLLBACK, [3-3](#)
communications
 between server, gateway and remote host, [1-7](#)
 needed for Solaris, [3-3](#)
configuration
 gateway directories, [11-2](#)

configuration verification
 OLTP
 on gateway using SNA, [9-13](#)
 on gateway using TCP/IP for IMS Connect, [11-12](#), [11-13](#)

configuring
 commit-confirm, [9-11](#), [9-13](#)
 gateway
 optional steps to allow multiple users using SNA, [9-8](#)
 optional steps to allow multiple users using TCP/IP for IMS Connect, [11-8](#)
 gateway directories, [9-2](#)
 Oracle database, [9-11](#)
 upgrading from previous releases, [9-6](#), [11-7](#)
 Oracle database for gateway using TCP/IP for IMS Connect
 pre-configuration steps, [11-1](#)
 SNAP-IX, [8-1](#)
 TCP/IP for IMS Connect
 on the gateway, [11-10](#)
 the gateway
 for TCP/IP for IMS Connect, [11-1](#)
 using SNA, [9-1](#)
 the OLTP, [10-1](#)
 your network
 using SNA, [5-1](#)

configuring a LAN device
 on Linux, [6-4](#)

Configuring APPC/MVS, [10-3](#)

Configuring CICS Transaction Server for z/OS, [10-1](#)

configuring your network, [5-1](#)

CONNECT clause, [12-4](#)
 for database link security, [12-3](#)
 in TCP/IP security, [12-6](#)

CPI-C, [9-18](#)

CPI-C profiles
 creating
 on Solaris, [8-5](#)
 creating, on Solaris, [8-5](#)

creating
 mode profiles, on AIX, [7-4](#)
 public database link, [9-5](#), [11-4](#)

creating devices
 on Linux systems, [6-4](#)

creating IBM communications server definitions for the gateway
 on Linux, [6-3](#)

creating local LUs
 on Linux, [6-5](#)

creating partner LUs
 on Linux, [6-5](#)

creating peer connections
 on Linux, [6-4](#)

creating the configuration
 on Linux systems, [6-4](#)

creating the CPI-C side information profile
 on Linux, [6-6](#)

creating the node
 on Linux systems, [6-4](#)

D

data dictionary
 See PG DD, [1-3](#)

data exchange
 PGAXFER function, [1-8](#)

database link, [1-7](#), [A-5](#)
 creating, [9-5](#), [11-4](#)
 when configuring Oracle database, [9-5](#), [11-4](#)
 in configuring the network, [5-1](#)
 in verifying gateway installation
 on gateway using SNA, [9-13](#)
 on gateway using TCP/IP for IMS Connect, [11-13](#)
 public and private, [12-2](#)
 security, CONNECT clause, [12-3](#)

database link name
 modifying .sql files, [9-13](#), [11-13](#)

datastores
 gateway access to, [1-2](#)

DBMS_OUTPUT packages, [9-4](#), [11-3](#)

DBMS_PIPE, [9-5](#), [9-8](#), [11-5](#), [11-6](#), [11-9](#)

de-installing
 the gateway, [4-6](#)

defining local LUs
 on Linux, [6-5](#)

defining partner LUs
 on Linux, [6-5](#)

defining the adjacent node
 on Linux, [6-5](#)

defining the link station
 on Linux, [6-5](#)

dependent LU
 on AIX, [7-2](#)
 on Solaris, [8-2](#)

describe statement
 DBMS_OUTPUT, [11-4](#)

DESCRIBE statement
 DBMS_OUTPUT, [9-4](#), [11-3](#)
 UTL_RAW, [9-4](#), [11-3](#)

dfhcsdup.jcl file, [10-2](#)

DFHRPL DD statement, [10-2](#)

DG4APPC
 known restrictions, [2-2](#)
 see gateway, [2-2](#)

dg4pwd, [C-1](#)
 dg4pwd utility
 definition, [C-1](#)
 recommended security utility feature, on gateway using SNA, [C-1](#)
 directories
 for installing gateway and OIS files, [9-2](#), [11-2](#)
 DISPLAY datatypes, [2-2](#)

E

EBCDIC language, [1-3](#)
 gateway know restrictions pertaining to, [2-2](#)
 necessary to change to ASCII when using TCP/IP, [11-14](#)
 enhancements
 using PGAU to automatically upgrade PG DD entries, [9-9](#), [11-10](#)
 error
 during commit processing, [A-6](#)
 obsolete parameters, [13-4](#)
 parameter name misspelled, [9-11](#), [11-12](#)
 treating incoming APPC CONFIRM requests as errors, [A-2](#), [A-6](#)

F

FDS_CLASS parameter, [13-3](#)
 FDS_CLASS_VERSION
 parameter added, [13-5](#)
 FDS_INSTANCE parameter, [13-3](#)
 file
 dfhcsdup.jcl, [10-2](#)
 initPGA.ora, [9-10](#), [11-11](#)
 initsid.ora, [1-6](#), [9-10](#), [9-12](#), [11-11](#), [13-1](#), [A-1](#), [B-1](#), [C-1](#)
 gateway parameters for gateway using SNA, [9-10](#)
 gateway parameters for gateway using TCP/IP for IMS Connect, [11-11](#)
 new parameters on gateway using SNA, [13-3](#)
 new startup shell parameters, [13-4](#)
 parameters changed since V4, [13-3](#)
 listener.ora, [5-1](#), [5-2](#), [A-5](#)
 pgaims.sql, [11-14](#)
 pgasna.export, [7-3](#), [7-4](#)
 PGAU control files, [4-2](#)
 pgavsn.sql, [9-13](#)
 pgddapub.sql, [9-9](#), [11-9](#)
 prvtppgpb.plb, [9-6](#), [11-7](#)
 tnsnames.ora, [A-5](#)
 utlpg.sql, [9-7](#), [11-8](#)
 utlraw.sql, [9-7](#), [11-8](#)

FLIP transaction
 OLTP configuration
 and verification for APPC/MVS, [9-16](#)
 and verification for CICS Transaction Server for z/OS, [9-14](#)
 and verification for IMS/TM on gateway using SNA, [9-15](#)
 and verification for IMS/TM on gateway using TCP/IP, [10-4](#), [11-14](#)

function
 put_line, [9-4](#), [11-4](#)

functions
 see RPC (remote procedural call), [1-7](#)
 See UTL_PG, [C-1](#)
 see UTL_RAW, [C-1](#)

G

gateway
 access to IBM datastores, [1-2](#)
 communication overview, [1-7](#)
 communications with all platforms, [1-2](#)
 compatibility with other SNA-enabled products, [3-3](#)
 components, [1-5](#), [9-1](#)
 for SNA and TCP/IP for IMS Connect, [1-6](#)
 configuring, [9-1](#)
 for multiple users, on gateway using TCP/IP for IMS Connect, [11-8](#)
 for SNA, [5-1](#)
 for TCP/IP for IMS Connect, [11-1](#), [11-10](#)
 configuring for multiple users
 on gateway using SNA, [9-8](#)
 creating
 SNA server profiles for, [7-3](#)
 SNAP-IX definitions for, [8-3](#)
 SNAP-IX profiles for, [8-2](#)
 de-installing, [4-6](#)
 directory locations for configuration, [9-2](#), [11-2](#)
 factors affecting memory requirements, [3-1](#)
 features
 application transparency, [1-2](#)
 code generator, [1-3](#)
 fast interface, [1-2](#)
 flexible interface, [1-2](#)
 location transparency, [1-2](#)
 Oracle database integration, [1-3](#)
 performs automatic conversions, [1-3](#)
 site autonomy and security, [1-3](#)
 support for tools, [1-3](#)
 functions, using SNA, [1-9](#)
 initialization files, [C-1](#)
 initialization parameters
 also see PGA parameters, [A-1](#)

- gateway (*continued*)

 initialization parameters (*continued*)

 described, [9-10](#), [11-11](#)

 for gateway using SNA, [A-1](#)

 new and changed since Version 4

 gateway, [13-2](#)

 renamed since V4, [13-3](#)

 SET, [9-11](#), [11-11](#), [A-1](#), [B-1](#)

 installation

 first-time install, configuring the Oracle

 database, [9-4](#), [11-3](#)

 pre-installation procedures, [4-3](#)

 steps, [4-4](#)

 verification, [9-13](#), [11-13](#)

 with Oracle Universal Installer action

 items, [4-5](#)

 installing, [4-1](#)

 known restrictions, when using SNA, [2-2](#)

 migrating to new release, using SNA, [13-1](#)

 network attachment requirements, [3-2](#)

 overview, [1-1](#)

 parameter files, [9-10](#), [11-11](#)

 also see gateway initialization

 parameters, and PGA

 parameters, [9-10](#)

 initPGA.ora, [9-10](#), [11-11](#)

 pre-installation steps for TPC/IP, [4-4](#)

 remote procedure, definition, [C-1](#)

 remote transaction initiation

 using SNA, [1-8](#)

 using TCP/IP, [1-8](#)

 remote transaction termination

 using SNA, [1-8](#)

 using TCP/IP, [1-8](#)

 requirements

 hardware, [3-1](#)

 restoring to previous releases, [4-3](#)

 security options and overview, [12-1](#)

 also see, security, [12-1](#)

 server

 restoring previous version, [4-3](#)

 setting up multiple gateway instances, [A-5](#)

 SNA

 SNAP-IX configuration on Solaris, [8-1](#), [12-3](#)

 SNA security validation, [12-3](#)

 startup shell parameters

 FDS_CLASS_VERSION, [13-5](#)

 steps to install, via Oracle Universal Installer, [4-5](#)

 upgrading

 from previous release, [4-2](#)

 preparing to upgrade, [4-2](#)

 gateway initialization parameters

 for commit-confirm support, [9-12](#)

 gateway initialization parameters (*continued*)

 new, [13-2](#)

 gateway security requirements, [12-1](#)

 gateway using TCP/IP for IMS Connect

 gateway initialization parameters needed, [B-1](#)

 transaction types, [1-9](#)

 gpglocal, [9-6](#), [11-6](#)

 needed to compile PGAU-generated TIP

 specifications, [9-6](#), [11-6](#)

 gpglocal package, [11-6](#)

 gpglocal.pkb script, [9-6](#), [11-6](#)

 gpglocal.pkh script, [9-6](#), [11-6](#)

 grant

 access, [9-8](#), [11-9](#)

 authorization, [9-6](#), [11-6](#)

 execute, [9-5](#), [11-5](#)

 explicit, [9-8](#), [11-9](#)

 private, [9-8](#), [11-9](#)

 public, [9-8](#), [11-9](#)

H

-
- hardware requirements, [3-1](#)

 Heterogeneous Services (HS), [13-3](#)

 and Oracle Net considerations, on gateway

 using SNA, [13-2](#)

 catalogs

 installing, on gateway using SNA, [9-5](#)

 installing, on gateway using TCP/IP for

 IMS Connect, [11-4](#)

 parameters needed for gateway using

 TCP/IP, [13-3](#)

 HS parameters, [13-3](#)

 description, [13-1](#), [13-3](#)

 see also, (HS=), [13-3](#)

 HS_DB_INTERNAL_NAME parameter, [13-3](#)

 HS_FDS_FETCH_ROWS parameter, [13-3](#)

I

-
- IBM mainframe requirements, [3-3](#)

 implementation of the gateway

 for SNA and TCP/IP for IMS Connect, [1-6](#)

 implied APPC, [A-6](#)

 IMS Connect

 and security, [12-5](#)

 mainframe requirements, for gateway using

 TCP/IP, [3-4](#)

 IMS FLIP transaction, [10-4](#)

 IMS/TM

 installation verification

 on gateway using SNA, [9-15](#)

 on gateway using TCP/IP for IMS

 Connect, [11-14](#)

IMS/TM (continued)

mainframe requirements for gateway using TCP/IP, [3-4](#)

TP name length, [A-3](#)

verification of configuration

on gateway using SNA, [9-15](#)

on gateway using TCP/IP for IMS Connect, [11-14](#)

independent LU, [10-1](#), [10-3](#)

independent LU, on AIX, [7-2](#)

independent versus dependent LUs

on Linux, [6-2](#)

initialization files

See gateway initialization files, also see PGA parameters, [C-1](#)

initiating remote transactions, [1-8](#)

initPGA.ora file, [9-10](#), [11-11](#)

initsid.ora file, [1-6](#), [9-10](#), [9-12](#), [11-11](#), [13-1](#), [A-1](#), [B-1](#), [C-1](#)

gateway parameters on gateway using SNA, [9-10](#)

gateway parameters on gateway using TCP/IP for IMS Connect, [11-11](#)

HS parameter descriptions, [13-1](#), [13-3](#)

new parameters, on gateway using SNA, [13-3](#)

new startup shell parameters, [13-4](#)

parameters changed since V4, [13-3](#)

installation

steps, [4-4](#)

installation verification

CICS on gateway using SNA, [9-14](#)

gateway

with SNA, [9-13](#)

with TCP/IP for IMS Connect, [11-13](#)

IMS/TM

on gateway using SNA, [9-15](#)

on gateway using TCP/IP for IMS Connect, [11-14](#)

OLTP, [9-14](#)

installing

and configuring the gateway, [4-1](#)

preinstallation steps, [4-3](#)

sample applications

on gateway for SNA protocol, [9-17](#)

on gateway with TCP/IP for IMS Connect, [11-15](#)

IPC

key, [5-1](#)

protocol, [5-1](#)

K**key**

IPC, [5-1](#)

known restrictions

for DG4APPC, [2-2](#)

for PGAU, [2-2](#)

L**Link Station profiles**

on AIX, [7-4](#)

using smit to start, on AIX, [7-7](#)

listener.ora file, [5-1](#), [5-2](#), [A-5](#)

sample file for gateway using SNA, [A-6](#)

sample for gateway using TCP/IP, [B-4](#)

Local LUs

creating

on AIX, [7-4](#)

on Solaris, [8-4](#)

LOG_DESTINATION parameter, [13-3](#)

for gateway using SNA, [13-2](#)

for gateway using TCP/IP, [B-2](#)

logmode entry name, [9-15](#)

LU name

assigning to the gateway, on AIX, [7-4](#)

LU6.2

and specifying SNA conversation security, [12-3](#)

LUs

(Logical Units), [7-4](#)

alias identified by side information profile, on AIX, [7-6](#)

and gateway security, [12-3](#)

in SNA security validation, [12-3](#)

independent

in configuring APPC/MVS on the gateway, [10-3](#)

in configuring CICS Transaction Server for z/OS, [10-1](#)

vs. dependent, on AIX, [7-2](#)

vs. dependent, on Solaris, [8-2](#)

local

profile, on AIX, [7-4](#)

on AIX, [7-2](#)

Partner

location profile, on AIX, [7-5](#)

on AIX, [7-5](#)

on Solaris, [8-5](#)

profile, [7-5](#)

Partner LU name

assigning alias, on AIX, [7-5](#)

M**mainframe requirements**

migrating

an existing gateway to use TCP/IP, [13-5](#)

migrating (*continued*)
 backout considerations when migrating to new release, [13-2](#)
 existing gateway instance to new release, using SNA, [13-1](#)
 to 10.1.0.2.0
 special parameters, [13-2](#)
 mode definitions
 creating
 on Solaris, [8-5](#)
 mode profiles
 creating on AIX server, [7-4](#)
 creating on Solaris, [8-5](#)
 multi-conversational transaction type, for gateway using TCP/IP, [1-9](#)

N

network
 configuring
 with SNA, [5-1](#)
 reconfiguring, [5-1](#)
 networking products required, [3-3](#)
 node profile, on AIX, [7-3](#)
 non-persistent socket transaction type for TCP/IP for IMS Connect, [1-9](#)

O

obsolete parameters, in gateway using SNA, [13-4](#)
 OLTP, [5-2](#), [A-4](#)
 and dependent LUs
 on AIX, [7-2](#)
 on Solaris, [8-4](#)
 and SECURITY=PROGRAM option, on all platforms, [12-4](#)
 and SECURITY=PROGRAM option, on all platforms using TCP/IP, [12-5](#)
 and SNA SECURITY=SAME option, on all platforms, [12-4](#)
 configuration, [10-1](#)
 configuration verification
 APPC/MVS, [9-16](#)
 CICS Transaction Server for z/OS, [9-14](#)
 on gateway using SNA, [9-13](#)
 on gateway using SNA IMS/TM, [9-15](#)
 on gateway using TCP/IP for IMS Connect, [11-12](#)
 configuring
 for gateway using TCP/IP for IMS Connect, [11-10](#)
 currently supported types, [9-14](#), [10-1](#)
 definition, [1-3](#)
 for TCP/IP for IMS Connect, [1-5](#)

OLTP (*continued*)
 in gateway architecture featuring SNA, [1-5](#)
 in gateway using TCP/IP, [1-5](#)
 installation verification, [9-14](#), [11-13](#)
 mode name specification, on AIX, [7-4](#)
 post-installation steps
 on gateway using SNA, [9-17](#)
 on gateway using TCP/IP for IMS Connect, [11-15](#)
 remote, [1-1](#)
 requirements, [3-3](#)
 security and inbound APPC session requests on Solaris, [8-1](#)
 security on the gateway, [12-1](#)
 SNA security option
 on all platforms, [12-4](#)
 user ID mapping, [12-3](#)
 verifying configuration
 on gateway using SNA, [9-14](#)
 on gateway using TCP/IP for IMS Connect, [11-13](#)
 OLTP for SNA
 mainframe requirements, [3-3](#)
 OLTP for TCP/IP
 IBM mainframe requirements, [3-4](#)
 one-shot transaction types, for gateway using SNA, [1-8](#)
 online transaction processor
 See OLTP, [1-5](#)
 Oracle database, [1-8](#), [4-1](#), [4-3](#), [4-4](#), [5-1](#), [9-5](#), [9-6](#), [11-4](#), [11-6](#), [12-2](#), [A-4](#), [A-5](#)
 and gateway security, [12-3](#)
 and TCP/IP for IMS Connect
 pre-configuration steps, [11-1](#)
 component of the gateway, [1-5](#)
 configuring for commit-confirm, [9-11](#)
 definition, [1-3](#)
 enabling DBMS_OUTPUT PL/SQL package, [9-4](#), [11-3](#)
 logon authentication needed, [12-2](#)
 multiple servers on the gateway
 using TPC/IP, [1-5](#)
 multiple servers on the gateway using SNA, [1-5](#)
 precompiles PL/SQL package, [1-2](#)
 READ_ONLY mode, [A-4](#)
 role
 in gateway communication, [1-7](#)
 in logon security, [12-3](#)
 in starting the gateway, [1-6](#)
 shipped with PL/SQL packages, [9-8](#), [11-9](#)
 stores PL/SQL, [C-1](#)
 upgrading
 from previous releases, [9-6](#), [11-7](#)

- Oracle database (*continued*)

verifying
 - APP/C/MVS configuration, [9-16](#)
 - gateway installation with SNA, [9-13](#)
 - gateway installation with TCP/IP for IMS Connect, [11-13](#)
 - IMS/TM, on gateway using SNA, [9-15](#)
 - IMS/TM, on gateway using TCP/IP for IMS Connect, [11-14](#)
 - version requirements, [3-3](#)
- Oracle Database Gateway for APPC

also see gateway, [1-9](#)
 - de-installing, [4-6](#)
 - development environment, [1-3](#)
 - functions, [1-9](#)
- Oracle Database Listener, [5-2](#)
- Oracle global transaction ID, [9-17](#)
- Oracle Heterogeneous Services

See Heterogeneous Services, [13-3](#)
- Oracle Net, [1-3, A-2](#)
 - considerations, when migrating gateway featuring SNA, [13-2](#)
 - Heterogeneous services/tnsnames.ora
 - considerations, on gateway using SNA, [13-2](#)
 - on gateway using TCP/IP, [B-2](#)
 - security considerations, [12-2](#)
 - to start the gateway, [1-6](#)
- Oracle Net Listener, [5-1](#)
- Oracle Universal Installer

using, [4-5](#)
 - to install gateway, [4-5](#)
- Oracle10g Database

and networking products needed, [3-3](#)
- override

user ID and password, [12-4, 12-6](#)
-
- P
- package

DBMS_OUTPUT, [9-4, 11-3](#)
 - ggplocal, [11-6](#)
 - UTL_PG, [9-4, 9-8, 11-4, 11-9](#)
 - invalidated or deinstalled, [9-7, 11-7](#)
 - UTL_RAW, [9-4, 11-3](#)
 - invalidated or deinstalled, [9-7, 11-7](#)
- package specifications

avoid reinstalling, [9-7, 11-7](#)
 - reinstalling, [9-7, 11-7](#)
- parameter files

See gateway initialization files, [A-1, B-1](#)
 - See PGA parameters, [A-1, B-1](#)
 - See RRM parameters, [A-1, B-1](#)
- parameters

changed since Release 4, on gateway using SNA or TCP/IP, [13-2](#)
 - FDS_CLASS_VERSION, [13-5](#)
 - gateway initialization parameter, described, [9-10, 11-11](#)
 - needed for commit-confirm support, [9-12](#)
 - new
 - FDS_CLASS (startup shell), [13-3](#)
 - FDS_INSTANCE (startup shell), [13-3](#)
 - obsolete, in gateway using SNA, [13-4](#)
 - PGA
 - described for SNA, [9-11](#)
 - described for TCP/IP, [11-11](#)
 - renamed since version 4 (gateway initialization), [13-3](#)
 - see PGA parameters and gateway, [11-11](#)
 - USING, [5-1](#)
- Partner LU

locations profile, on AIX, [7-5](#)
 - on Solaris, [8-5](#)
 - profile, [7-5](#)
 - see also LUs
 - Partner, [7-5](#)
- password

change using ALTER USER command, [9-5, 11-5](#)
 - Oracle authentication, [12-2](#)
 - Oracle password to be used by gateway, [A-2](#)
 - using TCP/IP, [B-2](#)
 - overrides, [12-4, 12-6](#)
 - with operating system authentication, [12-2](#)
- PDS (partitioned dataset), [10-3](#)
- persistent socket transaction type

for TCP/IP for IMS Connect, [1-9](#)
- persistent transaction type, for gateway using SNA, [1-9](#)
- PG Data Dictionary

See PG DD, [1-3](#)
- PG DD

after upgrade, [4-2](#)
 - allowing multiple users, [9-8, 11-8](#)
 - creating public synonyms for multiple users, [9-8, 11-8](#)
 - definition, [1-3](#)
 - install script, [9-5, 11-5](#)
 - installing for gateway configuration, [9-5, 11-4](#)
 - no access from earlier PGAU versions, [4-3](#)
 - restoring previous version, [4-3](#)
 - tables, [9-8, 11-8](#)
 - upgrade when upgrading gateway, [9-7, 11-8](#)
 - using PGAU to upgrade existing entries, [9-9, 11-10](#)
- pg4tcpmap table, [13-5](#)
 - see PGA_TCP_IMSC table, [11-12](#)

pg4tcpmap tool, [11-12](#), [12-5](#), [12-6](#), [13-5](#), [B-2](#), [B-3](#)
 definition, [C-1](#)
 function, [1-2](#), [11-12](#)
 in remote transaction initiation, [1-8](#)
 on gateway using TCP/IP, [11-5](#), [B-2](#)
 output sample, [B-3](#)

PGA
 definition, [C-1](#)
 initialization files
 initPGAI.ora and initPGAU.ora, [A-5](#)

PGA parameters
 described, [9-11](#), [11-11](#)
 list of, for gateway using TCP/IP, [B-2](#)
[LOG_DESTINATION](#), [A-2](#)
 on gateway using
 SNA, [A-1](#)
 TCP/IP, [B-1](#)

[PGA_CAPABILITY](#), [11-11](#), [A-2](#), [B-2](#)
 choosing settings, [A-4](#)
 options for updating foreign databases,
 [A-4](#)
 protections against data problems, [A-4](#)

[PGA_CAPABILITY](#), for gateway using
 TCP/IP, [B-2](#)

[PGA_CONFIRM](#), [A-2](#)
 choosing settings, [A-6](#)

[PGA_LOG_DB](#), [A-2](#), [B-2](#)

[PGA_LOG_PASS](#), [A-2](#), [B-2](#)
 on gateway using TCP/IP, [B-2](#)

[PGA_LOG_USER](#), [A-3](#), [B-3](#)

[PGA_RECOVERY_PASS](#), [A-3](#)

[PGA_RECOVERY_TPNNAME](#), [A-3](#)

[PGA_RECOVERY_USER](#), [A-3](#)

[PGA_SECURITY_TYPE](#), [12-3](#)–[12-5](#), [A-3](#),
[B-2](#)

[PGA_SECURITY_TYPE= SAME](#), on all
 platforms, [12-4](#)

[PGA_TCP_PASS](#)
 for gateway using TCP/IP, [B-2](#)

[TRACE_LEVEL](#), [A-4](#)
 on gateway using TCP/IP, [B-3](#)

[PGA_CAPABILITY](#)
 See PGA parameters, [A-4](#)

[PGA_CONFIRM](#)
 See PGA parameters for gateway using SNA
 or TCP/IP, [A-6](#)

[PGA_SECURITY_TYPE](#)
 See PGA parameters, [12-3](#), [12-5](#)

[PGA_SECURITY_TYPE parameter](#)
 and TCP/IP security, [12-5](#)

[PGA_SECURITY_TYPE= SAME](#), on all platforms,
[12-4](#)

[PGA_TCP_DB](#)
 PGA parameter
 for gateway using TCP/IP, [B-2](#)

[PGA_TCP_DB parameter \(TCP/IP only\)](#), [13-3](#)

[PGA_TCP_DB PGA parameter \(TCP/IP only\)](#),
[B-2](#)

[PGA_TCP_IMSC table](#), [13-5](#)
 for mapping SNA parameters to TCP/IP,
[11-12](#)
 loading, on gateway using TCP/IP, [11-12](#)

[PGA_TCP_PASS parameter \(TCP/IP only\)](#), [13-3](#)

[PGA_TCP_PASS PGA parameter \(TCP/IP only\)](#),
[B-2](#)

[PGA_TCP_USER parameter \(TCP/IP only\)](#),
[13-3](#), [B-3](#)

[PGAADMIN](#), [9-5](#), [9-6](#), [9-13](#), [11-6](#), [11-13](#)
 creating the gateway administrator user ID,
[9-5](#), [11-4](#)
 granting access to additional users, [9-8](#), [11-8](#)
 granting execution privileges on
[DBMS_PIPE](#), [9-5](#), [11-5](#)

initial password during creation, [9-5](#), [11-5](#)

[pgacr8au.sql script](#), [9-5](#), [11-5](#)

[PGAI](#)
 setting up, [A-5](#)

[pgaims.sql file](#), [9-15](#)
 on gateway using TCP/IP, [11-14](#)

[PGAINIT](#), [1-8](#)

[PGAINIT function](#), [1-8](#), [C-1](#)

[PGAINIT TIP](#), [11-12](#)

[pgasna.export file](#), [7-3](#), [7-4](#)

[PGATERM function](#), [C-1](#)

[PGAU](#), [5-2](#)
 -generated TIP specifications, [1-7](#)
 -generated TIP specifications use UTL_PG,
[9-4](#), [11-4](#)
 -generated TIP specifications use
[UTL_RAW](#), [9-4](#), [11-3](#)

accesses definitions in PG DD, [1-4](#)

control files, [4-2](#)

definition
 used to generate TIP specifications, [1-3](#)

known restrictions in this release, [2-2](#)

purpose of PGDL, [C-1](#)

restoring previous versions, [4-3](#)

setting up, [A-5](#)

upgrading existing PG DD entries, [9-9](#), [11-10](#)

[PGAU commands](#)

[DEFINE DATA](#)
 COBOL COPY REPLACE restrictions,
[2-2](#)

[GENERATE](#), [9-9](#), [11-10](#)
 produces TIP in output files, [9-10](#), [11-10](#)
 to upgrade existing TIPs, [9-9](#), [11-10](#)

[pgavsn.sql file](#), [9-13](#)

[PGAXFER function](#), [1-8](#), [C-1](#)

PGDD
 compatibility issues between new and older gateways, [9-8](#)

pgddapub.sql
 file, [9-9, 11-9](#)

pgddc8.sql script, [9-5, 11-5](#)

pgddc8s.sql script, [9-8](#)

PGDDDEF role, [9-8, 11-8](#)

PGDDGEN role, [9-8, 11-8](#)
 adding a privilege for upgrading PG DD entries, [9-9, 11-10](#)

pgdupgr.sql script, [9-7, 11-8](#)

PGDL (Procedural Gateway Definition Language)
 definition, [C-1](#)

PL/SQL, [1-4](#)
 code generator, [1-3](#)
 datatypes, [1-7](#)
 converted to RAW, [1-8](#)
 function in the gateway, [1-1, 1-7](#)
 running pgtiptr.sql script to create routines, [9-6, 11-6](#)

UTL_PG package function, [C-1](#)

UTL_RAW function, [C-1](#)

UTL_RAW package installation
 on gateway using SNA, [9-4](#)
 on gateway using TCP/IP for IMS Connect, [11-3](#)

PL/SQL package
 definition, [1-4, C-1](#)
 developer access to, [9-8, 11-9](#)
 enabled, [9-4, 11-3](#)
 functions, [1-7](#)
 See TIP, [1-7](#)

PL/SQL stored procedure, [9-12](#)
 used for logging transactions, [9-11](#)

PL/SQL stored procedure specification
 also called "TIP", [1-2](#)
 See PL/SQL package, [C-1](#)

post-installation steps for OLTP
 on gateway using SNA, [9-17](#)
 on gateway using TCP/IP, [11-15](#)

pre-installation steps
 for SNA, [4-3](#)

privileges
 needed to create TIPs, [9-9](#)

Procedural Gateway Administration
 See PGA, [C-1](#)

Procedural Gateway Administration Utility
 see PGAU, [1-7](#)

processing inbound connections
 Linux, [6-2](#)

processor OLTP
 configuring, for commit-confirm, [9-12](#)

profiles
 see SNA server profiles, [7-1](#)

protocol
 IPC, [5-1](#)
 TCP, [5-2](#)

prvtpgb.plb
 file, [9-6, 11-7](#)
 script, [9-6, 11-7](#)

prvtrawb.plb script, [9-4, 9-6, 11-3, 11-7](#)

public synonyms for multiple PG DD users, [9-8, 11-8](#)

put_line function, [9-4, 11-4](#)

R

RACF, [12-6, B-2](#)

READ_ONLY
 PGA_COMPATIBILITY setting, [A-4](#)

recompiling TIPs
 See TIP, [9-7, 11-7](#)

reinstallation of package specifications, [9-7, 11-7](#)

release-specific information
 restrictions, [2-2](#)

remote host transactions (RHT)
 types, [1-8](#)

remote procedural call
 See RPC, [1-1](#)

remote procedure
 definition, [C-1](#)

remote transaction initiation
 on gateway using SNA, [1-8](#)
 on gateway using TC/IP, [1-8](#)

remote transaction program
 See RTP, [1-1](#)

remote transaction termination
 on gateway using SNA, [1-8](#)
 on gateway using TCP/IP for IMS Connect, [1-8](#)

requirements
 hardware, [3-1](#)
 network attachments, [3-2](#)
 software, [3-2](#)
 system, [3-1](#)

restoring a previous release of the gateway, [4-3](#)

restrictions
 on gateway using SNA, [2-2](#)

resume configuration of the gateway
 on Linux, [6-7](#)

role
 PGDDDEF, [9-8, 11-8](#)
 PGDDGEN, [9-8, 11-8](#)

ROLLBACK, [3-3, A-4](#)

RPC
 definition, [1-3](#)
 function
 PGAINIT, [1-8, C-1](#)
 PGATERM, [C-1](#)

RPC (*continued*)
 function (*continued*)
 PGAXFER, [1-8](#), [C-1](#)
 within the gateway, [1-1](#), [1-7](#)
 processing, [1-1](#)

RTP
 definition, [1-3](#)
 executing, [1-3](#)
 function in the gateway, [1-1](#)
 PGA_CAPABILITY settings for read-only
 RTPs, [A-4](#)

S

sample applications
 included, [9-17](#), [11-15](#)
 installing, [9-17](#), [11-15](#)

script
 gpglocal.pkb, [9-6](#), [11-6](#)
 gpglocal.pkh, [9-6](#), [11-6](#)
 pgacr8au.sql, [9-5](#), [11-5](#)
 pgddcr8.sql, [9-5](#), [11-5](#)
 pgddcr8s.sql, [9-8](#)
 pgddupgr.sql, [9-7](#), [11-8](#)
 prvtpgb.plb, [9-6](#), [11-7](#)
 prvtrawb.plb, [9-4](#), [9-6](#), [11-3](#), [11-7](#)
 utlpg.sql, [9-5](#)
 utlraw.sql, [9-4](#), [11-3](#)

security
 and database links, [12-2](#)
 and SNA validation, [12-3](#)
 authenticating application logons, [12-2](#)
 authentication mechanisms
 in SNA security, [12-4](#)
 for TCP/IP for IMS Connect, [12-5](#)
 link accessibility for public and private
 databases, [12-2](#)
 links and CONNECT clauses, [12-3](#)
 overview of gateway security requirements,
 [12-1](#)
 processing inbound connections
 on AIX, [7-1](#)
 on Solaris, [8-1](#)
 specifying SNA conversation security, on all
 platforms, [12-3](#)

SECURITY=NONE, [12-4](#)

SECURITY=NONE SNA security option, on all
 platforms, [12-4](#)

SECURITY=NONE TCP/IP security option, [12-5](#)

SECURITY=PROGRAM SNA security option, on
 all platforms, [12-5](#)

SECURITY=SAME SNA security option, on all
 platforms, [12-4](#)

SET
 gateway initialization parameter, [9-11](#), [11-11](#)

SET command, [9-11](#), [11-11](#)

shells
 see Bourne, Korn, and C shells, [4-1](#)

Side Information Profiles, [7-4](#)
 function, on AIX, [7-6](#)
 function, on Solaris, [8-2](#)

Side Profile Name, [9-15](#)

SINGLE_SITE
 and PGA_CAPABILITY parameter, [A-4](#)

smit, [7-3](#)

System Management Interface Tool for AIX,
 [7-3](#)

SNA
 and gateway components, [1-5](#)
 implementation of the gateway, [1-6](#)
 location of gateway initialization parameters,
 [A-1](#)
 migrating existing gateway to the new
 release, [13-1](#)
 new gateway initialization parameters, [13-2](#)
 parameters, [1-8](#)
 PGA parameters, [A-1](#)
 preinstallation procedures, [4-3](#)
 remote transaction initiation, [1-8](#)
 remote transaction termination on the
 gateway, [1-8](#)
 required for Solaris, [3-3](#)
 transaction types, [1-8](#)

SNA APPC
 also see APPC, [1-1](#)
 function in the gateway, [1-1](#)

SNA communication package, [9-15](#), [9-16](#), [12-4](#)
 and SNA security option, on all platforms,
 [12-4](#)

and SNA security validation, all platforms,
 [12-3](#)

configuration for the gateway, [5-1](#)

SNA definitions
 creating SNAP-IX definitions, [8-3](#)

SNA node
 profile, on AIX, [7-3](#)

SNA profiles, [12-3](#)
 See SNA server profiles, [7-3](#)

SNA protocol
 gateway initialization parameters, [A-1](#)

SNA security options
 SECURITY=NONE, on all platforms, [12-4](#)
 SECURITY=PROGRAM, on all platforms,
 [12-4](#)
 SECURITY=SAME, on all platforms, [12-4](#)

SNA server, [7-1](#)
 dependent LUs, [7-2](#)
 independent LUs, on AIX, [7-2](#)
 profiles, [7-3](#), [7-6](#)
 activating and verifying, [7-7](#)

- SNA server (*continued*)
 profiles (*continued*)
 creating, on AIX, 7-3
 definition, 7-1
 for non APPN-capable nodes, on AIX, 7-5
 Link Station, 7-4
 local LU profile, 7-4
 mode profile, 7-4
 Partner LU, 7-5
 Partner LU location profile, 7-5
 Side Information Profiles, 7-6
 SNA node profile, 7-3
 types, on AIX, 7-3
 verify and activate profiles, on AIX, 7-7
- SNAP-IX
 communication support for Solaris, 8-1, 12-3
 configuring, 8-1, 8-3
 definitions stored, 8-3
 function in gateway communication, 8-1, 12-3
 profiles creating, 8-2
- socket file descriptor
 returned by TCP/IP network to PGAINIT, 1-8
- software requirements, 3-2
- Solaris
 APPC support provided by SNAP-IX, 8-1
 communication protocol needed, 3-3
 configuring SNA, 5-1
 configuring SNAP-IX, 8-1, 8-3
 LU types, 8-2
 SNA conversation security, 12-3
 SNA security option SECURITY =SAME, 12-4
 SNA security option SECURITY=NONE, 12-4
 SNA security option
 SECURITY=PROGRAM, 12-4
 SNA security validation, 12-3
 storing SNAP-IX SNA definitions, 8-3
- specifying LUs
 on Solaris, 8-5
- SQL*Plus
 sample, used when gateway and Oracle database share machine, 9-5, 11-4
 to configure Oracle database, 9-4, 11-3, 11-5
 use in configuring Oracle database for Commit-Confirm, 9-12
 using to connect to Oracle database, 9-5, 11-4
- statement
 describe, 11-4
 DESCRIBE, 9-4
 DFHRPL DD, 10-2
- system identifier
 choosing, 4-3
- system requirements, 3-1
-
- T
-
- TCP protocol, 5-2
- TCP/IP
 specifying conversation security, 12-5
 TCP/IP for IMS Connect
 and Remote Transaction Initiation, 1-8
 configuring
 for the gateway, 11-10
 gateway to permit multiple users, 11-8
 function in the gateway, 1-1
 gateway initialization parameters, list, B-1
 gateway preinstallation procedures, 4-4
 gateway support for, description, 1-2
 Heterogeneous Services parameters needed, 13-3
 HS parameter descriptions, 13-1, 13-3
 implementation of the gateway, 1-6
 IMS Connect release required on IBM mainframe, 3-4
 installing sample applications, 11-15
 loading the PGA_TCP_IMSC table, 11-12
 mapping SNA parameters to, 11-12
 migrating existing gateway using SNA to TCP/IP, 13-5
 necessary to recompile TIPs when changing communication protocol, 11-14
 new gateway initialization parameters, 13-2
 non-persistent socket transaction type, 1-9
 OLTP in gateway architecture, 1-5
 parameter files
 also see gateway initialization parameters, and PGA parameters, 11-11
 performing post-installation procedures, 11-15
 persistent socket transaction type, 1-9
 pg4tcpmap tool output sample, B-3
 PGA parameters, 11-11, B-1
 PGA_TCP_DB parameter, B-2
 PGA_TCP_USER parameter, B-3
 remote transaction initiation, 1-8
 remote transaction termination, 1-8
 security, 12-5
 security options
 SECURITY=NONE, 12-5
 SECURITY=NONE, on all platforms, 12-5
 SECURITY=PROGRAM, on all platforms, 12-5
- TIP recompile needed on upgrade, 4-3, 11-14
- TRACE_LEVEL parameter, B-3

TCP/IP for IMS Connect (*continued*)
 transaction types, [1-9](#)
 using pg4tcpmap tool, [B-2](#)
 verifying
 gateway installation, [11-13](#)
 OLTP configuration, [11-13, 11-14](#)

TCP/IP protocol adapter
 for SNA, [5-1](#)

terms, gateway terms defined, [1-3](#)

testing the connection
 on Linux, [6-6](#)

TIP
 Also called PL/SQL package, [1-7](#)
 body output files, [9-10, 11-10](#)
 conversions, [1-3](#)
 converting PL/SQL datatypes to RAW, [1-8](#)
 definition, [1-3, 1-4](#)
 developer access to PL/SQL packages, [9-8, 11-9](#)
 developer authorization on gpglocal, [9-6, 11-6](#)
 functions, [1-7](#)
 in Oracle database, [1-8](#)
 invalidated if package specifications
 reinstalled, [9-7, 11-7](#)
 override, on all platforms, [12-4](#)
 override, on all platforms, in security for
 TCP/IP, [12-6](#)
 recompile
 after reinstalling package specifications, [9-7, 11-7](#)
 on upgrade from Release 4.0.1, [4-2](#)
 recompile upon upgrade from SNA to
 TCP/IP, [4-3, 11-14](#)
 recompiling when changing from SNA to
 TCP/IP, [11-14](#)
 regenerate to upgrade function and
 maintenance, [9-9, 11-10](#)
 remote transaction initiation (PGAINIT), [1-8](#)
 specification output files, [9-10, 11-10](#)
 specifications, [9-6, 9-8, 9-9, 11-6, 11-8, 11-10](#)
 generated by PGAU, [1-7](#)
 use UTL_PG, [9-4, 11-4](#)
 use UTL_RAW, [9-4, 11-3](#)
 trace access PL/SQL routines, [9-5, 11-6](#)
 upgrade considerations from previous
 versions, [4-2](#)

tnsnames.ora file, [A-5](#)
 sample file for gateway using SNA, [A-7](#)
 sample for gateway using TCP/IP, [B-5](#)

TP name, [A-3](#)

trace access, [9-5, 11-6](#)

TRACE_LEVEL parameter, [13-3, A-4](#)
 on gateway using SNA, [A-4](#)

TRACE_LEVEL parameter (*continued*)
 on gateway using TCP/IP for IMS Connect, [B-3](#)

Transaction Interface Package
 See TIP, [1-3](#)

transaction socket
 transaction type for TCP/IP, [1-9](#)

transaction types
 for TCP/IP for IMS/Connect, [1-9](#)
 one-shot, persistent and multi-
 conversational, for SNA, [1-8](#)

transferring
 initsid.ora gateway initialization file
 parameters, [13-1](#)

transparency
 (application), [1-2](#)
 (location), on gateway using SNA, [1-2](#)

U

upgrading
 considerations, [4-2](#)
 existing TIP specifications, [9-9, 11-10](#)

user ID
 as security authentication, [12-2](#)
 uppercase translation, OLTP and SNA
 communications packages, on all
 platforms, [12-4](#)

user ID mapping
 OLTP, [12-3](#)

USING parameter, [5-1](#)

using SNA security validation
 Linux, [6-1](#)

utility
 dg4pwd, [C-1](#)
 UTL_PG, [9-8, 11-9](#)
 installing, [9-4, 11-4](#)
 package, [9-8, 11-9](#)
 definition, [C-1](#)
 invalidated or deinstalled, [9-7, 11-7](#)

UTL_RAW, [9-4, 9-8, 11-3, 11-9](#)

interface
 PL/SQL package, [11-9](#)

package
 invalidated or deinstalled, [9-7, 11-7](#)

PL/SQL package
 definition, [C-1](#)

utlpg.sql
 file, [9-7, 11-8](#)
 script, [9-5](#)

utlraw.sql
 file, [9-7, 11-8](#)
 script, [9-4, 11-3](#)

V

VTAM, [10-3](#)
 configuring for connection to the gateway,
 [10-1](#)
VTAM logmode table, [10-2](#), [10-3](#)

X

xsnaadmin
 invoking on Solaris, [8-3](#)

Z

z/OS, [1-2](#), [1-4](#), [3-3](#), [10-1](#), [10-2](#)