
Author: Cynthia Chin–Lee, Sai–Krishna–Kishore Gummaraj

Contributors: Thomas Arnold, Scott Burdman, Sudhakar Chitageri, Charles Congdon, DJ Cover, Kurt Engeleiter, Ronny Guthzer, Dave Hayter, Gary Huffman, Pierre Krabbendam, Thomas Leah–Martin, Tony Lekas, Saar Maoz, Alice Mulder, Sally Norton, Jitendra Pandey, Tony Purmal, Dan Ryan, David Schwab, Don Shellito, Peter Shih, Robert Smith.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government Agency or anyone licensing or using the Programs on behalf of the US Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication, and disclosure of the Programs should be subject to the restrictions in FAR 52.227–19, Commercial Computer Software – Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be licensee’s responsibility to take all appropriate fail-safe, back up, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle disclaims liability for any damages caused by such use of the Program.

Oracle is a registered trademark, and CASE Designer, CASE Dictionary, Pro*Ada, Pro*COBOL, Pro*FORTRAN, Pro*Pascal, Pro*PL/I, SQL*Loader, SQL*Net, SQL*Plus, and Developer/2000 are trademarks or registered trademarks of Oracle Corporation.

Alpha, Compaq, DEC, DECnet, Digital, OpenVMS, OpenVMS Clusters, VAX, VAX Clusters, and VMS are trademarks of Compaq Computer Corporation.

All other company or product names mentioned are used for identification purposes only and may be trademarks of their respective owners.
The SQL*Net for OpenVMS Configuration and User’s Guide is one of a set of Oracle documents that describes how to install, configure, maintain, and use SQL*Net on OpenVMS.

The purpose of this guide is to explain how to configure and use SQL*Net 2.3.4 for OpenVMS, including the following components:

- TNS Listener
- Oracle Names
- Oracle Intelligent Agent and SNMP Support
- Oracle Advanced Networking Option and SNMP Support

The configuration and use of the protocols: DECnet, TCP/IP, and OpenVMS Mailboxes are detailed in this guide.

The following topics are covered in this Preface:

- Audience
- How this guide is organized
- Conventions used in this guide
- Related publications
- Customer support information
- Your comments are welcome
Audience

This guide is intended for users who configure and use SQL*Net 2.3.4. This guide assumes that the user has a fundamental knowledge of the OpenVMS operating system; it does not document any features of OpenVMS except when they affect or are affected by SQL*Net. Only those SQL*Net features that are specific to the OpenVMS environment are documented in this guide. For general, non–platform–specific information about SQL*Net 2.3.4, refer to the documents listed in the section “Related Publications” later in this chapter.

How This Guide is Organized

This guide is divided into the following chapters and appendices:

Chapter 1: Introduction
This chapter provides general conceptual information about SQL*Net Version 2.3.4 in the OpenVMS environment. It presents a configuration overview, describes the possible SQL*Net installations, and explains the role of the Transparent Network Substrate.

Chapter 2: SQL*Net Configuration with Network Manager
This chapter explains how to use the Network Manager to configure SQL*Net in the OpenVMS environment. It tells how to start Network Manager, tells what to enter in the port–specific fields, and describes how to save and distribute configuration information.

Chapter 3: The Protocol Adapters
This chapter provides information about the Mailbox, DECnet, TCP/IP, and bequeath protocol adapters on OpenVMS.

Chapter 4: The TNS Listener
This chapter provides information about the TNS listener on OpenVMS.

Chapter 5: Oracle Names
This chapter provides OpenVMS–specific information about Oracle Names. It describes the requirements and tells how to start Oracle Names from the DCL prompt.
Chapter 6: Oracle Intelligent Agent and SNMP Support
This chapter provides OpenVMS–specific information about installing and running the Oracle Intelligent Agent and information about Simple Network Management Protocol (SNMP) Support.

Chapter 7: Advanced Networking Option
This chapter provides OpenVMS–specific information about installing the Advanced Networking Option (ANO).

Appendix A: Major Changes from Previous Versions of SQL*Net
This appendix lists the major changes from previous versions of SQL*Net to SQL*Net 2.3.4.

Appendix B: Sample ORASRV_NETV2 Command Procedure
This appendix shows the contents of the ORASRV_NETV2_COM.SAMPLE file.

Appendix C: Sample ORASRV_BEQ_<sid> Command Procedure
This appendix shows the contents of the ORASRV_BEQ_COM.SAMPLE file.

Appendix D: SQL*Net Configuration for OpenVMS Clusters
This appendix describes how to configure SQL*Net for easy operation in OpenVMS Clusters/shared disk configurations.

Conventions Used in this Guide
This section addresses the conventions used in this guide.

Syntax
This guide uses the following conventions:

Monospaced font is used to represent information displayed or entered on terminals. For example, menu screens that are displayed during the Oracle7 installation procedure are represented in this guide with monospaced font.

UPPERCASE in monospaced font represents a command name. Enter the text exactly as shown.

UPPERCASE words within the text refer to command names or file names.
Basic Terms Used in this Guide

This guide uses the following terms:

**Oracle**
Generally refers to the company, Oracle Corporation, that develops and distributes the Oracle7 Server, SQL*Net, and related products.

**Oracle7 Server**
Refers to a single Oracle product, the Oracle7 Server, that stores, manages, and manipulates data. It is sometimes referred to as the kernel, the RDBMS, or simply Oracle7. The product name includes such subproducts as Server Manager and SQL*Loader.

**Oracle7 System**
Refers to all the Oracle products installed at a site, including the Oracle7 Server and any optional tools such as Oracle Forms or SQL*Net, and to the structures created in shared OpenVMS memory when Oracle7 is running.

**Database**
In this guide, the reference is only to the physical data files that contain data stored by the Oracle7 Server.

SQL*Net Terms Used in this Guide

Readers of this guide should be familiar with the following terms as they are used in Oracle networking documents.

**TNS–based product**
A product based on the Transparent Network Substrate (TNS) refers to software that directly calls or uses TNS. An example would be SQL*Net 2.3.4.

**TNS community**
A TNS community corresponds to what is typically called a network. It consists of several clients or nodes that fulfill the following conditions:

- Connected and can communicate using one common network transport protocol (regardless of media or distance)
- Run TNS–based products, such as SQL*Net 2.3.4
Communities are given names, but because they are based on a single protocol they may also be referred to using phrases such as the TCP/IP community or the DECnet community.

<table>
<thead>
<tr>
<th><strong>TNS network/application network</strong></th>
<th>A TNS network (also called an application network) is a network that consists of one or more TNS communities running TNS–based products. A simple example is one TNS community in which two nodes are running SQL*Net Version 2.3.4.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TNS client</strong></td>
<td>A TNS client is an application that is a member of a TNS community and that is running TNS–based software such as SQL*Net Version 2.3.4. A TNS client may be either a requestor or a provider of a service.</td>
</tr>
<tr>
<td><strong>Initiator</strong></td>
<td>An initiator is a TNS client application that requests a connection to another client in a TNS network. Every TNS connection is originated by an initiator. An example is an Oracle Forms application that accesses an Oracle7 database using SQL*Net Version 2.3.4.</td>
</tr>
<tr>
<td><strong>TNS connection</strong></td>
<td>A TNS connection is an application-level connection between two TNS clients in the same TNS Community.</td>
</tr>
<tr>
<td><strong>Connect descriptor</strong></td>
<td>An initiator requests a TNS connection using a connect descriptor, which identifies the destination application and usually contains initial application connection information (such as a database system ID). Usually the connect descriptor is identified by a service name, a short alias for the connect descriptor. The connect descriptors and their corresponding service names are stored in the Oracle Names server for the network (if one exists) or in the TNSNAMES.ORA file.</td>
</tr>
<tr>
<td><strong>Path</strong></td>
<td>A path is the series of nodes traversed by a connection, beginning with the start point (the initiator) to the end point (the destination).</td>
</tr>
</tbody>
</table>
Related Publications

OpenVMS Publications

- SQL*Net for OpenVMS Configuration and User’s Guide
- Oracle7 for Alpha OpenVMS Installation Guide
- Oracle7 for VAX OpenVMS Installation Guide
- Oracle7 for OpenVMS Server and Tools Administrator’s Guide
- Release notes associated with Oracle product on OpenVMS
- Platform–specific README files in the product directories (for example, ORA_RDBMS:READMEVMS.DOC)

Product–Specific Publications

- Understanding SQL*Net
- Oracle Network Manager Administrator’s Guide
- Oracle Names Administrator’s Guide
- Oracle Network Products Troubleshooting Guide
- Oracle SNMP Support Reference Guide
- Oracle7 Server SQL Reference
- Oracle7 Server Utilities
- Oracle7 Server Messages
- PL/SQL User’s Guide and Reference
- Oracle7 Server Distributed Systems I: Distributed Data
- Oracle7 Server Distributed Systems II: Replicated Data
- Oracle7 Parallel Server
- Trusted Oracle7 Server Administrator’s Guide
- Product–specific README and DOC files in the product directories (for example, ORA_RDBMS:READMEDOC.DOC)
Customer Support Information

(Please copy this page and distribute within your organization as necessary.)

For Oracle Worldwide Customer Support Services (WCSS), contact your local number. (The hours are detailed in your support contract.)

Please prepare the following information before you call, using this page as a checklist:

- Your Customer Support Identification number (CSI) if applicable, or full contact details, including any special project information
- The complete version numbers of the Oracle7 Server and associated products (for example, Oracle7 Server version 7.3.4.3 or Oracle Forms version 4.5.6.3.2) Note: The version numbers for products can be obtained from the file ORA_UTIL:PRODUCTS.TXT
- The hardware type on which the problem occurs (for example, Compaq Alpha)
- The operating system name and version number (for example, OpenVMS 7.1)
- Details of error codes and associated descriptions. Please write these down as they occur, since they are critical in helping WCSS to quickly resolve your problem.
- A full description of the issue, including:
  - What – What happened? For example, the command used and result obtained.
  - When – When did it happen? For example, time of day, or after a certain command, or after an O/S upgrade.
  - Where – Where did it happen? For example, on a particular system or within a certain procedure or table.
  - Extent – What is the extent of the problem? For example, production system unavailable, or moderate impact but increasing with time, or minimal impact and stable.

Note: Keep in mind what did not happen, as well as what did happen. This type of information can help WCSS to more quickly resolve your problem.

- Keep copies of any trace files, core dumps, and redo log files recorded at or near the time of the incident, since WCSS will need these to further investigate your problem.

For installation-related problems, please have the following information available:

- Error returned by the installation procedure and/or OpenVMS
Your Comments Are Welcome

We value and appreciate your comments as an Oracle user and reader of the manuals. As we write, revise, and evaluate our documentation, your opinions are the most important input we receive. At the back of our printed manuals is a Reader’s Comment Form, which we encourage you to use to tell us what you like and dislike about this manual or other Oracle manuals. If the form is not available, please use the following address, phone number, or FAX number.

Compaq Products Division
Publications Manager
Oracle Corporation
500 Oracle Parkway
Box 659104
Redwood Shores, CA 94065
Phone: (650) 506-7000
FAX: (650) 506-7361
# Contents

## Chapter 1 Introduction
- Configuration Overview .............................................. 1–2
  - Centralized ....................................................... 1–3
  - Client/Server .................................................... 1–3
  - Distributed Database ............................................. 1–3
- SQL*Net Installations ............................................... 1–4
- SQL*Net and the Transparent Network Substrate (TNS) ........... 1–4
  - Using SQL*Net ................................................... 1–4
  - SQL*Net Architecture ........................................... 1–6

## Chapter 2 SQL*Net Configuration with Network Manager
- Pre–Configuration Tasks ............................................. 2–2
  - Before You Set Up OpenVMS Clients ......................... 2–2
  - Setting Up OpenVMS Clients .................................. 2–2
- Starting Network Manager ......................................... 2–3
  - Using Network Manager ......................................... 2–3
    - Protocol–Specific Information ........................... 2–3
    - Location–Specific Parameters ........................... 2–4
    - User–Defined Parameters ................................. 2–4
- Creating Tables for Oracle Names ................................. 2–6
  - Creating and Distributing the Configuration Files .......... 2–7
Chapter 6  Oracle Intelligent Agent and SNMP Support  .......... 6 – 1
  Introduction ......................................................... 6 – 2
  Installing the Oracle Intelligent Agent ...................... 6 – 2
  Oracle Intelligent Agent Setup and Discovery Option ......... 6 – 3
    Creating the Startup Scripts ............................... 6 – 3
    Agent Parameter Files and Discovery Option ............... 6 – 3
    Setting the Preferred Credentials ......................... 6 – 6
  Oracle Intelligent Agent Startup, Shutdown, and Status Query 6 – 6
    Startup of the Agent .................................. 6 – 6
    Shutdown of the Agent .................................. 6 – 7
    Status Query of the Agent ................................ 6 – 7
  Oracle Intelligent Agent Maintenance .......................... 6 – 7
  Oracle Intelligent Agent Product Capabilities ................. 6 – 7
  SNMP Support .................................................. 6 – 8

Chapter 7  Advanced Networking Option ......................... 7 – 1
  Documentation Set .............................................. 7 – 2
  Requirements .................................................. 7 – 2
    What’s in this Release? .................................. 7 – 2
    Media ..................................................... 7 – 2
    Installation Requirements ................................ 7 – 2
  Installation ................................................... 7 – 3
    Installation Warning .................................. 7 – 4
    Installation Tasks .......................................... 7 – 4
    Task 1: Responding to Installer Prompts .................... 7 – 5
    Task 2: Relinking Other Executables ....................... 7 – 5
    Task 3: Using with Oracle Names ......................... 7 – 5
    Task 4: Manual Steps for the Authentication Adapters .... 7 – 6
  De–Installation ................................................ 7 – 7
    De–Installation Warning .................................. 7 – 8
    De–Installation Tasks ...................................... 7 – 8
  Usage Notes for the Authentication Adapters ................. 7 – 10
    General Information ....................................... 7 – 10
    SecurID .................................................... 7 – 10
    Kerberos5 .................................................... 7 – 10
    Identix (Biometric) ......................................... 7 – 11
Introduction

This chapter provides general conceptual information about SQL*Net 2.3.4 in the OpenVMS environment. It covers the following topics:

- Configuration Overview
- SQL*Net Installations
- SQL*Net and the Transparent Network Substrate (TNS)
SQL*Net is a communications software product that allows you to create a data management environment to share information stored in Oracle databases. SQL*Net uses the communications protocols supported by various operating systems to provide a distributed processing and distributed database environment for Oracle. SQL*Net also refers to a set of products or adapters that support industry-standard protocols such as DECnet and TCP/IP.

An Oracle database management system can be configured in one of the following ways:

- Centralized
- Client/server
- Distributed database

In Figure 1–1 Ethernet is only an example. SQL*Net works with other network types.

Figure 1–1 Configuration Options
Centralized
In a centralized configuration, the Oracle7 Server and Oracle tool are located on the same machine. This machine is not necessarily on a network and users access the application through terminals. If you use a centralized configuration, you may use a simple SQL*Net adapter called the *bequeath adapter*, which requires no SQL*Net configuration. However, if you wish to use pre-spawned servers or multithreaded servers, you must configure SQL*Net even in centralized configurations.

Client/Server
In a client/server configuration, the Oracle7 Server resides on a multi-tasking server system, and the client side of the applications resides on another computer, such as a workstation or personal computer. Both the client and server are connected by a physical network and communicate via a network protocol such as DECnet. In a client/server environment, the Oracle application built with an application development tool makes database requests to the server over the network.

Distributed Database
In a distributed database configuration, users query separate databases as a single database. The major advantage of a distributed database is that users and applications are not required to know where data resides. You can query database tables by name, regardless of how the network protocols work together to access the appropriate remote database containing the table. Therefore, SQL*Net users can communicate and share database information stored in different locations, on different computers, with different operating systems. Distributed databases allow local administration of data and can reduce network traffic if the data that is accessed most often at a location can be stored locally.

SQL*Net allows the client and server to communicate over a variety of media and protocols. A client/server configuration allows DBAs to distribute CPU-intensive user interfaces to low-cost workstations. It also allows application users to be greeted with the graphical user interface (GUI) with which they are most familiar.
SQL*Net Installations

When installing SQL*Net on OpenVMS, you can choose which SQL*Net adapter to install. During installation, you may install SQL*Net DECnet, SQL*Net TCP/IP, or both.

In addition, the VMS Mailbox adapter is installed automatically, as is the bequeath adapter, which allows mailbox connections without a network configuration or running listener.

**Additional Information:** See the *Oracle7 for OpenVMS Installation Guide* for instructions on installing SQL*Net. Also see the files ORA_RDBMS:READMEVMS.DOC and ORA_NETCONFIG:README_NETCONFIG.DOC.

---

SQL*Net and the Transparent Network Substrate (TNS)

This section introduces SQL*Net in general terms and describes the components that make up SQL*Net Version 2.3.4.

**Using SQL*Net**

SQL*Net connects dissimilar networks together and allows client/server transactions to occur transparently. An end user does not have to know that a network exists, because SQL*Net hides the complexity of machine–level interactions by presenting a layer of interconnectivity to the user through its client/server architecture. This layer is called the Transparent Network Substrate, or TNS.

Figure 1 – 2 shows a client/server configuration using SQL*Net.
Figure 1 – 2 Client/Server Configuration

Figure 1 – 2 shows an OpenVMS computer that holds the physical Oracle7 database and a client workstation with an Oracle Forms application that needs to access the Oracle7 database. The OpenVMS computer (A in the figure) is the server and the workstation (B in the figure) is the client.

The transaction proceeds as follows:

1. The client requests some data.
2. SQL*Net packages the request and sends it to the Transparent Network Substrate (TNS).
3. TNS routes the packaged request to the server.
4. SQL*Net on the server side unpackages the request and sends it to Oracle7.
5. Oracle7 processes the request and sends the requested data to SQL*Net.
6. SQL*Net packages the data and sends it to TNS.
7. TNS routes the data to the client.
8. SQL*Net on the client side unpackages the data and sends it to the application.

**SQL*Net Architecture**  SQL*Net consists of the following components:
- SQL*Net Interface
- Transparent Network Substrate (TNS)
- Oracle Protocol Adapters

**SQL*Net Interface**  The SQL*Net interface bundles or unbundles messages received from TNS. The SQL*Net interface code resides on all nodes that use SQL*Net. On the client (application program) side, the interface bundles the messages received from the application and passes them to TNS for delivery. On the Oracle7 Server side, the interface unbundles the messages received from TNS and passes them to the Oracle7 Server.

**Transparent Network Substrate**  TNS allows peer-to-peer connectivity. It is a transparent layer to which different network protocols are connected. It provides a network of applications above the existing networks of computers.

**Oracle Protocol Adapters**  The Oracle Protocol Adapters allow TNS and its services to communicate over existing network *communication protocols*. The Protocol Adapters map the functions of the underlying protocol into the equivalent functions within TNS. This mapping of communication functions allows calls to or from TNS to be protocol nonspecific.

Figure 1–3 shows how TNS and the Oracle Protocol Adapters interface with existing network protocols. For any TNS client running an industry–standard protocol, the Oracle Protocol Adapter interfaces between the unique API of the underlying protocol and the consistent interface of Oracle’s TNS.
A single machine can support multiple protocols and protocol adapters simultaneously. A node that supports multiple protocols and protocol adapters is said to be a member of multiple TNS communities, one for each protocol installed.

A TNS client belonging to multiple communities is common in two cases:

- As a client application that needs to access other applications in more than one network. Installing two protocols and protocol adapters allows a client to connect to any server application in either community.

- As a server application that is being accessed by clients from multiple TNS communities. Installing two protocols and the protocol adapters would allow all clients from both communities to access a server application on that machine.

**Additional Information:** For further general information about SQL*Net, please refer to the following manuals:

- Oracle Network Products Messages Manual
- Oracle Network Manager Administrator’s Guide
- Understanding SQL*Net
- Oracle Names Administrator’s Guide
- Oracle SNMP Support Reference Guide
- Oracle Advanced Networking Option Administrator’s Guide
This chapter explains how to use Network Manager in an OpenVMS environment. It covers the following topics:

- Pre-Configuration tasks
- Starting Network Manager
- Using Network Manager
- Creating tables for Oracle Names
- Creating and distributing the configuration files

This chapter is intended as an OpenVMS-specific supplement to the Network Manager documentation. For general product information about Oracle Network Manager, please refer to the Oracle Network Manager Administrator’s Guide.
Pre-Configuration Tasks

This section lists the tasks you must perform before using Network Manager.

Before You Set Up OpenVMS Clients

Before you set up your clients, you should have completed the following procedures:

- Verified that your operating environment meets the SQL*Net for OpenVMS system requirements.
- Installed the networking software and hardware on your system.
- Verified that your network protocol is functioning properly.
- Installed SQL*Net on your systems.
- Shut down Oracle7 before loading and building SQL*Net when adding protocol adapters.

For more information about OpenVMS system requirements and installation procedures, see the installation guide for your platform.

For information on loading and operating your protocol, see your network protocol installation guide.

Setting Up OpenVMS Clients

Complete the following configuration tasks for the OpenVMS clients on your network.

- Identify the nodes on the network
- Identify the listener port number for TCP/IP (Oracle recommends using the number 1526, which is reserved for SQL*Net Version 2)
- Identify the listener object name for DECnet
- Define node name (for both TCP/IP and DECnet)

For instructions on performing these tasks, refer to the Compaq Network Documentation.
Starting Network Manager

Network Manager is available on Windows 95, Windows 98, and Windows NT only. Before using Network Manager to generate and distribute the configuration files to your OpenVMS node(s), you must first:

- Validate the connectivity between the PC used for Network Manager and your OpenVMS node.
- Verify that there exists a means by which to transfer the files from the PC to your OpenVMS node (this could be FTP or via Pathworks utilities.)
- If you intend to use an OpenVMS database as a Names Server, verify that the PC Names Server (or TNSNAMES.ORA) contains the connection address of the OpenVMS database to be used for Names. Also, you must first create the required tables (described below under ‘Creating tables for Oracle Names.’)

Using Network Manager

This section gives information about using the Network Manager.

Protocol–Specific Information

In preparing to use Network Manager, the Understanding SQL*Net guide suggests that you plan what specific information about your network you will enter. Aside from entering the names of your communities and listeners, the addresses of your servers, and system identifiers of your databases, you will need to enter protocol–specific information.

If you do not want to accept Network Manager’s default values for some of these protocol–specific keywords, enter your own values using the format suggested in Table 2–1.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Keyword</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECnet</td>
<td>NODE</td>
<td>&lt;nodename&gt;</td>
</tr>
<tr>
<td></td>
<td>OBJECT</td>
<td>&lt;alphanumeric string&gt;</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>HOST</td>
<td>&lt;hostname&gt;</td>
</tr>
<tr>
<td></td>
<td>PORT</td>
<td>&lt;number&gt;</td>
</tr>
</tbody>
</table>

Location-Specific Parameters

The Network Manager window called “Oracle Database” has a field called “Location-specific parameters,” with the following keyword:

PROGRAM=

After the keyword, enter the file specification of the command procedure that sets up the environment for the database and starts the database server.

The full entry should be in the following format:

PROGRAM=<directory><filename>

where:

Directory Full specification of the directory containing the command procedure. Do not use a logical name to refer to this directory.

Filename Name of the command file that sets up the database environment. A sample file, called ORASRV_NETV2_COM.SAMPLE, is in the TNS_ADMIN directory. You can use this sample file as a base, editing it so that it contains correct information about your own database.

User-Defined Parameters

This field, which is also in the Oracle Database window, asks whether there are any special parameters you wish to set.

This field is optional for configurations of Oracle7 on OpenVMS. If you wish, you can use this field to set preferences for the server process, such as process quotas and priority.

Syntax of the LISTENER.ORA File

```
listener_alias =
  (ADDRESS_LIST=
   (ADDRESS=
    (PROTOCOL=TCP)
    (PORT=number)
    (HOST=name|ip_address)
    (QUEUESIZE=number))
   (ADDRESS=
    (PROTOCOL=DECNET)
    (NODE=name)
    (OBJECT=object name|number)
    (QUEUESIZE=number))))
STOP_listener_alias=YES|NO
CONNECT_TIMEOUT_listener_alias=number
TRACE_LEVEL_listener_alias=number|USER|ADMIN|OFF
```
SID_LIST_listener_alias =
    (SID_DESC=
        (SID_NAME=sid_name)
        (PROGRAM=file-spec)
        (TIMEOUT=number) (optional)
        (OUT=file-spec) (optional)
        (OSDS=(PRIORITY=number) (optional)
            (QUOTA=(ASTLM=number) (optional)
                (BIOLM=number) (optional)
                (BYTLM=number) (optional)
                (CPULM=number) (optional)
                (DIOLM=number) (optional)
                (FILLM=number) (optional)
                (PGFLQUOTA=number) (optional)
                (PRCLM=number) (optional)
                (TQLM=number) (optional)
                (WSQUOTA=number) (optional)
                (WSDEFAULT=number) (optional)
                (ENQLM=number) (optional)
                (WSEXTENT=number) (optional)
                (JTQUOTA=number)))) (optional)

Sample LISTENER.ORA File SID_LIST

SID_LIST_listener1=
    (SID_DESC=
        (SID_NAME=PROD)
        (PROGRAM=DISK$NET1:[ORACLE732.DB_PROD]ORASRV_NETV2.COM)
        (TIMEOUT=24)
        (OSDS=(QUOTA=(ASTLM=250)
            (BIOLM=100)
            (BYTLM=150000)
            (CPULM=0)
            (DIOLM=100)
            (FILLM=100)
            (PGFLQUOTA=250000)
            (PRCLM=10)
            (TQLM=100)
            (WSQUOTA=4096)
            (WSDEFAULT=2048)
            (ENQLM=2000)
            (WSEXTENT=8192)
            (JTQUOTA=8192))))
In this example, note the following settings:

- The server process is set to time out after 24 minutes
- The OSDS (operating system dependent settings) consist of OpenVMS process quotas for the server process which the listener starts; had the user not specified these settings, the server process would use the settings of the account which starts the process.

Additional Information: If you are using an OpenVMS system to create files that will be used by another operating system, you should refer to the Oracle documentation for this operating system.

Creating Tables for Oracle Names

If you will be using Oracle Names, you must store information about your network in a database. To create the database tables for storing this configuration information, do the following:

- Copy the appropriate SQL script files to TNS_ADMIN for your Names database. The SQL install scripts are located on the PC (in ORAWIN/DBS) from where Network Manager will be run. The following files are required:
  - ROSBILD.SQL
  - ROSGRNT.SQL
  - NMCBILD.SQL
  - NMCGRNT.SQL

- After copying the files to TNS_ADMIN, run the scripts from within SQLPLUS in the following order:
  SQLPLUS> @tns_admin:rosbild
  SQLPLUS> @tns_admin:rosgrnt PUBLIC
  SQLPLUS> @tns_admin:nmcbild
  SQLPLUS> @tns_admin:rosgrnt PUBLIC
Creating and Distributing the Configuration Files

As described in *Understanding SQL* Net, the Network Manager generates several configuration files for the Oracle networking products. Appendix A of *Understanding SQL* Net describes the contents of the configuration files.

After you’ve generated these files using Network Manager, you must distribute them among the client and server nodes that will be using them.

The following table lists the configuration files and their appropriate directory locations. Copy each file to the location listed in Table 2–2.

<table>
<thead>
<tr>
<th>Configuration File</th>
<th>Node</th>
<th>Directory Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNSNAMES.ORA</td>
<td>Clients not using Oracle Names</td>
<td>ORA_ROOT: [NETWORK.ADMIN]</td>
</tr>
<tr>
<td>NAMES.ORA</td>
<td>Node on which Oracle Names is installed</td>
<td>ORA_ROOT: [NETWORK.ADMIN]</td>
</tr>
<tr>
<td>SQLNET.ORA</td>
<td>Clients and servers</td>
<td>ORA_ROOT: [NETWORK.ADMIN]</td>
</tr>
<tr>
<td>LISTENER.ORA</td>
<td>Servers</td>
<td>ORA_ROOT: [NETWORK.ADMIN]</td>
</tr>
<tr>
<td>TNSNAV.ORA</td>
<td>Not applicable on OpenVMS</td>
<td></td>
</tr>
<tr>
<td>TNSNET.ORA</td>
<td>Not applicable on OpenVMS</td>
<td></td>
</tr>
<tr>
<td>INTCHG.ORA</td>
<td>Not applicable on OpenVMS</td>
<td></td>
</tr>
<tr>
<td>PROTOCOL.ORA</td>
<td>Valid node verification</td>
<td>ORA_ROOT: [NETWORK.ADMIN]</td>
</tr>
</tbody>
</table>

Table 2–2 Configuration files and directory locations

If you are generating to an OpenVMS Names Server, once again, verify both the general connectivity between the PC and the OpenVMS host, as well as the connectivity between PC NetMan and your OpenVMS database.
This chapter gives information about the following protocol adapters on OpenVMS:

- Mailbox Adapter
- DECnet Adapter
- TCP/IP Adapter
- Bequeath Adapter
- Bequeath Listener

Note: This chapter is a supplement to the Understanding SQL*Net guide.
Mailbox Adapter

The Mailbox protocol adapter, or IPC adapter, is automatically configured for use when you install SQL*Net. It can be used for client/server connections when both client and server are on the same OpenVMS node. If the client and server are on different machines, then the connection must take place using DECnet or TCP/IP.

When configuring the TNS listener to listen for mailbox connections, you need to specify a KEY value in LISTENER.ORA for the IPC protocol. The listener then creates a mailbox which listens for connections and creates a system–wide logical name (the same as the KEY value) which translates to this mailbox device. It is via this logical name that clients find the listener’s mailbox.

When the SQL*Net parameter AUTOMATIC_IPC is not turned off, connect descriptors specifying the DECnet or TCP/IP protocols for a server on the same node are converted to mailbox connections. If you wish to prevent this, you must include the following line in the TNS_ADMIN:SQLNET.ORA file:

AUTOMATIC_IPC = OFF

Syntax

The following fields must be defined:

(PROTOCOL=IPC)
(KEY=<IPC logical name>)

where:

PROTOCOL The keyword that identifies the specific protocol adapter used; for this protocol, the value is IPC. The value can be entered in either uppercase or lowercase.

KEY The logical name used to connect to the listener via the Mailbox adapter.

Example

This example shows the two fields for the OpenVMS Mailbox adapter.

(PROTOCOL=IPC)
(KEY=ORA_IPC)

Note: A full example of a Mailbox connect descriptor can be found in the file TNS_ADMIN:TNSNAMES_ORA.SAMPLE.
The DECnet protocol adapter provides support for client/server connections using DECnet as a protocol. You can turn SQL*Net support for DECnet on or off via the NetConfig configuration screen (please refer to the Oracle7 for OpenVMS Installation Guide for your platform.)

When configuring the TNS listener to listen for DECnet connections, you need to specify an OBJECT value in the LISTENER.ORA file for the DECnet protocol. The listener registers itself as a DECnet object, using the name specified as the OBJECT value. After the listener is started, you can see the listener object if you issue the following command in the DECnet NCP utility:

```
NCP> SHOW KNOWN OBJECTS
```

The DECnet protocol adapter supports both DECnet Phase IV and DECnet Phase V (DECnet/OSI.) However, if you are using DECnet Phase V, depending on your configuration, you may need to use Phase IV aliases in your TNS connect descriptors.

**Syntax**

The following fields must be defined:

- (PROTOCOL=DECNET)
- (NODE=node)
- (OBJECT=object name)

The following field is optional:

- (QUEUESIZE=n)

where:

**PROTOCOL**

Keyword that identifies the specific protocol adapter used; for this protocol, the value is DECNET. The value can be entered in either uppercase or lowercase.

**NODE**

Name of the remote node to which you want to connect. This node must be included in the list of your system’s node names. Use the NCP utility to display these parameters by entering the following command in NCP:

```
NCP> SHOW KNOWN NODES
```
DECnet object name of the listener to which you want to connect on the specified node. The object is created when the listener starts and is deleted when the listener is stopped. The object of an alias used in the TNSNAMES.ORA file must have a corresponding address in the LISTENER.ORA file; it must have the same object, protocol, and node. An object name must be less that 15 characters in length.

Use upper case when entering object IDs.

Parameter that can be used to enlarge the connection mailbox. This parameter is optional; if it is not specified, the default value of 20 is used, allowing twenty DECnet connections to the listener to be pending. If many simultaneous connections are made to the listener, some connection requests may not be received if the DECnet connection mailbox is too small.

**Example**

In this example, the DECnet connect descriptor specifies a connection to the listener on VMS1 with object name LISTEN1.

(PROTOCOL=DECNET)
(NODE=VMS1)
(OBJECT=LISTEN1)

**Note:** A full example of a DECnet connect descriptor can be found in the file TNS_ADMIN:TNSNAMES_ORA.SAMPLE.
TCP/IP Adapter

The TCP/IP protocol adapter provides support for client/server connections using TCP/IP as a protocol. You can turn SQL*Net support for TCP/IP on or off via the NetConfig configuration screen (please refer to the Oracle7 for OpenVMS Installation Guide for your platform).

SQL*Net 2.3 on OpenVMS is developed and certified using Compaq’s TCP/IP Services for OpenVMS (UCX). If you wish to use the TCP/IP protocol adapter for SQL*Net, you should have Version 4.2 or higher of TCP/IP Services for OpenVMS installed. TCP/IP protocol stacks from other vendors may work with Oracle, but customers use these products at their own risk. Any TCP/IP problems that can not be reproduced using TCP/IP Services for OpenVMS will simply be referred to the TCP/IP vendor.

Syntax

The following fields must be defined:

(PROTOCOL=TCP)
(HOST=hostname)
(PORT=port#)

The following field is optional:

(QUEUESIZE=n)

where:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROTOCOL</td>
<td>Keyword that identifies the specific protocol adapter used; for this protocol, the value is TCP. The value can be entered in either uppercase or lowercase.</td>
</tr>
<tr>
<td>HOST</td>
<td>Host name or IP address.</td>
</tr>
<tr>
<td>PORT#</td>
<td>TCP/IP port number of the listener to which you wish to connect on the host specified.</td>
</tr>
<tr>
<td>QUEUESIZE</td>
<td>Parameter to increase the queue size. This parameter is optional; if it is not specified, the default value of 20 is used. If simultaneous connections are made to the listener, some connection requests may not be received if the listener socket queue size is too small.</td>
</tr>
</tbody>
</table>
Example

In this example, the TCP/IP connect descriptor specifies a listener on the VMS1 host.

(PROTOCOL=TCP)
(HOST=VMS1)
(PORT=1526)

Note: A full example of a TCP/IP connect descriptor can be found in the file TNS_ADMIN:TNSNAMES_ORA.SAMPLE.

Bequeath Adapter

Each database that you wish to connect to with the bequeath adapter must have a command file named ORASRV_BEQ_<sid>.COM in ORA_ROOT:[NETWORK.ADMIN]. For databases created with the Oracle7 Release 7.3.2 or later, this command file is generated when you create the database. You must create this command procedure manually for pre-existing databases. Edit the ORA_ROOT:[NETWORK.ADMIN]ORASRV_BEQ_COM.SAMPLE file or execute the command procedure ORA_RDBMS:CREATE_ORASRV_BEQ.COM as follows:

$ @ORA_RDBMS:CREATE_ORASRV_BEQ <ora_db> <sid> <dbname>

where:

<ora_db> is the database administration directory;

<sid> is the SID of the database, for example, PROD; and

<dbname> is the NAME of the database.

For example:

$ @ORA_RDBMS:CREATE_ORASRV_BEQ DKA400:[ORACLE73.DB_PROD] – PROD PRODDB

Note: You can find the contents of ORASRV_BEQ_COM.SAMPLE file in Appendix C of this manual.

A new mechanism for the bequeath adapter was implemented with Oracle7 Version 7.3.3. This mechanism uses the new BEQUEATH_LISTENER process.
The Bequeath Listener, running as a detached process, creates detached server processes to service clients on the same machine, using the bequeath adapter. This allows the Oracle server to run in a suitably privileged process. The alternative would be to have the server installed with privileges and run in a subprocess of the client. However, that would require the server to be linked without traceback information, making server trace information unusable if problems are encountered.

For each request from the client, the Bequeath Listener creates a detached server process and two mailboxes. It then sends the mailbox names to the client and the client establishes a connection to the server using these mailboxes.

By default, these mailboxes are created with a buffer quota of 8192 bytes and a maximum message size of 2048 bytes. You can change these parameters by defining logical names with other values. For example:

```
$ define ORA_BEQ_MBXSIZ n
$ define ORA_BEQ_MBXBFQ n
```

The maximum value for the mailbox buffer quotas is 60000 bytes. You should adjust these values carefully, and you should adjust them for performance reasons only.

The Bequeath Listener uses a known mailbox name to listen for client requests. This mailbox name is in the format:

```
ORA_BEQ_READ_MBX_xxxxxxxxxx_n
```

where:

`xxxxxxxxxx` is a unique number that is automatically generated when you install the Bequeath Listener.

`n` is a single-digit number (0–9) that is the Bequeath Listener number.

Starting up the Bequeath Listener

The Bequeath Listener starts automatically when INSORACLE is invoked (at installation time or later, usually during system startup). Unless you decide to invoke the REMORACLE command, the Bequeath Listener should be up and running all the time.

For a particular installation, up to 10 Bequeath Listeners may run. You might choose to have more than one Bequeath Listener if you have numerous, near-simultaneous connection requests using the bequeath
adapter. When there is more than one Listener, the client will randomly pick one to use for a connection request. To cause more than one Bequeath Listener to start, define a process logical name ORA_BEQ_NUM_OF_LISTENERS to a value between 2 and 10.

If the Bequeath Listener is down and you want to start it, execute the command:

```
BEQLSNR START [n]
```

If you do not provide the optional numeric parameter, then the number of Bequeath Listeners specified by ORA_BEQ_NUM_OF_LISTENERS will be started. If n is provided, a number between 0 and 9, then only that particular Bequeath Listener will be started.

**Bequeath Listener Status**

You can issue a status command to determine whether the Bequeath Listener is up and running. Issue the command:

```
BEQLSNR STATUS [n]
```

If you do not provide the optional numeric parameter, then Bequeath Listener 0 is queried. To query Bequeath Listeners 1 through 9, if they exist, supply the number on the command line.

**Shutting down the Bequeath Listener**

To stop the Bequeath Listener issue the command:

```
BEQLSNR STOP [n]
```

If you do not provide the optional numeric parameter, then all Bequeath Listeners for the installation are stopped. To stop a particular Bequeath Listener, provide its number in the command line.

**The ORASRV_BEQ_<sid>.COM Command File**

Every SID served by your Oracle installation required that a command file named ORASRV_BEQ_<sid>.COM exist in ORA_ROOT:[NETWORK.ADMIN]. When the Bequeath Listener starts a server process for a particular SID, it is this command file that the process runs.

A sample of this file is provided with your Oracle installation and is named ORA_ROOT:[NETWORK.ADMIN]ORASRV_BEQ_COM.SAMPLE. If you have instances created before Oracle7 Version 7.3.2.3.2, then you will need to use this sample file to create the ORASRV_BEQ_<sid>.COM file for
the instance. If you are upgrading from Oracle7 Version 7.3.2.3.2, then
the ORASRV_BEQ_<sid>.COM files already exist in the old installation, but
they must be modified. Use the sample file to make the modifications.
If you are upgrading from Oracle7 Version 7.3.3, then the
ORASRV_BEQ_<sid>.COM files can simply be copied from your old
installation. When you create a new database using
ORA_INSTALL:ORACLEINS.COM a new ORASRV_BEQ_<sid>.COM file is
created for you automatically.

Problem Resolution

Writing trace information

The Bequeath Listener writes some trace information, but because the
output of the detached processes is set to the null device (NL:], you
will not normally see it.

To get the trace information from the Bequeath Listener, you should do
the following:

1. Stop the Bequeath Listener
2. Edit the STARTUP_BEQLSNR.COM
3. Change the NL: to a file name
4. Restart the Bequeath Listener

Changing the quota for a Server Process that is created by the
Bequeath Listener

To change the quota, modify the file BEQLSNR.COM and remove the
comments for the quota parameter that you want to change. Be sure to
STOP/START the Bequeath Listener after modifying this file.

Hitting the problem: ORA–12203: TNS:unable to connect to
destination

If you hit this problem, issue the command BEQLSNR STATUS to
determine whether the Bequeath Listener is up and running. If the
Bequeath Listener does not respond, use the command BEQLSNR
STOP to stop the Bequeath Listener and use the command BEQLSNR
START to restart it.
Client hitting the problem: ORA-12203: TNS:unable to connect to destination

Choose one of the following solutions:

- Change the logical ORA_BEQ_TIMEOUT to something greater than 120 seconds (for example: 300 seconds). Before running the client program, define this logical also in the ORA_NETWORK:BEQLSNR.COM file or

- Define the logical ORA_BEQ_NUM_OF_LISTENERS to a value between 1 and 10 to increase the capacity when a number of clients are connecting at the same time to the Bequeath Listener.

With this method, you can increase the number of connections that the Bequeath Listeners can handle at one time. Each time that a client requests a connection, it will randomly pick one of the Bequeath Listeners that are running to serve it with the connection request. Note that you do not need to STOP/START the Bequeath Listener after defining this logical name. This logical name determines the number of Bequeath Listeners.

Bequeath Listener Privileges

The process in which the Bequeath Listener runs must have the OpenVMS privileges in the table below to be able to perform the associated function.

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMKRNL</td>
<td>Pass this privilege to server processes that the Listener creates.</td>
</tr>
<tr>
<td>DETACH</td>
<td>Create detached processes.</td>
</tr>
<tr>
<td>LOG_IO</td>
<td>Perform certain I/O functions.</td>
</tr>
<tr>
<td>NETMBX</td>
<td>Use DECnet.</td>
</tr>
<tr>
<td>PRMMBX</td>
<td>Create a permanent mailbox on which to listen. (The mailbox is permanent so that the logical name associated with it goes into the SYSTEM logical name table.)</td>
</tr>
<tr>
<td>SYSLCK</td>
<td>Lock system wide resources.</td>
</tr>
<tr>
<td>SYSPRV</td>
<td>Allow the listener to deassign logical names tables from a different UIC.</td>
</tr>
<tr>
<td>SYSNAM</td>
<td>Create SYSTEM logical names and shared logical name tables.</td>
</tr>
<tr>
<td>TMPMBX</td>
<td>Create temporary mailboxes.</td>
</tr>
</tbody>
</table>
Allow the Listener to get information about and to control processes that it may not have created, such as dispatchers and shared server processes.

**Note:** Before attempting to start the Bequeath Listener, the process that starts the Bequeath Listener must have the privileges in this table or be able to have them set.

<table>
<thead>
<tr>
<th>PRIVILEGE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORLD</td>
<td>Allow the Listener to get information about and to control processes that it may not have created, such as dispatchers and shared server processes.</td>
</tr>
</tbody>
</table>

**Table 3–1 Privileges and Their Functions**
This chapter gives information about the TNS Listener on OpenVMS. It covers the following topics:

- Introduction
- LSNRCTL
- Listener Privileges
- Process Quotas
- ORASRV_NETV2 Command File
- General Connections
Introduction

The function of the TNS Listener is to receive connection requests from local or remote clients and to provide the client with a Server process to which to connect. The Listener can service multiple instances. For each instance, the Listener keeps a list of services that provide access to that instance. If multithreaded servers are being used, the Listener may direct a client connection to a dispatcher. Otherwise, for dedicated servers, the Listener will direct the client connection to an existing prespawned server or will create a new server process to service the connection.

Additional Information: General information about the Listener and its configuration can be found in the generic SQL*Net documentation. This chapter provides only information about the Listener that is specific to OpenVMS.

LSNRCTL

The LSNRCTL utility is used to start and stop the Listener and to query its status or services. The LSNRCTL command executes the command procedure ORA_NETCONFIG:LSNRCTL.COM, which provides a shell to the executable program ORA_NETCONFIG:LSNRCTL.EXE.

The main function of the command procedure is to check that the privileges required to start the Listener are present (see the section "Listener Privileges"). If a LSNRCTL START command is entered and the required privileges are not present, an error is displayed and LSNRCTL exits.

Note: Start the listener using the Oracle account.

Warning: If you enter the LSNRCTL interactive mode by giving the LSNRCTL command without a subcommand and you have received a warning about inadequate privileges, DO NOT attempt to start the Listener. The Listener process may still start, depending on the privileges you do have, but it may not function properly.

Warning: Also, DO NOT start the Listener from a process that has a UIC in the system group, for example a group less than or equal to MAXSYSGROUP. If you enter a LSNRCTL START command from such a process, an error is displayed and LSNRCTL exits. If you enter a LSNRCTL command with no arguments, you are warned not to start the Listener from within the LSNRCTL utility. If the Listener did
run in a system group, any Server processes it creates would be in the system group and the Server would abort, because it does not allow itself to run in privileged groups.

**Listener Privileges**

The process in which the Listener runs must have the OpenVMS privileges in Table 4.1 to be able to perform the associated function.

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMKRNL</td>
<td>Pass this privilege to server processes that the Listener creates.</td>
</tr>
<tr>
<td>DETACH</td>
<td>Create detached processes.</td>
</tr>
<tr>
<td>LOG_IO</td>
<td>Perform certain I/O functions.</td>
</tr>
<tr>
<td>NETMBX</td>
<td>Use DECnet.</td>
</tr>
<tr>
<td>PRMMBX</td>
<td>Create a permanent mailbox on which to listen.</td>
</tr>
<tr>
<td></td>
<td>(The mailbox is permanent so that the logical name associated with it goes into the SYSTEM logical name table.)</td>
</tr>
<tr>
<td>SYSLCK</td>
<td>Lock system wide resources.</td>
</tr>
<tr>
<td>SYSNAM</td>
<td>Create SYSTEM logical names and shared logical name tables.</td>
</tr>
<tr>
<td>TMPMBX</td>
<td>Create temporary mailboxes.</td>
</tr>
<tr>
<td>WORLD</td>
<td>Allow the Listener to get information about and to control processes that it may not have created, such as dispatchers and shared server processes.</td>
</tr>
</tbody>
</table>

**Note:** Before attempting to start the Listener, the process that starts the Listener must have the privileges in this table or be able to have them set. As noted above, the LSNRCTL command file will attempt to set these privileges and warn the user if it was unable to do so.

Table 4–1 Privileges and Their Functions
Process Quotas

Process quotas for the Listener and for the Server processes which the Listener creates can be controlled by logical names. The logical names are:

ORACLNSR_<quotaname>

where:

<quotaname> is ASTLM, BIIOLM, BYTLM, CPULM, DIOLM, FILLM, PGFLQUOTA, PRCLM, TQELM, WSQUOTA, WSDEFAULT, ENQLM, WSEXTENT, or JTQUOTA.

Several of the logical names are defined in LSNRCTL.COM and control the quotas of the Listener process. They are defined in user mode so that they are not present after exiting LSNRCTL. If your Listener supports an especially large number of services, some of these quotas may need to be increased. For the quotas you determine to be deficient or under direction of Oracle Support, you can edit the quota values in LSNRCTL.COM.

To control the quotas of the processes that the Listener creates, specify the logical names in the file ORA_NETWORK:TNSLSNR.COM, the command file that runs in the Listener process. Statements to define these logical names are in TNSLSNR.COM, but are commented out.

If, for example, a very large file–backed SGA requires that Server processes have larger quotas, you can uncomment the appropriate logical name definition in TNSLSNR.COM and specify the quota value. Starting with Version 7.3.2.3.2 and the VLM feature, a file backed SGA is created when the INIT.ORA parameter VLM_BACKING_STORAGE_FILE is set to TRUE.

Quotas can also be specified for the Server processes in the LISTENER.ORA file on a SID–by–SID basis. This is done in the SID_DESC section for a Listener. For example:

SID_LIST_LISTENER =
 (SID_DESC =
    (SID_NAME = <name>)
    (PROGRAM = <disk:><directory>\ORASRV_NETV2.COM)
    (OSDS=)
       (PRIORITY=<number>)
    (QUOTA=
       (ASTLM=<number>)
       (BYTLM=<number>)
       (PGFLQUOTA=<number>)
    )
)
There are no restrictions on the number of quotas that you can specify in the QUOTA list. However, if any quota is specified in the QUOTA list, then none of the quotas specified by logical name will be used and quotas that are not specified in the list will assume the system default.

⚠️ **Warning:** The process priority of the Server process can also be specified, as in the example above, but this is not recommended.

### ORASRV_NETV2 Command File

In the LISTENER.ORA file, you provide a SID_LIST for each defined Listener, and for each SID you provide a program name. The program name is the full directory path, without Oracle logical names, to a command file, based on the file `TNS_ADMIN:ORASRV_NETV2_COM.SAMPLE`.

When you edit the file `TNS_ADMIN:ORASRV_NETV2_COM.SAMPLE` for a particular SID, it may have any name. The recommendation is to name it:

`TNS_ADMIN:ORASRV_NETV2_<sid>.COM`

For example:

```plaintext
(SID_LIST_LISTENER =
  (SID_DESC =
    (SID_NAME = PROD)
    (PROGRAM = DK:[O732.NETWORK.ADMIN]ORASRV_NETV2_PROD.COM)
  )
)

When the Listener starts a prespawned or dedicated server process, the process runs this command file. To create this file for a particular SID, change the following three lines in the sample command file:

```bash
$ ora_db = "database administration directory"
$ sid = "sid name"
$ dbname = "database name"
```

where:

"database administration directory" is the directory that contains the startup, shutdown, and orauser command files for the database.
Specify a full directory path, without using Oracle logical names.

“sid name” and “database name” are the names provided for the database when it was created.

**Note:** You can find a sample command file in Appendix B of this document.

---

### General Connections

Make sure that your SQL*Net task file defines any logical names used by the INIT.ORA parameters USER_DUMP_DEST and BACKGROUND_DUMP_DEST (if defined).

**Note:** If you define these logical names by calling ORA_DB:ORAUSER_<database_name>.COM, make sure that ORA_UTIL:ORAUSER.COM runs RDBMSUSER.COM **AFTER** PROGINTUSER.COM.

PROGINTUSER.COM also defines ORA_SLAX, making it into a search list with ORA_PROGINT_MESG and ORA_RDBMS.
This chapter provides information about Oracle Names on OpenVMS. It covers the following topics:

- Introduction
- NAMESCTL
- Names Server Privileges
- Requirements

Note: This chapter assumes that Oracle Names and all the related SQL*Net products have been installed at your site.
Introduction

The function of the Names Server is to resolve connection addresses in a homogeneous and centralized location. As a client issues a connection request, the Names Server is responsible for directing the client connection request to the appropriate listener for the specified SID. TNSNAMES.ORA can also resolve the listener address. However, the benefits of the centralized list of connection addresses that Oracle Names provides greatly eases the maintenance of large network definitions.

NAMESCTL

The NAMESCTL utility is used to start and stop the Names Server and to query its status or services. The NAMESCTL command executes the command procedure ORA_NETCONFIG:NAMESCTL.COM, which provides a shell to the executable program ORA_NETCONFIG:NAMESCTL.EXE.

The main function of the command procedure ORA_NETCONFIG:NAMESCTL.COM is to check that the privileges required to start the Names Server are present (see the section "Names Server Privileges"). If a NAMESCTL START command is entered and the required privileges are not present, an error is displayed and NAMESCTL exits.

Note: Start the Names Server using the Oracle account.

⚠️ Warning: If you enter the NAMESCTL interactive mode by giving the NAMESCTL command without a subcommand and you have received a warning about inadequate privileges, **DO NOT** attempt to start the Names Server. The Names Server process may still start, depending on the privileges you do have, but it may not function properly.

⚠️ Warning: **DO NOT** start the Names Server from a process that has a UIC in the system group, for example a group less than or equal to MAXSYSGROUP. If you give a NAMESCTL START command from such a process, an error is displayed and NAMESCTL exits. If you enter a NAMESCTL command with no arguments, you are warned not to start the Names Server from within the NAMESCTL utility.
Names Server Privileges

The process in which the Names Server runs must have the OpenVMS privileges in Table 5.1 to be able to perform the associated function.

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMKRN</td>
<td>Facilitate kernel mode processing</td>
</tr>
<tr>
<td>NETMBX</td>
<td>Use DECnet</td>
</tr>
<tr>
<td>PRMMBX</td>
<td>Create a permanent mailbox on which to listen (The mailbox is permanent so that the logical name associated with it goes into the SYSTEM logical name table.)</td>
</tr>
<tr>
<td>SYSNAM</td>
<td>Create SYSTEM logical names and shared logical name tables</td>
</tr>
<tr>
<td>TMPMBX</td>
<td>Create temporary mailboxes</td>
</tr>
</tbody>
</table>

**Note:** Before attempting to start the Names Server, the process that starts the Names Server must have the privileges in this table or be able to have them set. As noted above, the NAMESCTL command file will attempt to set these privileges and warn the user if it was unable to do so.

Table 5–1 Privileges and Their Functions

Requirements

Before you can use Oracle Names, the following configuration files must be in the TNS_ADMIN directory of the system that calls Oracle Names:

- NAMES.ORA
- SQLNET.ORA

**NAMES.ORA**

The NAMES.ORA file describes the Names Server and is read by NAMESCTL.EXE at Server startup. NAMES.ORA identifies, among other things:

- Address of the running server
- Trace information
- Protocols serviced
- Indication of whether the network definition is stored in file format or in a database
• SID of the Names database (if applicable)

**SQLNET.ORA**

The SQLNET.ORA file is read by both the client, as part of a connection request, and by NAMESCTL, for all operations other than server startup. This file identifies the Names service to be used to resolve the connection. Within SQLNET.ORA, the NAMES.PREFERRED_SERVERS list specifies to the client the address of the Names Server with which to connect.

**Note:** The presence of the NAMES.PREFERRED_SERVERS clause in SQLNET.ORA supersedes the presence of TNSNAMES.ORA in TNS_ADMIN. Therefore, if the use of a Names Server has been specified with SQLNET.ORA, then even if TNSNAMES.ORA exists and has a complete list of connection addresses, the client will not read TNSNAMES.ORA.
This chapter provides information about installing and running the Oracle Intelligent Agent (hereafter referred to as “the Agent”) on OpenVMS and information about Simple Network Management Protocol (SNMP) Support. Read this chapter carefully and completely before starting to install and use the Agent and SNMP Support on OpenVMS.

This chapter covers the following topics:

- Introduction
- Installing the Oracle Intelligent Agent
- Oracle Intelligent Agent Setup and Discovery Option
- Oracle Intelligent Agent Startup, Shutdown, and Status Query
- Oracle Intelligent Agent Maintenance
- Oracle Intelligent Agent Product Capabilities
- SNMP Support
Introduction

The Agent is a backend server process that communicates with the Oracle Enterprise Manager (OEM) running on a Windows NT machine.

The current port of the Agent is certified only when using the PEER Master SNMP Agent, supplied as part of the NETCONFIG distribution.

Additional Information: General information about the Oracle Intelligent Agent can be found in the Windows NT OEM Documentation.

Additional Information: General information about SNMP can be found in the Oracle SNMP Support Reference Guide.

Installing the Oracle Intelligent Agent

The Agent requires that TCP/IP be installed on your OpenVMS system. In addition, you must enable TCP/IP support for SQL*Net in the NetConfig configuration screen.

Additional Information: See Chapter 3, "The TCP/IP Adapter"

The Agent may be installed at the same time as other products or it may be installed later.

Installation of the Agent creates the directories ORA_ROOT:[OEMAGENT] and ORA_ROOT:[NETWORK.AGENT]. Most of the Agent files will reside in or beneath these directories.

If you are using the same Oracle Installation from more than one node in a VMS Cluster, you can only run the agent from this installation on one of the nodes. If you attempt to run the agent on multiple nodes from the same installation, there will be file name and file usage conflicts. This is a generic limitation of the Oracle Intelligent Agent on Clusters for all platforms.

For each additional node on which you wish to run the Agent, you must perform a client–only installation for the Agent (installing AGENT, NETCONFIG, and UTIL) and run the Agent from this client–only installation.
Oracle Intelligent Agent Setup and Discovery Option

To correctly set up the Agent environment, the following two kinds of files need to be created:

- Startup command procedures
- Parameter (*.ORA) files

Creating the Startup Scripts

Once the Agent has been successfully installed, create the following three files:

- `ORA_ROOT:[NETWORK.AGENT]AGENT_START.COM`
- `ORA_ROOT:[NETWORK.AGENT]DBSNMPC.COM`
- `ORA_ROOT:[NETWORK.AGENT]DBSNMPJ.COM`

To create these three files, use the sample files `AGENT_START_COM.SAMPLE`, `DBSNMPC_COM.SAMPLE`, and `DBSNMPJ_COM.SAMPLE` provided in `ORA_ROOT:[NETWORK.AGENT]`. Correctly fill in the values for `ora_db`, `sid`, and `dbname` in these three files.

When you startup the Agent, `AGENT_START.COM` is run as a detached process. `DBSNMPC_NETV2.COM` is invoked, again as a detached process, whenever the OEM console requests a connection on the address specified by the `dbsnmp.spawn_address` parameter in `SNMP_RW.ORA`.

Agent Parameter Files and Discovery Option

At startup, and when requested by the daemon thereafter, the agent runs a `Tcl` script called `NMICONF.TCL`, which resides in the `ORA_ROOT:[NETWORK.AGENT.CONFIG]` directory. This script starts by reading the `ORATAB` file, which is kept in the `TNS_ADMIN` directory.

The `ORATAB` file should be as follows:

```
<sid_name>,<tns_admin_directory_for_SID>
```

For example:

```
ora734,disk$d1:[oracle7.network.admin]
```

Note that you can specify any SID that you want the Agent to monitor and that exists on this node.
If the ORATAB.ORA file does not exist, then ORA_RDBMS:ORA_RDBMS_SIDS.DAT will be used to tell the Agent which services to monitor for this node.

Note that this file contains service entries just for the current installation where the Agent is running and not for all the SIDs that are on that node. If you want the Agent to monitor other SIDs in different installations, put them in the TNS_ADMIN:ORATAB.ORA file.

Then, for each database instance found in ORATAB, the tnsnames in that SID’s Oracle installation list is searched for an address on the local host with the appropriate SID in the CONNECT_DATA. The listener.ora found in that same directory is searched for the SID of the database. Again, the first listener that matches our SID becomes the listener active for that database.

Note: This generic discovery phase is impossible if the local database names are stored in Oracle Names instead of in a local TNSNAMES.ORA file. We cannot do the backwards SID–to–name matching through Names. As a result, if Oracle Names is in use for the host, an old–style SNMP.ORA must still be in TNS_ADMIN, with the parameter nmi.register_with_names set to FALSE. If this flag is detected at start–up, none of the generic discovery occurs. Instead, the information in the old–style snmp.ora is used to construct the new configuration files.

The configuration files SNMP_RO.ORA and SNMP_RW.ORA are created.

The file SNMP_RO.ORA should reside in TNS_ADMIN, the same location as the TNS config files. This file contains lines from the old SNMP.ORA that should never be touched by the user:

SNMP.VISIBLESERVICES=(SERVICE–1,...SERVICE–N)
SNMP.SID.<service name>=<SID for the database>
SNMP.ORACLEHOME.<service name>=<oracle_home>

The file SNMP_RW.ORA should reside in TNS_ADMIN, the same location as the TNS config files. This file contains the following lines from the old SNMP.ORA that are automatically generated, but the user may want to modify or add to them:

SNMP.INDEX.<service name>=<integer index>

Note: The integer must be unique on the host.

SNMP.CONTACT.<service name>=<free form text giving contact info>
NMIREGISTERWITHNAMES=[TRUE|FALSE]
Note: The choice of TRUE or FALSE determines whether the agent should try to register itself with Oracle Names.

NMI.TRACE_LEVEL=0
NMI.TRACE_MASK=(106)
DBSNMP.ADDRESS=<TNS address on which the agent can listen>
DBSNMP.SPAWNADDRESS=<TNS address for the services.ora>

The following lines are not automatically generated, but may be added to the SNMP_RW.ORA file:

SNMP.CONNECT.<service name>.NAME= <user name for the subagent to use>
SNMP.CONNECT.<service name>.PASSWORD= <password for the subagent to use>

Note: This is optional for ALL databases.

SNMP.DBPOLLTIME= <interval for polling the database, in seconds>

Note: Note: This line governs the time that the subagent polls the database to determine whether it has gone down, or the interval between retries if the database has gone down or was never connected. The default is 30 seconds.

NMI.TRACE_DIRECTORY= <directory to which NMD will have write access>
NMI.TRACE_FILE= <file name for the trace>
NMI.LOG_DIRECTORY= <directory to which NMD will have write access>
NMI.LOG_FILE= <file name for the log file>

Note: The address chosen is a reserved TCP port granted to Oracle by the IANA (Internet Assigned Number Authority). Changing this port will likely make the agent undetectable by the EM Console and force a manual configuration step!

The TCL script NMICONF.TCL can execute other TCL scripts written specifically to discover other Oracle services. If these other scripts exist, they should be installed with NMICONF.TCL in ORA_ROOT:[NETWORK.AGENT.CONFIG] and their names should be listed in a file in the same directory called NMICONF.LST, one script per line.
The file `ORA_AGENT:SERVICES.ORA` is created during the discovery phase, and will be used to tell the OEM which services the Agent is monitoring.

### Setting the Preferred Credentials

The preferred credentials are supported from the OEM console. To run a job on the HOST database, you MUST supply username/password in the preferred credentials fields in the OEM console. To check that the username/password is valid, login to the HOST node where the Intelligent Agent is running and issue the command:

```
$ SHOW PROCESS/RIGHTS
```

to see that the account is not disabled and that it has the `ORA_AGENT_ID` identifier.

### Oracle Intelligent Agent Startup, Shutdown, and Status Query

This section explains how to startup, shutdown, and status query the Agent.

#### Startup of the Agent

The Agent consists of the following two processes:

- WORK process, which runs as a background detached process
- COMM process, which also runs as a background detached process.

Additionally, a third JOB process is created by the WORK process whenever it is needed to execute jobs.

Use the following command to start the Agent:

```
$ LSNRCTL DBSNMP_START
```

This command creates a detached process with a process name of the form `ORA_AGENTWORK`, which will then spawn a subprocess.

If a nonzero trace level is specified in `SNMP_RW.ORA`, two trace files with the names `DBSNMFPC.TRC` and `DBSNMPW.TRC` will be created in the `ORA_ROOT:[NETWORK.TRACE]` directory.

Whenever a job is executed, a trace file of the form `DBSNMP[_<pid>].TRC` will also be created.
Shutdown of the Agent

Use the following command to shutdown the Agent:

```bash
$ LSNRCTL DBSNMP_STOP
```

*Note:* Use the Oracle7 account to stop or start the Agent.

Status Query of the Agent

Use the following command to verify whether the Agent is running:

```bash
$ LSNRCTL DBSNMP_STATUS
```

Oracle Intelligent Agent Maintenance

Unlike the listener process, the Agent processes are in a continuous loop, polling for incoming connections in each loop. This means that trace information is continuously being generated. Therefore, it is advisable to turn off tracing during normal operation and to turn it on only when a problem is encountered.

Oracle Intelligent Agent Product Capabilities

The following types of jobs are currently supported:

- Executing a host command from the OEM
- Starting up a database or Listener
- Shutting down a database or Listener
- Running a DBA or Tcl script task
- Running a SQL*Plus, Export, Import, or Load task
- Broadcasting a message on the host

The following types of events are currently supported:

- New errors are in the alert file
- Database users have been blocking other processes
- Database is up or down
- A new database connection can be established
- Buffer cache hit ratio is low
• Chained rows exist
• Data dictionary miss ratio is high
• Disk I/O rate is high
• Library cache miss ratio is high
• SQL*Net I/O rate is high
• Value of V$SYSSTAT parameter is too high
• Differences between parameters in V$SYSSTAT are too high
• Response time of user program is too low
• Exceeds datafile limit
• Exceeds lock resource limit
• Exceeds process limit
• Exceeds session limit
• Exceeds user limit
• Archive device is full
• Contiguous free space is too small
• Dump destination device is full
• Segments are reaching maximum
• SQL*Net listener is up or down
• Node is up or down
• CPU utilization is high
• Paging rate is high
• Disk is full

SNMP Support

SNMP is supported by the Intelligent Agent acting as a SNMP subagent. To enable SNMP support, choose \( Y \) in the SNMP support line when configuring the Agent.

The PEER master agent is currently supported. To run the PEER master agent, change the following two files, as follows:
1. Modify the three lines in the PEERAGENT_COM.SAMPLE file that is found in the ORA_NETWORK directory and save it as PEERAGENT.COM in the same directory.

2. Modify the CONFIG_MASTER.SAMPLE file that could be found in the TNS_ADMIN directory, and save it as CONFIG.MASTER in the same directory.

Then connect to Oracle using the SYS account to run the CATSNMP.SQL script from the ORA_AGENT directory, unless it was already run when the Agent was configured.

To use SNMP Support, start the SNMP master agent before starting the TNS Listener and before starting the Intelligent Agent

- To start the PEER master agent, execute the command:
  \$ @ORA_NETWORK:STARTUP_PEERAGENT
- To query the status of the PEER master agent, execute the command:
  \$ @ORA_NETWORK:STATUS_PEERAGENT
- To stop the PEER master agent, execute the command:
  \$ @ORA_NETWORK:SHUTDOWN_PEERAGENT

Additional Information: Refer to the Oracle SNMP Support Reference Guide.
This chapter provides OpenVMS-specific installation information for the current release of Advanced Networking Option (ANO) for Security and Single Sign-On.

Attention: A separate license is required to use ANO.

The topics covered are as follows:
- Documentation set
- Requirements
- Installation
- De-Installation
- Usage notes for the authentication adapters
Requirements

This section details installation requirements for ANO on OpenVMS.

The topics covered in this section are:

- What’s in this Release?
- Media
- Installation Requirements

What’s in this Release?

The Advanced Networking Option for Security and Single Sign–On (ANO) is the new name for the product released earlier under the name Secure Network Services. This release of ANO Alpha OpenVMS supports the following features:

- Encryption (to RSA and DES standards)
- Checksumming (MD5)
- Authentication (SecurID, Kerberos5, and Identix Adapters)

Attention: At this time, there is NO support for SQL*Net/DCE and Native Naming Adapters.

Media

Version 2.3.4 of ANO OpenVMS is available on CD-ROM.

Installation Requirements

This section summarizes all the requirements necessary before installing ANO OpenVMS.
System Requirements

See the *Oracle7 for Alpha OpenVMS Installation Guide* for system hardware and software requirements.

Server Authentication Adapter Requirements

Table 7–2 specifies the software requirements for Authentication Adapters.

<table>
<thead>
<tr>
<th>Adapter</th>
<th>Requirements for ANO</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIT Kerberos5</td>
<td>Kerberos v5.4.2 or higher</td>
</tr>
<tr>
<td></td>
<td>The Kerberos authentication server must be installed on a physically secure machine</td>
</tr>
<tr>
<td>SecurID</td>
<td>ACE/Server v1.2.4 or higher</td>
</tr>
<tr>
<td>Identix</td>
<td>Identix TouchNETII (Encrypt) 1.4</td>
</tr>
</tbody>
</table>

Table 7 – 1 Software Requirements for Authentication Adapters

Note: No additional authentication adapter software is required to relink Oracle products. However, Oracle does not provide an authentication server for Kerberos5, SecurID, or Identix. You must separately install and configure the appropriate authentication server.

Installation

This section describes the steps necessary to install ANO OpenVMS.

The topics covered in this section are:

- Installation Warning
- Installation Tasks

See Also: The *Oracle7 for Alpha OpenVMS Installation Guide* for more information about installing Oracle products using the Installer.

Note: Any reference to ANO in the following pages signifies one or more of the following options while choosing to build NETCONFIG using the Oracle Installer:

- Install ANO encryption
- Install SecurID Authentication Adapter
• Install Kerberos5 Authentication Adapter
• Install Identix Authentication Adapter

Installation Warning

When you install ANO, the Installer automatically relinks only the following Oracle components:

- **NETCONFIG** (lsnrctl, tnslsnr, names, namesctl)
- **RDBMS** (srv, imp, exp, sqlldr, ...)
- **SVRMGR**
- **UTIL**
- **PROGINT**
- **SQLPLUS**
- **OEMAGENT** (if installed)

If you do not wish to relink these components, do not choose the options to install ANO.

To use other Oracle products after installing ANO, you must relink them as a separate operation.

Installation Tasks

- Task 1: Responding to Installer Prompts
- Task 2: Relinking Other Executables
- Task 3: Using with Oracle Names
- Task 4: Manual Steps for the Authentication Adapters
Task 1: Responding to Installer Prompts

The following build option screen is displayed:

<table>
<thead>
<tr>
<th>Option</th>
<th>Current Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. System or Group installation? [S/G]</td>
<td>S</td>
</tr>
<tr>
<td>2. Install DECnet adapter? [Y/N]</td>
<td>N</td>
</tr>
<tr>
<td>3. Install TCP/IP adapter? [Y/N]</td>
<td>N</td>
</tr>
<tr>
<td>4. Build Oracle Names Server? [Y/N]</td>
<td>N</td>
</tr>
<tr>
<td>5. Install ANO encryption? [Y/N]</td>
<td>N</td>
</tr>
<tr>
<td>6. Install SecurID Authentication Adapter? [Y/N]</td>
<td>N</td>
</tr>
<tr>
<td>7. Install Kerberos5 Authentication Adapter? [Y/N]</td>
<td>N</td>
</tr>
<tr>
<td>8. Install Identix Authentication Adapter? [Y/N]</td>
<td>N</td>
</tr>
</tbody>
</table>

Enter (A)LL to select all options.
Enter (E)XIT to exit this menu with selected options.
Enter (Q)UIT to quit this menu with no action.
Enter the number of the option that you want to change:

Options 5, 6, 7, and 8 are related to ANO.

Task 2: Relinking Other Executables

Since ANO is integrated into SQL*Net during installation, it won’t be effective for all applications that use SQL*Net directly. If you have any applications that connect to the database via SQL*Net for which you would like to include Encryption/Checksumming in the connection, you need to relink these applications after installing ANO. This includes the Oracle tools, such as Reports 2.5, Forms 4.5, etc.

The list of products that are automatically relinked during ANO installation is given in the “Installation Warning” section.

Task 3: Using with Oracle Names

The Oracle Names executables are automatically relinked during the ANO build. To use ANO with Oracle Names, modify the file
TNS_ADMIN:NAMES.ORA by adding an entry for the SQLNET.CRYPTO_SEED parameter. You can do this by copying the line that begins with “SQLNET.CRYPTO_SEED=” from your TNS_ADMIN:SQLNET.ORA file into your TNS_ADMIN:NAMES.ORA file.

**Attention:** The complete line must be copied exactly or you will not be able to start the Oracle Names Server using ANO.

**Task 4: Manual Steps for the Authentication Adapters**

In the database server’s local INIT.ORA file, set the following parameters:

```
remote_os_authent = false
os_authent_prefix = ""
```

For SecurID Adapter

The logical ORA_VAR_ACE should point to the directory where the configuration file SDCONF.REC is available. By default, this logical will point to the [NETWORK.ACE] directory under ORA_ROOT. If your configuration file is located somewhere else, modify the definition of ORA_VAR_ACE in ORA_ROOT:[NETCONFIG]SECURID_USER.COM to point to the correct directory. Make sure that the directory is readable by all Oracle Server processes.

For Kerberos5 Adapter

The following file is required on the client side:

- **KRB.CONF** (configuration file that specifies the default realm of the client and maps all known realms to Key Distribution Centers (KDCs))

The following files are required on the server side:

- **KRB.REALMS** (maps hostnames and domains into realms)
- **V5SRVTAB** (contains key that the KDC uses to encrypt a service ticket for the client)

The location of all of the above files **MUST** be specified using corresponding parameters in SQLNET.ORA.
Additionally, the SQL*Net client also creates a credential cache file whose location needs to be specified in SQLNET.ORA on the client side.

The following is an example of the parameters in SQLNET.ORA for an installation that can act as both client and server:

```ora
SQLNET.AUTHENTICATION_KERBEROS5_SERVICE = ORACLE
SQLNET.AUTHENTICATION_SERVICES = (BEQ,KERBEROS5)
SQLNET.KERBEROS5_KEYTAB = DISK:[TST7323.NETWORK.ETC]V5SRVTAB.
SQLNET.KERBEROS5_CONF = DISK:[TST7323.NETWORK.KRB5]KRB.CONF
SQLNET.KERBEROS5_REALMS = DISK:[TST7323.NETWORK.KRB5]KRB.REALMS
SQLNET.KERBEROS5_CC_NAME = DISK:[TST7323.NETWORK.CCACHE]CCFILE.DAT
```

For Identix Adapter

The following is an example of the parameters in SQLNET.ORA for an installation that can act as both client and server when using the Identix Adapter:

```ora
SQLNET.AUTHENTICATION_SERVICES = (BEQ,IDENTIX)
SQLNET.IDENTIX_FINGERPRINT_DATABASE = <Alias for the Identix DB>
SQLNET.IDENTIX_FINGERPRINT_DATABASE_USER = OFM_CLIENT
SQLNET.IDENTIX_FINGERPRINT_DATABASE_PASSWORD = OFM_CLIENT
SQLNET.IDENTIX_FINGERPRINT_METHOD = ORACLE
```

## De–Installation

This section describes the steps necessary to de–install ANO from your system.

The topics covered in this section are:

- De–Installation Warning
- De–Installation Tasks

**Attention:** The de–install process will NOT modify any of the .ORA files under the TNS_ADMIN directory. This means that if your SQLNET.ORA file contained parameters to enable authentication or encryption, they may no longer work after the de–install. Remember to do the same actions on any of the client installs, if necessary, to retain the compatibility.
De–Installation Warning

⚠️ **Warning:** The de–install script does NOT automatically relink any of the executables linked during ANO install. You need to use ORACLEINS to relink all of these executables.

De–Installation Tasks

- Task 1: Preparing Your System
- Task 2: De–install

---

**Task 1: Preparing Your System**

To prepare your system to de–install ANO, do the following:

1. Shut down all running database instances normally.
2. Shut down all SQL*Net listener processes.
3. Log into the Oracle account.
4. Run ORAUSER.COM in your UTIL directory under ORA_ROOT. This will define the symbols and logicals for your oracle installation.

---

**Task 2: De–install**

De–installing ANO does NOT result in automatic relinking of the executables that were linked during ANO install. You need to relink these using ORACLEINS.

1. At the command prompt, type:
   ```bash
   $ ORACLEINS
   ```
2. Choose option 3 to go to the Main Menu.
3. Choose option 1 to go to the “Software Installation and Upgrade Menu”.
4. Choose option 2 “Select Build Configuration Options”. Then select product “NetConfig”. Your previous install options are remembered by ORACLEINS.

The following build option screen is displayed:
Options 5, 6, 7, and 8 are related to ANO. Choose N for the options that you want to de-install.

5. Exit back to the “Software Installation and Upgrade Menu” and choose option 4 to build the selected products. This causes the following products to be be relinked:

- **NETCONFIG** (lsnrctl, tnslsnr, names, namesctl, ...)
- **RDBMS** (srv, imp, exp, sqldlr, ...)
- **SVRMGR**
- **UTIL**
- **PROGINT**
- **SQLPLUS**
- **OEMAGENT** (if installed)

**Note:** De-install does NOT relink any other applications using SQL*Net. All such applications, including for example, Oracle Tools such as Reports 2.5 and Forms 4.5, must be relinked either through ORACLEINS, if possible, or manually.
Usage Notes for the Authentication Adapters

The usage notes are categorized into the following areas.

- General Information
- SecurID
- Kerberos5
- Identix (Biometric)

General Information

Include the following line in your LISTENER.ORA file:

```
SQLNET.AUTHENTICATION_SERVICES=(NONE)
```

The listener should not participate in the authentication service.

It is recommended that you always include BEQ as one of the authentication services in your SQLNET.ORA file. Here is an example:

```
SQLNET.AUTHENTICATION_SERVICES=(BEQ,KERBEROS5)
```

In this way, connections within the Server machine through the default bequeath adapter do not have to go through the authentication. This is especially important during database startups and shutdowns.

SecurID

If you expect excessive delays in your relink to access the ACE server from your client machine, use the following syntax to connect to the database, for example:

```
$ SQLPLUS USERNAME/"<nnnn><pppppp>+<qqqqqq>"@DATABASE
```

where:

- `<nnnn>` is the PIN number of your SecurID card.
- `<pppppp>` and `<qqqqqq>` are two successive codes displayed on the card.

Kerberos5

1. Make sure that the clock skew between the client machine and the machine running the KDC is less than one minute.

2. Oracle client and server processes use the Coordinated Universal Time (UTC) format (time elapsed since 00:00:00 Jan. 1, 1970 in
records). Make sure that your system is set to the correct time zone in terms of deviation from Greenwich Mean Time (GMT). Otherwise you will get the error “Clock skew too great” in your SQL*Net trace file.

3. Make sure that the value of the parameter
   
   SQLNET.AUTHENTICATION_KERBEROS5_SERVICE that you specify in SQLNET.ORA matches exactly, including case, with the value specified in the KDC.

Identix (Biometric)

Make sure that the alias that you are using in the
   SQLNET.IDENTIX_FINGERPRINT_DATABASE parameter is in the TNSNAMES.ORA file on the server side. This alias in the TNSNAMES.ORA file should contain the line:

   (security=(authentication_service=none))
Major Changes from Previous Versions of SQL*Net

This appendix lists the major changes from previous versions of SQL*Net to SQL*Net Version 2.3.4, as follows:

- List of changes
List of Changes

Most changes are specific to OpenVMS customers, except for the first item listed below, which is the most significant one. Generic enhancements have been made to SQL*Net functionality and these are explained in the “Understanding SQL*Net” documentation.

- SQL*Net Version 1 is no longer supported, starting with Oracle7 Version 7.3.2 on all platforms. You should remove any SQL*Net Version 1 aliases from your TNSNAMES.ORA file and replace any database links which use Version 1 connect strings with links that use Version 2 connect descriptors.

  The Network Manager (NETMAN) is no longer supported on server platforms. You must now configure your network with NETMAN on a Windows desktop system.

- Linking in “Single-task” is no longer supported on OpenVMS. All database client programs must use SQL*Net to connect to a server process.

- The DECnet and TCP/IP adapters are no longer provided in separate savesets and are part of the NETCONFIG saveset. They must still, however, be enabled on the NETCONFIG configuration screen.

- SQL*Net 2.3 on OpenVMS is developed and certified using Compaq’s TCP/IP Services for OpenVMS (UCX). TCP/IP protocol stacks from other vendors may work with Oracle, but customers use these products at their own risk. Any TCP/IP problems that can not be reproduced using TCP/IP Services for OpenVMS will simply be referred to the TCP/IP vendor.

- The bequeath adapter now provides the capability to connect to a local database without configuring SQL*Net. Please refer to Chapter 3, “The Protocol Adapters” for more information.

- The process quotas for the listener, dedicated servers, and prespawned servers can now be controlled via logical names. Please refer to Chapter 4, “The TNS Listener” for more information.

A – 2 SQL*Net for OpenVMS Configuration and User’s Guide
Sample ORASRV_NETV2 Command Procedure

This appendix shows the contents of the ORASRV_NETV2_COM.SAMPLE file.

You can find a copy of this file in the TNS_ADMIN directory.

Use this sample file as a base for creating an ORASRV_NETV2 command procedure which contains correct information about your database environment.
NAME: ORASRV_NETV2.COM sample file

USAGE: In (PROGRAM=...) argument in LISTENER.ORA

FUNCTION: Setup environment for RDBMS shadow processes and start them for SQL*Net V2.

MODIFIED:
SRavindh 07/30/97 – Set default to trace directory to create trace files
Rgutherz 07/28/97 – Add authentication support
Rgutherz 06/24/97 – Change process name for PRESPAWN/DEDICATED server proc
Rgutherz 06/24/97 – Grant all privileges needed to server process
DSchwab 08/03/96 – rewritten (using suggestions from DHayter)

This file serves as a template for the actual file that will be created on a per instance basis to setup environment for, and startup RDBMS shadow processes for SQL*Net V2 connections.

Name of the actual .COM file will be specified in (PROGRAM='...') name-value pair in the SID entry of LISTENER.ORA.

To create ORASRV_NETV2.COM files for different databases, edit the following lines to substitute the SID of the database and its administration directory.

ora_db = "database administration directory"
sid = "sid name"
dbname = "database name"

define/job/nolog ora_sid 'sid'
lmn = "ORA_RDBMS_LNM_"+sid

When a database is started up, Server Manager creates and populates a logical name table ORA_RDBMS_LNM_<sid>. If this logical name table doesn’t (yet) exist then pick up the required logical names from the ORAUSER_<dbname>.COM command procedure. This should only be the case if the database has not yet been started.

If the logical name table DOES exist, then it should contain all of the logicals required for the server process to access message files etc.

lnmx = f$strlnm(lmn,"LNMSYSTEM_DIRECTORY","TABLE")
if lnxm
then
  ora_rdbms = f$strlnm("ORA_RDBMS",lnm)
  if ora_rdbms .eqs. "" then lnxm = "FALSE"
endif
$ if lnm
$ then
$   if f$trnlnm("ORA_SYSTEM", lnm) .eqs. "" then define ora_system 'ora_rdbms'
$   if f$trnlnm("ORA_PLSQL", lnm) .eqs. "" then define ora_plsql 'ora_rdbms'
$   lnm = lnm + ","
$ else
$   v1 = f$verify(0)  ! We don’t want to see ORAUSER stuff
$   @'ora_db'orauser_'dbname'  ! Pick up the logicals this way
$   v2 = f$verify(v1)  ! Switch VERIFY back on if it was originally on
$   lnm = ""  ! Blank out logical name table
$ endif
$!
$ old_priv = f$setprv("SYSLCK,PRMMBX,LOG_IO,TMPMBX,SYSNAM,DETACH,CMKRNL,WORLD")
$!
$ orasrv := $ORA_RDBMS:SRV.EXE
$!
$! Form name of our logical name table
$!
$ pid = f$getjpi("", "PID")
$ tab = "TNS_" + pid
$ define/table=lnm$process_directory lnm$file_dev –
   'lnm''tab',lnm$process,lnm$job,lnm$group,lnm$system
$ arg1 = f$trnlnm("arg1")
$ arg2 = f$trnlnm("arg2")
$ arg3 = f$trnlnm("arg3")
$ arg4 = f$trnlnm("arg4")
$ arg5 = f$trnlnm("arg5")
$ arg6 = f$trnlnm("arg6")
$ arg7 = f$trnlnm("arg7")
$ arg8 = f$trnlnm("arg8")
$!
$!
$! Fix for bug 636327
$ rmbx = "ORA_NTP_IR_" + pid
$ rmbxdev = "''f$trnlnm(rmbx)''"
$ define 'rmbx' 'rmbxdev'
$ wmbx = "ORA_NTP_IW_" + pid
$ wmbxdev = "''f$trnlnm(wmbx)''"
$ define 'wmbx' 'wmbxdev'
$ tnsdev = f$parse("''f$trnlnm(""TNS_ADMIN"")'','"DEVICE","NO_CONCEAL")
$ tnsdir = f$parse("''f$trnlnm(""TNS_ADMIN"")'','"DIRECTORY")
$ define TNS_ADMIN 'tnsdev''tnsdir'
$ netdev = f$parse("''f$trnlnm(""ORA_NETWORK"")'','"DEVICE","NO_CONCEAL")
$ netdir = f$parse("''f$trnlnm(""ORA_NETWORK"")'','"DIRECTORY")
$ define ORA_NETWORK 'netdev''netdir'
$!
$! Execute the Authentication command scripts, if present
$ oraroot = f$trnlnm("ORA_NETWORK") - "NETWORK"
$ if f$search("''ORAROOT'netconfig]SECURID_USER.COM"') .nes. "" then -
   @'ORAROOT'netconfig]SECURID_USER.COM
$ if f$search("''ORAROOT'netconfig]KERBEROS_USER.COM"') .nes. "" then -
@'ORAROOT'netconfig\KERBEROS_USER.COM

$!
$!
$! Check/Change process name (Rgutherz)
$! Check for prespawn/dedicated server process
$ len = f$length(arg1)
$ if f$locate("prespawn",arg1) .ne. len
$ then
$ type = "P"
$ else
$ type = "C"
$ endif
$ sname = f$edit(sid,"upcase")
$!
$ i = 1
$retry:
$ if i .gt. 100
$ then
$ write sys$output "Error changing process name after 100 times"
$ exit
$ endif
$ on error then goto retry
$ time = f$time()
$ time = f$cvttime(time)
$ time = f$extract(17,5,time)
$ time = time - "."
$ i = i + 1
$ set process/name="ORA_''sname'''type'''time"
$ on error then goto CLEANUP
$!
$! Set default to trace directory
$ oranetwork = f$strnlm("ORA_NETWORK") - "]"
$ set default 'oranetwork'.TRACE]
$!
$ if arg1 .eqs. ""
$ then
$ orasrv "(LOCAL=NO)"
$ else
$ orasrv 'arg1' 'arg2' 'arg3' 'arg4' 'arg5' 'arg6' 'arg7' 'arg8'
$ endif
$!
$ CLEANUP:
$ sts = $status
$ severity = f$message(sts,"severity")
$!
$ if severity .eqs. "%E" .or. severity .eqs. "%F" .or. severity .eqs. "%W"
$ then
$! Check if DB is up. If not then don’t create trace file
$ db_up = f$strnlm("ORA_RDBMS_LNM_''sid'"),"LNMSYSTEM_DIRECTORY",,"TABLE")
$ if .not. db_up then goto CLEANUP2
$ status = sts
$ trace_dir = ora_db - "]" + ".trace]
$ open/write trace_file 'trace_dir/orasrv_netv2_'pid'.trc
$ write trace_file "This trace file was generated by ORASRV_NETV2.COM"
$ write trace_file "exit with STATUS='status'"
$ msg = f$message("'status'")
$ write trace_file "'msg'"
$ close trace_file
$ endif
$!
$CLEANUP2:
$ on error then continue
$ deassign/user/table=lnm$system_directory 'tab'
$ exit 'sts'
APPENDIX

Sample ORASRV_BEQ_<sid> Command Procedure

This appendix shows the contents of the ORASRV_BEQ_COM.SAMPLE file. You can find a copy of this file in the TNS_ADMIN directory.
NAME: ORASRV_BEQ_<sid>.COM template

FUNCTION: Setup environment for RDBMS shadow processes and start it for
SQL*Net V2 bequeath adapter.

MODIFIED:
Rguthertz 05/01/98 – Fix for bug 636327
Rguthertz 04/18/97 – Mods for BEQ listener
DSchwab 07/23/95 – Mods so that this will run in spawned process
DHayter 07/19/96 – Avoid running ORAUSER_<dbname>.COM if possible
DSchwab 07/16/96 – Created

This file serves as a template for the actual file that will be created on a
per instance basis to setup environment for, and startup RDBMS shadow
processes for SQL*Net V2 bequeath connections. It must exist as
ORA_ROOT:[NETWORK.ADMIN]ORASRV_BEQ_<sid>.COM.

ON ERROR THEN GOTO CLEANUP

Edit the following lines to substitute the SID of this database
and its administration directory:

Substitute <sid> with the actual sid of the database
sid    = "<sid>"                ! e.g. PROD

Substitute <dbname> with the name of the database
dbname = "<dbname>"            ! e.g. PRODDB

Substitute <ora_db> with the location of the database admin. directory
ora_db = "<ora_db>"            ! e.g. DKB300:[ORACLE7.DB_PROD]

define ora_sid 'sid'        ! Need the ORA_SID logical
lnm = "ORA_RDBMS_LNM_''sid''"   ! The database’s own logical name table

If the logical name table doesn’t (yet) exist then pick up the required
logical names from the ORAUSER_<dbname>.COM command procedure.
This should only be the case if the database has not yet been started.

If the logical name table DOES exist, then it should contain all of the
logicals required for the server process to access message files etc.

lnmx = f$trnlnm(lnm,"LNM$SYSTEM_DIRECTORY",,,"TABLE")
if lnmx ! If ORA_RDBMS_LNM_<sid> exists...
then
ora_rdbms = f$trnlnm("ORA_RDBMS",lnmx) ! If ORA_RDBMS isn’t there...
if ora_rdbms .eqs. "" then lnmx = "FALSE" ! ....pretend the table isn’t
endif
$!
If the ORA_RDBMS_LNM_<$sid> logical name table exists, and has ORA_RDBMS
defined within it, then we should be safe in assuming that it will have
all the other required logicals. ORA_SYSTEM and ORA_PLSQL may be
exceptions to this before Oracle7 Server release 7.3.2.2.0 so we’ll
make SURE that they get defined somewhere.
$!
$!
If we get to this point then we are forced to use ORAUSER_<$dbname> .COM
$!
to define the necessary logicals for us - but this may be rather slow.
$!
$v1 = $fverify(0) ! If logging output we don’t want to see ORAUSER stuff
$v2 = $fverify($v1) ! Switch VERIFY back on if it was originally on
$!
! Blank out logical name table as it doesn’t yet exist
$ endif
$!
$ orasrv := $ORA_RDBMS:SRV.EXE
$!
$!
$ Form name of our logical name table
$!
$ pid = $f$getjpi("", "PID")
$ tab = "TNS_" + $pid
$!
$ define/table=$lnm$process_directory $lnm$file_dev -
	'tab','$lnm' $lnm$process,$lnm$job,$lnm$group,$lnm$system
$!
$!

$!
Fix for bug 636327
$ rmbx = "ORA_NTP_IR_" + $pid
$ rmbxdev = "$fstrnlmn(rmbx)"
$ define 'rmbx' 'rmbxdev'
$ wmbx = "ORA_NTP_IW_" + $pid
$ wmbxdev = "$fstrlnm(wmbx)"
$ define 'wmbx' 'wmbxdev'
$ tnsdev = $fparse(""fstrlnm(""TNS_ADMIN"")"","DEVICE","NO_CONCEAL")
$ tnsdir = $fparse(""fstrlnm(""TNS_ADMIN"")"","DIRECTORY")
$ define TNS_ADMIN 'tnsdev''tnsdir'
$ netdev = $fparse(""fstrlnm(""ORA_NETWORK"")"","DEVICE","NO_CONCEAL")
$ netdir = $fparse(""fstrlnm(""ORA_NETWORK"")"","DIRECTORY")
$ define ORA_NETWORK 'netdev''netdir'
$! Execute the Authentication command scripts, if present
$ oraroot = fstrlnm(""ORA_NETWORK"") - "NETWORK"
$ if f$search(""ORAROOT''netconfig\]SECURID_USER.COM"") .nes. "" then -
  @'ORAROOT''netconfig\]SECURID_USER.COM'
$ if f$search(""ORAROOT''netconfig\]KERBEROS_USER.COM"") .nes. "" then -
  @'ORAROOT''netconfig\]KERBEROS_USER.COM'
$ if f$search(""ORAROOT''netconfig\]KERBEROS_USER.COM"") .nes. "" then -
  define SYS$SCRATCH 'ORAROOT''NETWORK.KRB5]
$!
$!
$!Change process name (Rgutherz)
$ sname = $fedit(sid,"upcase")
$ i = 1
$retry:
$ if i .gt. 100
$ then
  write sys$output "Error changing process name after 100 times"
  $ exit
$ endf
$ on error then goto retry
$ time = $ftime()
$ time = $cvtime(time)
$ time = $extract(17,5,time)
$ time = time - "."
$ i = i + 1
$ set process/name="ORA_''sname'B''time'"
$ on error then goto CLEANUP
$!
$ old_priv =
$fsetprv("SYSCLK,SYSPRV,PRMBX,LOG_IO,TMPMBX,SYSSAM,DETACH,CMKRNL,WORLD")
$ if arg1 .eqs. "" then
  $ orasrv 
  
$ else
  $ orasrv 'arg1' 'arg2' 'arg3' 'arg4' 'arg5' 'arg6' 'arg7' 'arg8'
$ endif
$!
$CLEANUP:
$ sts = $status
$ severity = $f$message(sts,"severity")
$!
$ if severity .eqs. "%E" .or. severity .eqs. "%F" .or. severity .eqs. "%W"
$ then
$! Check if DB is up. If not then don’t create trace file
$ db_up = f$trnlnm(“ORA_RDBMS_LNM_’’sid’’,”“LNM$SYSTEM_DIRECTORY”,”,”“TABLE”)
$ if .not. db_up then goto CLEANUP2
$ status = sts
$ trace_dir = ora_db - ”}" + ".trace}
$ open/write trace_file ’trace_dir’orarsrv_beq_’pid’.trc
$ write trace_file ”This trace file was generated by ORASRV_BEQ_’’sid’’.COM”
$ write trace_file ”exit with STATUS = ””status””
$ msg = f$message(””status””)
$ write trace_file ””msg””
$ close trace_file
$ endif
$!
$CLEANUP2:
$ on error then continue
$ deassign/user/table=lnm$system_directory ’tab’
$ exit ’sts’
This appendix describes how to configure SQL*Net for easy operation in OpenVMS Clusters/shared disk configurations. The topics are:

- Background/advantages of OpenVMS Clusters
- Standard configuration
- Clustered configuration
- SQL*Net infrastructure
Background/Advantages of OpenVMS Clusters

OpenVMS systems that share resources in a clustered configuration provide many advantages for the data center manager and user. A clustered configuration provides a useful resilience of service and can considerably reduce start-up costs through sharing peripheral resources.

In a clustered configuration, sharing resources is a prime consideration from a system management perspective. Sharing configuration files required by SQL*Net Version 2 is definitely a prime consideration. The configuration files are not initially set up to provide this common file facility, but with some simple coding and management techniques, the single directory, single file facility can be arranged.

Standard Configuration

The standard configuration has a minimum of three files in a directory, and these files are accessed via the logical name TNS_ADMIN. The three files are LISTENER.ORA, SQLNET.ORA and TNSNAMES.ORA, and they contain the following information:

- The LISTENER.ORA file contains details of the listener(s) configuration and the services available in addition to the files required to access them.
- The SQLNET.ORA file in a standard configuration contains information used to set the level of logging and tracing for the client and server.
- The TNSNAMES.ORA file contains the codes for the services available on the network and the routing information to establish contact.

Normally these files are concerned with only a single node and the default configuration is adequate. However, if a multiple node configuration is required, the directory containing the files must be duplicated on each of the nodes and a node–specific TNS–ADMIN location must be set up. This overhead can be considerable for the system or network managers, because configuration changes need to be repeated many times.

The Oracle Name Server provides the centralized addressing facility required to ease the management of complex configurations. If the Oracle Name Server is not going to be installed, changes can be made to the configuration files to ease the administration burden.
Clustered Configuration

The clustered configuration needs to overcome certain naming issues that prevent the listeners from being started on multiple nodes without introducing node–specific elements into the configuration. The changes are minor and can be greatly automated. The major problem with multiple listeners using the same file is that the logging files clash.

The second listener to be started then fails, because the file already exists and the first listener process holds it open. The solution is to configure the system so that the files do not clash. The next section, “SQL*Net Infrastructure,” explains the easiest way to configure SQL*Net files so that the files do not clash.

SQL*Net Infrastructure

OpenVMS clustered systems can be configured to provide identical services with various means of “load balancing” the user community, usually by the use of LAT services that have one or more systems in them. The relative LAT and OpenVMS systems negotiate which one is least active and direct the user accordingly.

The listeners can be set up to access the same databases on a cluster–wide basis in only one of the following two ways:

- The Oracle Parallel Server is installed on the OpenVMS Cluster, or
- The system is accessing some third–party database via the Transparent Gateway Products that support clustered access.

The advantage of this duplicate system is that “load–balancing” can be achieved using DECnet and the OpenVMS cluster alias node facility. The steps to make the parallel/clustered access will be mentioned during the set–up details.

To configure the multiple listeners, set the LISTENER.ORA file format to allow the multiple listeners with independent logging and tracing file facilities to start.

Starting multiple listeners from the same file is more manageable if a sensible standard naming methodology can be determined first. The listeners have been named after the nodes upon which they are active. For ease of management, a DCL command file should act as an automatic listener start and stop facility to distinguish the node being acted upon and to relate this to the listener in question.
The following commands are an example associated with such a facility:

$ NODE_ID = "''F$GETSYI("NODENAME")"
$ DEFINE ORA_SID 'NODE_ID'
$ LSNRCTL START LSNR'NODE_ID'

These commands invoke the listener process LSNR<nnnnnn>
where:
<nnnnnn> is the node name of the member of the OpenVMS Cluster.

The efficient way to submit this command procedure is to submit it to a batch queue on each of the nodes to run under the Oracle7 username.

Internally, the LISTENER.ORA file should be configured as follows:

```ora
LSNR<nnnnnn> = (ADDRESS_LIST =
  (ADDRESS =
    (PROTOCOL = DECNET)
    (NODE = <nnnnnn>)
    (OBJECT = LSNR_COMMON)
  )
)
STARTUP_WAIT_TIME_LSNR<nnnnnn> = 0
CONNECT_TIMEOUT_LSNR<nnnnnn> = 10

LSNR<mmmmmm> = (ADDRESS_LIST =
  (ADDRESS =
    (PROTOCOL = DECNET)
    (NODE = <mmmmmm>)
    (OBJECT = LSNR_COMMON)
  )
)
STARTUP_WAIT_TIME_LSNR<mmmmmm> = 0
CONNECT_TIMEOUT_LSNR<mmmmmm> = 10

SID_LIST_LSNR<nnnnnn> =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = SID1)
      (PROGRAM = DISK:[DIR]ORACLE_SID1_NETV2.COM)
    )
    (SID_DESC =
      (SID_NAME = SID2)
      (PROGRAM = 'DISK:[DIR]ORACLE_SID2_NETV2.COM')
  )
```
D – 5SQL*Net Configuration for OpenVMS Clusters

SID_LIST_LSNR<mmmmmm> =
  (SID_LIST =
   (SID_DESC =
     (SID_NAME = SID1)
     (PROGRAM = 'DISK:[DIR]ORACLE_SID1_NETV2.COM')
   )
   (SID_DESC =
     (SID_NAME = SID2)
     (PROGRAM = 'DISK:[DIR]ORACLE_SID2_NETV2.COM')
   )
   ...
  )

#LOG_LEVEL may be 0, ADMIN OR DEV
LOG_LEVEL_LSNR<nnnnnn> = 0
LOG_DIRECTORY_LSNR<nnnnnn> = DISK:[LOG_DIR]
LOG_FILE_LSNR<nnnnnn> = LSNR<nnnnnn>

#TRACE_LEVEL may be 0, ADMIN OR DEV
TRACE_LEVEL_LSNR<nnnnnn> = 0
TRACE_DIRECTORY_LSNR<nnnnnn> = DISK:[TRACE_DIR]
TRACE_FILE_LSNR<nnnnnn> = LSNR<nnnnnn>

LOG_LEVEL_LSNR<nnnnnn> = 0
LOG_DIRECTORY_LSNR<nnnnnn> = DISK:[LOG_DIR]
LOG_FILE_LSNR<nnnnnn> = LSNR<nnnnnn>

TRACE_LEVEL_LSNR<nnnnnn> = 0
TRACE_DIRECTORY_LSNR<nnnnnn> = DISK:[TRACE_DIR]
TRACE_FILE_LSNR<nnnnnn> = LSNR<nnnnnn>

where:
<nnnnnn> and <mmmmmmm> are indicators of the specific node, such as ALPHA1 and VAX1.

If you are using the Transparent Gateway to Rdb, the Oracle Parallel Server, or something similar that accepts connections from any node in a cluster, the initial portion of the LISTENER.ORA can be adapted. This allows the use of the DECnet alias feature when connection is made to an OpenVMS cluster node address and the load balancing happens automatically. The format requires that the listener object be identical, because there is no guarantee through which node the user will connect. The modified file portion is:

```
LSNR<nnnnnn> = (ADDRESS_LIST =
  (ADDRESS =
    (PROTOCOL = DECNET)
    (NODE = <nnnnnn>)
    (OBJECT = LSNR_COMMON) <––––––|
   )                               |
  )                                 |
  STARTUP_WAIT_TIME_LSNR<nnnnnn>  = 0     |
  CONNECT_TIMEOUT_LSNR<nnnnnn>  = 10    |
LSNR<mmmmmmm> = (ADDRESS_LIST =
  (ADDRESS =
    (PROTOCOL = DECNET)
    (NODE = <mmmmmmm>)
    (OBJECT = LSNR_COMMON)<––Changed to
     ) load balance
  ) using VAX
    ) cluster alias
```
Index

A
Advanced Networking Option. See ANO
ANO, 7–1
  authentication adapters, 7–6, 7–10
de-installation, 7–7, 7–8
documentation set, 7–2
features, 7–2
installation, 7–3
installer prompts, 7–5
media, 7–2
preparing for de-install, 7–8
relinking executables, 7–5
requirements, 7–2
using with Oracle Names, 7–5
authentication adapters
  general information, 7–10
  Kerberos5, 7–10
  manual steps, 7–6
  requirements with ANO, 7–3
  SecurID, 7–10

B
Bequeath Adapter, 3–6
Bequeath Listener, 3–7

C
Centralized configuration, 1–3
Client–server configuration, 1–3
Communications protocol, definition of, 1–6
Communities, protocol–dependent groups, 1–7
Configuration, 1–2
  centralized, 1–3
  client–server, 1–3
  distributed, 1–3
  configuration files, 2–7

D
DECnet Adapter, 3–3
Distributed database configuration, 1–3

E
Ethernet, 1–2

I
Identix, 7–7

K
Kerberos5, 7–3, 7–6, 7–10

L
Location–specific parameters, for Network Manager, 2–4
M
Mailbox Adapter, 3 – 2
Multiple protocols, on single machine, 1 – 7

N
native naming adapters, 7 – 2
Network Manager
  location–specific parameters, 2 – 4
  OpenVMS–specific information, 2 – 3
  Pre–Configuration Tasks, 2 – 2

O
Oracle Names, using with Advanced Networking Option, 7 – 5
Oracle Protocol Adapters, role in TNS network, 1 – 6
Oracle Worldwide Technical Support, how to contact, vii
ORASRV, command procedure, 2 – 4, B – 1
ORASRV_NETV2.COM.SAMPLE file, B – 1

P
Protocol Adapters
  See also Oracle Protocol Adapters
  Bequeath Adapter, 3 – 6
  DE/Cnet Adapter, 3 – 3
  Mailbox Adapter, 3 – 2
  TCP/IP Adapter, 3 – 5

Protocol adapters
  definition of, 1 – 6
  role of, 1 – 6
Protocol API differences, TNS implementation, 1 – 7
Protocol implementations, 1 – 7

R
requirements
  ANO system, 7 – 3
  authentication adapter with ANO, 7 – 3

S
SecurID, 7 – 3, 7 – 6, 7 – 10
SQL*Net/DCE, 7 – 2

T
TCP/IP Adapter, 3 – 5
TNS, Oracle network layer, 1 – 6
TNS client, community member, common cases, 1 – 7
TNS, Adapters, and Protocols, Protocol Adapters, 1 – 7
Transparent Network Substrate. See TNS

W
warning, ANO de–installation, 7 – 8
Reader’s Comment Form

SQL*Net for OpenVMS Configuration and User’s Guide
Part No. A69347–01

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

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

If you find any errors or have any other suggestions for improvement, please indicate the topic, chapter, and page number below:

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

Please send your comments to:
Compaq Products Division
Documentation Manager
Oracle Corporation
500 Oracle Parkway, Box 659104
Redwood City, CA 94065
Phone: (650) 506–7000    Fax: (650) 506–7361

If you would like a reply, please give your name, address, and telephone number below:

________________________________________________________________________
________________________________________________________________________

Thank you for helping us improve our documentation.