

Oracle9*i* Net Services

Administrator's Guide

Release 1 (9.0.1)

June 2001

Part No. A90154-01

ORACLE®

Part No. A90154-01

Copyright © 2001, Oracle Corporation. All rights reserved.

Primary Author: Deborah Steiner

Graphic Artist: Valarie Moore

Contributors: Henry Abrecht, Robert Achacoso, Jesse Anton, Mark Bauer, Pierre Baudin, David Chen, Howard Cohen, Toby Close, Stanley Guan, Harvey Eneman, Stella Kister, Shuvayu Kanjila, Cynthia Kibbe, Scot McKinley, Ed Miner, Peter Povinec, Richard Smith, Juliet Tran, and Norman Woo

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 or anyone licensing or using the programs on behalf of the U.S. 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 shall 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 the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle, SQL*Net, and SQL*Plus are registered trademarks of Oracle Corporation. Net8, Oracle9*i*, Oracle8*i*, and Oracle8 are trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	xix
Preface.....	xxi
Audience	xxii
Organization.....	xxiii
Related Documentation	xxv
Conventions.....	xxvi
Documentation Accessibility	xxxi
What's New in Oracle Net Services?	xxxiii
Oracle9 <i>i</i> Release 9.0 New Features in Oracle Net Services.....	xxxiv
Oracle8 <i>i</i> New Features in Oracle Net Services	xxxix
Part I Getting Started with Oracle Net Services	
1 Networking Challenges in the Internet Age	
What Is the Big Picture for Networking?	1-2
Oracle's Solution to Networking Issues.....	1-3
Connectivity	1-3
Manageability.....	1-9
Internet and Intranet Scalability	1-12
Network Security.....	1-19

A Suite of Networking Components.....	1-27
Oracle Net Services.....	1-27
Oracle Advanced Security	1-32

2 Connectivity Concepts

Database Service and Database Instance Identification.....	2-2
Database Services.....	2-2
Database Instances.....	2-4
Service Accessibility.....	2-6
Enhanced Service Accessibility with Multiple Listeners	2-8
Service Handlers	2-9
Naming.....	2-15

3 Configuration Management Concepts

Configuration Models	3-2
Localized Configuration File Support.....	3-2
Directory Server Support.....	3-4
Directory Naming Overview	3-4
Naming Configuration Storage in a Directory Server.....	3-6
Directory Entries	3-8
Adding or Modifying Entries in the Directory Server	3-9
Client Connections Using Directory Naming.....	3-10
Oracle Net Configuration and Directory Server Design.....	3-13
Oracle Names Support.....	3-18
Oracle Names Overview.....	3-19
Administrative Regions	3-21
Domains	3-21
Oracle Names As Data Repository.....	3-22
Data Stored in an Oracle Names Server	3-23
Organization of Network Components.....	3-25
Multiple Regions to Decentralize Administrative Responsibilities	3-27
Differences Between Versions of Oracle Names	3-30
About Discovery	3-33

4 Architecture of Oracle Net Services

Oracle Net Stack Communication Architecture	4-2
Stack Communication for Client/Server Application Connections	4-2
Stack Communication for Java Application Connections	4-8
Stack Communication for Web Client Connections	4-9
Listener Architecture.....	4-10
Database Server Process Architecture	4-14
Shared Server Processes.....	4-14
Dedicated Server Processes.....	4-16
Oracle Connection Manager Architecture	4-17
A Complete Architecture.....	4-20

5 Configuration and Administration Tools Overview

Oracle Net Manager	5-2
Starting Oracle Net Manager	5-3
Navigating Oracle Net Manager	5-4
Toolbar	5-7
Menus	5-8
Oracle Net Manager Wizards	5-10
Oracle Net Configuration Assistant	5-14
Oracle Net Control Utilities.....	5-16
Listener Control Utility.....	5-16
Oracle Connection Manager Control Utility	5-17
Oracle Names Control Utility	5-18
Configuration and Administration Tasks at a Glance	5-19

6 Quick Start to Oracle Net Connections

Prerequisites to Establishing Connectivity	6-2
Task 1: Confirm Network Availability.....	6-3
Task 2: Start the Listener and the Database Server	6-4
Task 3: Configure the Client to Use a Net Service Name	6-5
Task 4: Connect to the Database.....	6-7

Part II Configuration and Administration of Oracle Net Services

7 Planning the Network

Deploying a Network Inside an Organization.....	7-2
Scalability.....	7-3
Availability	7-5
Naming Methods.....	7-6
JDBC Drivers	7-6
Security.....	7-6
Tuning and Performance	7-7
Deploying a Network for the Internet.....	7-8
Scalability	7-9
Availability	7-9
Naming Methods	7-9
JDBC Drivers	7-9
Security.....	7-9
Naming Considerations	7-10
Performance Considerations.....	7-12
Listener Queue Size	7-12
Session Data Unit Size for Data Transfer Optimization.....	7-13
Persistent Buffer Flushing for TCP/IP	7-13
Planning Summary	7-14

8 Setting Up Directory Server Usage

Directory Configuration Overview.....	8-2
Configuring Directory Usage During Installation	8-2
Directory Usage Configuration During a Custom Installation on the Server	8-2
Directory Usage Configuration During a Client Installation.....	8-4
Configuring Directory Usage After Installation	8-5
Adding Users to and Removing Users from the OracleNetAdmins Group.....	8-9

9 Configuring Naming Methods

Naming Method Configuration Overview	9-2
About Connect Descriptors.....	9-2
Naming Methods.....	9-4
Configuring the Local Naming Method	9-5
Configuring the tnsnames.ora File During Installation.....	9-6
Configuring the tnsnames.ora File After Installation.....	9-6
Configuring the Directory Naming Method.....	9-12
Directory Naming Method Configuration Steps	9-13
Modifying Connectivity Information for Database Service Entries.....	9-18
Configuring the Oracle Names Method	9-19
Task 1: Consider Oracle Names Options	9-20
Task 2: Install Necessary Components.....	9-20
Task 3: Create an Oracle Names Server	9-21
Task 4: Configure Clients and Database Servers To Use Oracle Names Servers.....	9-34
Task 5: (Optional) Configure Client Caches	9-38
Task 6: Configure the Listener.....	9-41
Task 7: Register Data with the Oracle Names Server.....	9-42
Task 8: Delegate Domains to Delegated Administrative Regions.....	9-50
Task 9: Specify Domain Hints to Forward Requests to Remote Oracle Names Servers..	9-51
Task 10: Connect to the Database.....	9-53
Removing an Oracle Names Server	9-54
Configuring the Host Naming Method	9-56
Task 1: Configure the Listener with the Global Database Name	9-57
Task 2: Configure HOSTNAME as the First Naming Method	9-58
Task 3: Set Up Host Name Resolution Environment	9-58
Task 4: Connect to the Database.....	9-59
Configuring External Naming Methods.....	9-60
Configuring NIS.....	9-60

10 Exporting Naming Data to a Directory Server

Exporting Net Service Names from a tnsnames.ora File	10-2
Task 1: Create Structure in Directory Server	10-3
Task 2: Create Oracle Contexts	10-4
Task 3: Configure Directory Server Usage.....	10-4
Task 4: Export Objects To a Directory Server	10-5
Exporting Network Objects from an Oracle Names Server.....	10-6
Release 8.1.6 or Later Clients.....	10-6
Release 8.1.5 or Previous Clients	10-6
Exporting Data to a Directory Server.....	10-7
Configuring Oracle Names LDAP Proxy Servers.....	10-15
Considerations for Dissimilar DIT Structures	10-23

11 Configuring Profiles

Profile Configuration Overview	11-2
Profile Configuration During Installation.....	11-2
Configuring Client Attributes for Names Resolution	11-3
Configuring a Default Domain for Clients	11-3
Prioritizing Naming Methods	11-5
Routing Connection Requests.....	11-7
Configuring Database Access Control	11-8
Configuring Advanced Profile Information.....	11-9
Configuring Preferred Oracle Names Servers	11-11
Configuring External Naming Methods	11-12
Configuring Oracle Advanced Security.....	11-13

12 Configuring and Administering the Listener

Listener Configuration Overview	12-2
Listener Configuration During Installation.....	12-4
Customizing Listener Configuration.....	12-6
Configuring Listening Protocol Addresses.....	12-6
Configuring Protocol Addresses for Oracle9i JVM	12-8
Handling Large Volumes of Concurrent Connection Requests	12-9

Configuring Static Service Information.....	12-10
Configuring Password Authentication for the Listener	12-13
Configuring Service Registration	12-14
Listener Administration	12-19
Starting and Stopping the Listener	12-20
Monitoring Runtime Behavior.....	12-20
Monitoring Log Files.....	12-25
13 Configuring Oracle Connection Manager	
Oracle Connection Manager Configuration Overview	13-2
Configuring Oracle Connection Manager.....	13-2
Configuring the Oracle Connection Manager Computer.....	13-3
Configuring Clients for Oracle Connection Manager	13-5
Configuring the Oracle Database Server for Oracle Connection Manager.....	13-12
Enabling Oracle Connection Manager Features.....	13-12
Enabling Session Multiplexing.....	13-13
Enabling Access Control.....	13-14
Enabling Protocol Conversion Support.....	13-15
14 Configuring Shared Server	
Configuring Shared Server with the DISPATCHERS Parameter.....	14-2
Setting the Initial Number of Dispatchers	14-6
Enabling Connection Pooling	14-8
Allocating Resources.....	14-9
Using Shared Server on Clients	14-9
Overriding Shared Server on Clients	14-10
15 Enabling Advanced Features for Oracle Net Services	
Configuring Advanced Network Address and Connect Data Information	15-2
Creating a List of Listener Protocol Addresses.....	15-2
Configuring Address List Parameters.....	15-4
Configuring Advanced Connect Data Parameters	15-7
Configuring Connection Load Balancing.....	15-10
Example: Connection Load Balancing for Shared Server Configuration.....	15-12
Example: Connection Load Balancing for Dedicated Server Configuration	15-15

Configuring Transparent Application Failover	15-18
What TAF Restores.....	15-18
TAF Database Configurations.....	15-20
FAILOVER_MODE Parameters.....	15-21
TAF Implementation.....	15-22
Specifying the Instance Role for Primary and Secondary Instance Configurations	15-25
Configuring Session Data Unit	15-28
Client Configuration.....	15-28
Database Server Configuration.....	15-29
Configuring the VI Protocol for Application Web Server and Oracle Database Server Communication	15-30
Prerequisites to Using VI Protocol Support.....	15-30
Configuring a VI Protocol Address.....	15-31
Tuning VI	15-33
Configuring Connections to Non-Oracle Database Services	15-36
Configuring Oracle Net for External Procedures.....	15-36
Configuring Oracle Net for Oracle Heterogeneous Services	15-41
Configuring Oracle Net for an Oracle Rdb Database	15-44

Part III Testing and Troubleshooting Oracle Net Services

16 Establishing a Connection and Testing the Network

Connecting to a Database	16-2
Starting Oracle Net Services Components	16-2
Entering a Connect String.....	16-8
Initiating Connections.....	16-13
Testing the Network	16-15
Testing a Listener	16-16
Testing Oracle Connection Manager	16-16
Testing Configuration on the Database Server.....	16-16
Testing Network Connectivity from the Client.....	16-17
Testing an Oracle Names Server	16-23
Testing Network Objects with Oracle Net Manager or Oracle Names Control Utility	16-24

17 Troubleshooting Oracle Net Services

Diagnosing Oracle Net	17-2
Server Diagnostics	17-3
Client Diagnostics.....	17-4
Resolving the Most Common Error Messages for Oracle Net Services.....	17-7
Troubleshooting Directory Naming Errors	17-15
Oracle Names LDAP Proxy Server Error Reporting	17-16
Troubleshooting Tips from the Field for Oracle Net Services	17-18
Questions to Ask When Troubleshooting Oracle Net Services	17-19
Troubleshooting Network Problems Using Log and Trace Files.....	17-20
Logging Error Information for Oracle Net Services.....	17-20
Oracle Net Error Stacks.....	17-21
Oracle Net Services Log File Names.....	17-24
Setting Oracle Net Services Log Parameters	17-24
Setting Oracle Net Log Settings During Runtime of Control Utilities.....	17-28
Using Oracle Net Log Files	17-29
Analyzing Listener Log Files	17-30
Analyzing Oracle Connection Manager Logs	17-35
Oracle Net Tracing Error Information.....	17-39
Oracle Net Trace File Names	17-40
Setting Oracle Net Trace Parameters	17-40
Setting Oracle Net Trace Settings During Runtime of Control Utilities.....	17-48
Evaluating Oracle Net Traces	17-49
Contacting Oracle Support Services	17-54

Glossary

Index

List of Figures

1–1	Typical E-Commerce Architecture.....	1-2
1–2	Client/Server Application Connection	1-4
1–3	Java Application Connection.....	1-5
1–4	Web Client Connections through Application Web Server	1-6
1–5	Web Client Connections Through Java Application Web Server	1-7
1–6	Web Client Direct Connection Scenarios	1-8
1–7	Service Information Repository.....	1-10
1–8	Centralized Storage of Network Configuration with a Directory Server	1-11
1–9	Dedicated Server Architecture	1-13
1–10	Shared Server Architecture	1-14
1–11	Connection Pooling.....	1-15
1–12	Session Multiplexing.....	1-17
1–13	VI Protocol to Improve Performance.....	1-18
1–14	Intranet Network Access Control with Oracle Connection Manager	1-20
1–15	Internet Network Access Control with an Application Gateway	1-21
1–16	Encryption	1-23
1–17	User Authentication.....	1-24
1–18	Centralized Storage of Security Configuration with a Directory Server.....	1-26
1–19	Oracle Net on the Client	1-28
1–20	Oracle Net on the Server	1-28
1–21	Listener in a Connection Request.....	1-29
2–1	One Service Per Database.....	2-2
2–2	Multiple Services Associated with One Database	2-3
2–3	One Instance Per Database.....	2-4
2–4	Multiple Instances Associated with a Database.....	2-5
2–5	Direct Hand-Off to a Dispatcher	2-10
2–6	Redirected Connection to a Dispatcher.....	2-11
2–7	Connection to a Dedicated Server Process	2-13
2–8	Redirected Connection to a Dedicated Server Process	2-14
3–1	Client Using a Directory Server to Resolve a Connect Identifier	3-5
3–2	Database Service in a Directory Server	3-6
3–3	Domain Component DIT	3-8
3–4	X.500 Style DIT.....	3-8
3–5	Creating Entries in the Directory Server with Applications	3-9
3–6	Relative Naming	3-11
3–7	Oracle Names.....	3-20
3–8	Single Domain Naming Model.....	3-25
3–9	Hierarchical Naming Model	3-26
3–10	Delegated Administrative Regions.....	3-29
4–1	Layers Used in a Client/Server Application Connection.....	4-2

4-2	OSI Communication Layers	4-3
4-3	Layers Used for Java-Client Applications.....	4-8
4-4	Layers Used in Web Client Connections.....	4-9
4-5	Layers Used in an Initial Connection	4-10
4-6	Service Registration.....	4-11
4-7	Listener Architecture.....	4-13
4-8	Shared Server Architecture	4-15
4-9	Dedicated Server Architecture.....	4-16
4-10	Oracle Connection Manager Architecture	4-19
4-11	Scalable Architectural Solutions	4-21
5-1	Oracle Net Manager Interface.....	5-4
5-2	Service Attributes Property Sheet	5-6
5-3	Opening Page of the Net Service Name Wizard.....	5-11
5-4	Opening Page of the Directory Server Migration Wizard.....	5-12
5-5	Welcome Page of the Names Wizard	5-13
5-6	Opening Page of the Oracle Net Configuration Assistant.....	5-14
7-1	Internal Network Layout.....	7-2
7-2	Internet Network Layout.....	7-8
10-1	acme.com in Directory Server.....	10-3
10-2	Oracle Context.....	10-4
10-3	acme.com in Directory Server.....	10-8
10-4	Oracle Context.....	10-9
10-5	Single Domain Export to the Same DIT Node.....	10-12
10-6	Root and Subdomain Domain Export to Similar DIT	10-13
10-7	Single Domain Export to One DIT Node	10-24
10-8	Root and Subdomain Domain Export to Dissimilar DIT	10-25
10-9	Root and Subdomain Domain Export to One DIT Node	10-27
10-10	Multiple Domain Export to Two Subtrees	10-29
12-1	Example listener.ora File	12-5
12-2	listener.ora File with Queue Size.....	12-9
12-3	Listener Control Utility's STATUS Command Output.....	12-22
12-4	Listener Control Utility's SERVICES Command Output	12-24
13-1	Example cman.ora File.....	13-3
13-2	tnsnames.ora File with and without Oracle Connection Manager	13-10
13-3	tnsnames.ora File Configured for Protocol Conversion Support.....	13-11
15-1	Load Balancing Environment for a Shared Server Configuration	15-12
15-2	Load Balancing Example for a Shared Server Configuration	15-14
15-3	Load Balancing Environment for a Dedicated Server Configuration.....	15-15
15-4	Load Balancing Example for a Dedicated Server Configuration.....	15-17
15-5	Application Web Servers and Oracle Connection Manager	15-35
16-1	Absolute Name for X.500 Namespaces	16-10

16–2	Absolute Name for Domain Component Namespaces	16-12
17–1	Error Stack Components Mapped to Oracle Net.....	17-21
17–2	sqlnet.log File	17-23
17–3	cman_pid.log.....	17-35
17–4	cmadm_pid.log	17-35
17–5	Packet Information.....	17-50
17–6	Trace Example.....	17-51

List of Tables

3–1	Oracle Net Configuration Models.....	3-2
3–2	Oracle Net Configuration Files.....	3-2
3–3	LDAP Directory Common Attributes	3-7
3–4	LDAP Directory Authentication Methods.....	3-14
3–5	LDAP Directory User Groups	3-15
3–6	Oracle Net LDAP Main Object Classes	3-16
3–7	Oracle Net LDAP Derived Object Classes.....	3-17
3–8	Data Stored in an Oracle Names Server.....	3-23
3–9	Well-Known Oracle Names Servers	3-31
4–1	Oracle Connection Manager Processes	4-18
5–1	Oracle Net Configuration Assistant	5-15
5–2	Common Administrative Tasks for Oracle Net	5-19
7–1	Connection Pooling and Session Multiplexing.....	7-4
7–2	Naming Methods: Advantages and Disadvantages	7-10
7–3	SDU Considerations.....	7-13
7–4	Oracle Net Feature Summary	7-14
8–1	Directory Usage Configuration Page in Oracle Net Configuration Assistant.....	8-6
8–2	ldapmodify Arguments.....	8-9
9–1	Naming Methods.....	9-4
9–2	Oracle Names Checkpoint Files	9-21
9–3	Client Settings In sqlnet.ora for Oracle Names	9-37
10–1	Oracle Names Control Utility Structural Commands.....	10-10
10–2	Sub-Parameters of the NAMES.ADMIN_REGION Parameter for LDAP	10-19
10–3	Sub-Parameters of the NAMES.ADMIN_REGION Parameter for LDIF	10-20
11–1	Naming Method Values	11-5
11–2	Routing Settings in sqlnet.ora.....	11-7
11–3	Access Control Settings in sqlnet.ora	11-8
11–4	Advanced Settings in sqlnet.ora.....	11-9
11–5	External Naming Methods Settings in sqlnet.ora.....	11-12
12–1	Service Settings in listener.ora.....	12-10
12–2	Listener Control Utility STATUS Command	12-21
12–3	Listener Control Utility SERVICES Command	12-23
13–1	Session Multiplexing Levels	13-13
13–2	Per-Rule Parameters.....	13-14
14–1	Required Attributes of the DISPATCHERS Parameter	14-2
14–2	Optional Attributes of the DISPATCHERS Parameter	14-3
15–1	Address List Parameters	15-4
15–2	Address List Configuration for Naming Method Types	15-5
15–3	Address List Options Dialog Box.....	15-6

15–4	Advanced Connect Data Settings.....	15-8
15–5	Connect Data Configuration for Naming Method Types.....	15-9
15–6	Sub-Parameters of the FAILOVER_MODE Parameter	15-21
15–7	VI Protocol Settings in listener.ora.....	15-31
15–8	External Procedures Settings in listener.ora	15-38
15–9	Heterogeneous Services Settings in listener.ora	15-41
15–10	Oracle RDB Database Settings in a Connect Descriptor	15-44
17–1	ldapwrite Arguments.....	17-15
17–2	Error Stack Components.....	17-21
17–3	Oracle Net Log Files	17-24
17–4	Oracle Net Log Parameters	17-24
17–5	sqlnet.ora Log Parameters.....	17-25
17–6	listener.ora Log Parameters	17-26
17–7	names.ora Log Parameters	17-26
17–8	cman.ora Log Parameters	17-27
17–9	Service Registration Event Log Information	17-32
17–10	cman_pid.log Log Level 1 Event Codes	17-36
17–11	cman_pid.log Log Level 2 Event Codes	17-38
17–12	cman_pid.log Log Level 3 Event Codes	17-38
17–13	cman_pid.log Log Level 4 Event Codes	17-39
17–14	Oracle Net Trace Files	17-40
17–15	Oracle Net Trace Parameters	17-40
17–16	sqlnet.ora Trace Parameters	17-41
17–17	TNSPING Trace Parameters	17-43
17–18	listener.ora Trace Parameters.....	17-44
17–19	names.ora Trace Parameters	17-45
17–20	cman.ora Trace Parameters	17-46

Send Us Your Comments

Oracle9*i* Net Services Administrator's Guide, Release 1 (9.0.1)

Part No. A90154-01

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

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

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

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 506-7227 Attn: Information Development
- Postal service:
Oracle Corporation
Information Development Documentation Manager
500 Oracle Parkway, Mailstop 4OP11
Redwood Shores, CA 94065
USA

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

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

Preface

Oracle Net Services Administrator's Guide provides the information you need to understand and use Oracle Net Services and its related applications and components.

This preface contains these topics:

- [Audience](#)
- [Organization](#)
- [Related Documentation](#)
- [Conventions](#)
- [Documentation Accessibility](#)

Audience

Oracle Net Services Administrator's Guide is intended for the following kinds of readers:

- Network administrators
- Directory server administrators
- Database administrators
- Decision makers

This guide is especially targeted for network administrators who are responsible for ensuring connectivity. For network administrators, Oracle Corporation recommends reading all of Part I for a conceptual understanding of Oracle Net Services. In addition, read Chapter 3 in Part I and all of Part II for essential configuration instructions. Use Part III for troubleshooting.

Directory administrators will find Chapter 3 in Part I useful in understanding how Oracle Net Services uses a directory server. In addition, Chapter 8 and Chapter 10 in Part II provide instructions for establishing directory usage and exporting existing naming data to a directory server.

Database administrators should skim Chapter 1 and Chapter 2 to gain an understanding of the big networking picture. In addition, Chapter 5 will provide an overview of networking tools, which may be of use to the database administrator. Database administrator will also find Chapter 7, Chapter 12, and Chapter 14 useful in understanding how to configure Oracle9i database server features that require listener and shared server configuration.

Decision makers will find this guide useful in understanding how Oracle Net Services fits into the overall network architecture. Decision makers will find Chapter 1, Chapter 2, Chapter 4, and Chapter 7 useful for explaining the basics of Oracle Net Services.

Oracle Corporation recommends that all readers skim Part 1, to ensure that they have the background required to benefit from the rest of the guide.

Organization

This document contains:

Part I, "Getting Started with Oracle Net Services"

Chapter 1, "Networking Challenges in the Internet Age"

This chapter introduces main features of Oracle Net Services, including connectivity, centralized management, scalability, and security. In addition, Oracle Net Services, Oracle Connection Manager, and Oracle Advanced Security products and components are introduced.

Chapter 2, "Connectivity Concepts"

This chapter describes the concepts of service naming, connection establishment, naming, access control, and multiplexing.

Chapter 3, "Configuration Management Concepts"

This chapter describes localized configuration and centralized configuration methodologies.

Chapter 4, "Architecture of Oracle Net Services"

This chapter describes stack communication architecture and component architecture.

Chapter 5, "Configuration and Administration Tools Overview"

This chapter describes Oracle Net Services products and administration tools.

Chapter 6, "Quick Start to Oracle Net Connections"

This chapter describes how to set up a basic network configuration with Oracle Net Configuration Assistant.

Part II, "Configuration and Administration of Oracle Net Services"

Chapter 7, "Planning the Network"

This chapter describes considerations for planning a network using Oracle Net Services. It explains the relationships of the Oracle Net Services components, and options to help you better manage the network.

Chapter 8, "Setting Up Directory Server Usage"

This chapter describes how to configure access to a directory server.

Chapter 9, "Configuring Naming Methods"

This chapter describes how to configure naming methods.

Chapter 10, "Exporting Naming Data to a Directory Server"

This chapter describes how export data from a `tnsnames.ora` file or an Oracle Names server to a directory server.

Chapter 11, "Configuring Profiles"

This chapter describes how to configure client and database server profiles.

Chapter 12, "Configuring and Administering the Listener"

This chapter describes how to configure the listener to accept client requests, and how to perform common administrative tasks.

Chapter 13, "Configuring Oracle Connection Manager"

This chapter describes how to configure Oracle Connection Manager to provide access control, multiplexed connections, and protocol conversion.

Chapter 14, "Configuring Shared Server"

This chapter describes how to configure the Oracle database server to use the shared server connection model.

Chapter 15, "Enabling Advanced Features for Oracle Net Services"

This chapter describes how to configure advanced and optional connection features, such as load balancing, failover, and the VI protocol. It also explains how to configure connections to non-Oracle database services.

Part III, "Testing and Troubleshooting Oracle Net Services"

Chapter 16, "Establishing a Connection and Testing the Network"

This chapter describes how to start Oracle Net Services components, establish a connection, and test a connection.

Chapter 17, "Troubleshooting Oracle Net Services"

This chapter describes procedures to troubleshoot Oracle Net Services. It includes information on tracing and logging.

Glossary

Related Documentation

For more information, see these Oracle resources:

- *Oracle Net Services Reference Guide*.
- Oracle9*i* documentation set
- <http://www.ietf.org/> for information about the **Open Systems Interconnection (OSI)**
- <http://www.viarch.org/> for the VI specification

Many books in the documentation set use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle9*i* Sample Schemas* for information on how these schemas were created and how you can use them yourself.

In North America, printed documentation is available for sale in the Oracle Store at
<http://oraclestore.oracle.com/>

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

<http://www.oraclebookshop.com/>

Other customers can contact their Oracle representative to purchase printed documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://technet.oracle.com/membership/index.htm>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://technet.oracle.com/docs/index.htm>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Windows Operating Systems](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle9i Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
UPPERCASE monospace (fixed-width font)	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.

Convention	Meaning	Example
<i>lowercase monospace (fixed-width font)</i>	<p>Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values.</p> <p>Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.</p>	Enter <code>sqlplus</code> to open SQL*Plus. The password is specified in the <code>orapwd</code> file. Back up the datafiles and control files in the <code>/disk1/oracle/dbs</code> directory. The <code>department_id</code> , <code>department_name</code> , and <code>location_id</code> columns are in the <code>hr.departments</code> table. Set the <code>QUERY_REWRITE_ENABLED</code> initialization parameter to <code>true</code> . Connect as <code>oe</code> user. The <code>JRepUtil</code> class implements these methods.
<i>lowercase monospace (fixed-width font) italic</i>	Lowercase monospace italic font represents placeholders or variables.	You can specify the <code>parallel_clause</code> . Run <code>Uold_release.SQL</code> where <code>old_release</code> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
<code>[]</code>	Brackets enclose one or more optional items. Do not enter the brackets.	<code>DECIMAL (digits [, precision])</code>
<code>{}</code>	Braces enclose two or more items, one of which is required. Do not enter the braces.	<code>{ ENABLE DISABLE }</code>
<code> </code>	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	<code>{ ENABLE DISABLE }</code> <code>[COMPRESS NOCOMPRESS]</code>

Convention	Meaning	Example
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> ▪ That we have omitted parts of the code that are not directly related to the example ▪ That you can repeat a portion of the code 	<pre>CREATE TABLE ... AS <i>subquery</i>;</pre> <pre>SELECT <i>col1</i>, <i>col2</i>, ... , <i>coln</i> FROM employees;</pre>
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	<pre>acctbal NUMBER(11,2);</pre> <pre>acct CONSTANT NUMBER(4) := 3;</pre>
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	<pre>CONNECT SYSTEM/<i>system_password</i></pre> <pre>DB_NAME = <i>database_name</i></pre>
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	<pre>SELECT last_name, employee_id FROM employees;</pre> <pre>SELECT * FROM USER_TABLES;</pre> <pre>DROP TABLE hr.employees;</pre>
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	<pre>SELECT last_name, employee_id FROM employees;</pre> <pre>sqlplus hr/hr</pre> <pre>CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program. For example, to start Oracle Database Configuration Assistant, you must click the Start button on the taskbar and then choose Programs > Oracle - <i>HOME_NAME</i> > Database Administration > Database Configuration Assistant.	Choose Start > Programs > Oracle - <i>HOME_NAME</i> > Database Administration > Database Configuration Assistant
C:\>	Represents the Windows command prompt of the current hard disk drive. Your prompt reflects the subdirectory in which you are working. Referred to as the command prompt in this guide.	C:\oracle\oradata>
<i>HOME_NAME</i>	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	C:\> net start Oracle <i>HOME_NAME</i> NSListener

Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to 8.1, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default was:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin95 for Windows 95 ■ C:\orawin98 for Windows 98 <p>or whatever you called your Oracle home.</p> <p>In this Optimal Flexible Architecture (OFA)-compliant release, all subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. If you install release 9.0 on a computer with no other Oracle software installed, the default setting for the first Oracle home directory is C:\oracle\ora90. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>See <i>Oracle9i Database Getting Starting for Windows</i> for additional information on OFA compliances and for information on installing Oracle products in non-OFA compliant directories.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdbms\admin</i> directory.

Documentation Accessibility

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

<http://www.oracle.com/accessibility/>

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

What's New in Oracle Net Services?

This section describes the new networking features of Oracle9*i* release 9.0 and provides pointers to additional information. New features information from previous releases is also retained to help those users migrating to the current release.

The following sections describe the new networking features:

- [Oracle9i Release 9.0 New Features in Oracle Net Services](#)
- [Oracle8i New Features in Oracle Net Services](#)

Oracle9i Release 9.0 New Features in Oracle Net Services

The new features for Oracle Net Services in release 9.0 include:

- **Name Change to Networking Components**

To provide consistency with Oracle9i, several name changes to networking products, features, and parameters have been made in this release. The following table maps the former product and feature names to their new names.

Former Name	New Name
multi-threaded server (MTS)	shared server
Net8	Oracle Net to describe the software that provides connectivity
Net8	Oracle Net Services to encompass Oracle Net, the listener, and Oracle Connection Manager
Net8 communication stack layer	Oracle Net foundation layer
Net8 Configuration Assistant	Oracle Net Configuration Assistant
Net8 Manager	Oracle Net Manager

The following table maps the multi-threaded server (MTS) parameters and dynamic view names to the new names associated with **shared server**. Except for the MTS_MAX_SERVERS parameter, the old parameter names are maintained for backward compatibility to Oracle8i. Oracle Corporation recommends migrating to the new names.

MTS Parameter/View Name	New Shared Server Parameter/View Name
MTS_CIRCUITS initialization parameter	CIRCUITS initialization parameter
MTS_DISPATCHERS initialization parameter	DISPATCHERS initialization parameter
MTS_MAX_DISPATCHERS initialization parameter	MAX_DISPATCHERS initialization parameter
MTS_MAX_SERVERS initialization parameter	MAX_SHARED_SERVERS initialization parameter
MTS_SERVERS initialization parameter	SHARED_SERVERS initialization parameter
MTS_SESSIONS initialization parameter	SHARED_SERVER_SESSIONS initialization parameter
V\$MTS view	V\$SHARED_SERVER_MONITOR view

- **Connection Load Balancing for Dedicated Server Configurations**

Configurations that use dedicated servers can now use the **connection load balancing** feature that was previously available only for shared server configurations.

See Also: ["Connection Load Balancing" on page 2-9](#)
- **Multiple Oracle Context Creation in Oracle Net Configuration Assistant**

The Oracle Net Configuration Assistant enables you to create multiple Oracle Contexts to facilitate management of a complex naming structure in a directory server.

See Also: ["Configuring Directory Usage After Installation" on page 8-5](#)
- **Oracle Names LDAP Proxy Servers**

In future releases, Oracle Names will not be supported as a centralized naming method. As Oracle Names is deprecated in favor of **directory naming** with LDAP-compliant directory servers, **Oracle Names LDAP Proxy servers** provide a way for release 8.1.5 or previous clients that do not support directory naming to use the same data as is used for directory naming. Oracle Names LDAP Proxy servers are Oracle Names servers that have been configured to proxy for LDAP-compliant directory servers. Upon startup, Oracle Names LDAP Proxy servers obtain network object information from a directory server. This provides a single point of definition for all data in a directory server and does not require that both Oracle Names servers and directory servers be maintained separately and simultaneously.

See Also: ["Exporting Network Objects from an Oracle Names Server" on page 10-6](#)
- **VI Protocol Support**

Oracle protocol support now includes support for the Virtual Interface (VI) protocol. VI is a thin protocol that is more efficient than TCP/IP. Applications perform better using VI, because VI places most of the messaging burden upon high-speed network hardware, freeing the CPU for other tasks. The VI protocol is best deployed for application Web server and inter-database server communication where the servers are co-located.

See Also: ["Configuring the VI Protocol for Application Web Server and Oracle Database Server Communication" on page 15-30](#)

- **Unsupported Features**

In an effort to streamline configuration decisions for the Internet, the following subsections describe the features and the configuration file that are no longer being supported:

- Identix and SecurID authentication methods
- NDS External Naming and NDS Authentication
- Net8 OPEN
- Prespawned dedicated servers
- `protocol.ora` file
- SPX protocol
- Trace Assistant

Identix and SecurID Authentication Methods

If you are using Identix or SecurID **authentication methods**, provided by **Oracle Advanced Security**, Oracle Corporation recommends migrating to one of the following authentication methods:

- CyberSafe
- RADIUS
- Kerberos
- **SSL**

See Also: *Oracle Advanced Security Administrator's Guide*

Novell Directory Services (NDS) External Naming and NDS Authentication

Support for NDS as an authentication method and as an **external naming** method is no longer supported. If you are using NDS as an external naming method, Oracle Corporation recommends using **directory naming** instead.

See Also: "Configuring the Directory Naming Method" on page 9-12

Net8 OPEN

Net8 OPEN, which provided an application program interface (API) that enabled programmers to develop both database and non-database applications, is no longer supported.

Prespawned Dedicated Servers

Prestarted dedicated server processes are no longer supported. Instead, configure shared server to improve scalability and system resource usage.

protocol.ora File

The protocol.ora file is no longer supported.

Parameters in the protocol.ora file have been merged into the sqlnet.ora file. These parameters enable you to configure access control to the database, as well as no delays in TCP/IP buffer flushing. These parameters include:

- TCP.NODELAY
- TCP.EXCLUDED_NODES
- TCP.INVITED_NODES
- TCP.VALIDNODE_CHECKING

See Also: *Oracle Net Services Reference Guide* for a description of these parameters

If you have a protocol.ora file in the \$ORACLE_HOME/network/admin directory on UNIX, and the ORACLE_HOME\network\admin directory on Windows operating systems, Oracle Net Manager, when first started, automatically merges the protocol.ora parameters into the sqlnet.ora file.

There may be operating system-specific parameters in protocol.ora that are node specific. For this reason, Oracle Corporation recommends not sharing sqlnet.ora with other nodes after merging or adding these parameters.

SPX Protocol

Protocol addresses using the SPX protocol must be replaced. Oracle Net provides support for the following network protocols:

- TCP/IP
- TCP/IP with SSL
- Named Pipes
- LU6.2
- VI

See Also: *Oracle Net Services Reference Guide* for protocol parameter configuration

Trace Assistant

The Trace Assistant is no longer being supported. You must manually analyze trace files.

Oracle8i New Features in Oracle Net Services

These Oracle Net features introduced in Oracle8i also apply to Oracle9i:

- **Client Connection Features**

Client Load Balancing—When more than one listener supports a service, a client can randomize requests to the various listeners.

Connect-Time Failover—When more than one listener supports a service, a client can be configured to fail over the client request to a different listener if the first listener fails.

See Also:

- ["Client Load Balancing" on page 2-8](#)
- ["Connect-Time Failover" on page 2-8](#)

- **Directory Naming**

Network information can now be stored in a centralized LDAP-compliant directory service, including Oracle Internet Directory and Microsoft Active Directory.

See Also: ["Directory Server Support" on page 3-4](#)

- **Instance Role Specification for Primary and Secondary Configurations**

With the connect descriptor `INSTANCE_ROLE` parameter, you can specify a connection to the primary or secondary instance of Oracle9i Real Application Clusters and Oracle9i Real Application Clusters Guard configurations.

See Also: ["Specifying the Instance Role for Primary and Secondary Instance Configurations" on page 15-25](#)

- **Listener Features**

Service Naming—Oracle8i database services are identified by a service name rather than an Oracle System Identifier (SID). This impacts the way connect descriptors are defined.

Direct Hand-Off—The listener has the ability to hand off requests directly to an shared dispatcher, without issuing a redirect message back to the client.

Service Registration—Database instances register information with the listener during database startup.

Connection Load Balancing—The listener is able to balance the number of active connections among various instances and shared server dispatchers for the same service.

See Also:

- ["Database Service and Database Instance Identification" on page 2-2](#)
- ["Service Handlers" on page 2-9](#)
- ["Connection Load Balancing" on page 2-9](#)

- **Oracle Net Configuration Assistant**

[Oracle Net Configuration Assistant](#) is a post-installation tool that performs basic configuration. After installation, it automatically configures default configuration files.

See Also: ["Oracle Net Configuration Assistant" on page 5-14](#) and operating-system installation guides

- **Service Naming**

Database services are identified by service name and instance name rather than Oracle System Identifier (SID).

See Also: ["Database Service and Database Instance Identification" on page 2-2](#)

- **TCP/IP with Secure Sockets Layer (SSL) protocol**

A protocol for client/server authentication over a network using TCP/IP and the Secure Sockets Layer (SSL).

Part I

Getting Started with Oracle Net Services

Part I provides an overview of Oracle Net Services concepts, products, and tools.

This part contains the following chapters:

- [Chapter 1, "Networking Challenges in the Internet Age"](#)
- [Chapter 2, "Connectivity Concepts"](#)
- [Chapter 3, "Configuration Management Concepts"](#)
- [Chapter 4, "Architecture of Oracle Net Services"](#)
- [Chapter 5, "Configuration and Administration Tools Overview"](#)
- [Chapter 6, "Quick Start to Oracle Net Connections"](#)

Networking Challenges in the Internet Age

This chapter describes the networking issues that your system faces in the Internet age, and introduces the technology and products that can provide a complete network solution.

This chapter contains the following topics:

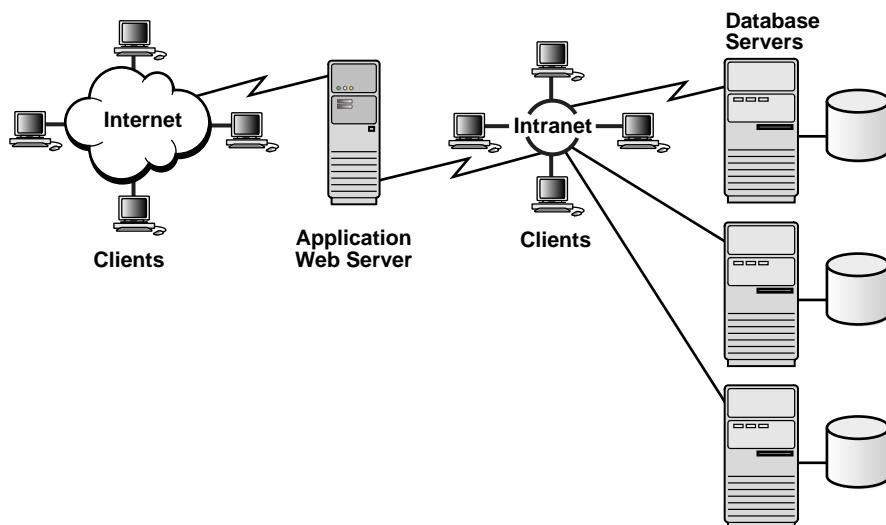
- [What Is the Big Picture for Networking?](#)
- [Oracle's Solution to Networking Issues](#)
- [A Suite of Networking Components](#)

What Is the Big Picture for Networking?

The e-business model creates new business requirements. To carry out electronic business successfully, Web sites must provide reliable connectivity and 24 by 7 availability. Corporate Web sites must also address user scalability and performance to simultaneously handle thousands of Internet connections to their data repositories. Solutions are needed as well to provide immediate Web browser access to existing applications and services.

Figure 1–1 shows a typical architecture in which Internet clients connect to a company's databases through an application Web server. The figure also shows the intranet architecture that enables a company's own clients to communicate with the databases. This basic architecture will be examined further to show how Oracle networking technologies are used throughout typical network environments.

Figure 1–1 Typical E-Commerce Architecture



Oracle's Solution to Networking Issues

Oracle Net Services provide enterprise wide connectivity solutions in distributed, heterogeneous computing environments. Oracle Net Services ease the complexities of network configuration and management, maximize performance, and improve network diagnostic capabilities.

This section introduces the basic networking concepts that come into play in a typical network configuration. The topics discussed include:

- [Connectivity](#)
- [Manageability](#)
- [Internet and Intranet Scalability](#)
- [Network Security](#)

Connectivity

Oracle Net, a component of Oracle Net Services, enables a network session from a client application to an Oracle database server. Once a network session is established, Oracle Net acts as the data courier for both the client application and the database server. It is responsible for establishing and maintaining the connection between the client application and database server, as well as exchanging messages between them. Oracle Net is able to perform these jobs because it is located on each computer in the network.

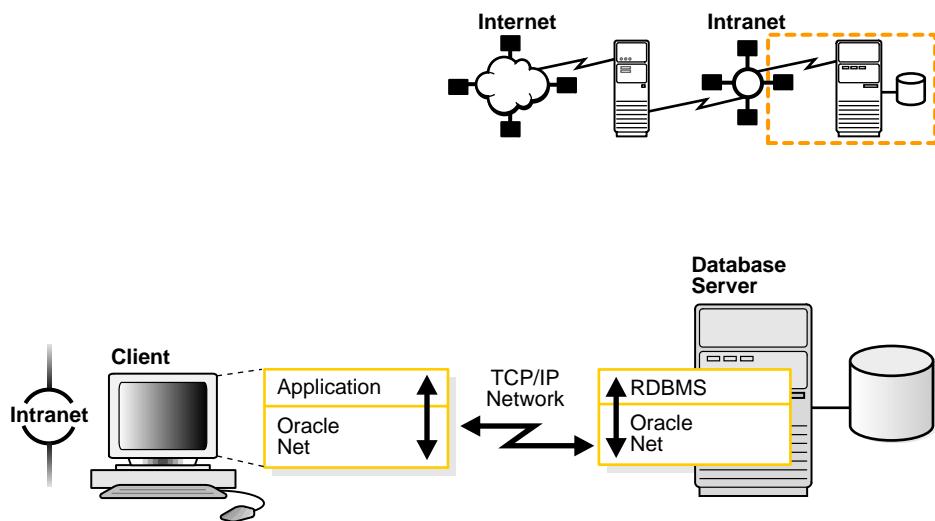
This section discusses the following connectivity topics:

- [Client/Server Application Connections](#)
- [Web Client Application Connections](#)

Client/Server Application Connections

Oracle Net enables connections from traditional client/server applications to Oracle database servers. **Figure 1–2** shows how Oracle Net enables a network connection between a client and a database server. Oracle Net is a software component that resides on both the client and the database server. Oracle Net is layered on top of a network **protocol**—rules that determine how applications access the network and how data is subdivided into packets for transmission across the network. In this illustration, Oracle Net communicates with the **TCP/IP protocol** to enable computer-level connectivity and data transfer between the client and the database server.

Figure 1–2 Client/Server Application Connection



Specifically, Oracle Net is comprised of the **Oracle Net foundation layer**, which establishes and maintains connections, and **Oracle protocol support**, which maps the foundation layer's technology to industry-standard protocols.

Java client applications access an Oracle database through a **Java Database Connectivity (JDBC) Driver**, a standard Java interface for connecting from Java to a relational database. Oracle offers the following drivers:

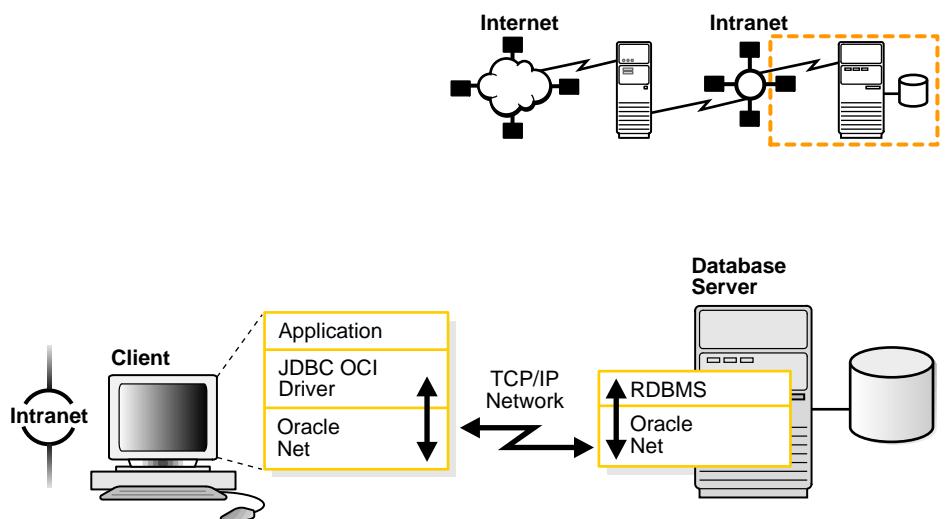
- The JDBC **Oracle Call Interface (OCI)** driver for client side use with an Oracle client installation
- The JDBC Thin driver for client side use without an Oracle installation, particularly with applets

These drivers use Oracle Net to enable connectivity between a client application and an Oracle database.

Note: The JDBC Thin driver is a 100 percent pure Java driver that requires no client installation.

Figure 1–3 shows a Java client application using a JDBC OCI driver and an Oracle database server. The Java client application makes calls to the JDBC OCI driver which in turn translates the JDBC calls directly into the Oracle Net layer. The client then uses Oracle Net to communicate with an Oracle database that is also configured with Oracle Net.

Figure 1–3 Java Application Connection

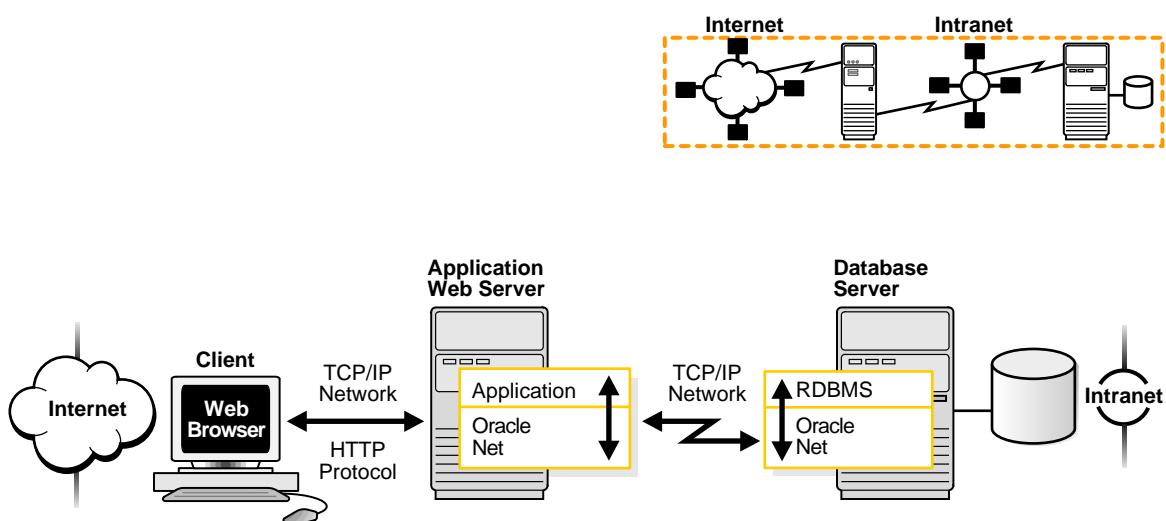


Web Client Application Connections

Internet connections from client Web browsers to an Oracle database server are similar to client/server applications, except for the architecture.

[Figure 1–4](#) shows the basic architecture for Web client connections, including a client Web browser, an application Web server, and an Oracle database server. The browser on the client communicates with HTTP to a Web server to make a connection request. The Web server sends the request to an application where it is processed. The application then uses Oracle Net to communicate with an Oracle database server that also is configured with Oracle Net.

Figure 1–4 Web Client Connections through Application Web Server



The basic components have the following characteristics:

- Hypertext Transport Protocol (HTTP)

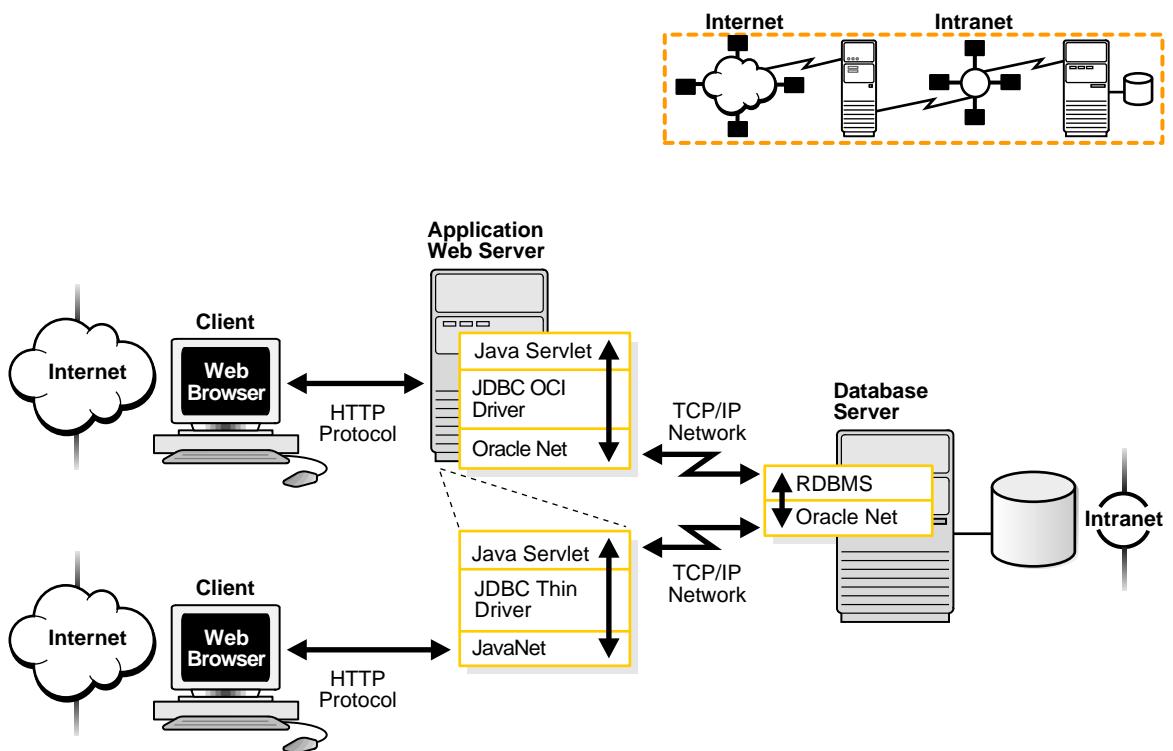
HTTP provides the language that enables Web browsers and application Web servers to communicate.

- Application Web Server

An application Web server manages data for a Web site, controls access to that data, and responds to requests from Web browsers. The application on the Web server communicates with the database and performs the job requested by the Web server.

An application Web server can host Java applications and servlets, as shown in [Figure 1–5](#). Web browsers make a connection request by communicating through HTTP to an application Web server. The application Web server sends the request to an application or a servlet, which in turn uses a JDBC OCI or a JDBC Thin driver to process the request. The driver then uses Oracle Net to communicate with an Oracle database server that also is configured with Oracle Net.

Figure 1–5 Web Client Connections Through Java Application Web Server



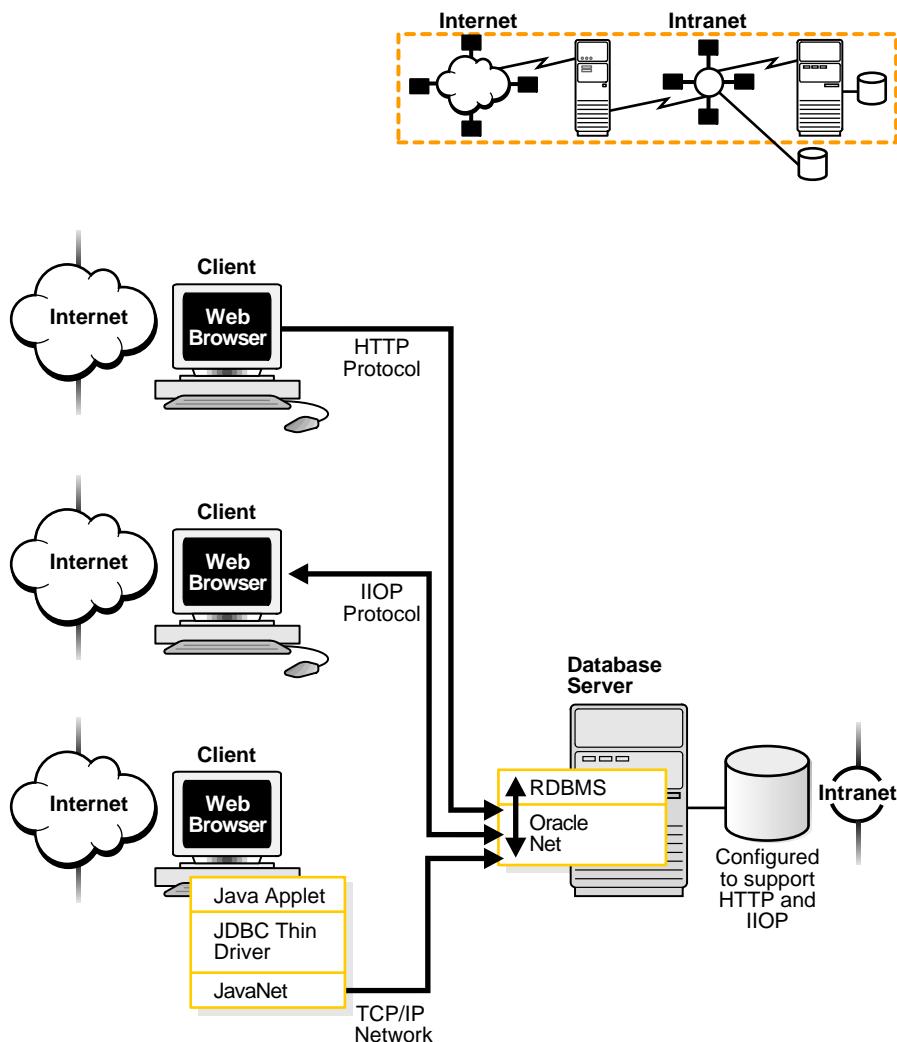
Web Client Connections Without an Application Web Server

Web clients that do not require an application Web server to access applications can access the Oracle database directly, for example, by using a Java applet. In addition to regular connections, the database can be configured to accept HTTP and [Internet Inter-ORB Protocol \(IIOP\)](#) connections. These protocols are used for connections to [Oracle9i JVM](#) in the Oracle9i instance.

[Figure 1–6](#) shows three different Web clients. The first Web client makes an HTTP connection to the database. The second Web client makes an IIOP connection. The third client uses a Web browser with a JDBC Thin driver, which in turn uses a Java

version of Oracle Net called JavaNet to communicate with the Oracle database server that is configured with Oracle Net.

Figure 1–6 Web Client Direct Connection Scenarios



Manageability

Oracle Net Services offer a number of manageability features that enable you to easily configure and manage networking components. These features are described in the following topics:

- [Location Transparency](#)
- [Centralized Configuration and Management](#)
- [Quick Installation and Configuration](#)

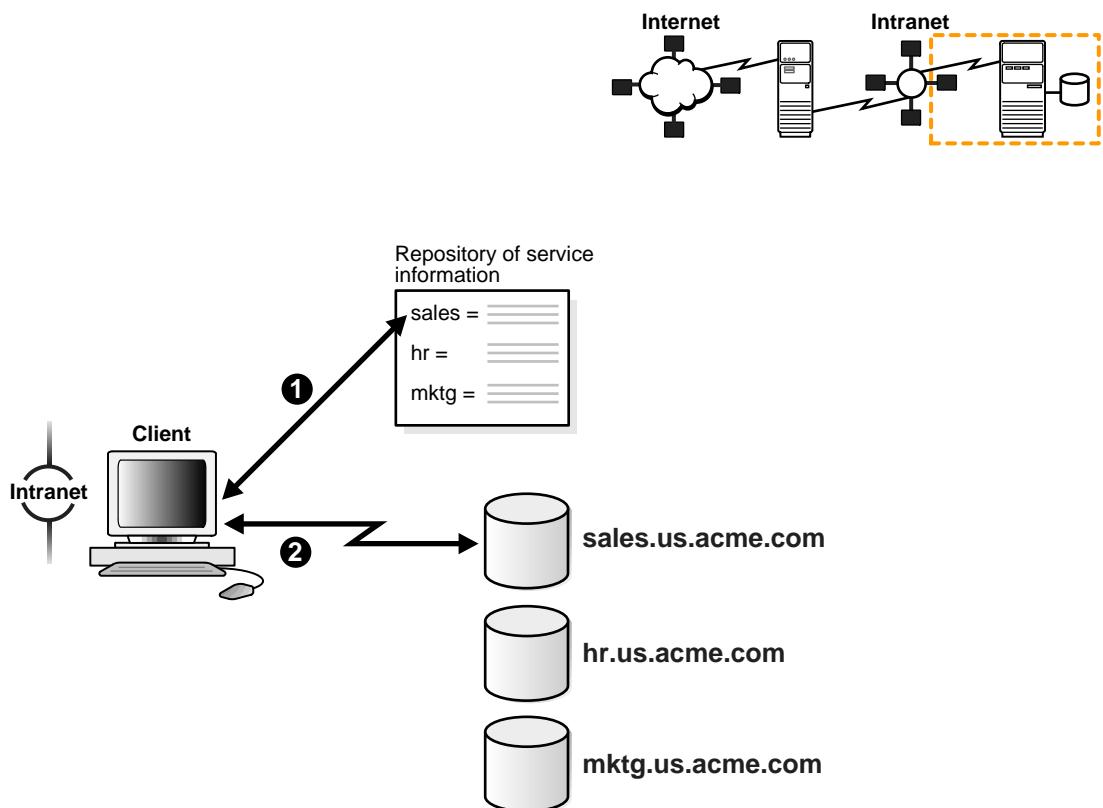
Location Transparency

A company can have several databases, each representing a specific type of service for various client applications. For example, a company may have three databases, which it uses for sales, human resources, and marketing applications. Each database is represented by one or more services. A **service** is identified by a **service name**, for example, `sales.us.acme.com`. A client uses this service name to identify the database it needs to access. The information about the database service and its location in the network is transparent to the client because the information needed for a connection is stored in a repository.

For example, in [Figure 1–7](#), a company has three databases that clients can access. Each database has a distinct service name: sales.us.acme.com, hr.us.acme.com, and mktg.us.acme.com.

1. The client uses the repository to find the information it needs for sales.us.acme.com.
2. Once the client has the information it needs, it connects to the database.

Figure 1–7 Service Information Repository



The repository is represented by one or more [naming methods](#). Oracle Net Services offer several types of naming methods that support localized configuration on each client, or centralized configuration that can be accessed by all clients in the network. Easy-to-use graphical user interfaces enable you to manage data stored in the naming methods.

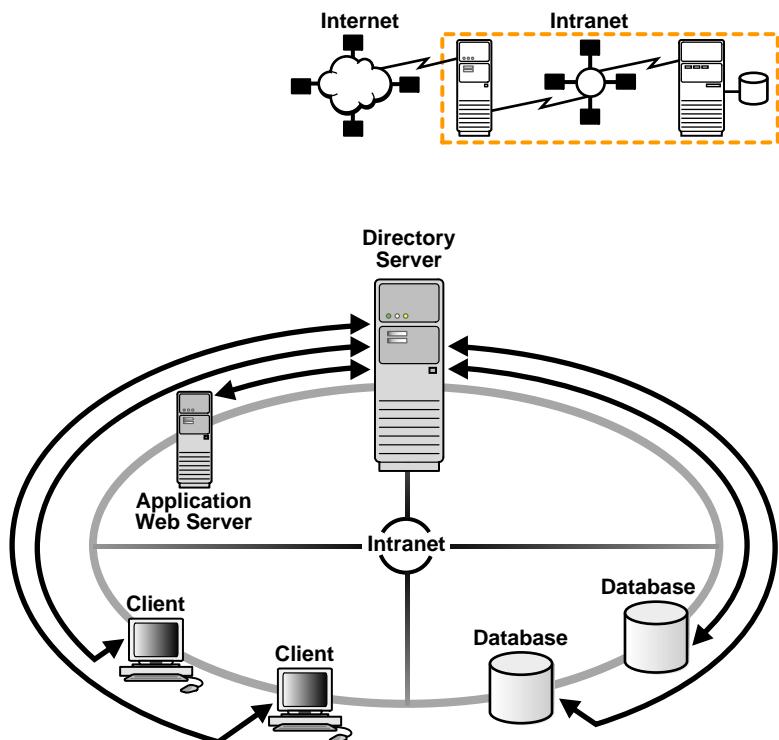
Centralized Configuration and Management

To manage large networking environments, administrators have to be able to easily access a centralized repository to specify and modify the network configuration. For this reason, the Oracle Net Services configuration can be stored in a LDAP-compliant directory server.

Support of LDAP-compliant directory servers provides a centralized vehicle for managing and configuring a distributed Oracle network. The directory can act as a central repository for all information on database network components, user and corporate policies, and user authentication and security, thus replacing clientside and serverside localized configuration files.

All computers on the heterogeneous network can refer to the directory for information. [Figure 1–8](#) shows clients, other servers (such as application Web servers) and Oracle database servers connecting to a centralized directory server.

Figure 1–8 Centralized Storage of Network Configuration with a Directory Server



See Also: ["Directory Server Support"](#) on page 3-4 for an in-depth overview of directory server concepts

Quick Installation and Configuration

Oracle Net Services install quickly and easily. Networking elements for the Oracle database server and clients are preconfigured for most environments. Information about an Oracle database service is populated in one or more naming methods. As a result, clients and servers are ready to immediately connect when installed, giving users the benefits of distributed computing.

Internet and Intranet Scalability

Oracle Net provides scalability features that enable you to maximize system resources and improve performance. These features are described in the following topics:

- [Shared Server](#)
- [Performance Between Application Web Servers and Oracle Databases](#)

Shared Server

Oracle's **shared server** architecture increases the scalability of applications and the number of clients that can be simultaneously connected to the database. The shared server architecture also enables existing applications to scale up without making any changes to the application itself.

When using shared server, clients do not communicate directly with a database's **server process**—a database process that handles a client's requests on behalf of a database. Instead, client requests are routed to one or more **dispatchers**. The dispatchers place the client requests on a common queue. An idle **shared server process** from the shared pool of server processes picks up and processes a request from the queue. This means a small pool of server processes can serve a large number of clients.

[Figure 1–9](#) on page 1-13 and [Figure 1–10](#) on page 1-14 show the basic difference between the shared server connection model and the traditional **dedicated server** connection model. In the shared server model, a dispatcher can support multiple client connections concurrently. In the dedicated server model, there is one server process for each client. Each time a connection request is received, a server process is started and dedicated to that connection until completed. This introduces a processing delay.

Shared server is ideal in configurations with a large number of connections because it reduces the server's memory requirements. Shared server is well suited for both Internet and intranet environments.

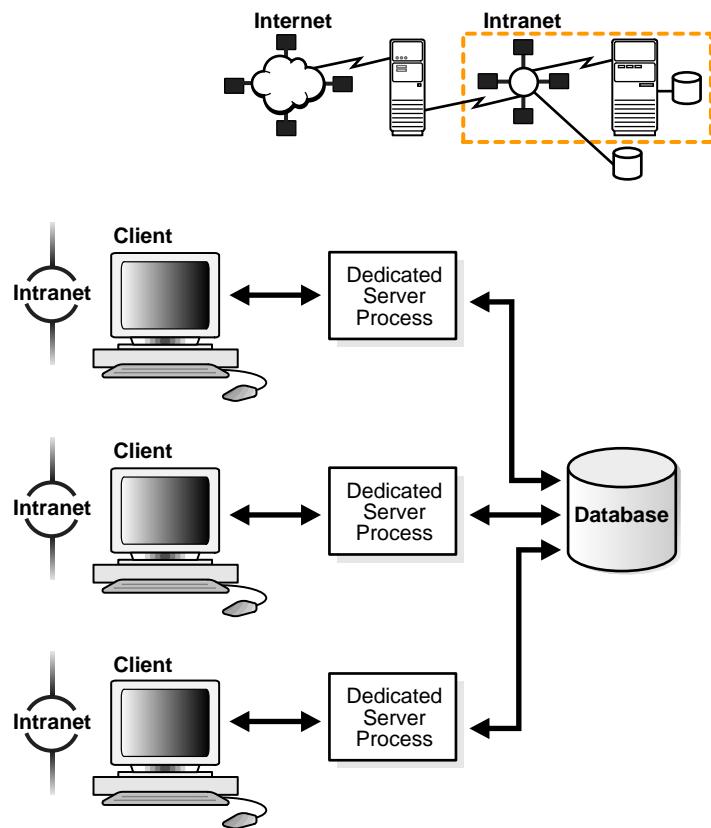
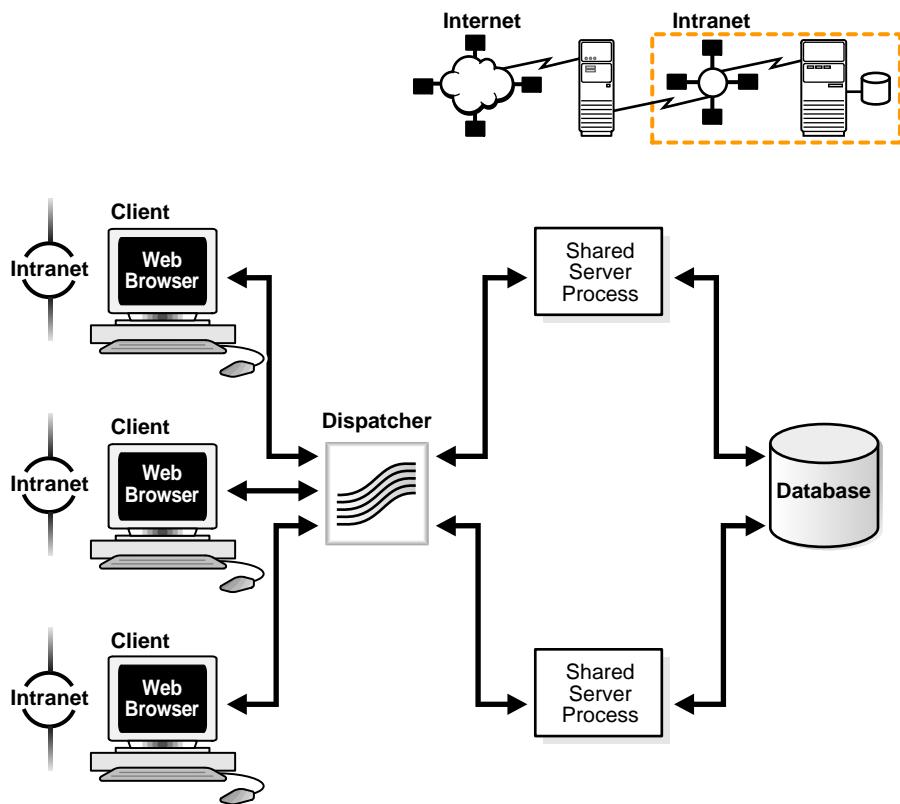
Figure 1–9 Dedicated Server Architecture

Figure 1–10 Shared Server Architecture



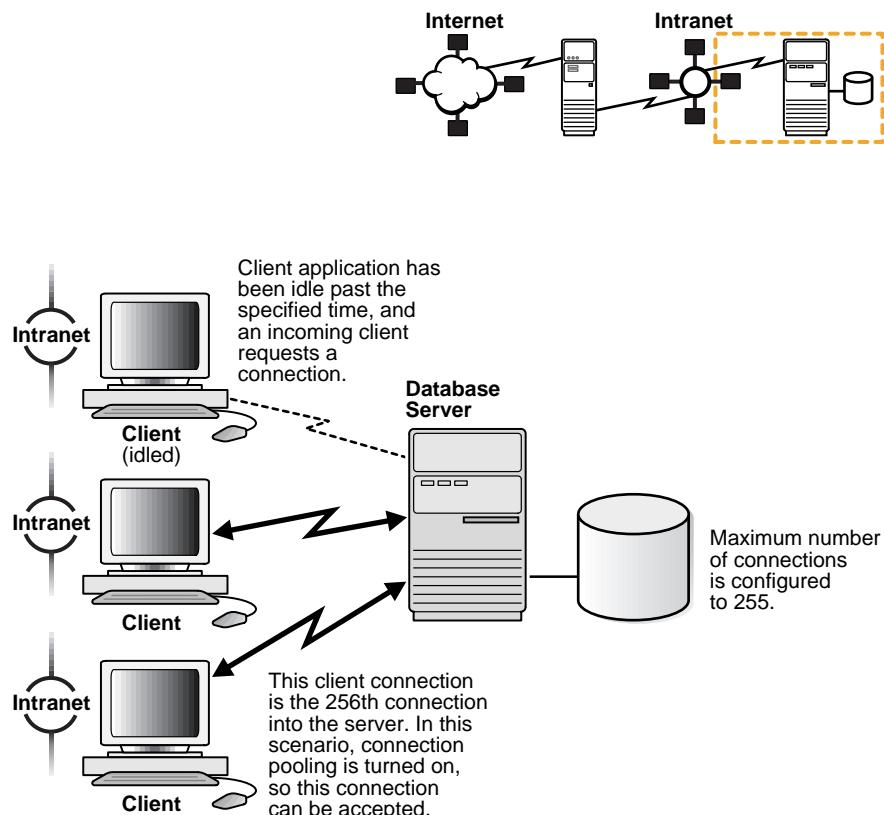
Utilization of server resources can be further enhanced with Oracle Net Services features that are configurable through shared server. These features are discussed in the following sections:

- [Connection Pooling](#)
- [Session Multiplexing](#)

Connection Pooling When thousands of clients are running interactive Web applications, many of these sessions may be idle at a given time. **Connection pooling** enables the database server to timeout an idle session and use the connection to service an active session. The idle logical session remains open, and the physical connection is automatically reestablished when the next request comes from that session. Therefore, Web applications can allow larger numbers of concurrent users to be accommodated with existing hardware.

Figure 1–11 shows how connection pooling works. In this example, the Oracle database server has been configured with 255 connections. One of the clients has been idle past a specified amount of time. Connection pooling makes this connection available to an incoming client connection, which is the 256th connection. When the idle client has more work to do, the connection is reestablished for that client with another client's idle connection.

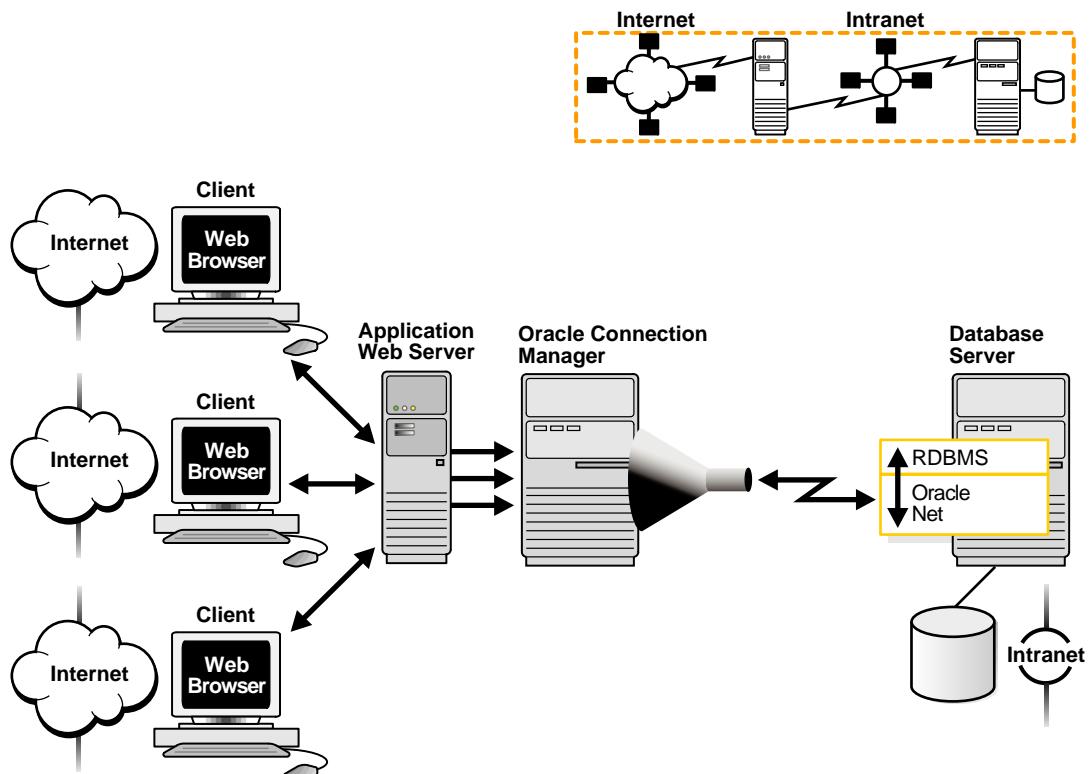
Figure 1–11 Connection Pooling



Session Multiplexing **Oracle Connection Manager**, an Oracle Net Services component, enables multiple client network sessions to be multiplexed, or funneled, through a single network connection to a database.

Session multiplexing reduces the demand on resources needed to maintain multiple network sessions between two processes by enabling the server to use fewer network connection endpoints for incoming requests. This enables you to increase the total number of network sessions that a server can handle. With multiple Oracle Connection Managers, thousands of concurrent users can connect to a server.

[Figure 1-12](#) on page 1-17 shows how session multiplexing can be used in a Web architecture. When Oracle Connection Manager is run on the same computer as an application Web server, the application Web server can route multiple client sessions through Oracle Connection Manager to ensure that those sessions have continuous access to an Oracle database server. This functionality is especially useful for Web applications where session availability and response time are major concerns.

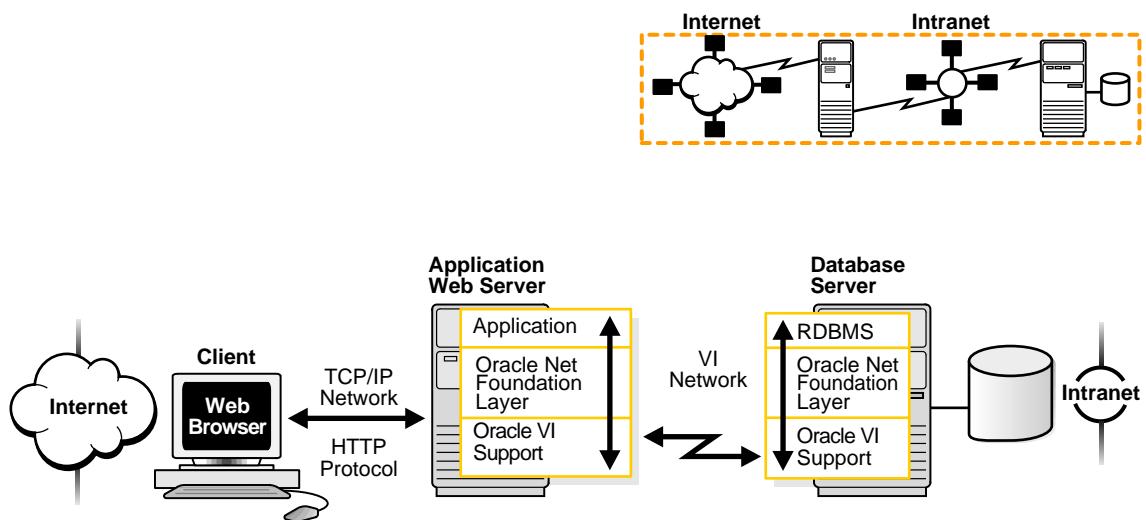
Figure 1–12 Session Multiplexing

Performance Between Application Web Servers and Oracle Databases

The **Virtual Interface (VI) protocol** reduces the overhead of TCP/IP by eliminating intermediate replication of data and transferring most of the messaging burden away from the CPU and onto the network hardware. The result is a low-latency, high-throughput interconnect protocol that reduces the amount of CPU cycles dedicated to network processing.

In an environment with an application Web server, the VI protocol can be used in place of TCP/IP between the application Web server and the database server. This configuration is shown in [Figure 1–13](#).

Figure 1–13 VI Protocol to Improve Performance



See Also: <http://www.viarch.org/> for the VI specification

Network Security

Data access and secure transfer of data are important considerations when deploying Oracle. Network security is enhanced with features described in the following topics:

- [Database Access Control](#)
- [Enhanced Network Security Through Oracle Advanced Security](#)

Database Access Control

Granting and denying access to a database is crucial for a secure network environment. Oracle Net Services enable database access control using features described in the following topics:

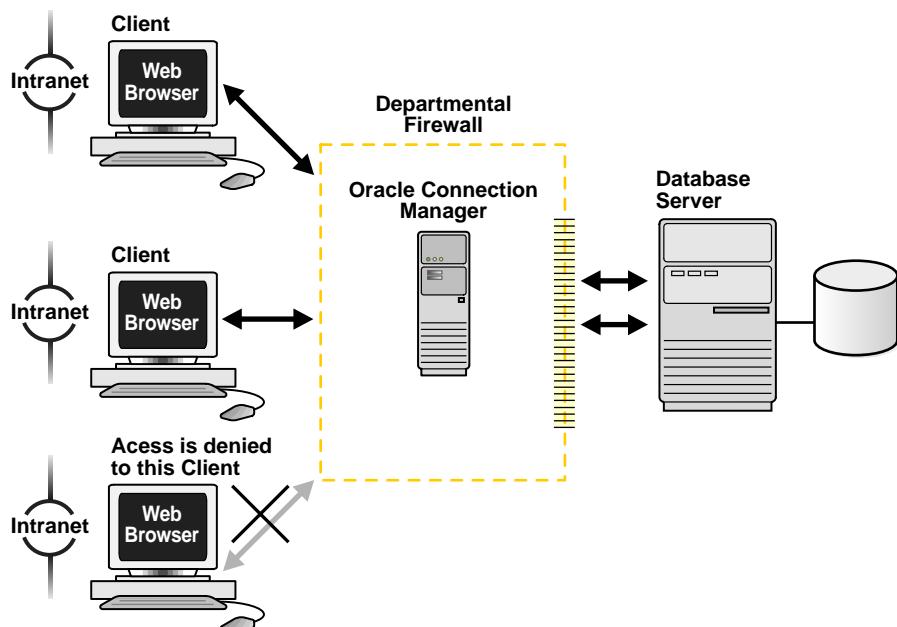
- [Firewall Access Control](#)
- [Protocol Access Control](#)

Firewall Access Control Oracle Connection Manager can be configured to grant or deny client access to a particular database service or a computer. By specifying filtering rules, you can allow or restrict specific client access to a server, based on the following criteria:

- Source host names or IP addresses for clients
- Destination host names or IP addresses for servers
- Destination database service names
- Client use of **Oracle Advanced Security**

Figure 1–14 shows an Oracle Connection Manager positioned between three Web clients and an Oracle database server. Oracle Connection Manager is configured to allow access to the first two Web clients and to deny access to the third. In order for this configuration to work, clients require the JDBC Thin driver.

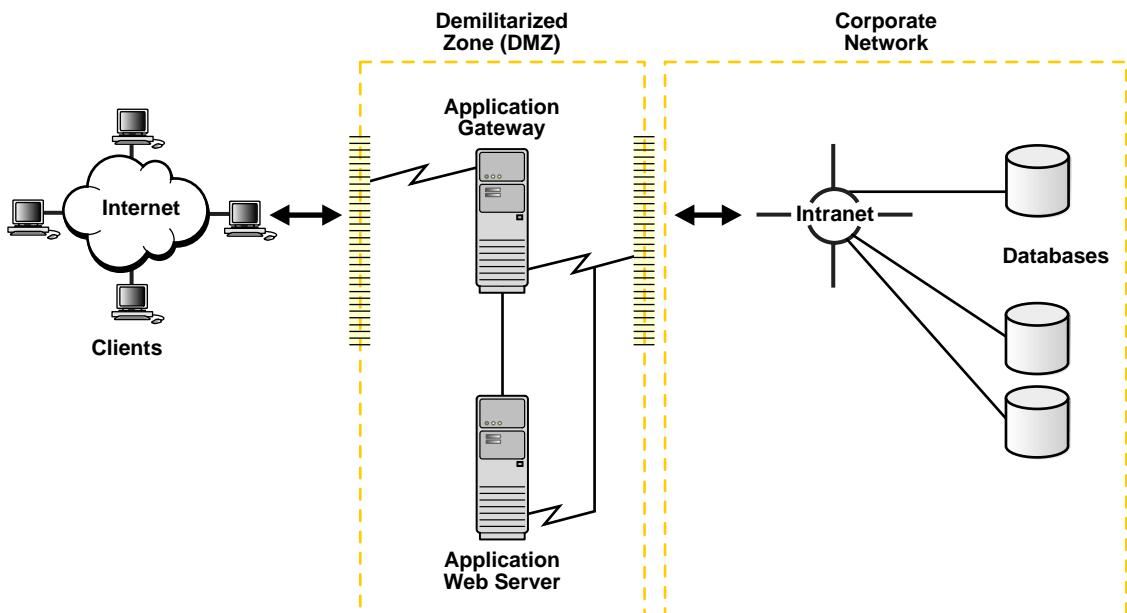
Figure 1–14 Intranet Network Access Control with Oracle Connection Manager



Oracle Connection Manager functionality is also offered by some firewall vendors through a software component called **Oracle Net Firewall Proxy**. A host computer, called an **application gateway**, runs the Oracle Connection Manager software.

Figure 1–15 shows an application gateway controlling traffic between internal and external networks and providing a single checkpoint for access control and auditing. As a result, unauthorized Internet hosts cannot directly access the database inside a corporation, but authorized users can still use Internet services outside the corporate network. This capability is critical in Internet environments to restrict remote access to sensitive data.

Figure 1–15 Internet Network Access Control with an Application Gateway



Protocol Access Control The database server can be configured with access control parameters in the `sqlnet.ora` configuration file. These parameters specify whether clients are allowed or denied access based on the protocol.

Enhanced Network Security Through Oracle Advanced Security

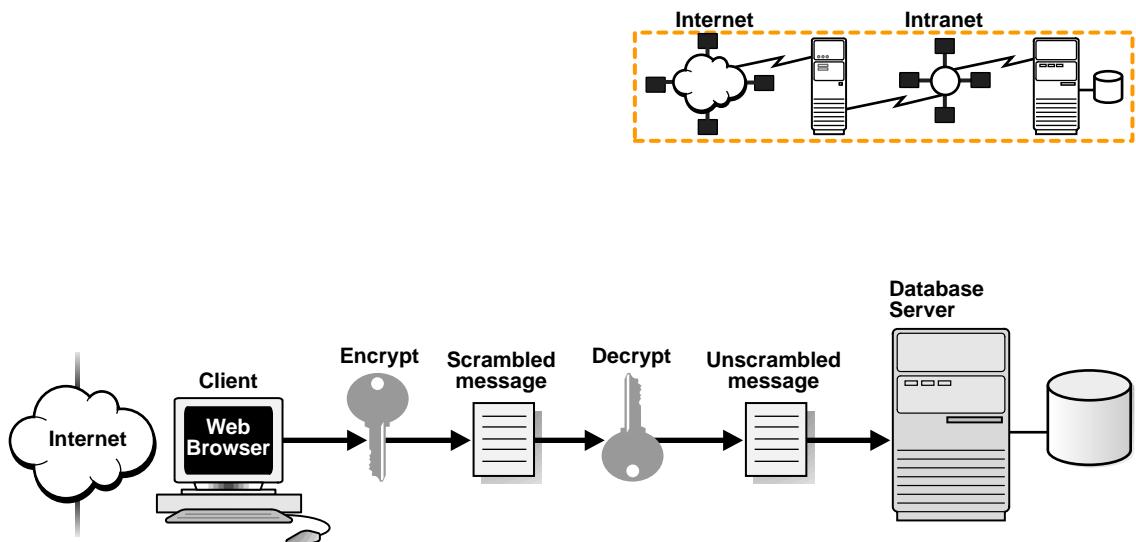
Oracle Advanced Security provides network security through the features described in the following topics:

- [Data Encryption](#)
- [User Authentication and Single Sign-On](#)
- [Enterprise User Security](#)

Data Encryption Sensitive information that travels over the Internet can be protected by encryption. Encryption is the mutation of information into a form readable only with a decryption key.

Figure 1–16 shows how encryption works. To ensure the security of the transaction, the buyer wishes to purchase a company’s product over the Internet using a credit card. The buyer’s credit card number is encrypted with an encryption key. The encrypted credit number is sent across the network to the database. Encryption scrambles the message, rendering it unreadable to anyone but the recipient. The server decrypts the message with a decryption key and reads the credit card number.

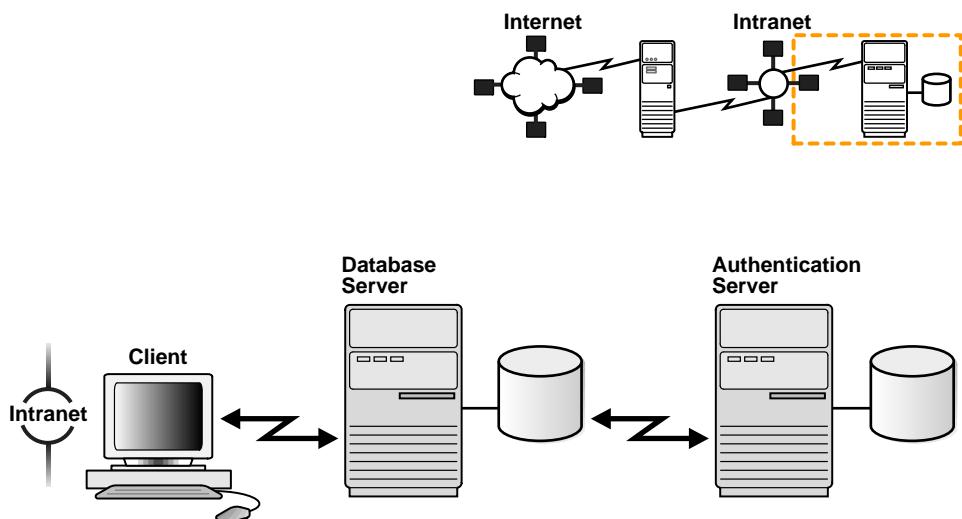
Figure 1–16 Encryption



User Authentication and Single Sign-On Authentication is used to prove the identity of the user. Passwords are the most common means of authentication. Oracle Advanced Security allows for enhanced authentication through Oracle authentication adapters that support various third-party authentication services.

[Figure 1-17](#) shows user authentication with an Oracle database configured with an authentication server. This architecture provides high confidence in the identity of users in distributed environments. Having a central facility to authenticate all members of the network (clients to servers, servers to servers, users to both clients and servers) is one effective way to address the threat of network nodes falsifying their identities.

Figure 1–17 User Authentication



Centralized authentication also provides the benefit of [single sign-on](#) for users. Single sign-on enables users to access multiple accounts and applications with a single password. A user only needs to log on once and can then automatically connect to any other service without having to give a user name and password again. Single sign-on eliminates the need for the user to remember and administer multiple passwords, and reduces the time spent logging into multiple services.

Authentication is provided through the following third-party services:

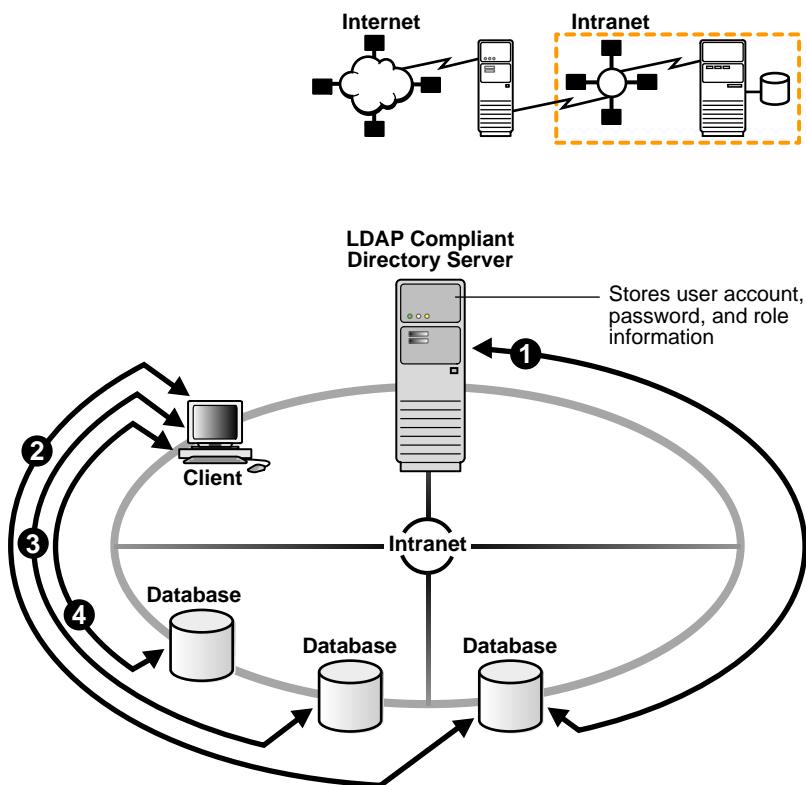
- **Secure Sockets Layer (SSL)**
- RADIUS
- CyberSafe
- Kerberos
- Distributed Computing Environment (DCE)

Enterprise User Security A centralized directory server can be used to store user, administration, and security information. This centralized configuration enables the administrator to modify information in one location, the directory. It also lowers the cost of administration and makes the enterprise more secure.

[Figure 1–18](#) shows how a directory server can be used to provide centralization of user account, user role, and password information.

1. A database server authenticates a user by accessing information stored in the directory.
- 2.-4. Once authenticated, a user can access the databases, which are configured for enterprise user security.

Figure 1–18 Centralized Storage of Security Configuration with a Directory Server



A Suite of Networking Components

The connectivity, manageability, scalability, and security features described in this chapter are provided by the following components and products, which are discussed in the following topics:

- [Oracle Net Services](#)
- [Oracle Advanced Security](#)

Oracle Net Services

This section describes the following components:

- [Oracle Net](#)
- [Listener](#)
- [Oracle Connection Manager](#)
- [Networking Tools](#)

Oracle Net

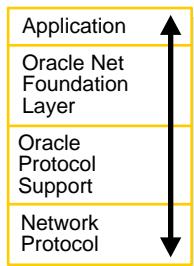
Oracle Net is a software layer that resides on the client and the Oracle database server. It is responsible for establishing and maintaining the connection between the client application and server, as well as exchanging messages between them, using industry-standard protocols. Oracle Net is comprised of two software components:

- Oracle Net foundation layer
- Oracle protocol support

Oracle Net Foundation Layer On the client side, applications communicate with Oracle Net foundation layer to establish and maintain connections. The Oracle Net foundation layer, in turn, uses Oracle protocol support that is able to communicate with an industry-standard network protocol, such as TCP/IP, to communicate with the Oracle database server.

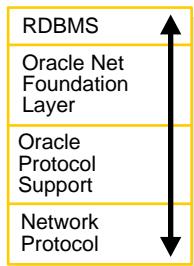
[Figure 1–19](#) illustrates the communication stack on the client.

Figure 1–19 Oracle Net on the Client



The Oracle database server side is similar to the client side as illustrated in [Figure 1–20](#). A network protocol sends client request information to an Oracle protocol support layer, which then sends information to the Oracle Net foundation layer. The Oracle Net foundation layer then communicates with the Oracle database server to process the client request.

Figure 1–20 Oracle Net on the Server



Oracle Protocol Support The Oracle Net foundation layer uses Oracle protocol support to communicate with the following industry-standard network protocols:

- TCP/IP
- TCP/IP with SSL
- Named Pipes
- LU6.2
- VI

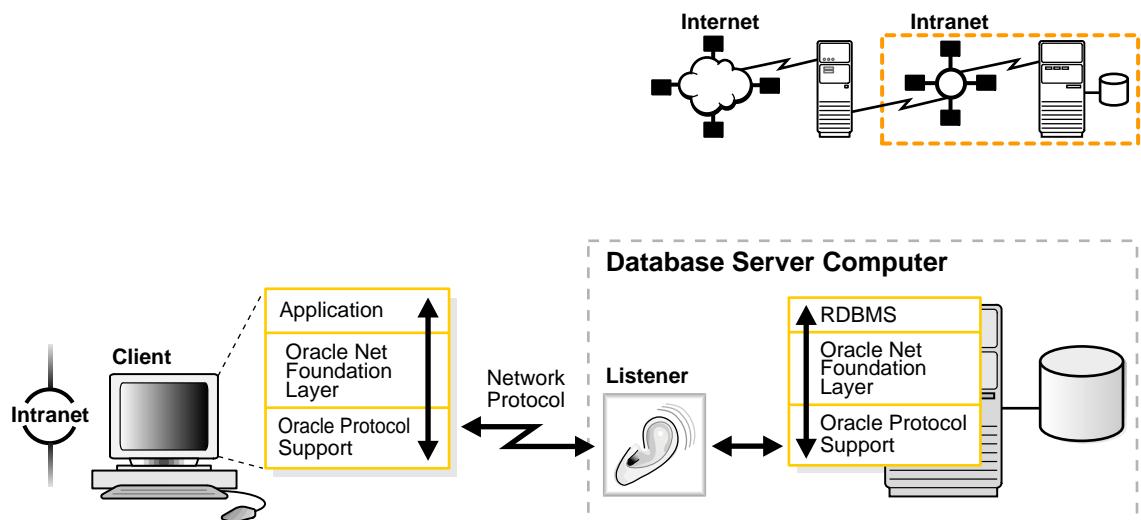
Oracle protocol support maps Oracle Net foundation layer functionality to industry-standard protocols used in client/server connections.

Listener

The one operation unique to the Oracle database server side is the act of receiving the initial connection through a process called the **listener**. The listener brokers a client request, handing off the request to the server. The listener is configured with a protocol address. Clients configured with the same protocol address can send connection requests to the listener. Once a connection is established, the client and Oracle database server communicate directly with one another.

[Figure 1–21](#) shows a listener accepting a connection request from a client and forwarding that request to an Oracle database server.

Figure 1–21 Listener in a Connection Request



Oracle Connection Manager

Oracle Connection Manager is a software component that resides on its own computer, separate from a client or an Oracle database server. It proxies requests destined for the database server. You can also configure Oracle Connection Manager to multiplex sessions, control access, or convert protocols.

In its session multiplexing role, Oracle Connection Manager funnels multiple sessions through a single transport protocol connection to a particular destination. This reduces the demand on resources needed to maintain multiple sessions between two processes by enabling the Oracle database server to use fewer connection end points for incoming requests.

As an access control filter, Oracle Connection Manager controls access to Oracle databases.

As a protocol converter, Oracle Connection Manager enables a client and an Oracle database server that have different networking protocols to communicate with each other.

See Also:

- ["Session Multiplexing"](#) on page 1-16 for an explanation of routing
- ["Firewall Access Control"](#) on page 1-20 for a description of filtering

Oracle Connection Manager functionality is also offered by some firewall vendors through Oracle Net Firewall Proxy. A host computer, or application gateway, runs the Oracle Connection Manager proxy software. An application gateway looks and acts like a real server from the client's point of view, and a real client from the server's point of view. The application gateway sits between the Internet and a company's internal network and provides middleman services to users on either side.

Note: Ask your firewall vendor if its product supports Oracle Net Firewall Proxy.

Networking Tools

Oracle Net Services provide graphical user interface tools and command-line utilities that enable you to easily configure, manage, and monitor the network.

Oracle Net Configuration Assistant is a tool that configures basic network components after installation, including:

- Listeners
- Naming methods
- Directory server usage

Oracle Net Manager combines configuration abilities with component control to provide an integrated environment for configuring and managing Oracle Net Services. With Oracle Net Manager, you can fine-tune the listener and naming method configuration created with Oracle Net Configuration Assistant. In addition, Oracle Net Manager offers built-in wizards and utilities that enable you to test connectivity, migrate data from one naming method to another, and create additional network components.

The command-line control utilities enable you to configure, administer, and monitor network components, such as listeners and Oracle Connection Managers.

See Also: [Chapter 5, "Configuration and Administration Tools Overview"](#)

Oracle Advanced Security

Oracle Advanced Security is a separately licensable product that provides a comprehensive suite of security features for the Oracle environment. This suite of security features protects enterprise networks and securely extends corporate networks to the Internet. It provides a single source of integration with network encryption and authentication solutions, single sign-on services, and security protocols. Oracle Advanced Security integrates industry standards and delivers unparalleled security to the Oracle network and other networks.

See Also: *Oracle Advanced Security Administrator's Guide*

2

Connectivity Concepts

This chapter explains how databases are identified and how clients access them.

This chapter contains these topics:

- [Database Service and Database Instance Identification](#)
- [Service Accessibility](#)
- [Enhanced Service Accessibility with Multiple Listeners](#)
- [Service Handlers](#)
- [Naming](#)

See Also: [Chapter 1, "Networking Challenges in the Internet Age"](#)
for an introductory level overview of networking concepts

Database Service and Database Instance Identification

This section contains these topics:

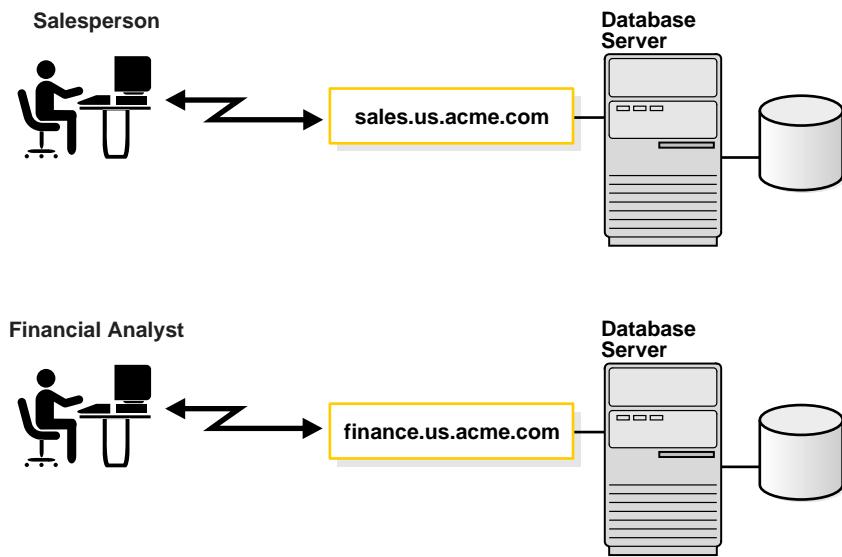
- [Database Services](#)
- [Database Instances](#)

Database Services

An Oracle database is represented to clients as a **service**; that is, the database performs work on behalf of clients. A database can have one or more services associated with it.

[Figure 2–1](#) shows two databases, each with its own database service for intranet clients. One service, `sales.us.acme.com`, enables salespersons to access the sales database. Another service, `finance.us.acme.com`, enables financial analysts to access the finance database.

Figure 2–1 One Service Per Database



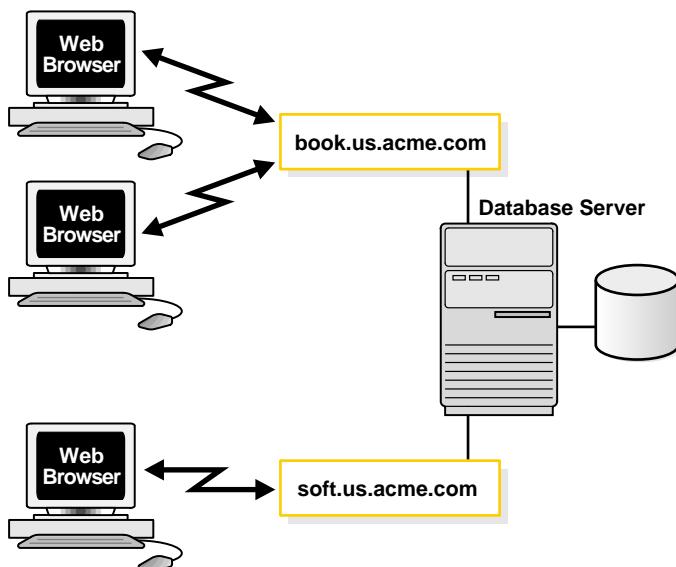
The sales and finance databases are identified by **service names**, `sales.us.acme.com` and `finance.us.acme.com` in this example. The service name is specified by the `SERVICE_NAMES` parameter in the initialization parameter file. The service name defaults to the **global database name**, a name comprised of the database name (`DB_NAME` parameter) and domain name (`DB_DOMAIN`)

parameter). In the case of `sales.us.acme.com`, `sales` is the database name and `us.acme.com` is the domain name.

Note: You can change the value of `SERVICE_NAMES` parameter dynamically with the SQL statement `ALTER SYSTEM` when the database is running. See *Oracle9i SQL Reference* for further information about the `ALTER SYSTEM` statement, and *Oracle9i Database Reference* for further information about the `SERVICE_NAMES` parameter.

A database can have multiple services associated with it. [Figure 2-2](#) shows one database that has two different services for Web clients. One service, `book.us.acme.com`, is dedicated to clients making book purchases. The other service, `soft.us.acme.com`, is dedicated to clients making software purchases.

Figure 2-2 Multiple Services Associated with One Database



Having multiple services associated with one database enables the following functionality:

- A single database can be identified in a number of different ways by different clients.
- A database administrator can limit or reserve system resources. This level of control enables better allocation of resources to clients requesting one of these services

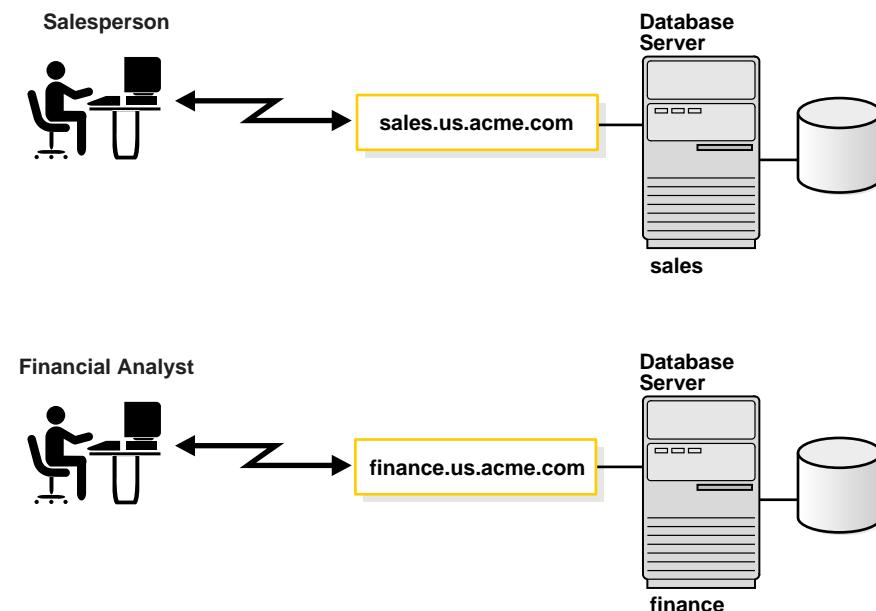
Database Instances

A database has one or more **instances**. An instance is comprised of a memory area called the **System Global Area (SGA)** and Oracle processes. The memory and processes of an instance efficiently manage the associated database's data and serve the database users.

Note: An instance also manages other services, such as the **Oracle9i JVM** and the Oracle Servlet Engine.

[Figure 2–3](#) shows two instances, sales and finance, associated with their respective databases.

Figure 2–3 One Instance Per Database

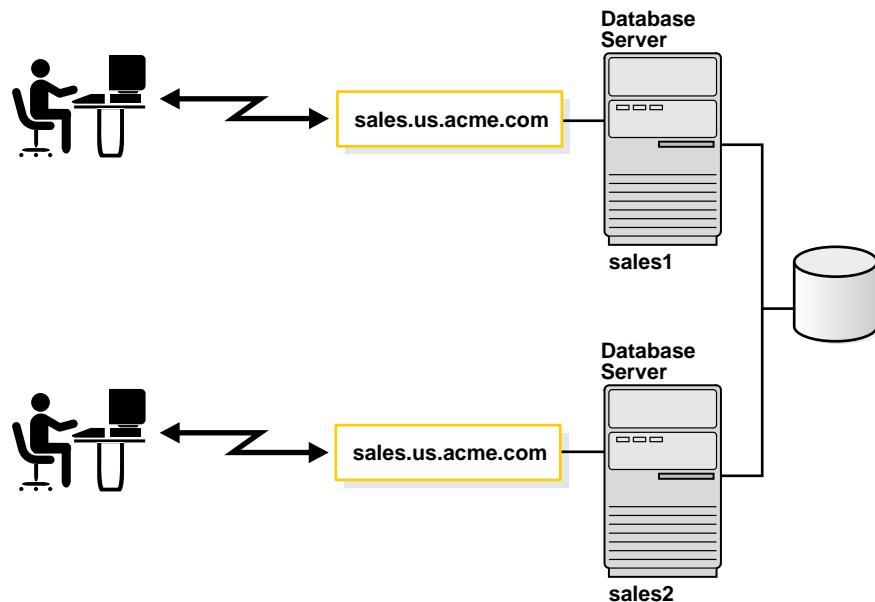


Like services, instances are identified by an **instance name**, sales and finance in this example. The instance name is specified by the `INSTANCE_NAME` parameter in the initialization parameter file. The instance name defaults to the **Oracle System Identifier (SID)** of the instance.

Some hardware architectures allow multiple computers to share access to data, software, or peripheral devices. Oracle9*i* Real Application Clusters can take advantage of such architecture by running multiple instances on different computers that share a single physical database.

Figure 2–4 shows an Oracle9*i* Real Application Clusters configuration. In this example, two instances, sales1 and sales2, are associated with one database service, sales.us.acme.com.

Figure 2–4 Multiple Instances Associated with a Database



Service Accessibility

To connect to a database service, clients use a **connect descriptor** that provides the location of the database and the name of the database service. The following example shows a connect descriptor that enables clients to connect to a database service called sales.us.acme.com.

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)))
```

The address portion of the connect descriptor is actually the protocol address of the **listener**. To connect to a database service, clients first contact a listener process that typically resides on the database server. The listener receives incoming client connection requests and hands these requests to the database server. Once the connection is established, the client and database server communicate directly.

Much like a business address, the listener is configured to accept requests from clients at a **protocol address**. This address defines the protocol the listener is listening on and any other protocol specific information. For example, the listener could be configured to listen at the following protocol address:

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)))
```

This example shows a TCP/IP protocol address that specifies the host of the listener and a port number. Clients configured with this same protocol address can send connection requests to this listener.

The connect descriptor also specifies the database service name with which clients seek to establish a connection. The listener knows which services can be sent connection requests, because an Oracle database dynamically registers this information with the listener. This process of registration is called **service registration**. It also provides the listener with information about the database instances and the **service handlers** available for each instance. Service handlers act as connection points to an Oracle database server. A service handler can be a **dispatcher** or a **dedicated server**.

See Also:

- "Service Handlers" on page 2-9 for a description of these service handler types
- "Listener Architecture" on page 4-10

If connecting to a specific instance of the database is required, clients can also specify the `INSTANCE_NAME` of a particular instance in the connect descriptor. This feature can be useful if you have an Oracle9i Real Application Clusters configuration with multiple instances. For example, the following connect descriptor specifies an instance name of `sales1` that is associated with `sales.us.acme.com`.

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)
    (INSTANCE_NAME=sales1)))
```

Optionally, clients that always want to use a particular service handler type can use a connect descriptor that specifies the service handler type. In the following example, a connect descriptor is configured to use a dispatcher for a shared server configuration, as indicated by (`SERVER=shared`).

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)
    (SERVER=shared)))
```

If you want the client to use a dedicated server, you can specify (`SERVER=dedicated`) in place of (`SERVER=shared`). If the `SERVER` parameter is not set, then shared server configuration is assumed. However, the client will use a dedicated server if not dispatchers are available.

When the listener receives the client request, it selects one of the service handlers that were previously registered. Depending on the type of handler selected, the communication protocol used, and the operating system of the database server, the listener performs one of the following actions:

- Hands the connect request directly off to a dispatcher.
- Sends a redirect message back to the client with the location of the dispatcher or dedicated server process. The client then connects directly to the dispatcher or dedicated server process.
- Spawns a dedicated server process and passes the client connection to the dedicated server process.

Once the listener has completed the connection operation for the client, the client communicates with the Oracle database server without the listener's involvement. The listener resumes listening for incoming network sessions.

Enhanced Service Accessibility with Multiple Listeners

For some configuration, such as Oracle9*i* Real Application Clusters, multiple listeners on multiple nodes can be configured to handle client connection requests for the same database service. A multiple-listener configuration also enables you to leverage the following failover and load balancing features:

- [Connect-Time Failover](#)
- [Transparent Application Failover](#)
- [Client Load Balancing](#)

These features can be implemented either singly or in combination with each other.

Connect-Time Failover

Connect-time failover enables clients to request a different listener if the first listener fails.

Transparent Application Failover

Transparent Application Failover (TAF) is a runtime failover for high-availability environments, such as Oracle9*i* Real Application Clusters and Oracle Real Application Clusters Guard. TAF fails over and reestablishes application-to-service connections. It enables client applications to automatically reconnect to the database if the connection fails and, optionally, resume a `SELECT` statement that was in progress. The reconnection happens automatically from within the Oracle Call Interface (OCI) library.

Client Load Balancing

Client load balancing enables clients to randomize requests to multiple listeners. This feature distributes the load so that individual listeners do not become overburdened.

Connection Load Balancing

Connection load balancing improves connection performance by balancing the number of active connections among multiple dispatchers. In a single-instance environment, the listener selects the least loaded dispatcher to handle the incoming client requests. In an Oracle9*i* Real Application Clusters environment, connection load balancing also has the capability to balance the number of active connections among multiple instances.

Due to dynamic service registration, a listener is always aware of all instances and dispatchers regardless of their location. Depending on the load information, a listener decides which instance and, if shared server is configured, which dispatcher to send the incoming client request to.

In a shared server configuration, a listener selects a dispatcher in the following order:

1. Least-loaded node
2. Least-loaded instance
3. Least-loaded dispatcher for that instance

In a dedicated server configuration, a listener selects an instance in the following order:

1. Least loaded node
2. Least loaded instance.

If a database service has multiple instances on multiple nodes, the listener chooses the least loaded instance on the least loaded node. If shared server is configured, then the least loaded dispatcher of the selected instance is chosen.

Service Handlers

This section contains these topics:

- [Dispatchers](#)
- [Dedicated Server Processes](#)

Dispatchers

The shared server architecture uses a dispatcher process to direct client connections to a common request queue. An idle shared server process from a shared pool of server processes picks up a request from the common queue. This approach enables a small pool of server processes to serve a large number of clients. A significant advantage of the shared server model over the dedicated server model is reduced system resources, enabling support of an increased number of users.

The listener uses the dispatcher as a type of service handler to which it can direct client requests. When a client request arrives, the listener performs one of the following actions:

- Hands the connection request directly to a dispatcher.
- Issues a redirect message to the client, containing the protocol address of a dispatcher. The client then terminates the network session to the listener and establishes a network session to the dispatcher, using the network address provided in the redirect message.

The listener uses direct hand off whenever possible. Redirect messages are used, for example, when dispatchers are remote to the listener.

[Figure 2–5](#) shows the listener handing a connection request directly off to a dispatcher.

1. The listener receives a client connection request.
2. The listener hands the connect request directly to the dispatcher.
3. The client is now connected to the dispatcher.

Figure 2–5 Direct Hand-Off to a Dispatcher

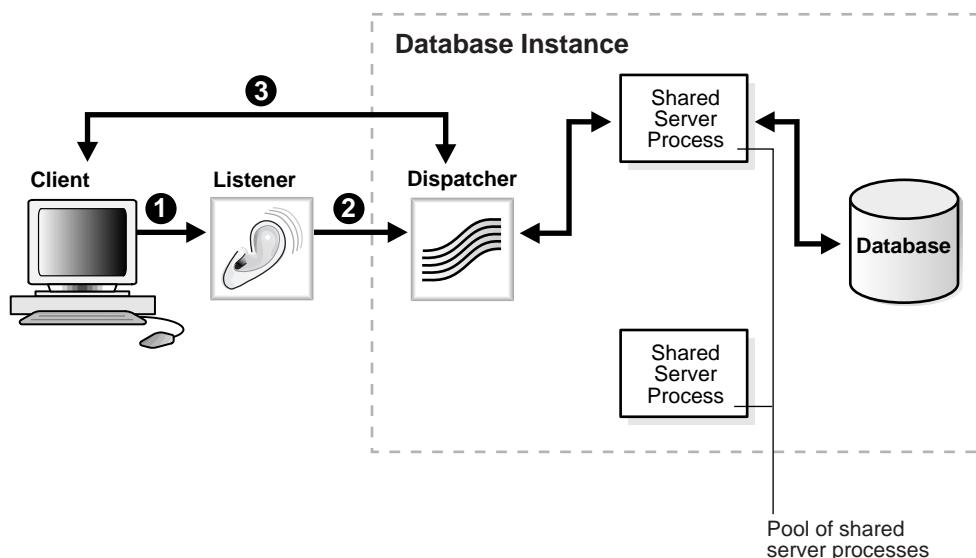
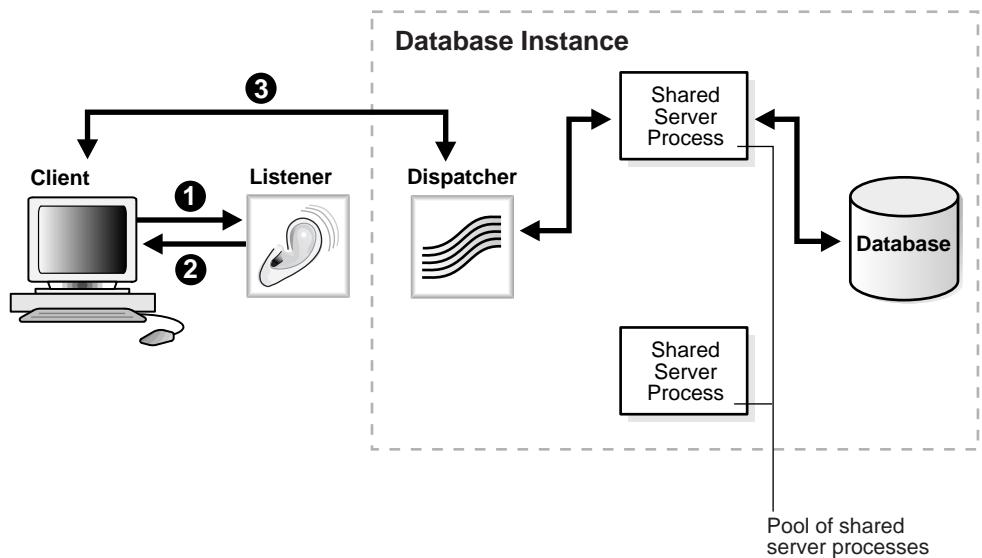


Figure 2–6 shows the role of a dispatcher in a redirected connection.

1. The listener receives a client connection request.
2. The listener provides the location of the dispatcher to the client in a redirect message.
3. The client connects directly to the dispatcher.

Figure 2–6 Redirected Connection to a Dispatcher



Dedicated Server Processes

In a dedicated server configuration, the listener starts a separate dedicated server process for each incoming client connection request dedicated to servicing the client. Once the session is complete, the dedicated server process terminates. Because a dedicated server process has to be started for each connection, this configuration may require more system resources than shared server configurations.

A dedicated server process is a type of service handler that the listener starts when it receives a client request. To complete a client/server connection establishment, one of the following actions occurs:

- The dedicated server inherits the connection request from the listener.
- The dedicated server informs the listener of its listening protocol address. The listener passes the protocol address to the client in a redirect message and terminates the connection. The client connects to the dedicated server directly using the protocol address.

Note: One of the options is selected based on the operating system and the transport protocol.

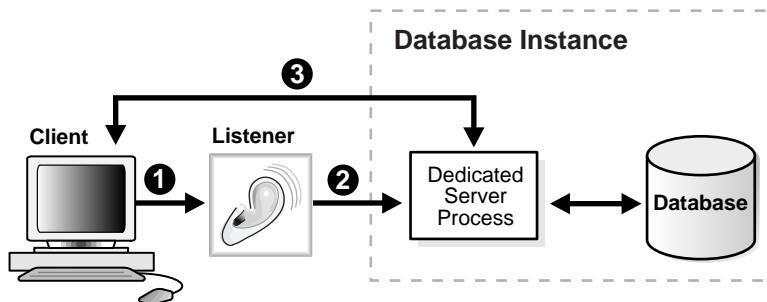
If the client and database server exist on the same computer, a client connection can be passed directly to a dedicated server process without going through the listener. The application initiating the session spawns a dedicated server process for the connection request. This happens automatically if the application that is used to start the database is on the same computer as the database.

Note: In order for remote clients to connect to dedicated servers, the listener and the database instance must be running on the same computer.

[Figure 2–7](#) shows the listener passing a client connection request to a dedicated server process.

1. The listener receives a client connection request.
2. The listener starts a dedicated server process, and the dedicated server inherits the connection request from the listener.
3. The client is now connected directly to the dedicated server.

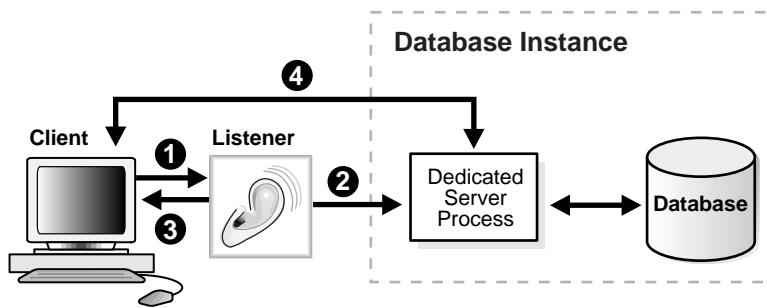
Figure 2–7 Connection to a Dedicated Server Process



[Figure 2–8](#) shows the role of a dedicated server in a redirected connection.

1. The listener receives a client connection request.
2. The listener starts a dedicated server process.
3. The listener provides the location of the dedicated server process to the client in a redirect message.
4. The client connects directly to the dedicated server.

Figure 2–8 Redirected Connection to a Dedicated Server Process



Naming

Naming is generally used to identify an entity with a simple name. Oracle Net uses this simple name, called a **connect identifier**, to identify a connect descriptor. A connect identifier is specified in several different ways. One of the most common ways is through use of a **net service name**, another name for the service, which maps to a connect descriptor.

The following example shows a net service name called `sales` mapped to a connect descriptor:

```
sales=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)))
```

Note: Besides complete connect descriptors, naming can also be used to map a name to an address or address list.

Users initiate a connect request by providing a **connect string**. A connect string includes a user name and password, along with a connect identifier, or the complete connect descriptor for the service to which they want to connect. The following examples demonstrate one connect string that uses a complete connect descriptor and another connect string that uses the `sales` net service name.

```
CONNECT
scott/tiger@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server1)(PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)))
```

```
CONNECT scott/tiger@sales
```

When a net service name is used, connection processing takes place by first mapping `sales` to the connect descriptor. This mapped information is stored in one or more repositories of information that are accessed with **naming methods**.

The process for establishing a client session with the aid of a naming method is as follows:

1. The client initiates a connect request providing a connect identifier.
2. The connect identifier is resolved to a connect descriptor by a naming method. This information is returned to the client.
3. The client makes the connect request to the address provided in the connect descriptor.
4. A listener receives the request and directs it to the appropriate database server.
5. The connection is accepted by the database server.

Oracle Net provides these naming methods:

- [Local Naming](#)
- [Directory Naming](#)
- [Oracle Names](#)
- [Host Naming](#)
- [External Naming](#)

These naming methods have the following characteristics:

Local Naming

Local naming stores net service names and their connect descriptors in a localized configuration file named `tnsnames.ora`.

Directory Naming

Directory naming stores net service names and database service names in a centralized LDAP-compliant [directory server](#) to access a database service.

Oracle Names

Oracle Names uses Oracle proprietary software to store the names and addresses of all database services on a network. Clients wishing to connect to a database server direct their connect requests to an Oracle Names server. Oracle Names servers resolve the name to a network address and return that information to the client.

Note: In future releases, Oracle Names will not be supported as a centralized naming method. Because no new enhancements are being added to Oracle Names, consider using directory naming instead or migrating an existing Oracle Names configuration to directory naming, as described in [Chapter 10, "Exporting Naming Data to a Directory Server"](#).

Host Naming

Host naming enables users to connect to an Oracle database server by using a host name alias. Host names are mapped to the database server's global database name in an existing names resolution service, such as **Domain Name System (DNS)**, **Network Information Service (NIS)**, or a centrally maintained set of `/etc/hosts` files.

External Naming

External naming stores net service names and their connect descriptors in a supported non-Oracle naming service. These supported third-party services include:

- NIS External Naming
- Distributed Computing Environment (DCE) Cell Directory Service (CDS)

Configuration Management Concepts

This chapter describes how configuration information for Oracle Net Services can be stored in localized configuration files or centralized in a directory server.

The topics covered include:

- [Configuration Models](#)
- [Localized Configuration File Support](#)
- [Directory Server Support](#)
- [Oracle Names Support](#)

Configuration Models

Configuration information can be stored in a localized configuration file or a centralized repository, as described in the [Table 3–1](#).

Table 3–1 Oracle Net Configuration Models

Network Configuration Model	Description
Localized management	Network address information stored in <code>tnsnames.ora</code> files on each computer in the network.
Centralized management	Network address information is stored in centralized directory services, including a LDAP-compliant directory server or an Oracle Names server . Note: In future releases, Oracle Names will not be supported as a centralized naming method.

Localized Configuration File Support

Depending on the configuration model used, network computers can be configured with the files described in [Table 3–2](#).

Table 3–2 Oracle Net Configuration Files

Configuration File	Description
<code>ldap.ora</code>	Located on the database server and client computers configured for centralized management features, this file contains parameters necessary to access a directory server.
<code>listener.ora</code>	Located on the database server, this configuration file for the listener includes: <ul style="list-style-type: none">■ Protocol addresses it is accepting connection requests on■ Database and nondatabase services it is listening for■ Control parameters used by the listener
<code>names.ora</code>	Located on the Oracle Names server, this file includes the location, domain information, and optional configuration parameters for an Oracle Names server.

Configuration File	Description
<code>sqlnet.ora</code>	<p>Located on client and database server computer, this file may contain:</p> <ul style="list-style-type: none"> ▪ Client domain to append to unqualified service names or net service names ▪ Order of naming methods the client should use when resolving a name ▪ Logging and tracing features to use ▪ Route of connections ▪ Preferred Oracle Names servers ▪ External naming parameters ▪ Oracle Advanced Security parameters ▪ Database access control parameters
<code>tnsnames.ora</code>	<p>Located on the clients, this file contains net service names mapped to connect descriptors. This file is used for the local naming method.</p>

Configuration files are typically created in `$ORACLE_HOME/network/admin` on UNIX, and `ORACLE_HOME\network\admin` on Windows operating systems. However, configuration files can be created in a variety of places, because Oracle Net searches for the configuration files in the following order:

1. For `sqlnet.ora` file, the current working directory from where an application is run.
2. The directory specified by the `TNS_ADMIN` environment variable
If the `TNS_ADMIN` environment variable is not defined as a variable on Windows NT, it may be in the registry.
3. The node's global configuration directory.
For Sun Solaris, this directory is `/var/opt/oracle`. Windows NT does not have a central directory.
4. The `$ORACLE_HOME/network/admin` directory on UNIX, and the `ORACLE_HOME\network\admin` directory on Windows operating systems.

See Also: Operating system-specific documentation

Directory Server Support

Today, network information is stored in multiple systems and in multiple directory formats. With new requirements for Internet computing and new e-business technologies, a common repository infrastructure is needed as a foundation for management and configuration of all data and resources. This kind of infrastructure reduces the cost of managing and configuring resources in heterogeneous networks.

Support of LDAP-compliant directory servers provides a centralized vehicle for managing and configuring a distributed Oracle network. The directory server can act as a central repository for all data on database network components, user and corporate policies, and user authentication and security, thus replacing clientside and serverside localized `tnsnames.ora` files.

This section contains these topics:

- [Directory Naming Overview](#)
- [Naming Configuration Storage in a Directory Server](#)
- [Directory Entries](#)
- [Adding or Modifying Entries in the Directory Server](#)
- [Client Connections Using Directory Naming](#)
- [Oracle Net Configuration and Directory Server Design](#)

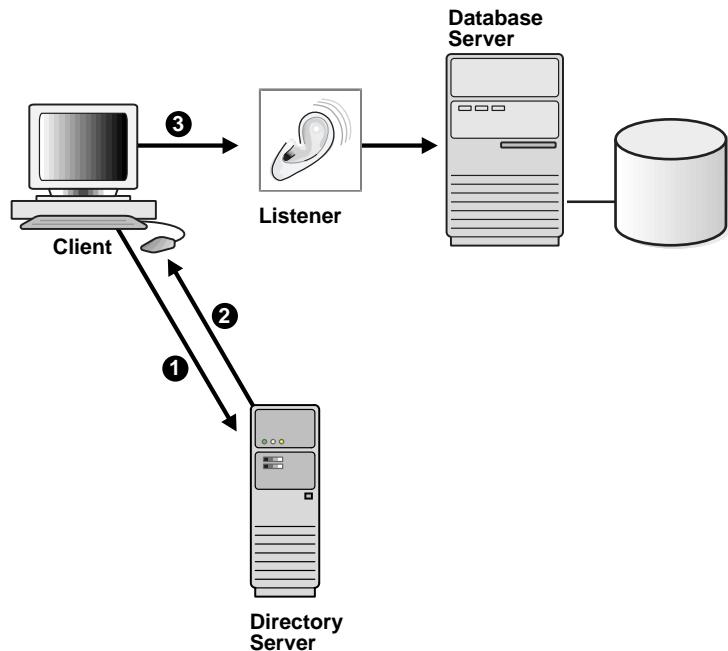
Directory Naming Overview

Oracle Net Services use a centralized directory server as one of the primary methods for storage of **connect identifiers**. Clients configured to use the directory server, use the connect identifiers in their connect string. The directory server resolves the connect identifier to a connect descriptor that is passed back to the client.

[Figure 3-1](#) on page 3-5 shows a client resolving a connect identifier through a directory server.

1. The client contacts the directory server to resolve a connect identifier to a connect descriptor.
2. The directory server resolves the connect identifier and retrieves the connect descriptor for the client.
3. The client sends the connection request to the listener, using the connect descriptor.

Figure 3–1 Client Using a Directory Server to Resolve a Connect Identifier



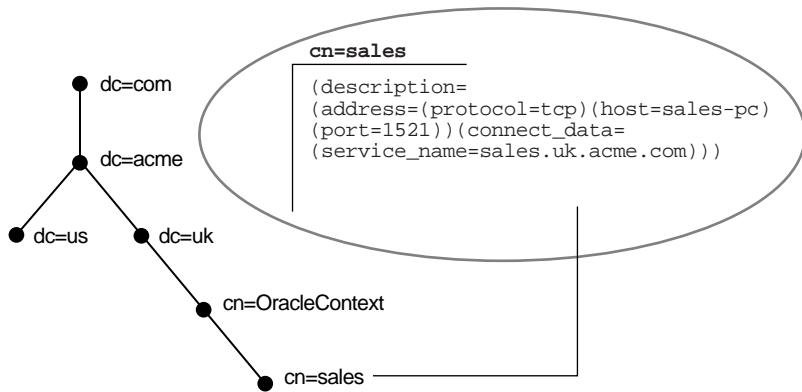
Oracle Net Services support Oracle Internet Directory and Microsoft Active Directory.

Note: JDBC drivers support directory naming. See the *Oracle9i JDBC Developer's Guide and Reference* for further information.

Naming Configuration Storage in a Directory Server

To understand how Oracle Net Services use a directory server, consider [Figure 3–2](#).

Figure 3–2 Database Service in a Directory Server



Each node in the tree is an **entry**. The branch on the right represents the entry for a database service called **sales**, which resides under a hierarchical domain structure of **dc=uk**, **dc=acme**, **dc=com**.

This hierarchy of entries is called the **directory information tree (DIT)**. Each entry is uniquely identified by a **distinguished name (DN)**. The DN tells you exactly where the entry resides in the directory server's hierarchy. The DN for **sales** is **(dn:cn=sales,cn=OracleContext,dc=uk,dc=acme,dc=com)**. Note that the format of a DN places the lowest component of the DIT to the left, then moves progressively up the DIT. Each DN is made up of a sequence of **relative distinguished names (RDNs)**, much the way a directory path contains a sequence of directories. In the entry for **sales**, the RDN is **cn=sales**. A RDN is made up of a set of **attributes**. For example, in **cn=sales**, **cn** is one of the entry's attributes. The attribute, along with its value, uniquely identifies the entry.

Table 3–3 lists the common attributes referred to in this chapter.

Table 3–3 LDAP Directory Common Attributes

Attribute	Description
commonName (cn)	Common name of an entry
country (c)	Name of country
domain component (dc)	Domain component
organization (o)	Name of organization
organizationalUnitName (ou)	Name of a unit within an organization

Notice that `sales` resides under `cn=OracleContext`. The common name `cn=OracleContext` is a special entry in the directory server called an **Oracle Context**. The entries under the Oracle Context support various directory-enabled features, such as **directory naming** and **enterprise user** security.

See Also: *Oracle Advanced Security Administrator's Guide* for further information about enterprise user security

In this example, the Oracle Context contains an entry for the `sales` database service.

During directory server usage configuration, a default directory entry that contains an Oracle Context (`cn=OracleContext`) is selected. The client uses this entry as the default place to locate and look up connect identifiers in the directory server.

The configured directory entry affects the connect string. For example, if a client needs to access the `sales` entry frequently, a reasonable default directory entry would be `dc=uk,dc=acme,dc=com`. Note that `cn=OracleContext` does not have to be explicitly specified, which enables the client to use the following connect string:

```
CONNECT scott/tiger@sales
```

The `sales` entry is relative to the Oracle Context.

If a client's directory entry does not match the directory entry where the service is located, the connect string must specify an entry's absolute name, as described in "Client Connections Using Directory Naming" on page 3-10.

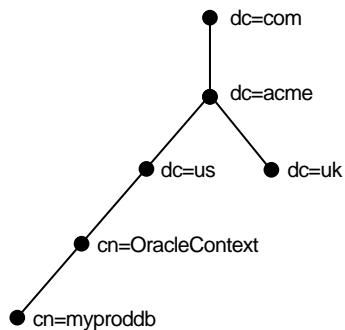
Directory Entries

DITs can be structured in many ways, using:

- Existing Domain Name Space (DNS)
- Internet naming scheme
- Geographical and organization lines

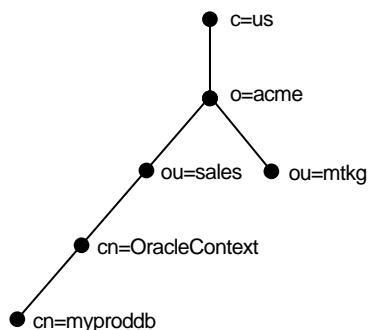
[Figure 3–3](#) shows a DIT structured according to DNS domain components.

Figure 3–3 Domain Component DIT



[Figure 3–4](#) shows a DIT structured according to country, organization, and organizational units. This structure is commonly referred to as an X.500 DIT.

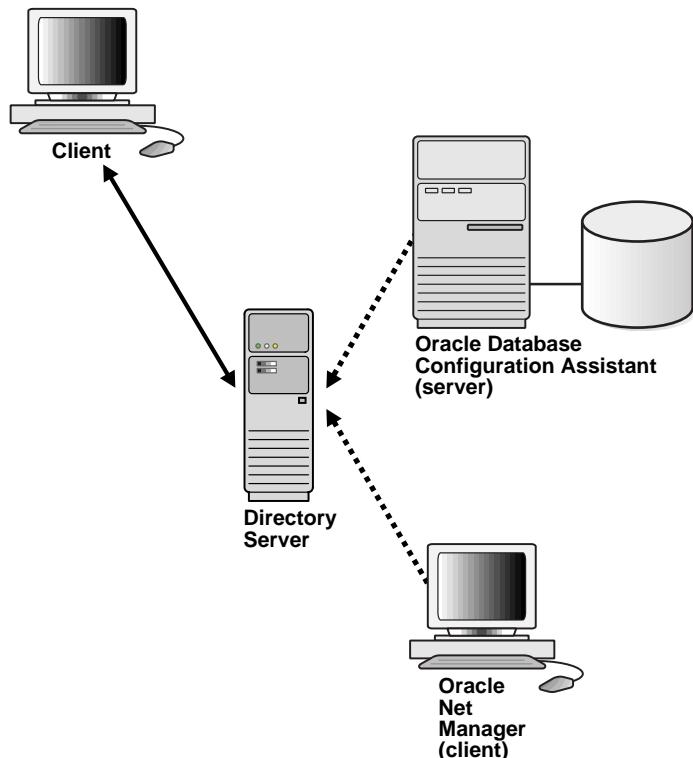
Figure 3–4 X.500 Style DIT



Adding or Modifying Entries in the Directory Server

A database service entry is created with [Oracle Database Configuration Assistant](#) during or after installation. Net service name entries can be created with [Oracle Net Manager](#). Oracle Net Manager can also be used to modify the Oracle Net attributes of the database service entries. [Figure 3–5](#) shows how Oracle Database Configuration Assistant and Oracle Net Manager interface with the directory server.

Figure 3–5 Creating Entries in the Directory Server with Applications



Clients configured for directory server access, as described in "[Client Connections Using Directory Naming](#)" on page 3-10, can connect to a database using entries created by these applications.

In order to use these applications to add entries, an Oracle Context (`cn=OracleContext`) must be created. The Oracle Context is created during directory server access configuration with [Oracle Net Configuration Assistant](#).

To create a database service entry, the administrator using Oracle Database Configuration Assistant must be a member of the OracleDBCreators group (`cn=OracleDBCreators, cn=Oraclecontext`). Likewise, the administrator using Oracle Net Manager to add net service names or Oracle Net attributes for database service or net service names entries must be a member of the OracleNetAdmins group (`cn=OracleNetAdmins, cn=OracleContext`). The directory user that created the Oracle Context is automatically added to both these groups, making this user the Oracle Context administrator. Other users can be added to this group by the directory administrator.

Client Connections Using Directory Naming

Most clients only need to perform name lookups in the directory server. To perform a lookup, the following minimum requirements must be met:

- The client must have anonymous authentication with the directory server.
- The client must be able to read access to Oracle Net entries located in the Oracle Context.
- The client must be configured for directory server usage.

Oracle Net Configuration Assistant typically performs the necessary directory server access configuration during client installation and stores the following information in a read-only `ldap.ora` file:

- Type of directory server
- Location of the directory server
- Default location in the directory server that contains an Oracle Context from which this client can look up connect identifiers

The client reads this file to locate the directory server and Oracle Net entries to access.

See Also: [Chapter 8, "Setting Up Directory Server Usage"](#) for information about configuring directory server access

In the same way they might use other naming methods, clients make connections to a database using one of the two types of connect identifiers stored in the directory server. Database service and net service entries can be referred to by their common names, or they can require additional directory location information in the connect string. How the connect identifier is specified in a connect string depends on how the client's default directory entry is configured.

With directory naming, an entry may be identified in one of two ways:

- [Using the Entry's Relative Name](#)
- [Using the Entry's Absolute Name](#)

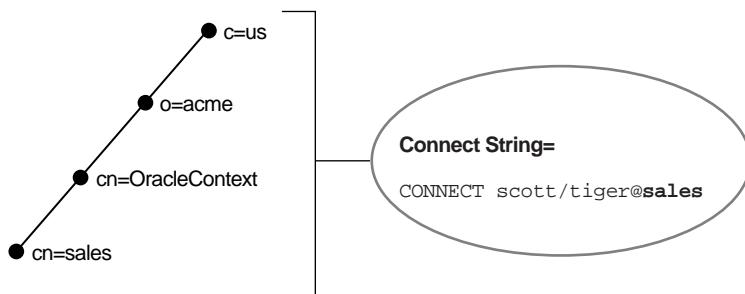
Note: JDBC OCI drivers support both relative and absolute naming. JDBC Thin drivers support absolute naming only when the complete DN is used. See the *Oracle9i JDBC Developer's Guide and Reference* for further information.

Using the Entry's Relative Name

In the following example, an entry is identified by its relative name, and the service can be referred to by its common name. A relative name can be used if the client is configured with a default directory entry that matches the directory location for the entry.

Consider a directory server that contains an entry for a database called `sales` with a DN of (`dn:cn=sales,cn=OracleContext,o=acme,c=us`), as shown in [Figure 3–6](#). If the client is configured with a default directory entry of `o=acme,c=us`, a connect string that uses `sales` can be used.

Figure 3–6 Relative Naming



Using the Entry's Absolute Name

Consider the same directory structure as shown [Figure 3–6](#) on page 3-11, but with a client configured with a default directory entry of `o=acme, c=uk`.

Because the client is configured with a directory entry that does not match the location of `sales` in the directory server, a connect string that uses `sales` does not work. Instead, the client must specifically identify the location of `sales`, which can be done in one of two ways:

- The entry's complete DN can be used in the connect string, for example:

```
CONNECT scott/tiger@cn="sales,cn=OracleContext,o=acme,c=us"
```

- The entry can be referred to by its absolute name, a name that includes the name of the object and its location in the directory server, much the way an absolute path is specified, for example:

```
CONNECT scott/tiger@sales.acme.us
```

Note: JDBC Thin drivers support absolute naming only when the complete DN is used.

See Also: ["Absolute Name Specification for Directory Naming"](#) on page 16-9 for further information about absolute names

Oracle Net Configuration and Directory Server Design

If you are responsible for designing directory servers for directory naming, consider the following issues:

- [Performance](#)
- [Security](#)
- [Schema](#)

Performance

Connect identifiers are stored in a directory server for all clients to access. Oracle Net uses a directory lookup for the requested name, resulting in heavy use of the directory server.

During a lookup, a name is searched under a specific Oracle Context. Because of the scope of the lookup, you probably want users to experience relatively quick performance so that the database connect time is not affected. Users may begin to notice slow connect times if lookups takes more than one second.

You can resolve performance problems changing the network topology or implementing replication.

See Also: Directory server vendor documentation for details on resolving performance issues

Security

Because administrative clients can create and modify entries in the directory server, security is essential. This section covers the following security-related topics:

- [Authentication Methods](#)
- [Access Control Lists](#)

Authentication Methods Clients that perform lookups for information in the directory server typically use anonymous authentication.

Clients that add or modify entries in a directory must authenticate with the directory server. Oracle Database Configuration Assistant or Oracle Net Manager may be used to add or modify the entries. Only authenticated users with proper privileges can modify entries. Use one of the authentication methods listed in [Table 3–4](#).

Table 3–4 LDAP Directory Authentication Methods

Authentication Method	Description
Native Authentication	The directory server uses operating system user credentials.
Simple Authentication	The client identifies itself to the directory server by means of a DN and a password, which are sent in the clear over the network. The server verifies that the DN and password sent by the client match the DN and password stored in the directory server.
Strong Authentication	Directories provide strong authentication by using public-key encryption available with Secure Sockets Layer (SSL). In public-key encryption, the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the recipient decrypts the message using the recipient's private key.

Access Control Lists Authentication is used with **Access Control Lists (ACLs)** to make decisions about whether clients can modify or add information in the directory server. ACLs are created at the same time as the Oracle Context with Oracle Net Configuration Assistant during directory server access configuration.

ACLs specify the following:

- The entries that the user can access
- The authentication method used to access the entry
- The access rights, or what the user can do with the object (read/write)

ACLs are established for a group of users. During Oracle Context creation, the OracleDBCreators and OracleNetAdmins groups are created under the Oracle Context entry.

Note: Additional groups are created during Oracle Context creation, as described in the *Oracle Directory Service Integration and Deployment Guide*.

The user who creates the Oracle Context with Oracle Net Configuration Assistant is automatically added as the first member of these groups.

Table 3–5 describes ACL requirements for these groups and anonymous users and their relation to Oracle Net entries in the directory server.

Table 3–5 LDAP Directory User Groups

Group	ACL Requirements
OracleNetAdmins group users	<p>Members of OracleNetAdmins (<code>cn=OracleNetAdmins,cn=OracleContext</code>) have create, modify, and read access to Oracle Net objects and attributes. Oracle Net Configuration Assistant establishes these access rights for this group during Oracle Context creation.</p> <p>In addition to the Oracle Context creator, other users can be added to this group by the directory administrator.</p> <p>See Also: "Adding Users to and Removing Users from the OracleNetAdmins Group" on page 8-9 for information on adding users to the OracleNetAdmins group</p>
Anonymous users	<p>All Oracle Net attributes and objects in the directory server have read access for the anonymous user. Read access of these objects for anonymous is also applied to the Oracle Context. This enables anonymous users to browse directory naming entries contained within the <code>cn=OracleContext</code> RDN. This does not include objects used for enterprise user security.</p> <p>Oracle Net Configuration Assistant sets up this access right during client installation.</p>

Group	ACL Requirements
OracleDBCreators group users	<p>Members of OracleDBCreators (<code>cn=OracleDBCreators,cn=OracleContext</code>) have create and read access to database service objects and attributes. Oracle Net Configuration Assistant establishes these access rights for this group during Oracle Context creation.</p> <p>In addition to the Oracle Context creator, other users can be added to this group by the directory administrator with Oracle Enterprise Security Manager.</p> <p>See Also: <i>Oracle Advanced Security Administrator's Guide</i> for further information about the OracleDBCreators group</p>

Schema

Directories must be populated with the correct version of the Oracle schema before Oracle Contexts or a database service or net service name entry can be created. The Oracle schema defines the type of objects, called **object classes**, that can be stored in the directory server and their attributes. [Table 3–6](#) for database service and net service name entries.

Table 3–6 Oracle Net LDAP Main Object Classes

Object Class	Description
<code>orclDbServer</code>	Defines the attributes for database service entries
<code>orclNetService</code>	Defines the attributes for net service name entries

Table 3–7 lists the object classes used by `orclDbServer` and `orclNetService`.

Table 3–7 Oracle Net LDAP Derived Object Classes

Object Class	Description
<code>orclNetAddress</code>	Defines a listener protocol address
<code>orclNetAddressList</code>	Defines a list of addresses
<code>orclNetDescription</code>	Specifies a connect descriptor containing the protocol address of the database and the connect information to the service
<code>orclNetDescriptionList</code>	Defines a list of connect descriptors

These object classes use attributes that specify the contents of connect descriptors.

See Also: *Oracle Net Services Reference Guide* for further information about these object classes and their attributes

Oracle Names Support

Like directory naming support, **Oracle Names** provides a distributed naming service to help simplify the setup and administration of global, client/server computing networks. Directory naming uses an LDAP-compliant directory server, which can have other functions in addition to supporting Oracle services. Oracle Names, however, is Oracle proprietary, providing support only for Oracle services.

Note: In future releases, Oracle Names will not be supported as a centralized naming method. Because no new enhancements are being added to Oracle Names, consider using directory naming or migrating an existing Oracle Names configuration to directory naming, as described in [Chapter 10, "Exporting Naming Data to a Directory Server"](#). The material presented here is primarily for reference to enable you to maintain your current Oracle Names environment.

This section contains these topics:

- [Oracle Names Overview](#)
- [Administrative Regions](#)
- [Domains](#)
- [Oracle Names As Data Repository](#)
- [Data Stored in an Oracle Names Server](#)
- [Organization of Network Components](#)
- [Multiple Regions to Decentralize Administrative Responsibilities](#)
- [Differences Between Versions of Oracle Names](#)
- [About Discovery](#)

Oracle Names Overview

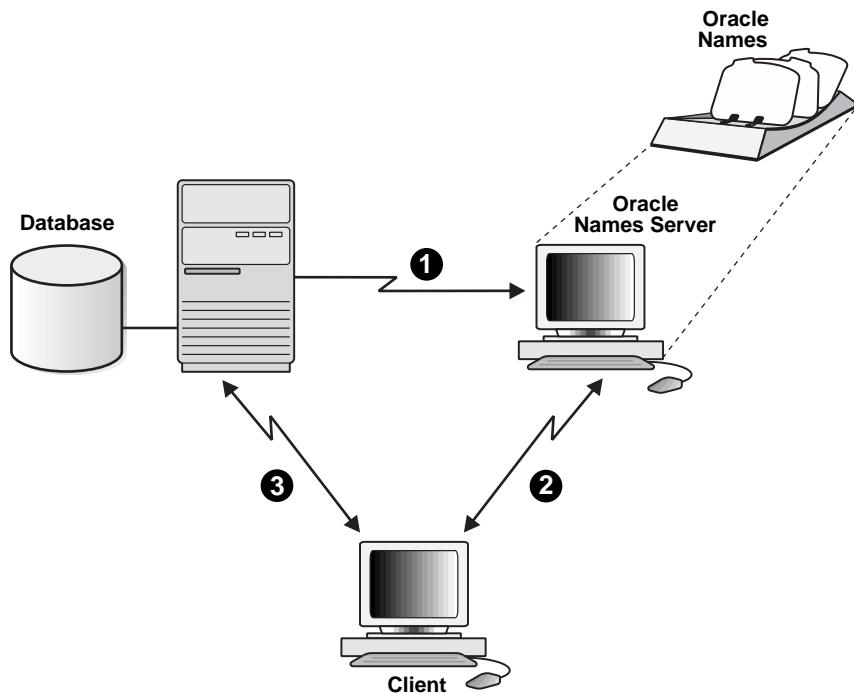
Oracle Names establishes and maintains an integrated system of **Oracle Names servers** which work together like a directory service. The system stores addresses for all the services on a network and makes them available to clients wishing to make a connection.

Much like a caller who uses directory assistance to locate a telephone number, clients configured to use Oracle Names refer their connection requests to an Oracle Names server. The Oracle Names server attempts to resolve the service name provided by the client to a network address. If the Oracle Names server finds the network address, it then returns that information to the client. The client can then use that address to connect to the service.

[Figure 3–7](#) shows how Oracle Names works to help establish a connection between a client and server:

1. A listener registers database service with an Oracle Names server.
2. A client seeks to locate a service on the network by contacting an Oracle Names server to retrieve the protocol address.
3. The client transparently connects to the service.

Figure 3–7 Oracle Names



See Also: ["Configuring the Oracle Names Method"](#) on page 9-19
for configuration information

Administrative Regions

Many networks have one central point of administration, that is, one **administrative region**. With Oracle Names, an administrative region refers to a collection of Oracle Names servers that administer services in a network. All connect information is stored in a single data repository that has the authority to interpret a service name. All Oracle Names servers within an administrative region query information from this data repository. If the administrative region uses a database for storage, there is one database for each administrative region. There can be any number of Oracle Names servers.

Oracle Names provides support for one or more administrative regions. This enables each data center to independently define and manage the services in its own environment. At the same time, all service addresses remain available to all clients across the enterprise. Oracle Names servers transparently forward client name resolution requests in one administrative region to the region which can service the request.

Domains

A **domain** is a logical group of computers and network services. Within each domain, all names must be unique: Across domains, simple service names can be repeated.

Any administrative region contains one or more domains. Network domains are similar to file directories used by many operating systems in that they are hierarchical. Unlike file system, however, network domains may or may not correspond to any physical arrangement of databases or other objects in a network. They are simply name spaces developed to prevent name space conflicts.

Note: Although they appear similar, the domains of an Oracle network are completely independent of DNS name spaces. For convenience, you may choose to mirror the DNS directory structure in the Oracle Names domain structure.

Oracle Names As Data Repository

Data in Oracle Names servers is updated through continuous replication between all the Oracle Names servers in the region or by writing to and reading from a common Oracle database. For smaller workgroup environments, administrators can configure Oracle Names servers to replicate data continuously among themselves.

Administrators can also choose to store the administrative region's data in an Oracle database, called the **region database**. A region database consists of tables that store Oracle Names information. Each Oracle Names server in a given administrative region periodically polls the region database for updated registrations. In this way, new registrations are communicated in a timely manner to all of the Oracle Names servers in a given administrative region. Using a region database also relieves Oracle Names servers of the necessity to communicate directly with each other and provides better reliability.

Data Stored in an Oracle Names Server

Table 3–8 describes the types of data stored in an Oracle Names server.

Table 3–8 Data Stored in an Oracle Names Server

Data	Description
Net service names	If you register net service names with the Oracle Names Control utility or Oracle Net Manager, an Oracle Names server stores them. An Oracle Names server also stores gateways to non-Oracle databases and Oracle RDB databases.
Global database names and addresses	The Oracle Names server retrieves information about the database, including the global database name (database name and domain) and address, from the listener. Global database names and addresses can be registered with the Oracle Names server, using either the Oracle Names Control utility or Oracle Net Manager.
Other Oracle Names server names and addresses	An Oracle Names server stores the names and addresses of all other Oracle Names servers in the same administrative region. If a network has more than one administrative region, the Oracle Names server stores the name and address of at least one Oracle Names server in the root administrative region and each of the immediate subregions. You do not need to register this information.

Data	Description
Global database links	<p>Database links enable a database to communicate with another database. The name of a database link is the same as the global database name of the database to which the link points. Typically, only one database link should exist for the database.</p>
	<p>The following types of database links can be created:</p> <ul style="list-style-type: none"><li data-bbox="577 445 1239 532">■ A private database link in a specific schema of a database. Only the owner of a private database link can use it.<li data-bbox="577 540 1239 592">■ A public database link for a database. All users in the database can use it.<li data-bbox="577 601 1239 670">■ A global database link in an Oracle Names server. Anyone in the network can use it.
	<p>Global database links, as well as user name and password credentials for them, can be registered with the Oracle Names server using Oracle Net Manager. These global database links may be supplemented with link qualifiers defined through the Oracle Net Manager.</p>
	<p>Global database links can be superseded with private and public database links created by individual users.</p>
	<p>See Also: <i>Oracle9i Database Administrator's Guide</i></p>
Aliases	<p>An Oracle Names server stores aliases or alternative service names for any defined net service name, database service, or global database link. Aliases can be registered with the Oracle Names server, using either the Oracle Names Control utility or Oracle Net Manager.</p>

Organization of Network Components

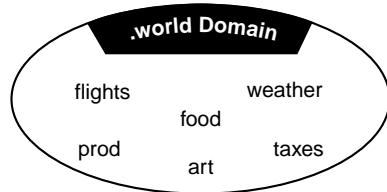
When you use Oracle Names, objects such as databases in a networked environment need to be named in a way that ensures they are unique within the network. The two basic models for naming objects in a network include:

- [Single Domain Model](#)
- [Hierarchical Naming Model](#)

Single Domain Model

The single domain naming model is useful if your network is small, and names are not duplicated. [Figure 3–8](#) shows a typical flat naming structure using a single domain name, `world`.

Figure 3–8 Single Domain Naming Model



In this environment, database service names are automatically appended with a `.world` extension (for example, `prod.world`, `flights.world`, and so forth).

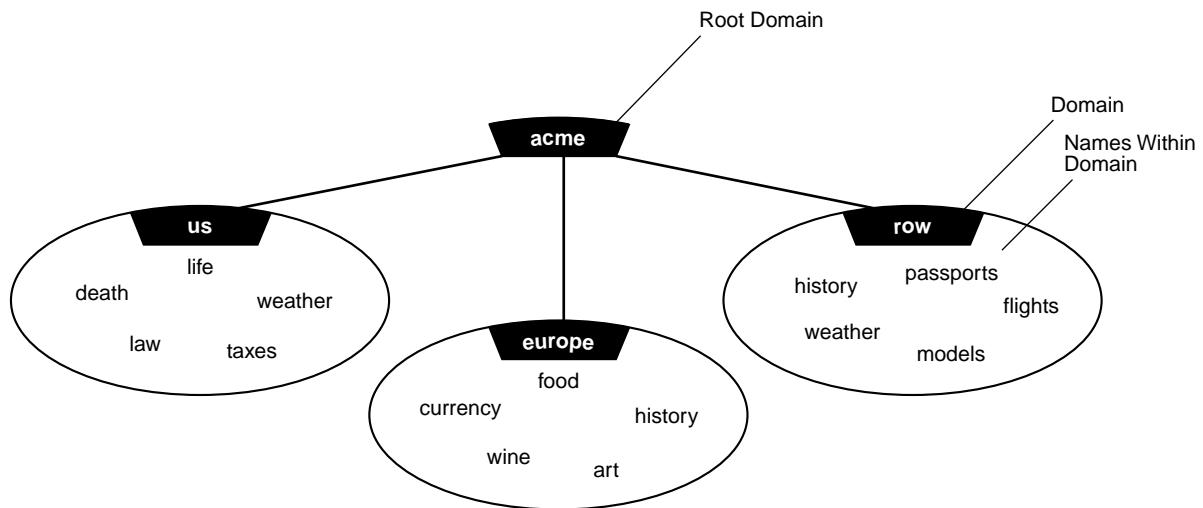
Note: In previous releases of SQL*Net and Oracle Names, a network with only one domain would default to `.world`. This convention is no required with Oracle Net Services and Oracle Names version 8 and 9. You may, however, want to keep this convention to be backward compatible, as well as to avoid having to rename all your databases.

Hierarchical Naming Model

Hierarchical naming models divide names into a hierarchical structure to allow for future growth or greater naming autonomy. This type of naming model enables more than one database with the same simple name in different domains.

[Figure 3–9](#) shows a hierarchical structure of domains including root domain acme and subdomains us.acme, europe.acme, and row.acme.

Figure 3–9 Hierarchical Naming Model



Notice in [Figure 3–9](#) both **weather** and **history** are repeated, but the names remain unique, that is, **history.row.acme** and **history.europe.acme**.

Default Domains The default domain is the domain within which most of the client's name requests are conducted. This is usually the domain in which the client resides, though it could also be another domain from which the client most often requests services. A client can request a network service within its default domain by using the service's simple, unqualified name, that is, without specifying a domain name. If a user requests a name without a **"."** character in it, the default domain name is automatically appended to the database service or database link name requested.

Consider for example, a client that is configured with a default domain of **europe.acme.com**. When this client makes a request for the service name **wine** in [Figure 3–9](#), the default domain name **europe.acme.com** is appended to the requested name so that the name becomes **wine.europe.acme.com**.

Multiple Domains Multiple domains are related hierarchically to a root domain (the highest-level domain in the hierarchy) in a series of parent-child relationships. For example, under the root might be several domains, one of which is called `com`. Under the `com` domain might be several more domains, one of which is `acme`. Under the `acme` domain might be several domains, such as `us`, `europe`, and so forth.

Multiple Regions to Decentralize Administrative Responsibilities

If you are using Oracle Names and your network is large or widely distributed geographically, you may choose to have multiple administrative regions. For example, if your network includes both the United States and Europe, you might want to have administrative decisions about the network made locally. To subdivide, you must delegate regions and domains from a parent to a child or subregion.

To delegate administrative regions, you must use a hierarchical naming model in which each administrative region controls one or more different domains.

Networks with multiple administrative regions are composed of one **root administrative region** and one or more **delegated administrative regions**.

Root Administrative Regions

The root administrative region contains the **root domain**. There can only be one root administrative region. The root administrative region contains the following information:

- Oracle Names servers in the root region
- Domains that are administered in this region: the root domain and other domains.
- Delegated administrative region Oracle Names servers: the domains and Oracle Names server addresses in any alternate regions that act as direct child regions of the root.
- Data definitions for the root region: All of the net service names, database service names, database links, and aliases associated with the root administrative region.

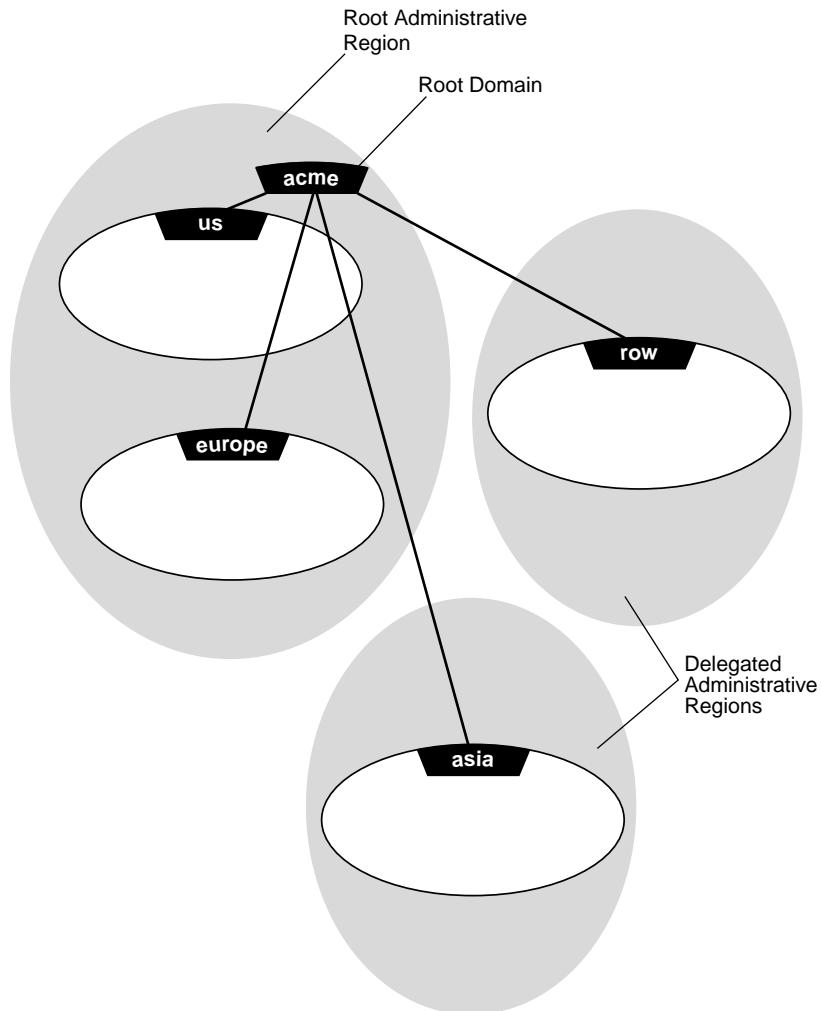
Delegated Administrative Regions

Administrative regions can be delegated from the top of the hierarchy down to other domains in the naming model. For example, a network with 10 domains can have between one and 10 administrative regions.

All administrative regions other than the root are hierarchically delegated directly or indirectly from it.

Figure 3–10 shows a network with five domains and three administrative regions: the root, acme, and two delegated regions, row and asia.

Figure 3–10 Delegated Administrative Regions



Delegated Administrative Regions Below Root

All administrative regions below the root are considered delegated administrative regions. Delegated administrative regions receive administrative responsibilities for a domain from other regions, such as the root administrative region. A delegated administrative region contains the following information:

- All Oracle Names servers and domains in the region.
- Domains and Oracle Names server addresses in any of this administrative region's child regions.
- Addresses of the Oracle Names servers in the root region. Having this data enables Oracle Names servers in delegated regions to contact any other region (through the root region).
- Data definitions, including all of the net service names, database service names, database links, and aliases for all of the domains in this local (delegated) administrative region.

Differences Between Versions of Oracle Names

Oracle Names version 8 and 9 differs significantly from earlier versions. The differences between the versions are described in the following topics:

- [Oracle Names Version 1](#)
- [Oracle Names Version 2](#)
- [Oracle Names Version 8 and 9 \(this release\)](#)

Oracle Names Version 1

In Oracle Names version 1, administrators configured Oracle Names servers using Oracle Network Manager and stored all topology data in a database. All the Oracle Names servers in a region shared the same information because they accessed the same database.

The clients had a list of **preferred Oracle Names servers** specified in the `sqlnet.ora` file. This list was created by the user, listing the order of preferred Oracle Names to contact. The first Oracle Names server in the list would be contacted first by a client.

Preferred Oracle Names servers may still be configured.

See Also: ["Configuring Connections to Non-Oracle Database Services"](#) on page 15-36 for configuration information

Oracle Names Version 2

In Oracle Names version 2, the administrator could choose between continuing Oracle Names server configuration, as in version 1 or using the Dynamic Discovery Option. The Dynamic Discovery Option was recommended only for a network with a single region and single DNS domain. The Dynamic Discovery Option uses **well-known Oracle Names servers**, which are precise names hard-coded into DNS or the hosts file on both the Oracle Names Server and its clients:

Table 3–9 Well-Known Oracle Names Servers

The well-known host names for TCP connections	The well-known computer names for Named Pipes connections ¹
oranamesrvr0	ORANAMESRV0
oranamesrvr1	ORANAMESRV1
oranamesrvr2	ORANAMESRV2
oranamesrvr3	ORANAMESRV3
oranamesrvr4	ORANAMESRV4

¹ Well-known Oracle Names server names for Named Pipes must be in all uppercase.

Oracle Names servers then become available at these well-known addresses, so clients do not need to be told, by way of a preferred Oracle Names server list, where to find an Oracle Names server.

If the Dynamic Discovery Option was chosen, each Oracle Names server automatically replicated its data to all other well-known Oracle Names servers in the administrative region. Listeners were configured to register themselves with well-known Oracle Names servers.

See Also: *Oracle Names Administrator's Guide*, Release 2.0, for configuration information

Oracle Names Version 8 and 9 (this release)

In Oracle Names version 8 and 9, the administrator may choose between continuing Oracle Names Server configuration as in version 1 or version 2, or using the new functionality. Oracle Names version 8 and 9 incorporates version 2 Dynamic Discovery Option features without the constraints of a single region and single domain. The main features of Oracle Names version 8 and 9 include:

- A service can register itself with any Oracle Names server it can find, and its name and address are made available to all Oracle Names servers in the region. Similarly, if an administrator manually registers a service to any Oracle Names server, that service information is available to all other Oracle Names servers. The address information is shared in one of two ways:
 - Region Database: If an Oracle database is used as the registration repository, the registration information is stored in the database and from there is accessible to all the Oracle Names servers.
 - Service Replication: In service replication, service information is stored in the cache of an Oracle Names server and is instantly replicated to the caches of all other Oracle Names servers.
- A list of Oracle Names servers to contact is created on a client through a discovery process rather than manual configuration, as described in "[About Discovery](#)" on page 3-33.
- A clientside process may be run to create a client cache version of the data stored in Oracle Names servers, including database addresses and Oracle Names server protocol addresses, as well as other service information. As a result, the client does not have to contact an Oracle Names server for address information, which reduces address lookup time.
- Oracle Names version 8 and 9 does not require a database to hold topology information. However, an administrator may choose to use one, and it is recommended.

See Also: ["Configuring the Oracle Names Method"](#) on page 9-19 for configuration information

About Discovery

A list of Oracle Names servers can be created that enables a client or another Oracle Names server to contact an Oracle Names server. The process of creating the list is called discovery.

When a client tries to discover an Oracle Names server with the Oracle Names Control utility or Oracle Net Manager, one Oracle Names server is found first. Once the client finds an Oracle Names server, it pings all other Oracle Names servers in the region. A list of Oracle Names servers is then created on the client and saved to `.sdns.ora` on UNIX and `sdns.ora` on Windows operating systems. This list is sorted in order of response time.

To find the first Oracle Names server, discovery searches in the following order:

1. A [preferred Oracle Names server](#) configured in the `sqlnet.ora` file
2. A [well-known Oracle Names server](#)
3. A local Oracle Names server configured with TCP/IP on port 1575

If the client is unable to find the first Oracle Names server using the above methods, one of the following alternate methods can be used:

- If you are using Oracle Net Manager, you will be prompted to specify the protocol address of an Oracle Names server
- If you are using the Oracle Names Control utility, the utility will fail. Rerun the command with the protocol address of a specified Oracle Names server.

Architecture of Oracle Net Services

This chapter describes the architecture of Oracle Net, the **listener**, **shared server**, **dedicated server**, and **Oracle Connection Manager**.

This chapter contains these topics:

- [Oracle Net Stack Communication Architecture](#)
- [Listener Architecture](#)
- [Database Server Process Architecture](#)
- [Oracle Connection Manager Architecture](#)
- [A Complete Architecture](#)

See Also: [Chapter 1, "Networking Challenges in the Internet Age"](#)
for an introductory level overview of Oracle Net architecture

Oracle Net Stack Communication Architecture

The primary function of Oracle Net is to establish and maintain connections between a client application and an Oracle database server. Oracle Net is comprised of several communication layers that enable clients and database servers to share, modify, and manipulate data.

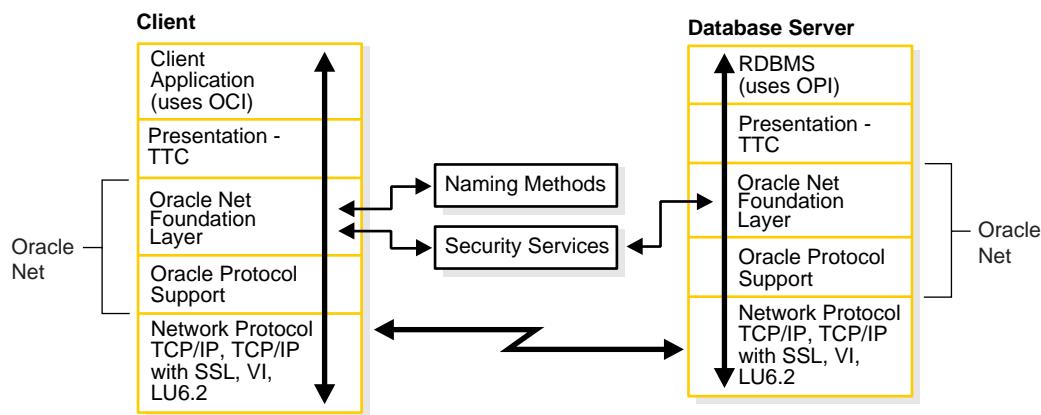
This section contains these topics:

- [Stack Communication for Client/Server Application Connections](#)
- [Stack Communication for Java Application Connections](#)
- [Stack Communication for Web Client Connections](#)

Stack Communication for Client/Server Application Connections

[Figure 4-1](#) illustrates the various layers on the client and on the database server after a connection has been established.

Figure 4-1 Layers Used in a Client/Server Application Connection



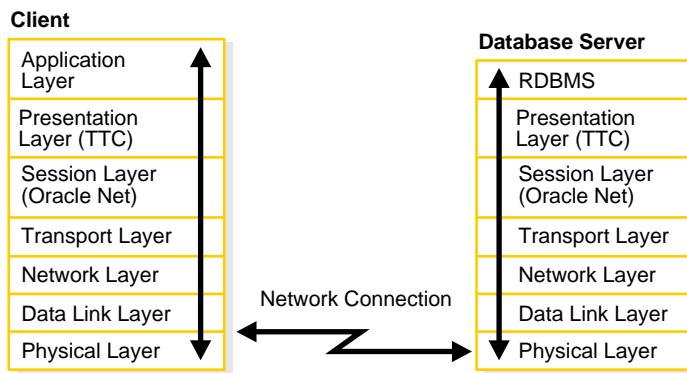
This communication architecture is based on the Open Systems Interconnection (OSI) model. In the OSI model, communication between separate computers occurs in a stack-like fashion with information passing from one node to the other through several layers of code, including:

1. Physical layer
2. Data link layer
3. Network layer

4. Transport layer
5. Session layer
6. Presentation layer
7. Application layer

Figure 4–2 shows how Oracle Net software—Oracle Net Foundation layer and Oracle Protocol Support—fits into the Session layer of the OSI model.

Figure 4–2 OSI Communication Layers



See Also: <http://www.ietf.org/> for information about the OSI stack

As shown in [Figure 4-1](#) on page 4-2, the client/server stack is comprised of the following:

- [Client Application](#)
- [Presentation](#)
- [Oracle Net Foundation Layer](#)
- [Oracle Protocol Support](#)
- [Network Protocol](#)
- [RDBMS](#)

Client Application

During a session with the database, the client uses [Oracle Call Interface \(OCI\)](#) to interact with the database server. OCI is a software component that provides an interface between the client application and the SQL language the database server understands.

See Also: [Oracle Call Interface Programmer's Guide](#)

Presentation

Character set differences can occur if the client and database server are running on different operating systems. The presentation layer resolves any differences. It is optimized for each connection to perform conversion only when required.

The presentation layer used by client/server applications is [Two-Task Common \(TTC\)](#). TTC provides character set and data type conversion between different character sets or formats on the client and database server.

At the time of initial connection, TTC is responsible for evaluating differences in internal data and character set representations and determining whether conversions are required for the two computers to communicate.

Oracle Net Foundation Layer

The Oracle Net foundation layer is responsible for establishing and maintaining the connection between the client application and database server, as well as exchanging messages between them. The Oracle Net foundation layer is able to perform these tasks because of a technology called [Transparent Network Substrate \(TNS\)](#). TNS provides a single, common interface functioning over all industry-standard protocols. In other words, TNS enables peer-to-peer application connectivity. In a peer-to-peer architecture, two or more computers (called **nodes**)

when they are employed in a networking environment) can communicate with each other directly, without the need for any intermediary devices.

On the client side, the Oracle Net foundation layer receives client application requests and resolves all generic computer-level connectivity issues, such as:

- The location of the database server or destination
- Whether one or more protocols are involved in the connection
- How to handle interrupts between client and database server based on the capabilities of each

On the server side, the Oracle Net foundation layer performs the same tasks as it does on the client side and also works with the listener to receive incoming connection requests.

In addition to establishing and maintaining connections, the Oracle Net foundation layer communicates with naming methods to resolve names and uses security services to ensure secure connections.

Oracle Protocol Support

Positioned between the Oracle Net foundation layer and the network protocol layer, the Oracle protocol support layer is responsible for mapping TNS functionality to industry-standard protocols used in the client/server connection. This layer supports the following network protocols:

- TCP/IP
- TCP/IP with SSL
- Named Pipes
- LU6.2
- VI

Network Protocol

All Oracle software in the client/server connection process requires an existing network protocol stack to establish the computer-level connection between the two computers for the transport layer. The network protocol is responsible for transporting data from the client computer to the database server computer, at which point the data is passed to the serverside Oracle protocol support layer.

The protocols have the following characteristics:

TCP/IP Protocol The Transmission Control Protocol/Internet Protocol (TCP/IP) is the de facto standard communication protocol used for client/server conversation over a network.

TCP/IP with SSL Protocol The TCP/IP with **Secure Sockets Layer (SSL)** protocol enables an Oracle application on a client to communicate with remote Oracle databases through TCP/IP and SSL. **Oracle Advanced Security** is required in order to use TCP/IP with SSL.

SSL stores authentication data, such as certificates and private keys, in an Oracle Wallet. When the client initiates a connection to the database server, SSL performs a handshake between the two using the certificate. During the handshake, the following processes occur:

- The client and database server negotiate a cipher suite—a set of authentication, encryption, and data integrity types—to apply to the messages they exchange.
- Depending on its configuration, the database server sends its certificate to the client in a message encrypted with the client's public key. The database server may also send a request for the client's certificate in the same message. The client decrypts this message by using its own private key, then verifies that the database server's certificate bears the certificate authority's signature.
- If required, the client may send the user's certificate to the database server. The certificate ensures that the user's information is correct and that the public key actually belongs to that user.

The database server checks the user's certificate to verify that it bears the certificate authority's signature.

See Also: *Oracle Advanced Security Administrator's Guide*

Named Pipes Protocol The **Named Pipes protocol** is a high-level interface providing interprocess communications between clients and database servers using distributed applications. One severside process creates the pipe, and the other clientside process opens it by name. What one side writes, the other can read, and vice versa. Named Pipes is specifically designed for PC LAN environments.

Named Pipes enables client/server conversation over a network using Named Pipes. This combination of Oracle products enables an Oracle application on a client to communicate with remote Oracle databases through Named Pipes (if the Oracle database is running on a host system that supports network communication using Named Pipes).

LU6.2 Protocol The Logical Unit Type 6.2 (LU6.2) protocol is part of the IBM Advanced Program-to-Program Communication (APPC) architecture.

APPN is the IBM peer-to-peer (program-to-program) protocol for the System Network Architecture (SNA) network. SNA is an IBM reference model similar to the OSI model.

APPN architecture lets the client and host communicate over an SNA network without forcing the client to emulate a terminal (as in terminal-to-host protocols). APPN architecture enables peer-to-peer communication; the client can initiate communication with the database server.

An SNA network with the LU6.2 and Physical Unit Type 2.1 (PU2.1) protocols provides APPN. The LU6.2 protocol defines a session between two application programs; LU6.2 is a product-independent LU type.

LU6.2 enables an Oracle application on a PC to communicate with the database server. This communication occurs over an SNA network with the Oracle database on a host system that supports APPN.

VI Protocol The Virtual Interface (VI) protocol can be used for application Web server and database server communication. VI is the de facto standard communication protocol for clustered server environments. Specifically designed for clusters, VI is a thin protocol that is more efficient than TCP/IP. Applications perform better using VI, because VI places most of the messaging burden upon high-speed network hardware, freeing the CPU for other tasks.

VI reduces the overhead of TCP/IP by eliminating intermediate copies and by transferring most of the messaging burden away from the CPU and onto the network hardware. The result is a low-latency, high-throughput interconnect protocol that reduces the amount of CPU cycles dedicated to messaging.

See Also: <http://www.viarch.org/> for the VI specification

RDBMS

Information passed from a client application across a network protocol is received by a similar communications stack on the database server side. The process flow on the database server side is the reverse of the process flow on the client side, with information ascending through the communication layers.

Instead of OCI, the database server uses **Oracle Program Interface (OPI)**. For each statement sent from OCI, OPI provides a response. For example, an OCI request to fetch 25 rows would elicit an OPI response to return the 25 rows once they have been fetched.

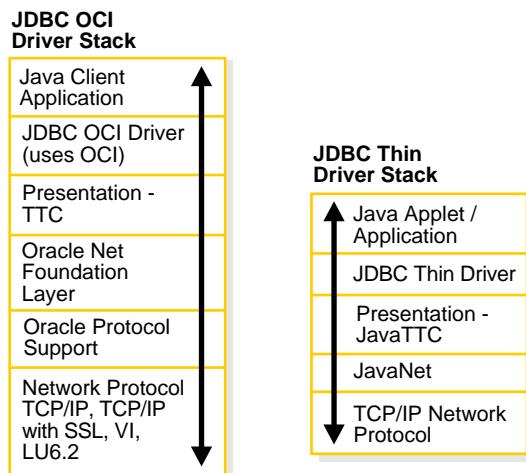
Stack Communication for Java Application Connections

Oracle's Java Database Connectivity (JDBC) drivers provide Java applications access to an Oracle database. Oracle offers two JDBC drivers.

- JDBC OCI driver is a level 2 JDBC driver which is used by client/server Java applications. The JDBC OCI driver converts JDBC invocations to calls to OCI which are then sent over Oracle Net to the Oracle database server.
- JDBC Thin driver is a level 4 driver which is used by Java applets. The JDBC Thin driver establishes a direct connection to the Oracle database server over Java sockets. Access to the database is assisted with a lightweight implementation of TTC and Oracle Net.

[Figure 4-3](#) shows the stack communication layers used by JDBC drivers.

Figure 4-3 Layers Used for Java-Client Applications



Note: The Named Pipes protocol is supported but is not represented in this figure.

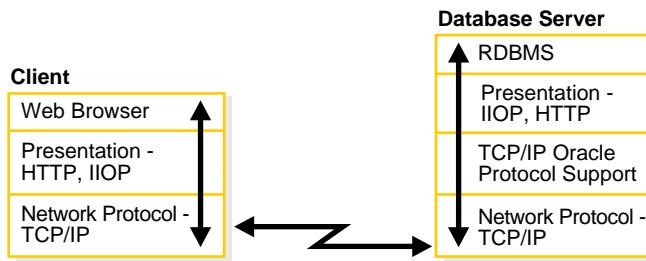
The JDBC OCI driver uses a communication stack similar to a standard client/server communication stack. The JDBC Thin driver uses a Java implementation of the Oracle Net foundation layer called JavaNet and a Java implementation of TTC called JavaTTC.

See Also: *Oracle9i JDBC Developer's Guide and Reference*

Stack Communication for Web Client Connections

In addition to TTC presentation, the Oracle database server supports many other presentations that can be used for Web clients accessing features inside the database. For example, the database can be configured to accept HTTP and **Internet Inter-ORB Protocol (IIOP)** connections. The HTTP presentation is used to access Oracle Internet File System, and the IIOP presentation is used for connections to Enterprise JavaBeans (EJBs) and Common Object Request Broker Architecture (CORBA) applications in the database.

Figure 4–4 Layers Used in Web Client Connections

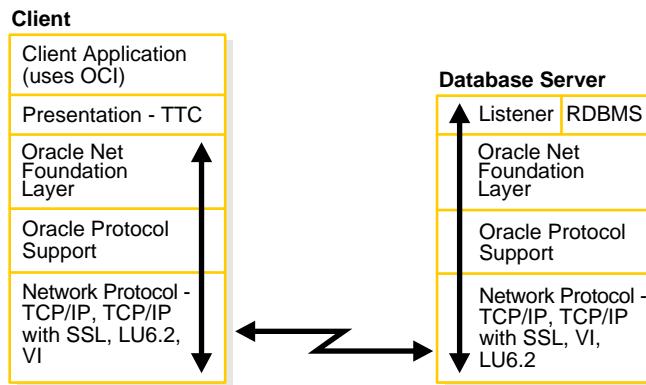


See Also: *Oracle9i Enterprise JavaBeans Developer's Guide and Reference*

Listener Architecture

The database server receives an initial connection from a client application through the **listener**. The listener is an application positioned on top of the Oracle Net foundation layer. [Figure 4–5](#) illustrates the various layers on the client and database server during an initial connection.

Figure 4–5 Layers Used in an Initial Connection



Note: The Named Pipes protocol is supported but is not represented in this figure.

The listener brokers client requests, handing off the requests to the Oracle database server. Every time a client requests a network session with a database server, a listener receives the initial request.

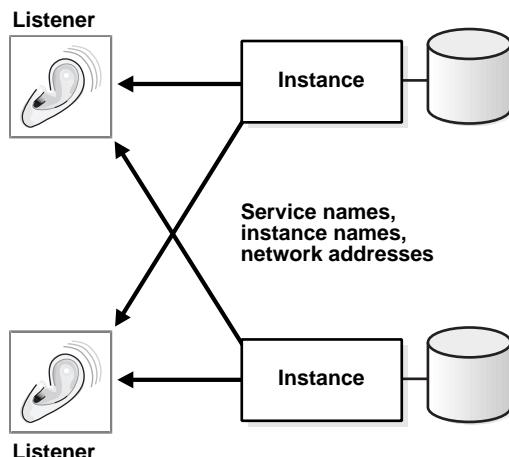
Each listener is configured with one or more **protocol addresses** that specify its listening endpoints. Clients configured with one of these protocol addresses can send connection requests to the listener.

Once a client request has reached the listener, the listener selects an appropriate **service handler** to service the client's request and forwards the client's request to it. The listener determines if a database service and its service handlers are available through **service registration**. During service registration, the **PMON process**—an instance background process—provides the listener with information about the following:

- Names of the database services provided by the database
- Name of the **instance** associated with the services and its current and maximum load
- Service handlers (**dispatchers** and **dedicated servers**) available for the instance, including their type, protocol addresses, and current and maximum load

This information enables the listener to direct a client's request appropriately. [Figure 4–6](#) shows instances registering information with listeners. Note that it does not represent all the information that can be registered.

Figure 4–6 Service Registration



Optionally, listening endpoints—port numbers—can be dynamically registered with the listener. For example, if [Oracle9i JVM](#) is installed, HTTP and IIOP listening endpoints are registered with the listener.

See Also: *Oracle9i Enterprise JavaBeans Developer's Guide and Reference* for information about dynamically registering HTTP and IIOP listening endpoints

If the listener is not running when an instance starts, PMON is not able to register the service information. PMON attempts to connect to the listener periodically, however, it may take up to 60 seconds before PMON registers with the listener after it has been started. To initiate service registration immediately after the listener is started, use the SQL statement `ALTER SYSTEM REGISTER`. This is especially useful in high-availability configurations.

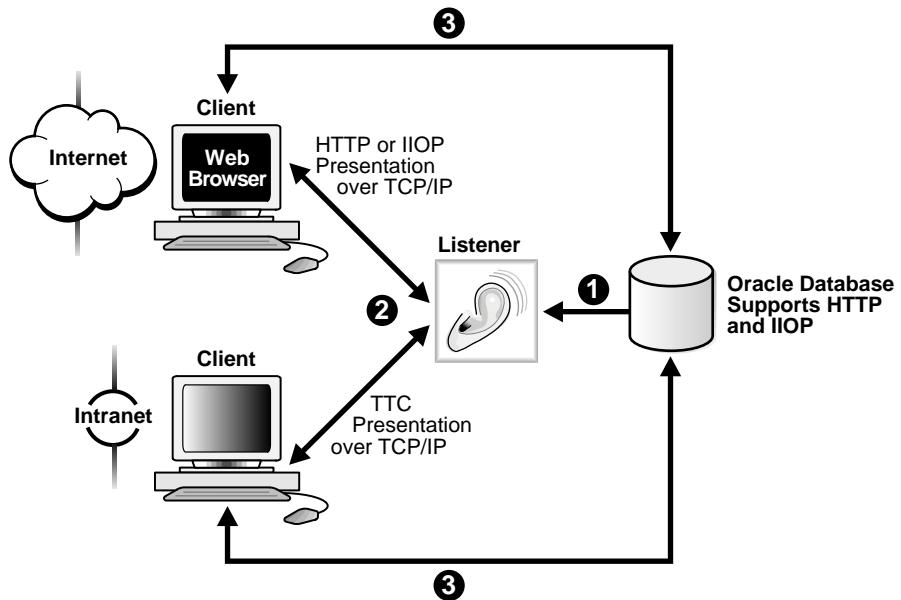
If a listener receives an incoming request before the respective instance has been registered, the listener rejects the request.

See Also: *Oracle9i SQL Reference* for further information about the `ALTER SYSTEM REGISTER` statement

[Figure 4–7](#) shows the role of a listener during connection establishment:

1. The database registers information about the services, instances, and service handlers with the listener.
2. The client makes an initial connection with the listener.
3. The listener parses the client request and forwards it to the service handler for the database service requested.

Figure 4–7 Listener Architecture



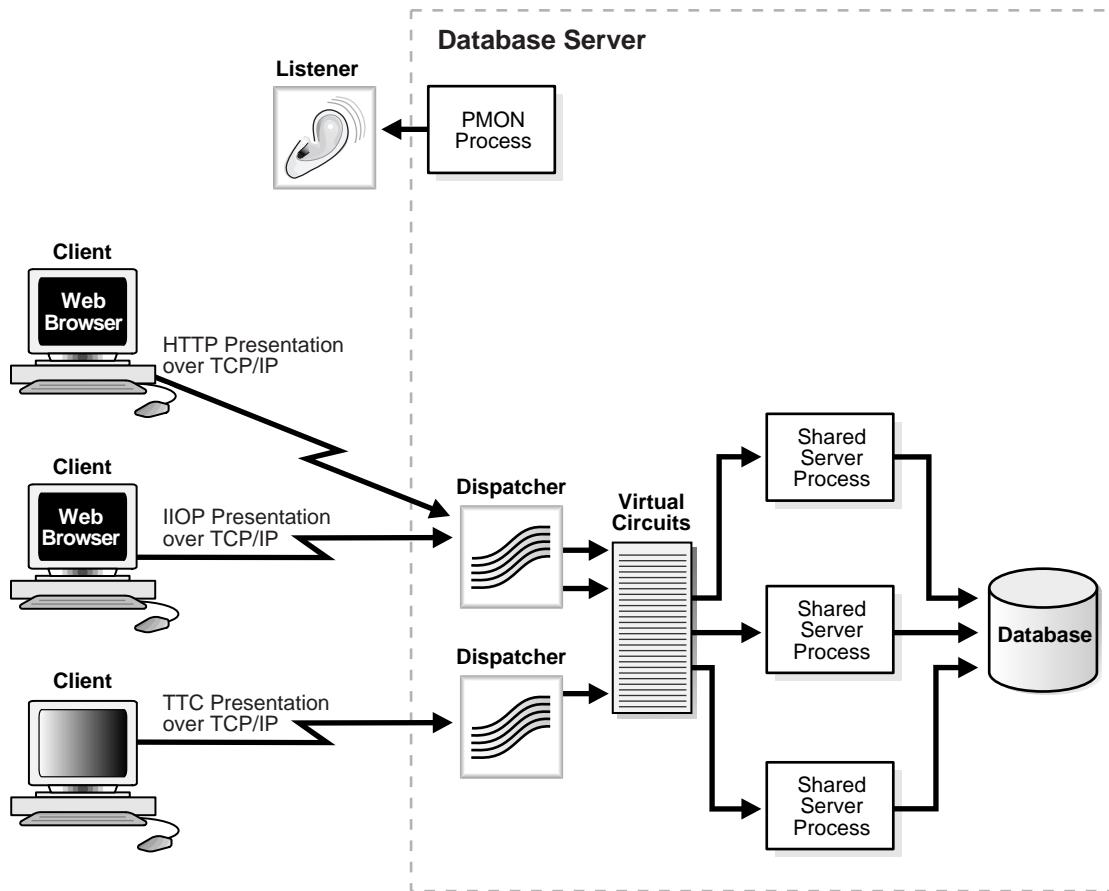
Database Server Process Architecture

Based on the service handler type registered with the listener, the listener forwards requests to either a **shared server process** or a **dedicated server** process.

Shared Server Processes

Shared server processes are utilized in the **shared server** architecture. Figure 4-8 on page 4-15 depicts a shared server architecture. With shared server architectures, client processes ultimately connect to a **dispatcher**. The PMON process registers the location and load of the dispatchers with the listener, enabling the listener to forward requests to the least loaded dispatcher.

A dispatcher can support multiple client connections concurrently. Each client connection is bound to a **virtual circuit**. A virtual circuit is a piece of shared memory used by the dispatcher for client database connection requests and replies. The dispatcher places a virtual circuit on a common queue when a request arrives. An idle shared server picks up the virtual circuit from the common queue, services the request, and relinquishes the virtual circuit before attempting to retrieve another virtual circuit from the common queue. This approach enables a small pool of server processes to serve a large number of clients.

Figure 4–8 Shared Server Architecture

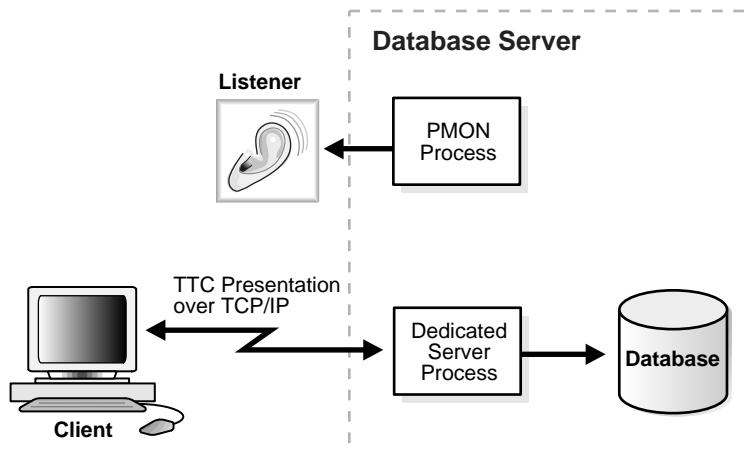
Dedicated Server Processes

Figure 4–9 depicts a dedicated server architecture. With a dedicated server architecture, each client process connects to a dedicated server process. The server process is not shared by any other client.

PMON registers information about dedicated server processes with the listener. This enables the listener to start up a dedicated server process when a client request arrives and forward the request to it.

Note: Dedicated server architectures do not support HTTP and IIOP clients. Only TTC clients are supported.

Figure 4–9 Dedicated Server Architecture



Oracle Connection Manager Architecture

Oracle Connection Manager is a router through which a client connection request may be sent either to its next hop or directly to the database server. Clients who route connection requests through an Oracle Connection Manager can take advantage of the connection multiplexing, access control, and protocol conversion features configured on that Oracle Connection Manager.

Oracle Connection Manager has three process types:

- Gateway process
- Administrative process

The gateway process, CMGW, receives client connections and evaluates against a set of rules whether to deny or allow access. If access is allowed, the gateway process forwards the requests to the next hop, typically the database server. In addition to allowing or denying access, the CMGW process can also multiplex or funnel multiple client connections through a single protocol connection.

The CMGW process registers with the administrative process, CMADMIN. CMADMIN is a multi-threaded process that is responsible for all administrative functions of Oracle Connection Manager.

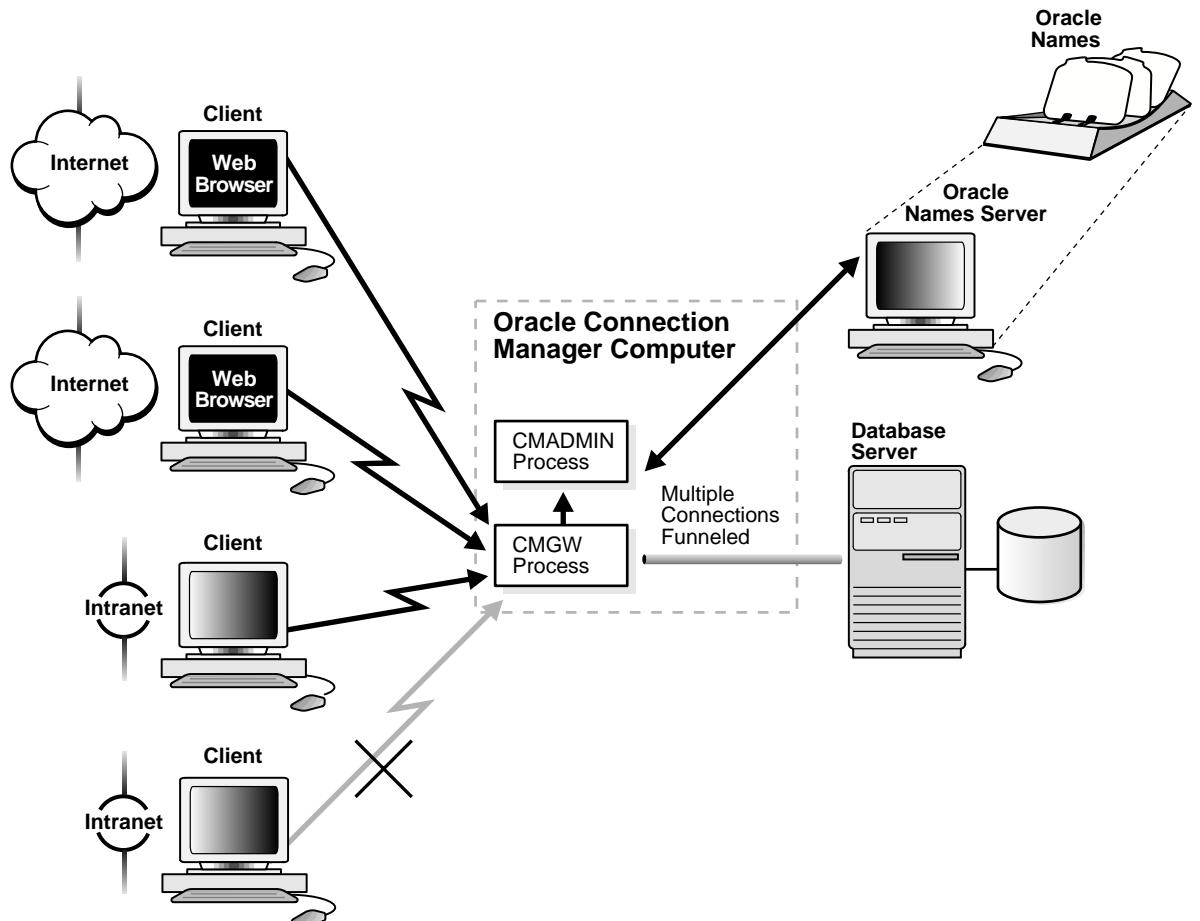
Table 4–1 describes the detailed responsibilities of the CMGW and CMADMIN processes.

Table 4–1 Oracle Connection Manager Processes

Process	Description
CMGW (Oracle Connection Manager Gateway Process)	<p>The CMGW process is responsible for the following tasks:</p> <ul style="list-style-type: none">■ Registering with the CMADMIN process■ Listening for incoming connection requests. By default it listens on TCP/IP with port 1630.■ Initiating connection requests to listeners for clients■ Relaying data between the client and database server■ Answering requests initiated by the Oracle Connection Manager Control utility
CMADMIN (Oracle Connection Manager Administrative Process)	<p>The CMADMIN process is responsible for the following tasks:</p> <ul style="list-style-type: none">■ Processing the CMGW registration■ Identifying all listeners serving at least one database instance■ Registering source route address information about the CMGW process and listeners■ Answering requests initiated by the Oracle Connection Manager Control utility■ Locating Oracle Names servers■ Monitoring registered listeners with Oracle Names servers■ Maintaining address information in Oracle Names servers for clients■ Periodically sending a request to Oracle Names servers to update their cache of available services

Figure 4-10 shows the CMGW process registering with the CMADMIN process and the CMGW process handling client requests. Notice that the CMGW process has denied access to the fourth client. The three client connections are then multiplexed through a single network protocol connection to the database.

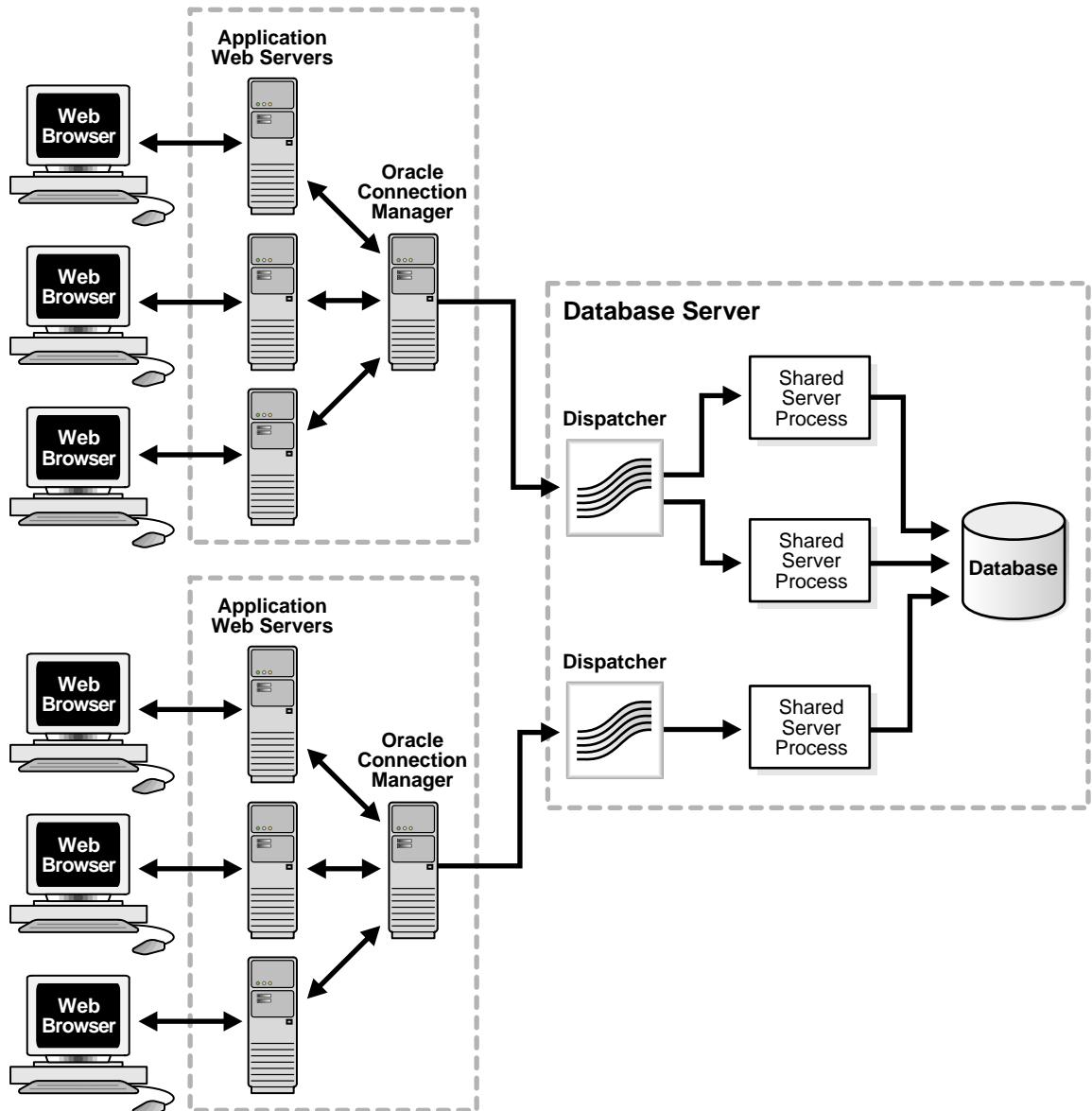
Figure 4-10 Oracle Connection Manager Architecture



A Complete Architecture

Oracle Net provides an architectural solution that allows for greater scalability in Internet and intranet environments.

[Figure 4-11](#) on page 4-21 shows how multiple connections to an Oracle database server are made more scalable with Oracle Connection Manager and a shared server architecture. Oracle Connection Manager is used to offload some of the network I/O of the application Web servers, and shared server is used to serve more concurrent users.

Figure 4–11 Scalable Architectural Solutions

5

Configuration and Administration Tools Overview

This chapter introduces the various administration tools of Oracle Net Services. It discusses the main administration application, Oracle Net Manager, and describes how to launch and navigate through it. It also introduces the command line control utilities.

This chapter contains these topics:

- [Oracle Net Manager](#)
- [Oracle Net Configuration Assistant](#)
- [Oracle Net Control Utilities](#)
- [Configuration and Administration Tasks at a Glance](#)

Oracle Net Manager

Oracle Net Manager is a graphical user interface tool that combines configuration abilities with Oracle Names component control to provide an integrated environment for configuring and managing Oracle Net. It can be used on either the client or server. Oracle Net Manager is also integrated with Oracle Enterprise Manager.

You can use Oracle Net Manager to configure the following network components:

- Naming—Define simple names, **connect identifiers**, and map them to **connect descriptors** to identify the network location and identification of a service. Oracle Net Manager supports configuration of connect descriptors in local `tnsnames.ora` files, a centralized LDAP-compliant directory service, or an Oracle Names server.
- Naming Methods—Configure the different ways in which connect identifiers are resolved into connect descriptors.
- Profiles—Configure preferences for enabling and configuring Oracle Net features on the client or server.
- Listeners—Create and configure listeners to receive client connections.

If an Oracle Names server is configured, you can start, stop, tune, or gather statistics for it with Oracle Net Manager.

This section introduces you to the features of Oracle Net Manager. However, the primary documentation for using Oracle Net Manager is the accompanying online help. This section contains these topics:

- [Starting Oracle Net Manager](#)
- [Navigating Oracle Net Manager](#)
- [Toolbar](#)
- [Menus](#)
- [Oracle Net Manager Wizards](#)

See Also: The Oracle Enterprise Manager documentation set and online help for information about using Oracle Enterprise Manager

Starting Oracle Net Manager

You can start Oracle Net Manager using the Oracle Enterprise Manager Console or as an independent application.

To start Oracle Net Manager from the Oracle Enterprise Manager console, on the Oracle Enterprise Manager Console, choose Tools > Service Management > Oracle Net Manager.

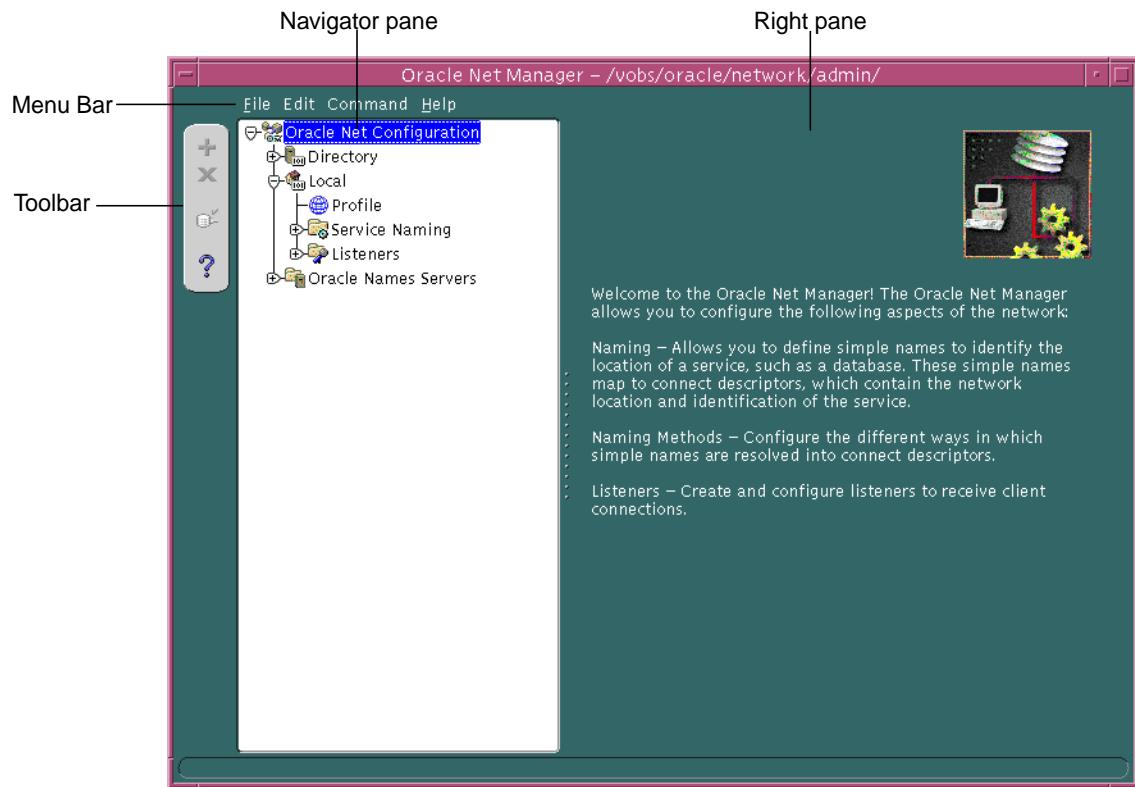
To start Oracle Net Manager as standalone application:

- On UNIX, run `netmgr` from `$ORACLE_HOME/bin`
- On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Oracle Net Manager

Navigating Oracle Net Manager

The Oracle Net Manager interface includes two panes, a toolbar, and various menu items.

Figure 5–1 Oracle Net Manager Interface



Panes

The Oracle Net Manager interface has two panes: the navigator pane and the right pane.

Navigator Pane The navigator pane provides a graphical tree view of network objects and the objects they contain, organized in hierarchies of folders. You can use the navigator pane to view, modify, add, or delete objects in each folder.

The navigator pane functions the same way as it does in other Oracle Enterprise Manager applications. That is, the navigator pane lets you:

- Expand and contract folders so that you can navigate to the network object you want to monitor or manage. Examples of objects are connect identifiers, listeners, profiles, and Oracle Names servers.
- Right-click on an object to perform operations on the object.

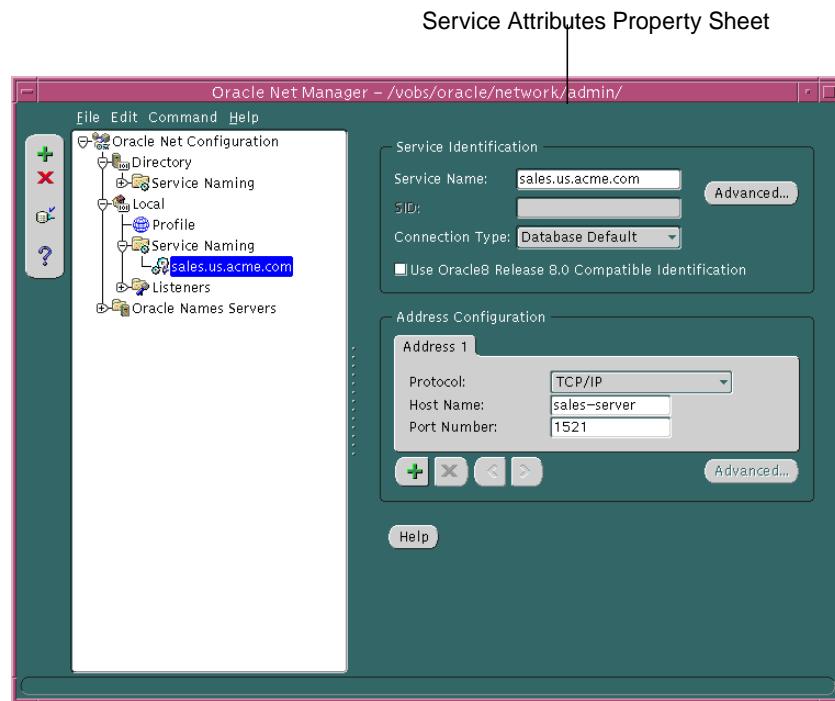
When you expand a folder, you see a nested list of objects and folders. When a object is selected, information about the object is displayed in the right pane of the Oracle Net Manager.

The navigator pane has the following main folders:

Directory	Displays connect identifiers configured in a directory server
Local	Displays networking elements configured in local configuration files: <ul style="list-style-type: none">■ Net service names in the <code>tnsnames.ora</code> file■ Listeners in the <code>listener.ora</code> file■ Profiles in the <code>sqlnet.ora</code> file
Oracle Names Servers	Displays Oracle Names servers

Right Pane The right pane contains property sheets that enable you to configure network components. [Figure 5–2](#) shows the Service Attributes property sheet used for connect identifier configuration.

Figure 5–2 Service Attributes Property Sheet



Toolbar

The toolbar contains buttons that correspond to the network objects in the folder hierarchies. The toolbar buttons are enabled depending on the objects viewed or selected in a pane. Move the mouse cursor over a toolbar button to display the description of the button's function. The toolbar buttons are:

Create

This button creates new connect identifiers, listeners, and Oracle Names servers objects under the Directory or Local > Service Naming, Listeners, and Oracle Names Servers folders.

Delete

This button deletes connect identifier and listener objects under the Directory or Local > Service Naming and Listeners folders.

Verify Connectivity

This button verifies the connectivity of a selected connect identifier to an Oracle database service. During a connectivity test, a connection to a database service is made by using connect descriptor information stored in a directory server or a `tnsnames.ora` file.

Help

The Help button opens the Oracle Net Manager online help.

Menus

You use Oracle Net Manager menus to perform actions, such as testing connectivity to an Oracle database, and to open wizards and dialog boxes. The following list describes the items available under each menu.

File Menu

The File menu has the following menu items:

- | | |
|-------------------------------|--|
| Open Network Configuration | Opens the local configuration files |
| Save Network Configuration | Saves the current configuration settings to the default location |
| Save As | Saves the current configuration settings to a nondefault location |
| Revert to Saved Configuration | Does not save any of the current configuration settings. Instead, retrieves the last saved configuration |
| Exit | Closes the Oracle Net Manager |

Edit Menu

The Edit menu has the following menu items:

- | | |
|--------|--|
| Create | Creates new connect identifiers, listeners, and Oracle Names servers objects under the Directory or Local > Service Naming, Listeners, and Oracle Names Servers folders. |
| Delete | Deletes connect identifier and listener objects under the Directory or Local > Service Naming and Listeners folders. |
| Rename | Renames a network object |

Command Menu

The Command menu has the following menu items:

- | | |
|-------------------------------|--|
| Directory | The Directory submenu contains the following menu items:

Change Current Context: Enables you to change the current Oracle Context. The Oracle Context is a subtree in the directory that stores Oracle entries, including database services and net service names for use with directory naming. When a new Oracle Context is selected, objects in the navigator's Oracle Net Configuration > Directory > Service Naming folder display for the new Oracle Context.

Set Authentication: Enables you to set authentication credentials to the directory server

Import Net Service Names: Launches the Directory Server Migration Wizard. This wizard imports net service names stored in a <code>tnsnames.ora</code> file to a directory server. |
| Test Service | Connects to a database service using connect descriptor information stored in a directory server or a <code>tnsnames.ora</code> file |
| Discover Oracle Names Servers | Finds Oracle Names servers. Once the client finds an Oracle Names server, it queries for all other Oracle Names servers in the domain. Based on the Oracle Names servers found, a list of Oracle Names servers is created on the client. The list is created in order of response time.

See Also: " About Discovery " on page 3-33 |
| Reload All Names Servers | The Oracle Names server checks immediately for data changes in its administrative region. If there are changes, the Oracle Names server reloads all stored data. |

Help Menu

The Help menu has the following menu items:

- | | |
|--------------------------|--|
| Contents | Opens Oracle Net Manager online help |
| About Oracle Net Manager | Opens a window that displays the Oracle Net Manager version number and copyright information |

Oracle Net Manager Wizards

The Oracle Net Manager wizards provide step-by-step guidance for tasks that require many steps. The wizards simplify complex tasks by guiding you through the task in manageable steps. The wizards are not intended to provide all configuration options. Once you have completed a task with a wizard, use other components of Oracle Net Manager to modify the configuration.

The following topics describe the Oracle Net Manager wizards:

- [Net Service Name Wizard](#)
- [Directory Server Migration Wizard](#)
- [Names Wizard](#)

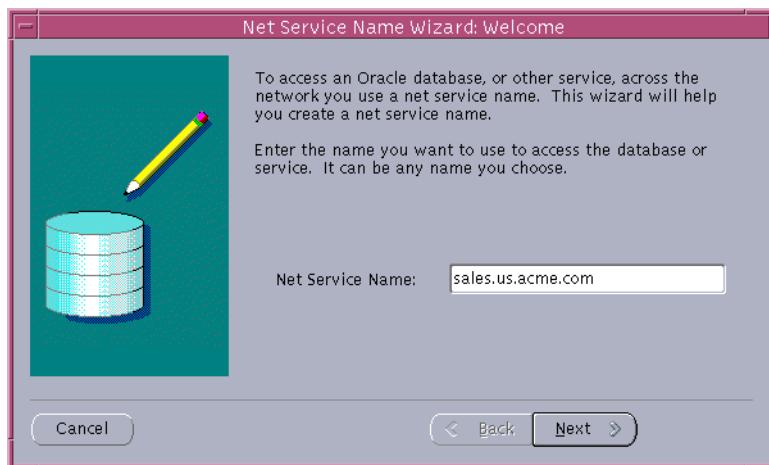
Net Service Name Wizard

The Net Service Name Wizard guides you through creating a basic net service name in a directory server or a `tnsnames.ora` file.

To start the Net Service Name Wizard to create net service names:

1. In the navigator pane, choose Directory or Local > Service Naming.
2. Choose plus (+) from the toolbar, or choose Edit > Create from the menu bar.

Figure 5–3 Opening Page of the Net Service Name Wizard



See Also: "Create a Net Service Name" topic in the Oracle Net Manager online help for detailed information about using the Net Service Name Wizard to create a net service name. To access this topic in the online help, choose Local or Directory > Service Naming > How To in the online help, and then select Create a Net Service Name.

Directory Server Migration Wizard

If a `tnsnames.ora` file already exists, its net service names can be exported to a directory server with the Directory Server Migration Wizard.

To start the Directory Server Migration Wizard, choose Command > Directory > Import Net Service Names in the Oracle Net Manager menu bar.

Figure 5–4 Opening Page of the Directory Server Migration Wizard



See Also: ["Exporting Net Service Names from a tnsnames.ora File" on page 10-2](#)

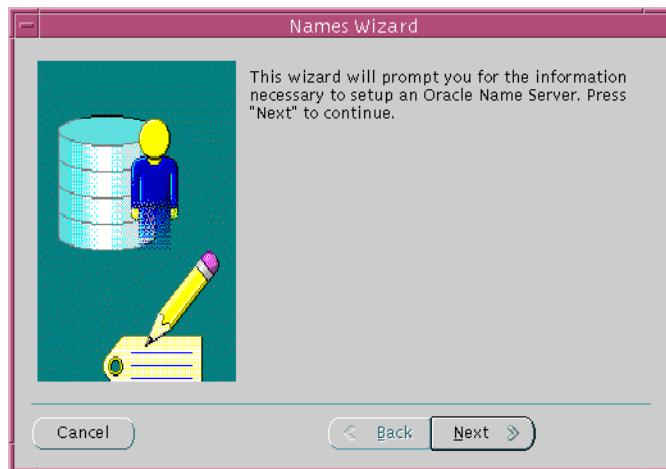
Names Wizard

The Names Wizard guides you through creating and configuring an Oracle Names server.

To start the Net Service Name Wizard to create an Oracle Names server:

1. In the navigator pane, choose Oracle Names Servers.
2. Choose plus (+) from the toolbar, or choose Edit > Create from the menu bar.

Figure 5–5 Welcome Page of the Names Wizard



See Also: ["Task 3: Create an Oracle Names Server"](#) on page 9-21
for more information about the Names Wizard

Note: In future releases, Oracle Names will not be supported as a centralized naming method. Because no new enhancements are being added to Oracle Names, consider using directory naming or migrating an existing Oracle Names configuration to directory naming, as described in [Chapter 10, "Exporting Naming Data to a Directory Server"](#).

Oracle Net Configuration Assistant

Oracle Net Configuration Assistant is provided primarily to configure basic network components during installation, including:

- Listener names and protocol addresses
- Naming methods the client will use to resolve **connect identifiers**
- Net service names in a `tnsnames.ora` file
- Directory server access

Oracle Net Configuration Assistant runs automatically during software installation, as described in your Oracle installation guide.

It can also be run after installation in standalone mode to configure naming method usage, the listener, net service names in the `tnsnames.ora` file, and directory server usage in a similar way that is provided during installation.

To start Oracle Net Configuration Assistant:

- On UNIX, run `netca` from `$ORACLE_HOME/bin`.
- On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Oracle Net Configuration Assistant.

See Also: Oracle Net Configuration Assistant online help

Figure 5–6 Opening Page of the Oracle Net Configuration Assistant



Table 5–1 describes the configuration options in the Welcome page:

Table 5–1 Oracle Net Configuration Assistant

Option	Description
Listener configuration	Choose this option to create, modify, delete, or rename a listener.
Naming Methods configuration	Choose this option to configure this computer to resolve connect identifiers to connect descriptor through one or more of following naming methods: <ul style="list-style-type: none">▪ Local naming▪ Directory naming▪ Oracle Names▪ Host naming▪ Network Information Service▪ Cell Directory Service
Local Net Service Name configuration	Choose this option to create, modify, delete, rename, or test connectivity of a connect descriptor stored in a local <code>tnsnames.ora</code> file.
Directory Usage configuration	Choose this option to configure a directory server for directory-enabled features.

Oracle Net Control Utilities

Oracle Net provides the following tools to help you start, stop, configure, and control each network component:

- [Listener Control Utility](#)
- [Oracle Connection Manager Control Utility](#)
- [Oracle Names Control Utility](#)

Listener Control Utility

The Listener Control utility enables you to administer the listener. The basic syntax for this utility is as follows:

```
lsnrctl command [listener_name]
```

For example, the following command starts a listener named lsnr:

```
lsnrctl START lsnr
```

You can also issue Listener Control utility commands at the LSNRCTL> program prompt. To obtain the prompt, enter `lsnrctl` with no arguments at the operating system command line. When you run `lsnrctl`, the program is started. You can then enter the necessary commands from the program prompt.

For example:

```
lsnrctl
LSNRCTL> START lsnr
```

See Also: *Oracle Net Services Reference Guide* for further information about the Listener Control utility

Oracle Connection Manager Control Utility

The Oracle Connection Manager Control utility enables you to administer an Oracle Connection Manager. The basic syntax for this utility follows:

```
cmctl command [process_type]
```

where *process_type* is the name of the Oracle Connection Manager process. The choices are as follows:

- cman for both the gateway process, CMGW, and the administrative process, CMADMIN
- cm for the CMGW process
- adm for the CMADMIN process

For example, the following command starts both the CMGW and the CMADMIN processes:

```
cmctl START cman
```

You can also issue Oracle Connection Manager utility commands at the CMCTL> program prompt. To obtain the prompt, enter `cmctl` with no arguments at the operating system command line. When you run `cmctl`, the program is opened. You can then enter the necessary commands from the program prompt.

For example:

```
cmctl  
CMCTL> START cman
```

See Also:

- "[Oracle Connection Manager Architecture](#)" on page 4-17 for an overview of the Oracle Connection Manager processes
- *Oracle Net Services Reference Guide* for a complete listing of Oracle Connection Manager Control utility commands

Oracle Names Control Utility

The Oracle Names Control utility enables you to administer an Oracle Names server. The basic syntax for this utility follows:

```
namesctl command [arguments]
```

For example, the following command starts Oracle Names server configured in the local node's names.ora file:

```
namesctl START
```

You can also issue Oracle Names Control utility commands at the NAMESCTL> program prompt. To obtain the prompt, enter namesctl with no arguments at the operating system command line. When you run namesctl, the program is started. You can then enter the necessary commands from the program prompt.

For example:

```
namesctl  
NAMESCTL> START
```

See Also: *Oracle Net Services Reference Guide* for further information about the Oracle Names Control utility

Configuration and Administration Tasks at a Glance

Network configuration and administration tasks are described throughout this guide. [Table 5–2](#) lists the common tasks, the tools associated with them, and points you to the topic in the guide that describes the task.

Table 5–2 Common Administrative Tasks for Oracle Net

Task	Tool(s) to Perform Task	See Also
Configuring Directory Server for Oracle Net Usage		
Configure directory server usage.	Oracle Net Configuration Assistant	Chapter 8, "Setting Up Directory Server Usage" on page 8-1
Add users to the OracleNetAdmins group.	Oracle Net Configuration Assistant	"Adding Users to and Removing Users from the OracleNetAdmins Group" on page 8-9
Authenticate with the directory.	Oracle Net Manager	Choose Directory > Service Naming > How To > Change the Oracle Context in the online help
Change the Oracle Context.	Oracle Net Manager	Choose Directory > Service Naming > How To > Set Authentication Credentials in the online help
Configuring Naming Methods		
Configure the local naming method.	Oracle Net Configuration Assistant Oracle Net Manager	"Configuring the Local Naming Method" on page 9-5
Configure the directory naming method.	Oracle Net Configuration Assistant Oracle Net Manager	"Configuring the Directory Naming Method" on page 9-12

Task	Tool(s) to Perform Task	See Also
Configure the Oracle Names method. Note: In future releases, Oracle Names will not be supported as a centralized naming method. Instead, consider using directory naming or migrating an existing Oracle Names configuration to directory naming, as described in Chapter 10, "Exporting Naming Data to a Directory Server".	Oracle Net Manager	"Configuring the Oracle Names Method" on page 9-19
Configure the host naming method.	Oracle Net Manager	"Configuring the Host Naming Method" on page 9-56
Configure external naming methods.	Oracle Net Manager	"Configuring External Naming Methods" on page 9-60
Migrating to Directory Naming		
Export from tnsnames.ora files	Oracle Net Manager	"Exporting Net Service Names from a tnsnames.ora File" on page 10-2
Export from Oracle Names	Oracle Names Control utility	"Exporting Network Objects from an Oracle Names Server" on page 10-6
Configuring Profiles		
Prioritize naming methods.	Oracle Net Manager Oracle Net Configuration Assistant	"Prioritizing Naming Methods" on page 11-5
Configure a default domain that is automatically appended to any unqualified net service name.	Oracle Net Manager Oracle Net Configuration Assistant	"Configuring a Default Domain for Clients" on page 11-3
Route connection requests.	Oracle Net Manager Oracle Net Configuration Assistant	"Routing Connection Requests" on page 11-7
Configure authentication methods available with Oracle Advanced Security .	Oracle Net Manager	"Configuring Oracle Advanced Security" on page 11-13 Choose Oracle Advanced Security > How To in the online help See Also: <i>Oracle Advanced Security Administrator's Guide</i>

Task	Tool(s) to Perform Task	See Also
Configuring Listeners		
Configure listening protocol addresses.	Oracle Net Manager Oracle Net Configuration Assistant	"Configuring Listening Protocol Addresses" on page 12-6
Configure dynamic service registration.	Automatic	"Configuring Service Registration" on page 12-14
Configure static service registration.	Oracle Net Manager	"Configuring Static Service Information" on page 12-10
Administering Listeners		
Start and stop listeners.	Listener Control utility	"Starting and Stopping the Listener" on page 12-20
View registered information.	Listener Control utility	"Monitoring Runtime Behavior" on page 12-20
Configuring Oracle Connection Manager		
Configure connection multiplexing.	Manual configuration	"Enabling Session Multiplexing" on page 13-13
Configure access control.	Manual configuration	"Enabling Access Control" on page 13-14
Configure protocol conversion support.	Manual configuration	"Enabling Protocol Conversion Support" on page 13-15
Configuring Oracle Names Servers		
Configure Oracle Names LDAP proxy servers.	Oracle Names Control utility Oracle Net Manager	"Configuring Oracle Names LDAP Proxy Servers" on page 10-15
Create Oracle Names servers.	Oracle Names Control utility Oracle Net Manager	"Task 3: Create an Oracle Names Server" on page 9-21
Create a list of preferred Oracle Names servers.	Oracle Net Manager	"Configuring Preferred Oracle Names Servers" on page 11-11
Register objects with Oracle Names servers.	Oracle Names Control utility Oracle Net Manager	"Task 7: Register Data with the Oracle Names Server" on page 9-21
Delegate domains to delegated administrative regions.	Oracle Names Control utility Oracle Net Manager	"Task 8: Delegate Domains to Delegated Administrative Regions" on page 9-50
Specify domain hints.	Manual configuration	"Task 9: Specify Domain Hints to Forward Requests to Remote Oracle Names Servers" on page 9-51

Task	Tool(s) to Perform Task	See Also
Administering Oracle Names Servers		
Start and stop Oracle Names servers, as well as gather statistics	Oracle Names Control utility Oracle Net Manager	<i>Oracle Net Services Reference Guide</i> Choose Oracle Names Servers > How To > Administer an Oracle Names server in the online help
Remove an Oracle Names server.	Oracle Names Control utility Oracle Net Manager	"Removing an Oracle Names Server" on page 9-54

6

Quick Start to Oracle Net Connections

This chapter is designed to help novice users set up and test a simple but common configuration—one between a client application and a database over a TCP/IP network.

This chapter contains these topics:

- [Prerequisites to Establishing Connectivity](#)
- [Task 1: Confirm Network Availability](#)
- [Task 2: Start the Listener and the Database Server](#)
- [Task 3: Configure the Client to Use a Net Service Name](#)
- [Task 4: Connect to the Database](#)

Prerequisites to Establishing Connectivity

The tasks in this quick start show a TCP/IP connection between a client computer and a database server. The following about the database server and client computers is assumed:

- Database Server Computer
 - It is running on the same network as the client
 - An Oracle database is installed
 - TCP/IP protocol support is installed
 - A listener is configured
- Client Computer
 - It is running on the same network as the database server
 - Oracle Client is installed
 - TCP/IP protocol support is installed

Task 1: Confirm Network Availability

Before using Oracle Net to connect a client computer to a database server, confirm that the client computer can successfully communicate with the database server computer. Evaluating network connectivity can eliminate network-based errors.

To confirm network connectivity:

1. Confirm that the database server computer can communicate with itself with a **loopback test**.

A loopback test is a connection from the database server back to itself. Many network protocols provide a means of testing network connections. The utility PING can be used for TCP/IP network.

In a TCP/IP network, each computer has a unique **IP address**. A name resolution service, such as **Domain Name System (DNS)**, can be used to map the IP address of a computer with its host name. If a name resolution service is not used, then the mapping is typically stored in a centrally maintained file called `hosts`. This file is located in the `/etc` directory on UNIX and the `\winnt` directory on Windows NT. For example, an entry for a database server computer named `sales-server` may look like the following:

```
#IP address of server      host name      alias  
144.25.186.203            sales-server    sales.us.acme.com
```

To use PING, enter the following at the command line:

```
ping database_server_host
```

The `database_server_host` is the host name of the database server computer. For example:

```
ping sales-server
```

If the loopback was unsuccessful, try using the IP address of the database server. For example:

```
ping 144.25.186.203
```

2. Verify the client computer can successfully communicate with the database server computer.

This varies according to the network protocol. For TCP/IP, you can use PING, FTP or TELNET utilities. If the client computer cannot reach the server, verify

that the network cabling and network interface cards are correctly connected. Contact your network administrator to correct these problems.

Task 2: Start the Listener and the Database Server

The listener and database server must be running in order for the database server to receive connections.

1. Start the listener with the Listener Control utility. From the command line, enter:

```
lsnrctl  
LSNRCTL> START [listener_name]
```

where *listener_name* is the name of the listener defined in the *listener.ora* file. It is not necessary to identify the listener if you are using the default listener, named LISTENER.

A status message indicating that the listener has started successfully displays.

2. Start the database:
 - a. Start SQL*Plus without connecting to the database:

```
sqlplus /nolog
```
 - b. Connect to the database as SYSDBA:

```
SQL> CONNECT username/password as sysdba
```

For example, SYSTEM/MANAGER is a SYSDBA user.
 - c. Enter the STARTUP command, specifying the database name and full path of the parameter file:

```
SQL> STARTUP database_name pfile=file
```

If you do not specify the PFILE option, the Oracle database uses the standard initialization parameter file located in the \$ORACLE_BASE/admin/*db_name*/pfile/*sid* directory on UNIX platforms, and ORACLE_BASE\admin*db_name*\pfile*sid* directory on Windows NT. If you do not specify a database name, then the database uses the value of the DB_NAME parameter specified in the initialization parameter file.

See Also: *Oracle9i Database Administrator's Guide* for further information about starting the database

3. Confirm that database **service registration** with the listener has completed. From the Listener Control utility, enter:

```
LSNRCTL> SERVICES [listener_name]
```

The SERVICES command lists the services supported by the database, along with at least one available **service handler**.

See Also: "SERVICES Command" on page 12-23

Task 3: Configure the Client to Use a Net Service Name

Once network connectivity has been verified, you can use **Oracle Net Configuration Assistant** to create a **net service name**, a simple name for the database service. The net service name resolves to the **connect descriptor**, that is, the network address of the database and the name of the database service. The client will use the net service name to connect to the database.

The following example shows the net service name sales mapped to a connect descriptor for a database called sales.us.acme.com. A client can use sales mapped to connect to sales.us.acme.com.

```
sales=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=sales.us.acme.com) ) )
```

To configure a client with a net service name:

1. Start Oracle Net Configuration Assistant:
 - On UNIX, run netca from \$ORACLE_HOME/bin.
 - On Windows NT, choose Start > Programs > Oracle - HOME_NAME > Network Administration > Oracle Net Configuration Assistant.

The Welcome page appears.

2. Select "Local Net Service Name Configuration."
3. Choose Next.

The Net Service Name Configuration page appears.

4. Choose Add, and then choose Next.

The Net Service Name Configuration, Database Version page appears:

5. If the destination service is an Oracle9*i* or Oracle8*i* database, then choose "Oracle8*i* or later database or service". If destination service is an Oracle8 release 8.0 database, then choose "Oracle8 release 8.0 database or service".
6. Choose Next.

The Net Service Name Configuration, Service Name appears.

7. Enter the name that identifies the database.

The **service name** is defined during database creation. If you are unsure what the service name is, ask the Database Administrator who created the database.

8. Choose Next.

The Net Service Name Configuration, Select Protocols page appears.

9. Select the protocol on which the listener is configured to listen. Note that this protocol must also be installed on the client. The listener is configured to listen on TCP/IP by default.

10. Choose Next.

The page appropriate for the selected protocol appears.

11. Enter the appropriate protocol parameters for the selected protocol in the fields provided, and then choose Next.

The Net Service Name Configuration, Test page appears.

12. Choose "Yes perform a test."

During a test, Oracle Net Configuration Assistant contacts the remote database service, establishes a connection, and ends contact.

A successful test results in the following message:

Connecting...Test successful.

If the test fails, it can be because the:

- Default user name (`scott`) and password (`tiger`) are not valid
- Protocol address information does not match the listener information
- Listener is not running
- Destination database service is down

Depending on your diagnosis of the problem, perform one of the following tasks:

- Choose Change Login to change the user name and password for the connection.
 - Choose Back to review protocol address information.
 - Start the listener or database on the server, as described in "[Task 2: Start the Listener and the Database Server](#)" on page 6-4.
- 13.** Choose Next.
- The Net Service Name Configuration, Net Service Name page appears.
- 14.** Accept the default net service name or enter another net service name in the Net Service Name field. The name you enter should be unique to the client.
- 15.** Choose Next.
- The Net Service Name Configuration, Another Net Service Name page appears.
- 16.** Choose No, and then choose Next.
- The Net Service Name Configuration, Configuration Done page appears.
- 17.** Choose Next, and then choose Finish.

Task 4: Connect to the Database

From the client computer, connect to the database server as follows.

- 1.** Start SQL*Plus:

```
sqlplus
```

- 2.** Connect to the database as follows:

```
CONNECT username/password@net_service_name
```

where *username* and *password* are the database user and password, and *net_service_name* is the net service name that you created in "[Task 3: Configure the Client to Use a Net Service Name](#)" on page 6-5.

Task 4: Connect to the Database

Part II

Configuration and Administration of Oracle Net Services

Part II describes how to set up and configure Oracle Net Services.

This part contains these chapters:

- [Chapter 7, "Planning the Network"](#)
- [Chapter 8, "Setting Up Directory Server Usage"](#)
- [Chapter 9, "Configuring Naming Methods"](#)
- [Chapter 10, "Exporting Naming Data to a Directory Server"](#)
- [Chapter 11, "Configuring Profiles"](#)
- [Chapter 12, "Configuring and Administering the Listener"](#)
- [Chapter 13, "Configuring Oracle Connection Manager"](#)
- [Chapter 14, "Configuring Shared Server"](#)
- [Chapter 15, "Enabling Advanced Features for Oracle Net Services"](#)

Planning the Network

Oracle Net Services provide a variety of options to help you design and manage networks that are both flexible and easy to use. With Oracle Net Services enhanced scalability and manageability features, you can develop a network to support a wide range of environments, whether they be simple workgroups or large mission critical enterprises.

This chapter describes considerations for planning a network using Oracle Net Services. It explains the relationships of the network products, and options for expanding and better managing your future network.

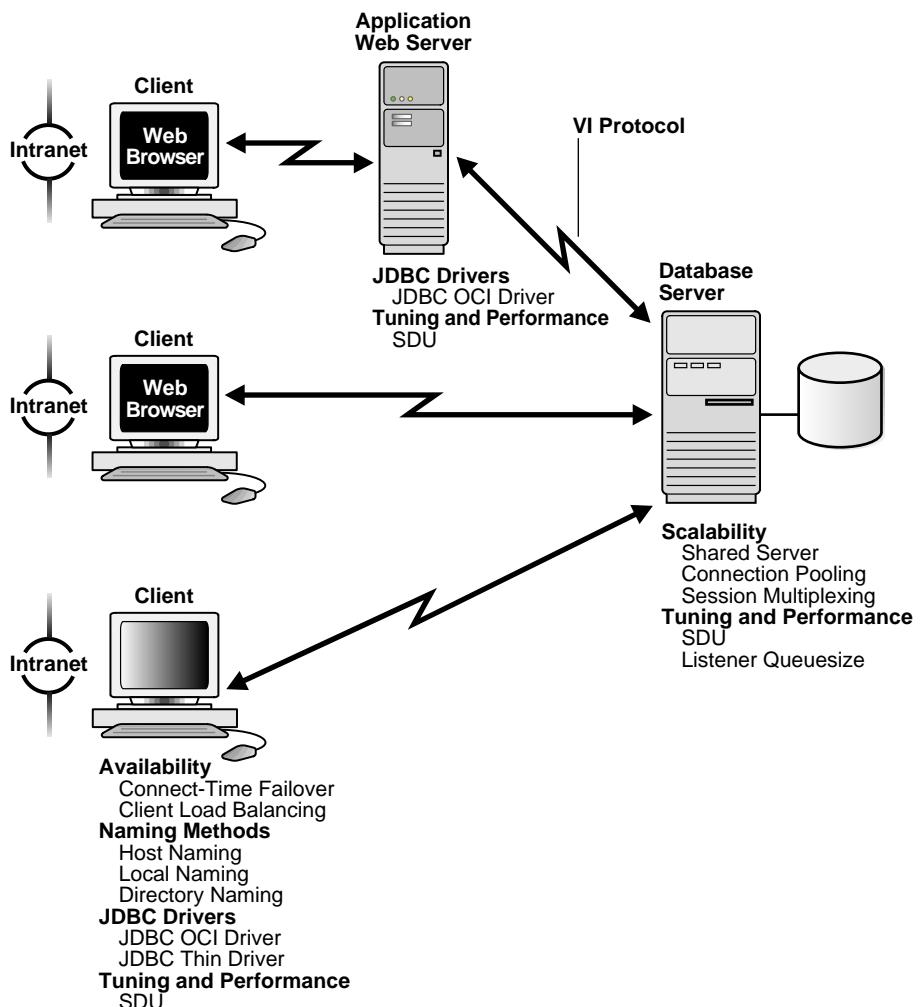
This chapter contains these topics:

- [Deploying a Network Inside an Organization](#)
- [Deploying a Network for the Internet](#)
- [Naming Considerations](#)
- [Performance Considerations](#)

Deploying a Network Inside an Organization

As Figure 7-1 shows, an internal network often has a diverse client makeup. Besides traditional clients that need access to the database servers, Web clients also need access. The type and number of clients, as well as other networking issues, affect the types of features to consider deploying.

Figure 7-1 Internal Network Layout



This section addresses deployment considerations for an internal network in terms of the following categories:

- [Scalability](#)
- [Availability](#)
- [Naming Methods](#)
- [JDBC Drivers](#)
- [Security](#)
- [Tuning and Performance](#)

Scalability

For an organization with hundreds or thousands of clients, scalability is of the utmost importance. Oracle Net Services offer a number of features that can improve database server scalability, including **shared server**, **connection pooling**, and **session multiplexing**.

Shared Server

With a shared server architecture, a small pool of server processes can serve a large number of clients. This reduces the server's memory requirements. Connection pooling and session multiplexing features are enabled through shared server.

Connection Pooling

Connection pooling enables the database server to time-out protocol connections and to use those connections to service an active session.

Session Multiplexing

Session multiplexing, available with Oracle Connection Manager, enables multiple client sessions to funneled through a single protocol connection.

Together, these features increase the total number of client sessions the server can handle.

The following table summarizes the relative advantages of using connection pooling and session multiplexing and provides recommendations for using them in the network.

Table 7–1 Connection Pooling and Session Multiplexing

Feature	Advantages	Disadvantages	Recommended for
Connection pooling	<ul style="list-style-type: none">■ Limits the number of network resources used for each process■ Supports large client populations■ Maximizes the number of client/server sessions over a limited number of process connections■ Optimizes resource utilization■ Enables identification and monitoring of real users■ Enables mid-tier applications to support additional services■ Requires only a single transport for clients with multiple applications■ Requires only a single network connection for database links	Database sessions should use the <code>IDLE_TIME</code> resource parameter.	Networks where many clients run interactive "high think/search time" applications such as messaging and OLAP

Feature	Advantages	Disadvantages	Recommended for
Session multiplexing	<ul style="list-style-type: none"> ▪ Limits the number of network resources used for each process ▪ Supports large client populations ▪ Maximizes the number of client/server sessions over a limited number of process connections ▪ Optimizes resource utilization ▪ Enables identification and monitoring of real users ▪ Enables mid-tier applications to support additional services ▪ Requires only a single transport for clients with multiple applications ▪ Requires only a single network connection for database links ▪ Provides support for pre-Oracle8 clients 	Clients must connect to Oracle Connection Manager.	Networks where continuous connectivity is required.

Availability

Availability to the database is crucial for any internal network. Multiple listeners can be configured to handle client connection requests for the same database service. This is especially ideal in an Oracle9i Real Application Clusters configuration, where each instance has a listener associated with it. Multiple listener configurations enable you to utilize **connect-time failover** and **connection load balancing** features.

Connect-Time Failover

Connect-time failover enables clients to request a different listener (usually on a different node) if the first listener fails.

Client Load Balancing

Client load balancing enables clients to randomize requests to the multiple listeners (usually on different nodes).

These features can be used together or separately. Together, they ensure access to the database and distribute the load so as not to overburden a single listener.

Naming Methods

Selecting the appropriate **naming method** for mapping names to **connect descriptors** depends upon the size of the organization.

For a small organization with only a few databases, use **host naming** to store names in an existing names resolution service, or **local naming** to store names in `tnsnames.ora` file on the clients.

For large organizations with several databases, use **directory naming** to store names in a centralized LDAP-compliant directory server.

See Also: "Naming Considerations" on page 7-10 for further information about selecting a naming method

JDBC Drivers

Java client applications access an Oracle database through a **Java Database Connectivity (JDBC) Driver**—a standard Java interface for connecting from Java to a relational database. Oracle Corporation offers the following drivers:

- OCI driver for clientside and application Web server use with an Oracle client installation
- Thin driver for clientside use without an Oracle installation, particularly with applets

Security

Internal network should be inside a firewall.

See Also: *Oracle Advanced Security Administrator's Guide* for further information about providing security for the internal network

Tuning and Performance

Oracle Net Services offers a number of features that can help reduce round-trip time across the network, increase listener performance, and reduce the number of protocols used.

VI Protocol

You can reduce round-trip time between application Web servers and database server with the **VI protocol**. The VI protocol can be used in place of TCP/IP, which has significant messaging overhead.

Session Data Unit (SDU) Size

Before sending data across the network, Oracle Net buffers and encapsulates data into the **session data unit (SDU)**. Oracle Net sends the data stored in this buffer when the buffer is full, flushed, or when database server tries to read data. When large amounts of data are being transmitted or when the message size is consistent, adjusting the size of the SDU buffers can improve performance, network utilization, or memory consumption. You can deploy SDU at the client, the application Web server, and the database server.

Listener Queue Size

If you anticipate receiving a large number of connection requests for a listening process, you can increase the size of the listener queue.

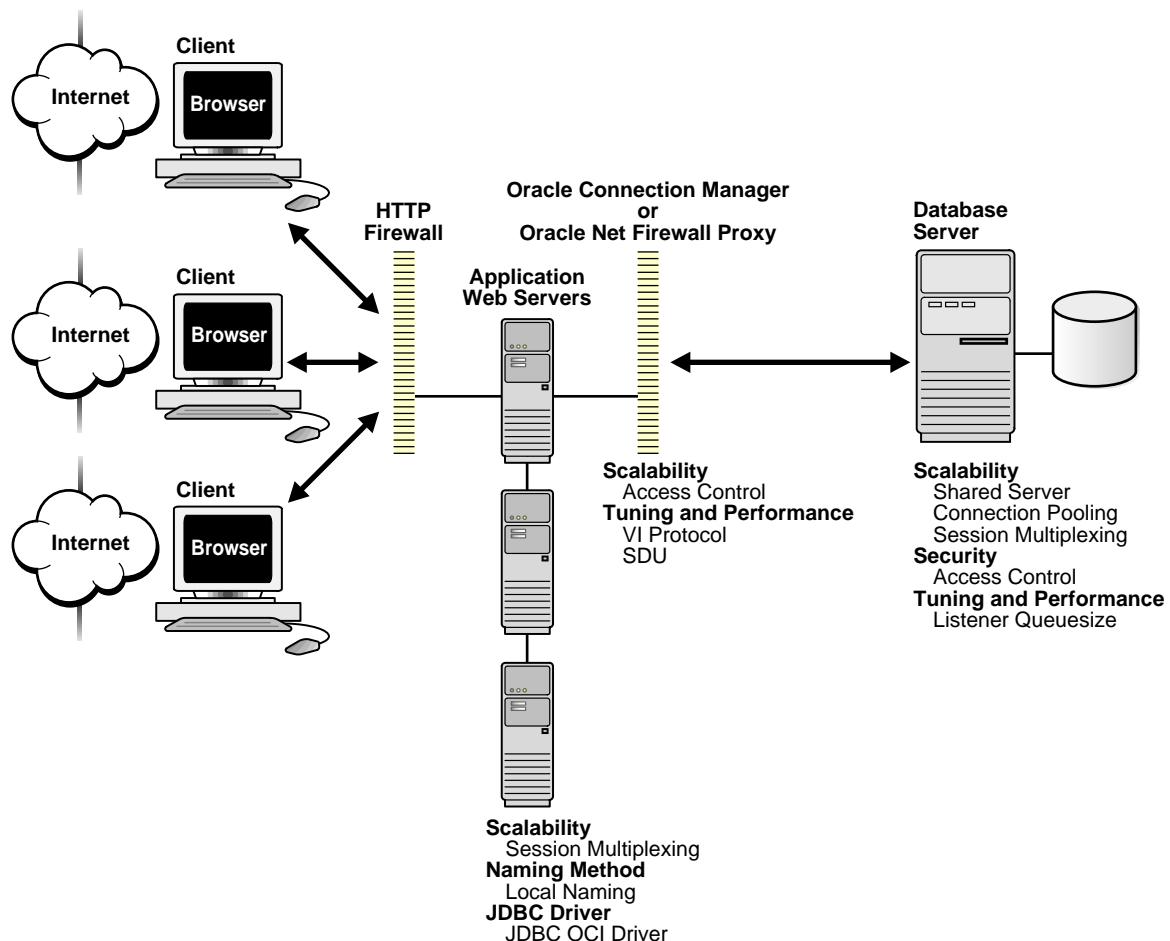
Protocol Conversion

The database only needs to be configured to listen on one protocol address, even though clients may use other protocols. Oracle Connection Manager provides a **protocol conversion** feature that enables a client and database server configured with different networking protocols to communicate with one another.

Deploying a Network for the Internet

As Figure 7–2 shows, an Internet network is usually consists of Web clients that access firewall-protected application Web servers, that then connect to a database. An Internet network has many of the same requirements as an internal network, but it also has its own unique set of requirements. This section examines both requirements.

Figure 7–2 Internet Network Layout



Scalability

Like an internal network, scalability can be improved in an Internet network with shared server, connection pooling, and session multiplexing features configured on the database server. Session multiplexing can also be configured on the application Web server tier. This can help offload some of the network I/O of the application Web servers, increasing throughput.

Availability

Availability to the database is crucial for any Internet network with a firewall. It is important to deploy at least two Oracle Connection Manager firewalls or Oracle Net Firewall Proxies in the event that one firewall goes down.

Naming Methods

For an Internet network, configure the application Web servers needed to connect to the databases with the local naming method.

See Also: ["Naming Considerations"](#) on page 7-10 for further information about selecting a naming method

JDBC Drivers

Java applications that reside on the application Web server require the JDBC OCI driver.

Security

Security in an Internet network is crucial to control access to the database.

Access Control

Granting and denying access to the a database is crucial for a secure network environment. Access control can be configured through a firewall or on the database. Oracle Connection Manager can be configured to act as a firewall, granting or denying clients access to a particular database service or a computer based on a set of filtering rules. The database can also be configured with parameters in the `sqlnet.ora` file to specify that clients using a particular protocol are allowed or denied access.

VI Protocol

Because the VI protocol is not an IP technology like TCP/IP—that is, there are few applications, such as `telnet` and `ftp`, enabled to use VI—it is secure by nature.

Tuning and Performance

The performance and tuning features described in "[Tuning and Performance](#)" on page 7-7 can also be deployed for an Internet network.

Naming Considerations

Table 7-2 summarizes the relative advantages and disadvantages of each naming method and provides recommendations for using them in the network.

Table 7-2 Naming Methods: Advantages and Disadvantages

Naming Method	Advantages/Disadvantages	Recommended for:
Local Naming	<p>Advantages:</p> <ul style="list-style-type: none">■ Provides a relatively straightforward method for resolving net service name addresses■ Resolves net service names across networks running different protocols <p>Disadvantage: Requires local configuration of all net service name and address changes</p>	Simple distributed networks with a small number of services that change infrequently.
Directory Naming	<p>Advantages:</p> <ul style="list-style-type: none">■ Centralizes network names and addresses in a single place, facilitating administration of name changes and updates. This eliminates the need for an administrator to make changes to what potentially could be hundreds or even thousands of clients.■ Directory stores names for other services.■ Tools provide simple configuration. <p>Disadvantage: Requires access to a directory server</p>	Large, complex networks (over 20 databases) that change on a frequent basis.

Naming Method	Advantages/Disadvantages	Recommended for:
Oracle Names	<p>Advantage: Centralizes network names and addresses in a single place, facilitating administration of name changes and updates. For example, whenever a change is made to an existing server or a new server is added to the network, the change is made only once on one Oracle Names server. This eliminates the need for an administrator to make changes to what potentially could be hundreds or even thousands of clients.</p> <p>Disadvantages:</p> <ul style="list-style-type: none"> ▪ Oracle Names stores network names and addresses for Oracle services only ▪ Requires additional setup and administration of Oracle Names servers <p>Note: In future releases, Oracle Names will not be supported as a centralized naming method. Instead, consider using directory naming or migrating an existing Oracle Names configuration to directory naming, as described in Chapter 10, "Exporting Naming Data to a Directory Server".</p>	Large, complex networks (over 20 databases) that change on a frequent basis.
Host Naming	<p>Advantages:</p> <ul style="list-style-type: none"> ▪ Requires minimal user configuration. The user can provide only the name of the host to establish a connection. ▪ Eliminates the need to create and maintain a local names configuration file (<code>tnsnames.ora</code>) <p>Disadvantage: Available only in a limited environment, as indicated in the Recommended for column</p>	<p>Simple TCP/IP networks that meet the criteria listed:</p> <ul style="list-style-type: none"> ▪ Your client and server are connecting using TCP/IP. ▪ The host name is resolved through an IP address translation mechanism such as Domain Name Services (DNS), Network Information Services (NIS), or a centrally maintained TCP/IP hosts file. ▪ No Oracle Connection Manager features are requested.

Naming Method	Advantages/Disadvantages	Recommended for:
External Naming	Advantage: Enables administrators to load Oracle net service name into their native name service using tools and utilities with which they are already familiar Disadvantage: Requires a third-party naming services that cannot be administered using Oracle Net products	Networks with existing name services.

Performance Considerations

This section covers performance considerations. It includes the following topics:

- [Listener Queue Size](#)
- [Session Data Unit Size for Data Transfer Optimization](#)
- [Persistent Buffer Flushing for TCP/IP](#)

Listener Queue Size

If you anticipate receiving a large number of connection requests for a listening process (such as a listener, Oracle Connection Manager, or Oracle Names server) over TCP/IP, Oracle Net enables you to configure the listening queue to be higher than the system default.

Session Data Unit Size for Data Transfer Optimization

Tuning your application to reduce the number of round trips across the network is the best way to improve your network performance. If this is done, it is also possible to optimize data transfer by adjusting the size of the session data unit (SDU).

The SDU is a buffer that Oracle Net uses to place data into before transmitting it across the network. Oracle Net sends the data in the buffer either when requested or when it is full.

Table 7–3 outlines considerations for modifying the size of the SDU.

Table 7–3 SDU Considerations

Modify SDU size when:	Do not modify SDU size when:
<ul style="list-style-type: none">■ The data coming back from the server is fragmented into separate packets■ You are on a wide area network (WAN) that has long delays■ Your packet size is consistently the same■ Large amounts of data are returned	<ul style="list-style-type: none">■ Your application can be tuned to account for the delays■ You have a higher speed network where the effect of the data transmission is negligible■ Your requests return small amounts of data from the server

Persistent Buffer Flushing for TCP/IP

Under certain conditions for some applications using TCP/IP, Oracle Net packets may not get flushed immediately to the network. Most often, this behavior occurs when large amounts of data are streamed. The implementation of TCP/IP itself is the reason for the lack of flushing, causing unacceptable delays. To remedy this problem, specify no delays in the buffer flushing process.

See Also: *Oracle Net Services Reference Guide* for further information about the TCP .NODELAY parameter

Planning Summary

Table 7-4 summarizes many of the options you may have chosen as you planned your network.

Table 7-4 Oracle Net Feature Summary

Feature	See Also
Scalability Features	
Connection pooling	"Enabling Connection Pooling" on page 14-8
Session multiplexing	"Enabling Session Multiplexing" on page 13-13
Shared server	"Configuring Shared Server with the DISPATCHERS Parameter" on page 14-2
Availability Features	
Client load balancing	"Configuring Address List Parameters" on page 15-4
Connect-time failover	"Configuring Address List Parameters" on page 15-4
Naming Method Features	
Directory Naming	"Configuring the Directory Naming Method" on page 9-12
Host Naming	"Configuring the Host Naming Method" on page 9-56
Local Naming	"Configuring the Local Naming Method" on page 9-5
Oracle Names	"Configuring the Oracle Names Method" on page 9-19
JDBC Drivers	
<i>Oracle9i JDBC Developer's Guide and Reference</i>	
Security Features	
Access control	"Enabling Access Control" on page 13-14 to configure Oracle Connection Manager "Configuring Database Access Control" on page 11-8 to configure access control parameters in <code>sqlnet.ora</code>
VI protocol	"Configuring the VI Protocol for Application Web Server and Oracle Database Server Communication" on page 15-30

Feature	See Also
Tuning and Performance Features	
Listener queue size	"Starting and Stopping the Listener" on page 12-20
Protocol conversion	"Enabling Protocol Conversion Support" on page 13-15
SDU size	"Configuring Session Data Unit" on page 15-28
VI protocol	"Configuring the VI Protocol for Application Web Server and Oracle Database Server Communication" on page 15-30

Setting Up Directory Server Usage

This chapter explains how to configure access to an LDAP-compliant directory server.

This chapter contains these topics:

- [Directory Configuration Overview](#)
- [Configuring Directory Usage During Installation](#)
- [Configuring Directory Usage After Installation](#)
- [Adding Users to and Removing Users from the OracleNetAdmins Group](#)

See Also: ["Directory Server Support"](#) on page 3-4 for an overview of directory server support of Oracle Net Services

Directory Configuration Overview

Many Oracle products have features that use an LDAP-compliant directory server to centrally store entries. Examples of features that use a directory are Oracle Net **directory naming** and **Oracle Advanced Security enterprise user**. If you want to use these features, you must establish a directory server for them, as well as enable your computers to use the directory server.

Directory server usage can be configured during or after installation, as described in the following sections:

- [Configuring Directory Usage During Installation](#)
- [Configuring Directory Usage After Installation](#)

Configuring Directory Usage During Installation

Oracle Universal Installer launches **Oracle Net Configuration Assistant** during software installation. Oracle Net Configuration Assistant enables you to configure usage of a directory server. Directory server usage configuration varies depending upon the installation mode you selected during installation, as described in these topics:

- [Directory Usage Configuration During a Custom Installation on the Server](#)
- [Directory Usage Configuration During a Client Installation](#)

Directory Usage Configuration During a Custom Installation on the Server

After a Custom installation on the server, Oracle Net Configuration Assistant prompts you to configure usage to a directory server. Directory server usage configuration enables:

- Oracle Database Configuration Assistant, which runs after Oracle Net Configuration Assistant, to register a database service entry in the directory server
- Oracle Net Manager to create net service names in the directory server, as well as to modify Oracle Net attributes of the database service entry and the net service name entries
- The server to look up database service and net service name entries in the directory server

Note: Directory usage configuration is not performed during a Enterprise Edition or Standard Edition installation on the server. For these installation types, Oracle Net Configuration Assistant can be run in standalone mode. See "[Configuring Directory Usage After Installation](#)" on page 8-5 for details.

During directory server usage configuration, Oracle Net Configuration Assistant prompts you to:

- Select the type of directory server, that is, Oracle Internet Directory or Microsoft Active Directory
- Identify the location of the directory server
- Select a location in the directory server that contains an **Oracle Context** from which this server can look up, create, and modify connect identifiers

The Oracle Context (`cn=OracleContext`) is the root of a directory subtree under which all Oracle software relevant information is kept.

The configuration information is stored in an `ldap.ora` file that the server reads to locate the directory server and access Oracle entries.

If an Oracle Context does not exist in the directory under the selected administrative context, then Oracle Net Configuration Assistant prompts you to create it. During Oracle Context creation, you are prompted for directory administrator authentication credentials. If the Oracle Context is created successfully, then the authenticated user is added to the following groups:

- **OracleDBCreators** (`cn=OracleDBCreators, cn=OracleContext`)
As a member of OracleDBCreators, a user can use Oracle Database Configuration Assistant to register a database service entry.
- **OracleNetAdmins** (`cn=OracleNetAdmins, cn=OracleContext`)
As a member of OracleNetAdmins, a user can use Oracle Net Manager to create, modify, and delete net service names, as well as modify Oracle Net attributes of database services.

A directory administrator can add other users to these groups.

Note: Additional groups are created during Oracle Context creation, as described in the *Oracle Directory Service Integration and Deployment Guide*.

In addition, Oracle Net Configuration Assistant verifies that the **Oracle schema** was created. The Oracle schema defines the Oracle entries and their attributes. If the schema does not exist or is an older version, you are prompted to create or upgrade it. During Oracle schema creation, you are prompted for authentication credentials.

After Oracle Net Configuration Assistant completes configuration, Oracle Database Configuration Assistant creates the database. The service name for the database is automatically created under the Oracle Context.

See Also:

- ["Adding Users to and Removing Users from the OracleNetAdmins Group" on page 8-9](#) to add users to the NetAdmins group
- *Oracle Advanced Security Administrator's Guide* for further information about adding users to the OracleDBCreators group
- Oracle installation guide

Directory Usage Configuration During a Client Installation

During client installation, Oracle Net Configuration Assistant prompts you to configure the use of a directory server. Directory server usage configuration enables the client to look up connect identifier entries in the directory. If directory server access is not configured, the client cannot use directory naming.

Oracle Net Configuration Assistant typically performs the necessary directory server usage configuration during client installation and stores the following in a read-only `ldap.ora` file.

During directory server access configuration, Oracle Net Configuration Assistant prompts you to:

- Specify the type of directory server
- Identify the location of the directory server
- Select a location in the directory that contains an Oracle Context from which this client can look up connect identifiers

This setting information is stored in a `ldap.ora` file that the client reads to locate the directory server and to access Oracle entries.

In addition, Oracle Net Configuration Assistant verifies that the Oracle schema was installed. If an Oracle Context or the Oracle schema was not configured by the server, you cannot complete directory server usage configuration on the client.

See Also: ["Directory Usage Configuration During a Custom Installation on the Server" on page 8-2](#)

Configuring Directory Usage After Installation

Directory usage can be configured with Oracle Net Configuration Assistant at any time.

To configure directory server usage:

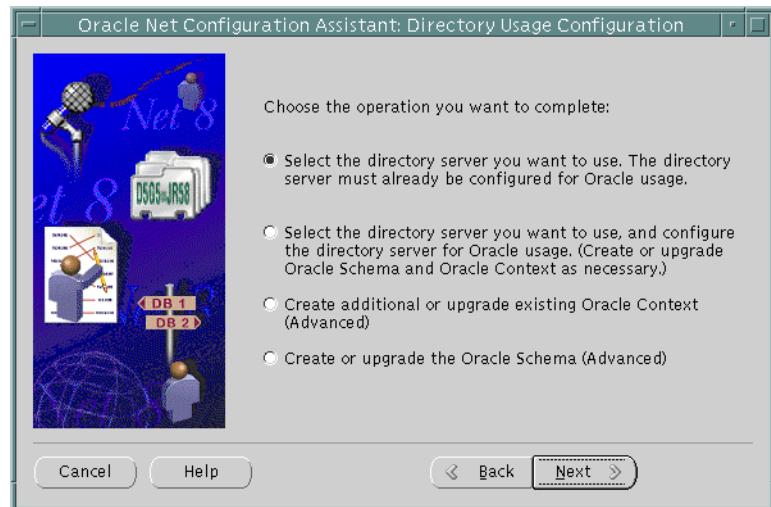
1. Start Oracle Net Configuration Assistant.

See Also: ["Oracle Net Configuration Assistant"](#) on page 5-14

The Welcome page appears.

2. Select Directory Service Usage Configuration, and then choose Next.

The Directory Usage Configuration page appears.



The Directory Usage Configuration page options are described in [Table 8–1](#).

Table 8–1 Directory Usage Configuration Page in Oracle Net Configuration Assistant

Option	Description
Select the directory server you want to use	<p>Select this option to enable this computer to use a directory server that is already configured to use directory-enabled features. This option is ideal for clients that use a directory server that has already been configured for these features.</p> <p>Once configuration is complete, this option enables this computer to look up entries in the directory. This option prompts you to:</p> <ul style="list-style-type: none">▪ Select the type of directory server▪ Identify the location of the directory server▪ Select a location in the directory server that contains an Oracle Context from which this client can look up connect identifiers <p>Note: If no Oracle Context or Oracle schema exists, then you cannot complete usage configuration using this option. You must first use the "Select the directory server you want to use, and configure the directory server for Oracle usage" option to create the Oracle Context and Oracle schema.</p>

Option	Description
<p>Select the directory server you want to use, and configure the directory server for Oracle usage.</p>	<p>Select this option to configure a directory server for directory-enabled features and enable this computer to use that directory. This option is designed for administrators when first configuring these features.</p>

Once configuration is complete, this computer can then look up entries in the directory server. This option prompts you to:

- Select the type of directory server
- Identify the location of the directory server
- Select or enter a location in the directory server that contains an Oracle Context from which this server can access and create Oracle entries

If an Oracle Context does not exist under the selected location, then Oracle Net Configuration Assistant prompts you to create one. Likewise, if the Oracle schema does not exist or is an older version, you are prompted to create or upgrade it. During Oracle Context or Oracle schema creation or upgrade, you are prompted for directory administrator authentication credentials. To create an Oracle Context, the following must exist in the directory server:

- A directory entry under which you want the Oracle Context to be created
- Oracle schema

If the Oracle Context is created successfully, then the authenticated user is added to the following groups:

- OracleDBCreators (`cn=OracleDBCreators, cn=OracleContext`)
- OracleNetAdmins (`cn=OracleNetAdmins, cn=OracleContext`)

See Also:

- Directory server vendor documentation for directory entry configuration instructions
- *Oracle Directory Service Integration and Deployment Guide* for further information about all the user groups
- "[Adding Users to and Removing Users from the OracleNetAdmins Group](#)" on page 8-9 to add users to the OracleNetAdmins group
- *Oracle Advanced Security Administrator's Guide* for further information about adding users to the OracleDBCreators group

Option	Description
Create additional or upgrade existing Oracle Context	<p>Select this option to create an additional Oracle Context in the directory, or upgrade the Oracle Context to the current release.</p> <p>To create an Oracle Context, the following must exist in the directory server:</p> <ul style="list-style-type: none"><li data-bbox="437 364 1288 393">▪ A directory entry under which you want the Oracle Context to be created<li data-bbox="437 404 1288 433">▪ Oracle schema <p>During Oracle Context creation or upgrade, you are prompted for directory administrator authentication credentials.</p> <p>If the Oracle Context is created successfully, then the authenticated user is added to the following groups:</p> <ul style="list-style-type: none"><li data-bbox="437 571 1288 601">▪ OracleDBCreators (<code>cn=OracleDBCreators,cn=OracleContext</code>)<li data-bbox="437 611 1288 641">▪ OracleNetAdmins (<code>cn=OracleNetAdmins,cn=OracleContext</code>)
Create or upgrade the Oracle Schema	Select this option to create the Oracle schema in the directory, or upgrade the Oracle schema to the current release. During Oracle schema creation or upgrade, you are prompted for authentication credentials.

3. Select the appropriate option, and then follow the prompts in the wizard and online help to complete directory server access configuration.

Adding Users to and Removing Users from the OracleNetAdmins Group

The directory user who creates the Oracle Context is a member of the OracleNetAdmins (`cn=OracleNetAdmins, cn=OracleContext`) group. Using directory tools, such as `ldapmodify`, a directory administrator or the directory user who created the Oracle Context can add users to this group.

To add a user to the OracleNetAdmins group with `ldapmodify`:

1. Create an **LDAP Data Interchange Format (LDIF)** file that specifies that you want to add a user to the OracleNetAdmins group. You can use the following sample LDIF file. Use the appropriate **distinguished name (DN)** for `cn=OracleNetAdmins` and the user that you want to add.

```
dn: cn=OracleNetAdmins,cn=OracleContext, ...
changetype: modify
add: uniquemember
uniquemember: <DN of user being added to group>
```

2. Enter the following `ldapmodify` syntax to add a user:

```
ldapmodify -h directory_host -p port -D binddn -w password -f ldif_file
```

Table 8–2 *ldapmodify Arguments*

Argument	Description
<code>-h <i>directory_host</i></code>	Specify the directory server host.
<code>-p <i>port</i></code>	Specify the listening TCP/IP port for the directory server. If you do not specify this option, the default port (389) is used.
<code>-D <i>binddn</i></code>	Specify the directory administrator or user DN.
<code>-w <i>password</i></code>	Specify the password for the directory administrator or directory user.
<code>-f <i>ldif_file</i></code>	Specify the input file name.

To remove a user from the OracleNetAdmins group with `ldapmodify`:

1. Create an LDIF file that specifies that you want to add a user to the OracleNetAdmins group. You can use the following sample LDIF file. Enter the appropriate DN for `cn=OracleNetAdmins` and the user that you want to add.

```
dn: cn=OracleNetAdmins,cn=OracleContext,...
```

```
changetype: modify
```

```
delete: uniquemember
```

```
uniquemember: <DN of user being delete from group>
```

2. Use the following `ldapmodify` syntax to delete the user:

```
ldapmodify -h directory_host -p port -D binddn -w password -f ldif_file
```

Configuring Naming Methods

This chapter describes how to configure **naming methods**.

This chapter contains these topics:

- [Naming Method Configuration Overview](#)
- [About Connect Descriptors](#)
- [Configuring the Local Naming Method](#)
- [Configuring the Directory Naming Method](#)
- [Configuring the Oracle Names Method](#)
- [Configuring the Host Naming Method](#)
- [Configuring External Naming Methods](#)

See Also: "Naming" on page 2-15 for an overview of naming methods

Naming Method Configuration Overview

To connect to a service, clients use a simple name, called a **connect identifier** in their connect string to connect to a service, rather than a complete **connect descriptor**. The connect descriptor contains:

- Network route to the service, including the location of the listener through a protocol address
- Service name of an Oracle9*i* or Oracle8*i* database or **Oracle System Identifier (SID)** of an Oracle release 8.0 database

A connect identifier can be **net service name** (a simple name for a service) or the actual name of the service. A connect identifier is resolved to a connect descriptor by a **naming method** specified for the client.

Once the connect identifier is resolved, the client then forwards the connection request to the listener. The listener compares the client information with the information it has received from the database service, as well as information it has stored in its own configuration file, `listener.ora`. If the information matches, a connection is granted.

Naming method configuration consists of creating connect descriptors for services and configuring clients to access the naming method.

About Connect Descriptors

A connect descriptor is comprised of one or more protocol addresses of the listener and connect data information for the destination service. The following example shows a typical connect descriptor:

```
sales=
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=sales.us.acme.com)) )
```

The ADDRESS section contains the listener protocol address, and the CONNECT_DATA section contains the destination service information. In this example, the destination service is a database service named `sales.us.acme.com`.

When creating a connect descriptor for an Oracle9*i* or Oracle8*i* database service, you must identify the service with the `SERVICE_NAME` parameter. Optionally, you can identify an instance with the `INSTANCE_NAME` parameter, as shown in the following:

```
sales=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=sales.us.acme.com)
      (INSTANCE_NAME=sales)))
```

The values for these parameters come from the `SERVICE_NAMES` (with an S) and `INSTANCE_NAME` parameters in the [initialization parameter file](#). The `SERVICE_NAMES` parameter in the initialization parameter file is typically the [global database name](#), a name comprised of the database name and domain name, entered during installation or database creation. For example, `sales.us.acme.com` has a database name of `sales` and a domain of `us.acme.com`. The `INSTANCE_NAME` parameter in the initialization parameter file is defaulted to the SID entered during installation or database creation.

See Also: "Database Service and Database Instance Identification" on page 2-2

When creating a connect a descriptor for an Oracle release 8.0 or version 7 database, you identify the service with the `SID` parameter. The following example shows a connect descriptor for an Oracle release 8.0 database with a `SID` of `sales`:

```
sales=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
    (CONNECT_DATA=
      (SID=sales)))
```

Naming Methods

[Table 9–1](#) describes the naming methods that Oracle Net supports.

Table 9–1 Naming Methods

Naming Method	Description
Local Naming	<p>Resolves net service names stored in a <code>tnsnames.ora</code> file stored on a client</p> <p>Local naming is most appropriate for simple distributed networks with a small number of services that change infrequently.</p> <p>See Also: "Configuring the Local Naming Method" on page 9-5</p>
Directory Naming	<p>Resolves services and net service names stored in a centralized LDAP-compliant directory server, for example, Oracle Internet Directory</p> <p>See Also: "Configuring the Directory Naming Method" on page 9-12</p>
Oracle Names	<p>Resolves services and net service names stored in an Oracle Names server</p> <p>See Also: "Configuring the Oracle Names Method" on page 9-19</p> <p>Note: In future releases, Oracle Names will not be supported as a centralized naming method.</p>
Host Naming	<p>Enables users to connect to an Oracle server by using a host name alias. Host names are mapped to a connect descriptor in an existing names resolution service, such as Domain Name System (DNS), Network Information Service (NIS), or a centrally-maintained set of <code>/etc/hosts</code> files.</p> <p>No client configuration is required to take advantage of this feature. This method is recommended for simple TCP/IP environments.</p> <p>See Also: "Configuring the Host Naming Method" on page 9-56</p>
External Naming	<p>Resolves service information stored in an a third-party naming service, for example, NIS</p> <p>See Also: "Configuring External Naming Methods" on page 9-60</p>

See Also: ["Naming"](#) on page 2-15

Configuring the Local Naming Method

The local naming method adds net service names to the `tnsnames.ora` file. Each net service name is mapped to a connect descriptor. The example that follows shows a `tnsnames.ora` file's basic syntax for a net service name mapped to a connect descriptor:

```
sales=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com) ))
```

In this example, the net service name `sales` is mapped to the connect descriptor contained in `DESCRIPTION`. `DESCRIPTION` contains the protocol address and identifies the destination database service.

You can configure local naming during or after installation, as described in these topics:

- [Configuring the tnsnames.ora File During Installation](#)
- [Configuring the tnsnames.ora File After Installation](#)

Configuring the tnsnames.ora File During Installation

Oracle Net Configuration Assistant enables you to configure net service names for clients. Oracle Universal Installer launches Oracle Net Configuration Assistant after software installation. The configuration varies depending on the installation mode.

Typical or Minimal Installation

Oracle Net Configuration Assistant prompts you to configure net service names in the `tnsnames.ora` file to connect to an Oracle database service.

Custom Installation

Oracle Net Configuration Assistant prompts you to select naming methods to use. If Local is selected, then Oracle Net Configuration Assistant prompts you to configure net service names in a `tnsnames.ora` file to connect to an Oracle database service.

Configuring the tnsnames.ora File After Installation

You can add net service names to the `tnsnames.ora` file at any time. To configure the local naming method, perform the following tasks:

[Task 1: Configure Net Service Names](#)

[Task 2: Configure TNSNAMES as the First Naming Method](#)

[Task 3: Distribute Configuration](#)

[Task 4: Configure the Listener](#)

[Task 5: Connect to the Database](#)

Note: The underlying network connection must be operational before attempting to configure connectivity with Oracle Net.

Task 1: Configure Net Service Names

To configure with the local naming method, use either of the following tools:

- [Oracle Net Manager](#)
- [Oracle Net Configuration Assistant](#)

Oracle Net Manager To configure net service names in the `tnsnames.ora` file with **Oracle Net Manager**:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

2. In the navigator pane, expand Local > Service Naming.
3. Choose plus (+) from the toolbar, or choose Edit > Create from the menu bar.
The Welcome page of the Net Service Name Wizard appears.
4. Enter any name in the Net Service Name field.

You can qualify the net service name with the client's domain. The net service name is automatically domain qualified if the `sqlnet.ora` file parameter `NAMES.DEFAULT_DOMAIN` is set.

See Also: ["Configuring a Default Domain for Clients"](#) on page 11-3

5. Choose Next.

The Protocol page appears.

6. Select the protocol on which the listener is configured to listen. Note that this protocol must also be installed on the client.
7. Choose Next.

The Protocol Settings page appears.

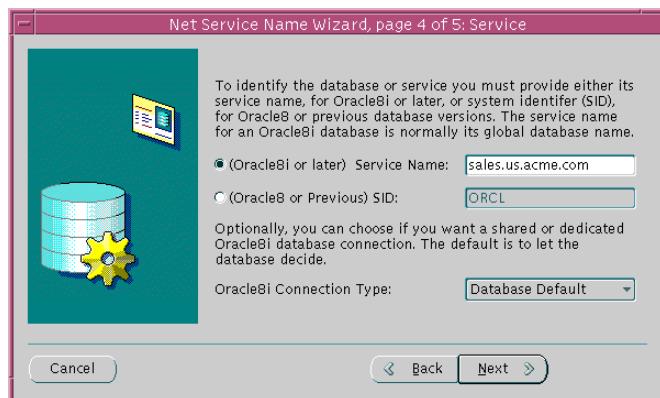
8. Enter the appropriate parameter information for the selected protocol in the fields provided.

See Also: [Oracle Net Services Reference Guide](#) for protocol parameter settings

9. Choose Next.

The Service page appears.

10. Select a release, enter a destination service, and optionally, select a database connection type.



If the destination service is *Oracle9i* or *Oracle8i* database, then choose (*Oracle8i* or later), and enter a service name in the Service Name field. If destination service is an Oracle release 8.0 database, then choose (*Oracle8* or Previous), and enter an Oracle System Identifier for an instance in the Database SID field.

See Also: ["About Connect Descriptors"](#) on page 9-2 for further information about the service name string to use

Oracle Corporation recommends that you use the default setting of Database Default for the connection type. If **shared server** is configured in the initialization parameter file, you can select Dedicated Server to force the listener to spawn a dedicated server, bypassing shared server configuration. If shared server is configured in the initialization parameter file and you want to guarantee the connection always uses shared server, select Shared Server.

See Also: [Chapter 14, "Configuring Shared Server"](#) for further information about shared server configuration

11. Choose Next.

The Test page appears.

12. Choose Test to verify that the net service name works, or choose Finish to save your configuration and dismiss the Net Service Name Wizard.

If you choose Test, Oracle Net connects to the database server by using the connect descriptor information you configured. Therefore, the database and the listener must be running for a successful test. If they are not, see ["Starting Oracle Net Services Components"](#) on page 16-2 to start components before

testing. During testing, a Connection Test dialog box appears, providing status and test results. A successful test results in the following message:

The connection test was successful.

If the test was successful, choose Close to dismiss the Connect Test dialog box, and proceed to Step 13.

If the test was not successful:

- a. Ensure that the database and listener are running, and then choose Test.
- b. Choose Change Login to change the user name and password for the connection, and then choose Test.

13. Choose Finish to save your configuration and dismiss Net Service Name Wizard.

See Also:

- ["Creating a List of Listener Protocol Addresses" on page 15-2](#) to configure multiple protocol addresses
- ["Configuring Advanced Connect Data Parameters" on page 15-7](#) to configure additional CONNECT_DATA options

Oracle Net Configuration Assistant To configure net service names in the `tnsnames.ora` file with Oracle Net Configuration Assistant:

1. Start Oracle Net Configuration Assistant.

See Also: ["Oracle Net Configuration Assistant"](#) on page 5-14

The Welcome page appears.

2. Select "Local Net Service Name Configuration", and then choose Next.

The Net Service Name Configuration page appears.

3. Choose Add, and then choose Next.

The Net Service Name Configuration, Database Version page appears.

4. If the destination service is an Oracle9*i* or Oracle8*i* database, then choose "Oracle8*i* or later database or service." If destination service is an Oracle8 release 8.0 database, then choose "Oracle8 release 8.0 database or service."

5. Choose Next.

6. Follow the prompts in the wizard and online help to complete net service name creation.

Task 2: Configure TNSNAMES as the First Naming Method

Configure local naming as the first method specified in the `NAMES.DIRECTORY_PATH` parameter in the `sqlnet.ora` file. This parameter specifies the order of naming methods Oracle Net uses to resolve connect identifiers to connect descriptors.

To specify local naming as the first naming method:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select Naming.
4. Choose the Methods tab.
5. From the Available Methods list, select TNSNAMES, and then choose the right-arrow button.

6. In the Selected Methods list, select TNSNAMES, and then use the Promote button to move the selection to the top of the list.
7. Choose File > Save Network Configuration.

The `sqlnet.ora` file updates with the `NAMES.DIRECTORY_PATH` parameter, listing `tnsnames` first:

```
NAMES.DIRECTORY_PATH=(tnsnames, onames, hostname)
```

Task 3: Distribute Configuration

After one client is configured, it is best simply to copy the `tnsnames.ora` and `sqlnet.ora` configuration files to the same location on the other clients. This ensures that the files are consistent. Alternatively, you can use Oracle Net Manager or Oracle Net Configuration Assistant on every client.

Task 4: Configure the Listener

Ensure that the listener (located on the server) is configured to "listen on" the same protocol address you configured for the net service name. By default, the listener should already be configured for the TCP/IP protocol on port 1521.

See Also: [Chapter 12, "Configuring and Administering the Listener"](#) for listener configuration details

Task 5: Connect to the Database

Clients can connect to the database using the following syntax:

```
CONNECT username/password@net_service_name
```

Configuring the Directory Naming Method

With the directory naming method, connect identifiers are mapped to connect descriptors contained in an LDAP-compliant directory server, including Oracle Internet Directory, Microsoft Active Directory, or Novell Directory Services. A directory provides central administration of database services and net service names, making it easier to add or relocate services.

A database service entry is created with [Oracle Database Configuration Assistant](#) during installation; net service name entries can be created with [Oracle Net Manager](#). To modify Oracle Net attributes of a database service entry and the net service name entries., use Oracle Net Manager.

Clients configured to access the directory can use these entries to connect to the database.

This section contains these topics:

- [Directory Naming Method Configuration Steps](#)
- [Modifying Connectivity Information for Database Service Entries](#)

Directory Naming Method Configuration Steps

To configure the directory naming method, perform the following tasks:

[Task 1: Configure Directory Usage on Server and Clients](#)

[Task 2: Create Net Service Name Entries \(Optional\)](#)

[Task 3: Configure LDAP as the First Naming Method](#)

[Task 4: Configure the Listener](#)

[Task 5: Connect to the Database](#)

Task 1: Configure Directory Usage on Server and Clients

Before a database service or net service name can be added to a directory, you must complete directory access configuration. You can configure directory server access during or after installation.

See Also: [Chapter 8, "Setting Up Directory Server Usage"](#) for further information about configuration directory server access

Task 2: Create Net Service Name Entries (Optional)

You can configure clients to use a net service name rather than the database service entry created by Oracle Database Configuration Assistant. Net service names are created under the Oracle Context (`cn=OracleContext`).

Note: Only users that are members of the OracleNetAdmins group can create net service entries in a directory. The user that created the Oracle Context during directory access configuration on the server is automatically a member of this group. To add or remove users from this group, see ["Adding Users to and Removing Users from the OracleNetAdmins Group"](#) on page 8-9.

See Also:

- ["Exporting Net Service Names from a tnsnames.ora File"](#) on page 10-2
- ["Exporting Network Objects from an Oracle Names Server"](#) on page 10-6

To create a net service name in a directory server:

1. Start Oracle Net Manager on a computer that has been configured with directory access for a server:

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

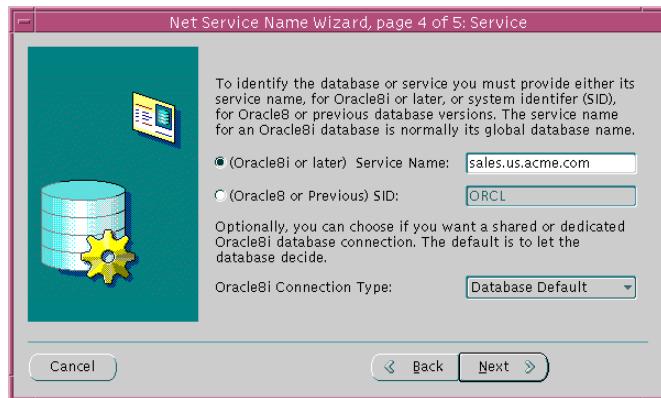
2. In the navigator pane, choose Directory > Service Naming.
3. Choose plus (+) from the toolbar, or choose Edit > Create from the menu bar.
The Welcome page of the Net Service Name Wizard appears.
4. Enter any name in the Net Service Name field.
5. Choose Next.
The Protocol page appears.
6. Select the protocol on which the listener is configured to listen. Note that this protocol must also be installed on the client.
7. Enter the appropriate parameter information for the selected protocol in the fields provided.

See Also: *Oracle Net Services Reference Guide* for protocol parameter settings

8. Choose Next.

The Service page appears.

9. Select a release, enter a destination service, and optionally, select a database connection type.



If the destination service is Oracle9*i* or Oracle8*i* database, then choose (Oracle8*i* or later), and enter a service name in the Service Name field. If destination service is an Oracle release 8.0 database, then choose (Oracle8 or Previous), and enter an Oracle System Identifier for an instance in the Database SID field.

See Also: ["About Connect Descriptors"](#) on page 9-2 for further information about the service name string to use

Oracle Corporation recommends that you use the default setting of Database Default for the connection type. If **shared server** is configured in the initialization parameter file, you can select Dedicated Server to force the listener to spawn a dedicated server, bypassing shared server configuration. If shared server is configured in the initialization parameter file and you want to guarantee the connection always uses shared server, select Shared Server.

See Also: [Chapter 14, "Configuring Shared Server"](#) for further information about shared server configuration

10. Choose Next.

The Test page appears.

11. Choose Test to verify that the net service name works, or choose Finish to save your configuration and dismiss Net Service Name Wizard.

If you choose Test, Oracle Net connects to the database server by using the connect descriptor information you configured. Therefore, the database and the listener must be running for a successful test. If they are not, see "[Starting Oracle Net Services Components](#)" on page 16-2 to start components before testing. During testing, a Connection Test dialog box appears, providing status and test results. A successful test results in the following message:

The connection test was successful.

If the test was successful, choose Close to dismiss the Connect Test dialog box, and proceed to Step 13.

If the test was not successful:

- a. Ensure that the database and listener are running, and then choose Test.
- b. Choose Change Login to change the user name and password for the connection, and then choose Test.

12. Choose Finish to save your configuration and dismiss Net Service Name Wizard.

See Also:

- ["Creating a List of Listener Protocol Addresses"](#) on page 15-2 to configure multiple protocol addresses
- ["Configuring Advanced Connect Data Parameters"](#) on page 15-7 to configure additional connect data options

Task 3: Configure LDAP as the First Naming Method

Configure directory naming as the first method specified in the NAMES.DIRECTORY_PATH parameter in the `sqlnet.ora` file. This parameter specifies the order of naming methods Oracle Net uses to resolve connect identifiers to connect descriptors.

To specify directory naming as the first naming method:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select Naming.
4. Choose the Methods tab.

5. From the Available Methods list, select LDAP, and then choose the right-arrow button.
6. In the Selected Methods list, select LDAP, and then use the Promote button to move the selection to the top of the list.
7. Choose File > Save Network Configuration.

The `sqlnet.ora` file updates with the `NAMES.DIRECTORY_PATH` parameter, listing `ldap` first:

```
NAMES.DIRECTORY_PATH=(ldap, tnsnames, onames, hostname)
```

Task 4: Configure the Listener

Ensure that the listener (located on the server) is configured to listen on the same protocol address configured for the net service name. By default, the listener is configured to listen on the TCP/IP protocol, port 1521.

See Also: [Chapter 12, "Configuring and Administering the Listener"](#) for listener configuration details

Task 5: Connect to the Database

Clients that are configured with a default directory entry that matches the directory location of the database service or net service name can connect to the database using the following syntax:

```
CONNECT username/password@connect_identifier
```

Clients that are configured with a default directory entry that does not match the entry's directory location cannot use the connect identifier in the connect string. Instead, these connections require the entry's distinguished name or its absolute name.

See Also:

- ["Connect Identifier and Connect Descriptor Syntax Characteristics"](#) on page 16-9 for connect identifier syntax rules
- ["Absolute Name Specification for Directory Naming"](#) on page 16-9 for absolute name usage

Modifying Connectivity Information for Database Service Entries

Note: Only users that are members of the OracleNetAdmins group can create net service entries in a directory. The user that created the Oracle Context during directory access configuration on the server is automatically a member of this group. To add or remove users from this group, see "[Adding Users to and Removing Users from the OracleNetAdmins Group](#)" on page 8-9.

A database service entry is created by Oracle Database Configuration Assistant during database creation after directory access has been configured on the server. The entry is contained under an Oracle Context (`cn=OracleContext`).

A database service entry stored in a directory may not contain any network route information. This information can be deleted by a directory administrator. A network route that includes the location of the listener through a protocol address must be included in the connect descriptor.

See Also:

- ["Exporting Net Service Names from a tnsnames.ora File"](#) on page 10-2
- ["Exporting Network Objects from an Oracle Names Server"](#) on page 10-6

To create or modify network route information for a database service:

1. Start Oracle Net Manager on a computer that has been configured with directory access for a server:

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

2. In the navigator pane, expand Directory > Service Naming.
3. Select the database service. The right pane displays the current destination service name.
4. In the Address Configuration box, choose plus (+).
A new Address tab appears.
5. Select a protocol, and then enter the appropriate parameter information for the selected protocol in the fields provided.

See Also: *Oracle Net Services Reference Guide* for protocol parameter settings

6. In the right pane, choose Apply.

Configuring the Oracle Names Method

Note: In future releases, Oracle Names will not be supported as a centralized naming method. Because no new enhancements are being added to Oracle Names, consider using directory naming or migrating an existing Oracle Names configuration to directory naming, as described in [Chapter 10, "Exporting Naming Data to a Directory Server"](#). The material presented here is primarily for reference to enable you to maintain your current Oracle Names environment.

See Also: *Oracle9i Database Migration* if you have an existing release 8.0 or release 7.x configuration

Oracle Names simplifies the setup and administration of global, client/server computing networks. Oracle Names makes network address and database link information available to all nodes throughout the network. Each database server network address is identified with a simple service name. Client applications then can request a database connection with that name rather than a lengthy address. Oracle Names shields users and applications from changes to the network infrastructure. It provides for centralized administration of network service names.

Configuring Oracle Names involves these tasks:

[Task 1: Consider Oracle Names Options](#)

[Task 2: Install Necessary Components](#)

[Task 3: Create an Oracle Names Server](#)

[Task 4: Configure Clients and Database Servers To Use Oracle Names Servers](#)

[Task 5: \(Optional\) Configure Client Caches](#)

[Task 6: Configure the Listener](#)

[Task 7: Register Data with the Oracle Names Server](#)

[Task 8: Delegate Domains to Delegated Administrative Regions](#)

[Task 9: Specify Domain Hints to Forward Requests to Remote Oracle Names Servers](#)

[Task 10: Connect to the Database](#)

Task 1: Consider Oracle Names Options

Prior to creating an Oracle Names server, you must decide whether:

- You need service registration data replicated continuously among Oracle Names servers, or want all Oracle Names servers within a region to store their registration data in an Oracle database.
- You need support for one or more **administrative region**. An administrative region is a collection of Oracle Names servers in one or more **domains**, a grouping of network objects, such as databases. Networks with multiple administrative regions must have one **root administrative region** and one or more **delegated administrative regions**. Delegated administrative regions contain the domains and Oracle Names server addresses in any alternate regions which act as direct child regions of the root.

See Also: "Oracle Names Support" on page 3-18

Task 2: Install Necessary Components

Ensure that the following are installed:

- Oracle Names on its own computer that is designated as the Oracle Names server
- Oracle Net Services on the clients
- Oracle Net Services on the database server

Important: The Oracle Universal Installer does not install the \$ORACLE_HOME/network/names directory on UNIX operating systems. This directory is necessary for successful configuration. If it does not exist, you must manually create it.

Task 3: Create an Oracle Names Server

How an Oracle Names server is created depends upon how you want region data stored. You can have the data:

- Stored in tables, called a **region database**, in an Oracle database accessible to Oracle Names servers

When information is added to an Oracle Names server, the information is stored in the database and in the Oracle Names server checkpoint files. If the database is unavailable, the information in these files is used.

- Saved to checkpoint files and replicated among Oracle Names servers

When information is added to an Oracle Names server, the information is stored in checkpoint files, and then replicated to the other Oracle Names servers.

By default, the checkpoint files are stored in `$ORACLE_HOME/network/names` on UNIX operating systems, and `ORACLE_HOME\network\names` on Windows NT. **Table 9–2** describes the checkpoint files.

Table 9–2 Oracle Names Checkpoint Files

Checkpoint File Name	Description
<code>ckpcfg.ora</code>	Contains a backup copy of the configuration parameters stored in the <code>ONRS_CONFIG</code> table in the region database
<code>ckpcch.ora</code>	Contains all current non-authoritative data that has been retrieved and cached from remote regions and has not expired yet
<code>ckptop.ora</code> and <code>ckpdom.ora</code>	These files contain all authoritative data for the region. The <code>ckptop.ora</code> file defines the domains in the administrative region and the Oracle Names servers authoritative for each domain. The <code>ckpdom.ora</code> file contains the authoritative data for each domain.
	If the Oracle Names server uses a region database, then these files serve as a copy of the region data in the tables as of the last reload. This data is used when the Oracle Names server starts if the database is inaccessible.
	If the Oracle Names server is not using the database, these files are its only persistent storage and are loaded by the Oracle Names server at startup. The Oracle Names server considers the data in these file to be current if there are no other Oracle Names servers in the region. The files are kept current as of the last update to the region.

The following sections cover both modes, as well as a default Oracle Names server that requires no configuration:

- [Default Oracle Names Server](#)
- [Create Tables in a Database](#)
- [Checkpoint Files/Replicated Updates](#)

Whichever method you choose, Oracle Corporation recommends that you create more than one Oracle Names server for the network, in case one should go down.

Default Oracle Names Server

An Oracle Names server can run without any configuration. Its name defaults to `ONAMES_host` if its name is configured in the `names.ora` file. The protocol address defaults to TCP/IP, port 1575 on the local host, or other well-known addresses for protocols.

See Also: Your operating system-specific documentation for default address information

If the `NAMES.DOMAINS` parameter is not configured in the `names.ora` file, the Oracle Names server assumes authority for the root domain.

If you would like to use this Oracle Names server, proceed to "[Task 2: Discover Oracle Names Servers](#)" on page 9-33.

Create Tables in a Database

To store service data in an Oracle database, perform the following tasks:

- [Task 1: Configure the Database Server](#)
- [Task 2: Create Oracle Names Servers](#)
- [Task 3: Start the Oracle Names Server](#)

Task 1: Configure the Database Server To configure the database server:

1. Start the database if it is not currently running; otherwise, go to Step 2.

See Also: "[Task 4: Start the Database](#)" on page 16-6

2. Connect to the database as the SYSTEM user:

```
SQL> CONNECT system/password
```

where *password* is MANAGER for the SYSTEM user account by default.

3. Create a user that can create tables.

- a. Create user. In the following example syntax, the default tablespace is users and the temporary tablespace is temp:

```
CREATE USER user
IDENTIFIED BY password
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp;
```

- b. Grant the user the ability to connect to the database:

```
GRANT CREATE SESSION TO user;
```

- c. Grant the user the ability to update tables in the users tablespace:

```
GRANT RESOURCE users TO user;
```

- d. Grant the ability to create synonyms in the user's schema:

```
GRANT CREATE SYNONYM TO user;
```

See Also: *Oracle9i Database Administrator's Guide* for further information about creating users

4. Run the namesini.sql script located in \$ORACLE_HOME/network/admin on UNIX operating systems and ORACLE_HOME\Network\Admin\Names on Windows NT. This script creates the tables needed by Oracle Names to store information. Optionally, run namesupg.sql to upgrade old tables.

```
SQL> CONNECT user/password
SQL> @oracle_home/network/admin/namesini.sql;
```

Task 2: Create Oracle Names Servers For each computer where Oracle Names is installed and where you want an Oracle Names server, create an Oracle Names server from Oracle Net Manager.

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, choose Oracle Names Servers.
3. Choose Edit > Create, or choose the plus (+) button to create a new Oracle Names server.

The Names Wizard starts.

The wizard guides you through the creation and configuration process, prompting for:

- A unique Oracle Name Server name
- A protocol address for the Oracle Names server
 - If you choose TCP/IP, Oracle Corporation recommends using the default and officially registered port of 1575 for the Oracle Names server.
- A choice to store information in a database or replicate information among Oracle Names servers
 - (Choose "Use a region database. ")
- A protocol address for a database's listener
 - If you choose TCP/IP, Oracle Corporation recommends using the default and officially registered port of 1521 for the listener.
- Database user ID, password, and service name or SID
- Whether or not this Oracle Names server is in the root administrative region

If you specify that this Oracle Names server is in the root administrative region, configuration completes.

If you specify that this Oracle Names server is not in the root administrative region, it assumes that this Oracle Names server is in a delegated administrative region. The wizard then prompts you for the local administrative region's domain name and the address of an Oracle Names server in the root administrative region before completing.

When the wizard completes, the following message appears:

A Names Server, onames_server, has been created with default settings. Use the Configure Server section to modify the default configuration.

4. Choose File > Save Network Configuration.
5. Repeat Steps 2 through 4 to create additional Oracle Names servers in a region. Oracle Net Manager does not support creation of multiple Oracle Names servers on one computer.

Oracle Net Manager creates a names.ora file with the following settings:

- Oracle Names server name (NAMES.SERVER_NAME)
- Oracle Names server listening protocol address (NAMES.ADDRESSES)
- Database information (NAMES.ADMIN_REGION)
- Root administrative region information (NAMES.DOMAIN_HINTS)
- Domains in this administrative region (NAMES.DOMAINS)

The names.ora file is created in the \$ORACLE_HOME/network/admin directory on UNIX, and the *ORACLE_HOME*\network\admin directory on Windows NT.

A names.ora file follows with annotations of content:

```
#Oracle Names server name. The name should include the name of the domain
#this Oracle Names server is in.
NAMES.SERVER_NAME=namesvr2.us.acme.com

#Oracle Names server listening protocol address
NAMES.ADDRESSES=
  [ (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp)(HOST=namesrv2-server)(PORT=1575))
    [ (ADDRESS=...)][])

#Database repository information
NAMES.ADMIN_REGION=
  (REGION=
    (DESCRIPTION=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
      (CONNECT_DATA=
        (SERVICE_NAME=sales.us.acme.com))
      (USERID=system)
      (PASSWORD=password)
      (NAME=local_region)
      (REFRESH=86400)
      (RETRY=60)
      (EXPIRE=600))

#If an Oracle Names server is in a delegated administrative region, identify
#the address of an Oracle Names server in the root administrative region.
NAMES.DOMAIN_HINTS=
  (HINT_DESC=
    (HINT_LIST=
      (HINT=
        (NAME=namesvr1)
        (ADDRESS=(PROTOCOL=tcp)(HOST=namesvr1-server)(PORT=1575)))

# Specify the domain controlled by this region and the time to live (TTL).
# If this is the root administrative region, specify NAME=(null) or NAME=. to
# identify the root domain.
NAMES.DOMAINS=
  (DOMAIN=
    (NAME=us.acme.com)
    (MIN_TTL=86400))
```

See Also: *Oracle Net Services Reference Guide* for names.ora file parameters

Administering Multiple Domains

If you want the region to administer more than one domain, specify the additional domains in the NAMES.DOMAINS parameter with Oracle Net Manager:

1. In the navigator pane, expand Oracle Names Servers.
2. Select the Oracle Names server.
3. From the list in the right pane, select Configure Server.
4. Choose the Domains tab.
5. Enter the domain name in the Domain Name field and time-to-live information, then choose Add.
6. Repeat Step 5 for each additional domain.
7. Choose File > Save Network Configuration.

In the following example, NAMES.DOMAINS contains a listing for the root (value of null), com, acme.com, and hq.acme.com domains. All the domain precedent to hq.acme.com must be defined in order to define hq.acme.com.

```
NAMES.DOMAINS=
(DOMAIN_LIST=
(DOMAIN=
(NAME=)
(MIN_TTL=86400))
(DOMAIN=
(NAME=com)
(MIN_TTL=86400))
(DOMAIN=
(NAME=acme.com)
(MIN_TTL=86400))
(DOMAIN=
(NAME=hq.acme.com)
(MIN_TTL=86400)))
```

Task 3: Start the Oracle Names Server To start the Oracle Names server, use either Oracle Net Manager or Oracle Names Control utility:

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none">1. In the navigator pane, expand Oracle Names Servers.2. Select the Oracle Names server.3. From the list in the right pane, select Manage Server.4. Choose the Control tab.5. Choose Start.6. Choose Apply. <p>The following message appears: Server started successfully. On Windows NT, a service called OracleORACLE_HOMENames onames_ server is created.</p>	<p>From the operating system command line, enter: <code>namesctl</code> If the following error messages appear, ignore them. <code>NNL-00024: warning: no preferred names servers in SQLNET.ORA</code> <code>NNL-00018: warning: could not contact default name server</code></p> <p><code>NAMESCTL> START</code> The START command loads the Oracle Names server into memory and tells it to begin executing. At startup, the Oracle Names server loads its configuration and data.</p> <p>On Windows NT, a service called OracleORACLE_HOMENames onames_ server is created</p>

Checkpoint Files/Replicated Updates

To replicate data among Oracle Names server in checkpoint files, perform these tasks:

- [Task 1: Create an Oracle Names Server](#)
- [Task 2: Discover Oracle Names Servers](#)

Task 1: Create an Oracle Names Server For each computer where Oracle Names is installed and where you want an Oracle Names server, create an Oracle Names server from Oracle Net Manager:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Oracle Names Servers.
3. Choose Edit > Create or choose plus (+) to create a new Oracle Names server.

The Names Wizard starts.

The wizard guides you through the creation and configuration process, prompting you for:

- A unique Oracle Name Server name
- A listening protocol address for the Oracle Names server
If you choose TCP/IP, Oracle Corporation recommends using the default and officially registered port of 1575 for the Oracle Names server.
- A choice to store information in a database or replicate information among Oracle Names servers
(Choose "Don't use a region database.")
- Verification that this is the first Oracle Names server in the region
If this is not the first Oracle Names server in the region, the wizard then prompts you to discover the other Oracle Names servers or to specify the address of another Oracle Names server in the region.
- Verification that this Oracle Names server is in the root administrative region

If you specify that this Oracle Names server is in the root administrative region, configuration completes.

If you specify that this Oracle Names server is not in the root administrative region, it assumes that this Oracle Names server is in a delegated administrative region. The wizard then prompts you for the local administrative region's domain name and the address of an Oracle Names server in the root administrative region before completing.

When the wizard completes, the following message appears:

A Names Server, onames_server, has been created with default settings. Use the Configure Server section to modify the default configuration.

4. Select Save Network Configuration from the File menu.
5. Repeat Steps 2 through 4 to create additional Oracle Names servers in a region. Oracle Net Manager does not support creation of multiple Oracle Names servers on one computer.

Oracle Net Manager creates a `names.ora` file with the following settings:

- Oracle Names server name (`NAMES.SERVER_NAME`)
- Oracle Names server listening protocol address (`NAMES.ADDRESSES`)
- Root administrative region information (`NAMES.DOMAIN_HINTS`)
- Domains in this administrative region (`NAMES.DOMAINS`)

The `names.ora` file is created in the `$ORACLE_HOME/network/admin` directory on UNIX, and the `ORACLE_HOME\Network\Admin` directory on Windows NT.

A names.ora file follows with annotations of content:

```
#Oracle Names server name. The name should include the name of the domain
this Oracle Names server is in.
NAMES.SERVER_NAME=namesrv2.us.acme.com

NAMES.ADDRESSES=
[ (ADDRESS_LIST=
  (ADDRESS=(PROTOCOL=tcp)(HOST=namesrv2-server)(PORT=1575))
  [(ADDRESS=...)])
[]]

#If an Oracle Names server is in a delegated administrative region, identify
the address of an Oracle Names server in the root administrative region.
NAMES.DOMAIN_HINTS=
(HINT_DESC=
(HINT_LIST=
(HINT=
(NAME=namesrv1)
(ADDRESS=(PROTOCOL=tcp)(HOST=namesrv1-svr)(PORT=1575))

# Specify the domain controlled by this region and the time to live (TTL).
# If this is the root administrative region, you must have NAME= (null) to
# identify the root domain properly.
NAMES.DOMAINS=
(DOMAIN=
(NAME=us.acme.com)
(MIN_TTL=86400))
```

See Also: *Oracle Net Services Reference Guide* for names.ora file parameters

Administering Multiple Domains

If you want the region to administer more than one domain, specify the additional domains in the NAMES.DOMAINS parameter with Oracle Net Manager:

1. In the navigator pane, expand Oracle Names Servers.
2. Select the Oracle Names server.
3. From the list in the right pane, select Configure Server.
4. Choose the Domains tab.
5. Enter the domain name in the Domain Name field and time-to-live information, and then choose Add.
6. Repeat Step 5 for each additional domain.
7. Choose File > Save Network Configuration.

In the following example, NAMES.DOMAINS contains a listing for the root (value of null), com, acme.com, and hq.acme.com domains. All the domain precedent to hq.acme.com must be defined in order to define hq.acme.com.

```
NAMES.DOMAINS=
(DOMAIN_LIST=
(DOMAIN=
(NAME=)
(MIN_TTL=86400))
(DOMAIN=
(NAME=com)
(MIN_TTL=86400))
(DOMAIN=
(NAME=acme.com)
(MIN_TTL=86400))
(DOMAIN=
(NAME=hq.acme.com)
(MIN_TTL=86400)))
```

Task 2: Discover Oracle Names Servers After all the Oracle Names servers are created, have each Oracle Names server, except the first one in the region, discover the other Oracle Names servers in a region.

To discover other Oracle Names servers, use either Oracle Net Manager or Oracle Names Control utility:

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none"> 1. Start the first Oracle Names server in the region: <ol style="list-style-type: none"> a. In the navigator pane, expand Oracle Names Servers. b. Select the Oracle Names server. c. From the list in the right pane, select Manage Server. d. Choose the Control tab. e. Choose Server Operations, and then choose Start. f. Choose Apply to start the Oracle Names server. The following message appears: Server started successfully. 2. From the second Oracle Names server, discover the first Oracle Names server. Choose Command > Discover Oracle Names servers. The following message appears: Discovered Oracle Names Server in the region. Please exit the tool and start again. The Discover command produces an .sdns.ora file in \$ORACLE_HOME/network/names on UNIX operating systems or a sdns.ora file in ORACLE_HOME\Network\Names on Windows. This file contains the names and addresses of the Oracle Names servers. If an Oracle Names server does not respond, a dialog prompts you for another Oracle Names server address. If you know the network address of a particular Oracle Names server, enter it. 3. Start the second Oracle Names server, following Step 1. 4. For each Oracle Names server added to the region, repeat Steps 2-3. 	<ol style="list-style-type: none"> 1. Start the first Oracle Names server in the region: namesctl NAMESTCL> START 2. From the second Oracle Names server, discover the first Oracle Names server. namesctl NAMESTCL> REORDER_NS The REORDER_NS command produces an .sdns.ora file in \$ORACLE_HOME/network/names on UNIX operating systems or a sdns.ora file in ORACLE_HOME\Network\Names on Windows. This file contains the name and address of the first Oracle Names server. If an Oracle Names server cannot be found, and you know the network address of a particular Oracle Names server, enter the address at the prompt. For example: NAMESTCL> REORDER_NS (ADDRESS=(PROTOCOL=tcp)(HOST=mail-server)(PORT=1575)) 3. Start the second Oracle Names server in the region: NAMESTCL> START 4. Connect to the first Oracle Names server in the region, and query for the second Oracle Names server to make sure it has registered itself. NAMESTCL> QUERY onames_server 5. For each Oracle Names server added to the region, repeat Steps 2-4.

See Also: ["About Discovery"](#) on page 3-33

Task 4: Configure Clients and Database Servers To Use Oracle Names Servers

To configure client and database server computers to use Oracle Names servers:

1. Create a list of Oracle Names servers to contact, using one of the following methods:

- Create a static list of preferred Oracle Names server in the `sqlnet.ora` file, as described in "[Configuring Preferred Oracle Names Servers](#)" on page 11-11.
- Discover Oracle Names servers by dynamically creating an `.sdns.ora` file in `$ORACLE_HOME\network\names` on UNIX operating systems, and `sdns.ora` file in `ORACLE_HOME\network\names` on Windows operating systems. This file contains the order of names and addresses of the Oracle Names server(s) to contact. Client computers read this file to find the addresses of Oracle Names servers to contact.

Preferred Oracle Names servers take precedence over the Oracle Names servers configured in the `.sdns.ora` or `sdns.ora` file.

See Also: ["About Discovery"](#) on page 3-33 for a description of discovery

To create an `.sdns.ora` file or an `sdns.ora` file, use either Oracle Net Manager or Oracle Names Control utility:

Use Oracle Net Manager...	Use Oracle Names Control utility...
<p>1. Start Oracle Net Manager.</p> <p>See Also: "Starting Oracle Net Manager" on page 5-3</p> <p>2. Choose Command > Discover Oracle Names servers.</p> <p>The following message appears:</p> <p>Discovered Oracle Names Server in the region. Please exit the tool and start again.</p> <p>The Discover command produces an <code>.sdns.ora</code> file in <code>\$ORACLE_HOME/network/names</code> on UNIX operating systems or a <code>sdns.ora</code> file in <code>ORACLE_HOME\Network\Names</code> on Windows. This file contains the names and addresses of the Oracle Names server(s). This file is read to find the addresses of Oracle Names servers.</p> <p>If an Oracle Names server does not respond, a dialog prompts you for another Oracle Names server address. If you know the network address of a particular Oracle Names server, enter it.</p>	<p>Discover the other Oracle Names servers:</p> <pre>namesctl NAMESTL> REORDER_NS</pre> <p>The <code>REORDER_NS</code> command produces an <code>.sdns.ora</code> file in <code>\$ORACLE_HOME/network/names</code> on UNIX operating systems or a <code>sdns.ora</code> file in <code>ORACLE_HOME\Network\Names</code> on Windows. This file contains the names and addresses of the Oracle Names server(s). This file is read to find the addresses of Oracle Names servers.</p> <p>If an Oracle Names server cannot be found, and you know the network address of a particular Oracle Names server, enter the address at the prompt. For example:</p> <pre>NAMESTL> REORDER_NS (ADDRESS=(PROTOCOL=tcp) (HOST=mail-server) (PORT=1575))</pre>

2. If you want Oracle Names to be the first method to reconcile net service names, set the NAMES.DIRECTORY_PATH parameter in the `sqlnet.ora` file in the following manner:
 - a. In the navigator pane, expand Local > Profile.
 - b. From the list in the right pane, select Naming.
 - c. Choose the Methods tab.
 - d. In the Available Methods list, select ONAMES, and then choose the right-arrow.
 - e. In the Selected Methods list, select ONAMES, and then use the Promote button to move the selection to the top of the list.
 - f. Choose File > Save Network Configuration.

The `sqlnet.ora` file should contain an entry that lists `onames` first in the `NAMES.DIRECTORY_PATH` parameter:

```
NAMES.DIRECTORY_PATH=(onames, tnsnames, hostname)
```

3. You may configure additional features for clients that use Oracle Names. [Table 9–3](#) describes the client `sqlnet.ora` file settings for Oracle Names.

Table 9–3 Client Settings In `sqlnet.ora` for Oracle Names

Oracle Net Manager Field	<code>sqlnet.ora</code> Parameter	Description
Default Domain	<code>NAMES.DEFAULT_DOMAIN</code>	Indicates the domain name space from which the client most often requests an Oracle Names server. When set, this name is automatically appended to any unqualified name in an Oracle Names request. Oracle Net sets the default domain to NULL by default.
Maximum Wait Each Attempt	<code>NAMES.INITIAL_RETRY_TIMEOUT</code>	Specifies how long a client waits for a response from an Oracle Names server before reiterating the request to another Oracle Names server. Oracle Net waits for 15 seconds by default.
Attempts Per Names Server	<code>NAMES.REQUEST_RETRIES</code>	Specifies the number of times a client attempts to iterate through the list of Oracle Names servers before allowing the operation to fail. Oracle Net attempts to iterate through the list of Oracle Names servers once before allowing the operation to fail by default.
Maximum Open Connections	<code>NAMES.MAX_OPEN_CONNECTIONS</code>	Specifies how many connections an Oracle Names client may have open at one time. Oracle Net enables a client to have 10 connections open at any one time by default. This default value should be sufficient for almost all situations.
Initial Preallocated Requests	<code>NAMES.MESSAGE_POOL_START_SIZE</code>	Enables you to preallocate an initial number of messages in a client's message pool. These messages may be used for future requests to Oracle Names servers. Oracle Net allocates 10 messages in the pool by default. This default value should be sufficient for almost all situations.

To add or configure these features:

- a. In the navigator pane, expand Local > Profile.
- b. From the list in the right pane, select Naming.
- c. Choose the Oracle Names tab.
- d. Enter data for the options indicated in the table on the previous page.
- e. Choose File > Save Network Configuration.

Task 5: (Optional) Configure Client Caches

To avoid clients sending requests to Oracle Names server for name lookups, you can create a client cache process local to the client. A client cache:

- Optimizes response time for clients by putting their most frequently referenced name and address definitions closer to them
- Reduces the request load on existing Oracle Names servers

The most optimal environment for the client cache is a large multiuser computer where the users are running a particular Oracle client application. Clients on a single-user computer with wide-ranging interests in the databases will also derive some benefit from a cache.

A client cache involves running a local Oracle Names cache process on the client. The client cache has no authoritative data but uses the discovery list file (.sdns.ora or sdns.ora) to find Oracle Names server in the local region. Requests from clients are forwarded to the Oracle Names server, and the results are cached by the cache process so that subsequent requests for that object can be answered directly by the cache process.

In a sense, the client cache is a client version of the Oracle Names server. The client cache is distinguished from an Oracle Names server in the following ways:

- It is not authoritative for any administrative region
- It uses a protocol address that is local to the computer. By default this address is (ADDRESS=(PROTOCOL=ipc)(KEY=onames)).
- Its parameter are configured in the sqlnet.ora file rather than the names.ora file
- Different executable named onrsd that is smaller in size and functionality

Creating a Client Cache

To create and start a client cache:

1. Ensure discovery of Oracle Names servers has been performed, as described in ["Task 4: Configure Clients and Database Servers To Use Oracle Names Servers"](#) on page 9-34. The information gathered during discovery is used for the client's cache.
2. (Optional) Configure additional parameters in the `sqlnet.ora` file.

The client cache does not require anything beyond what is usually present in `sqlnet.ora` file or the discovery list file (`.sdns.ora` or `sdns.ora`). However, many of the parameters from `names.ora` file can also be set in `sqlnet.ora` for the client cache.

See Also: *Oracle Net Services Reference Guide* for `names.ora` file parameters for more information about `names.ora` file parameters

3. Create and start the client cache on the client.

```
namesctl  
NAMECTL> START_CLIENT_CACHE
```

On Windows NT, the client cache may also be started through the Control Panel:

- a. Double-choose the Services icon in the Control Panel window.
- b. Select the Oracle`HOME_NAME`ClientCache service.
- c. Choose Start to start the service.
- d. In the Services window, choose Close.

The `START_CLIENT_CACHE` command starts a client cache process. The client cache process finds a list of the local Oracle Names servers in static list of preferred Oracle Names server in the `sqlnet.ora` file or in the discovery list file (`.sdns.ora` or `sdns.ora`) the first time it runs. Afterwards, it relies on its cache checkpoint file if restarted. Once the client cache has contact with the local region server(s), it will begin to cache data on all local servers, various foreign servers, and all the data it has been asked to find.

Administering a Client Cache

Once configuration is complete, administration for a client cache is essentially the same as it is for an Oracle Names server, that is, you can use the Oracle Names Control utility commands to control and administer the client cache.

See Also: *Oracle Net Services Reference Guide* for a complete list of administrative commands

Starting and Stopping the Client Cache In terms of starting and stopping, the client cache behaves essentially the same as an Oracle Names server, except that it is usually accessible locally through an IPC protocol address.

Starting the client cache requires the START_CLIENT_CACHE command rather than the START command.

Stopping the client cache requires specifying the protocol address of the client cache with the SET SERVER command or the STOP command:

```
namesctl  
NAMESCTL> SET SERVER (ADDRESS=(PROTOCOL=ipc)(KEY=onames))  
NAMESCTL> STOP
```

```
namesctl  
NAMESCTL> STOP (ADDRESS=(PROTOCOL=ipc)(KEY=onames))
```

On Windows NT, stopping the Oracle`HOME_NAME`ClientCache service stops caching for all Oracle Names servers.

Verifying the Client Cache is Working To verify that the client cache is working, query an Oracle Names server from the client as follows:

```
namesctl  
NAMESCTL> SET SERVER onames_server
```

If the client cache process does not find an Oracle Names server, every query it receives will fail with the following error message:

NNC-00419: authoritative server is unreachable

The STATUS command shows how many requests the client cache has received, how many of those were forwarded, and how many items are cached. For example:

```
namesctl
NAMESTL> STATUS
Version banner is "Oracle Names Client Cache for Solaris: Version 9.0...
Server name:          ""
Server has been running for:      2 minutes 39.92 seconds
Request processing enabled:       yes
Request forwarding enabled:      yes
Requests received:             53
Requests forwarded:            2
Foreign data items cached:       4
...
...
```

Task 6: Configure the Listener

If you are not using your operating system's primary protocol, configure the listener with listening protocol addresses.

See Also: [Chapter 12, "Configuring and Administering the Listener"](#) for listener configuration details

Task 7: Register Data with the Oracle Names Server

As described in [Table 3–8](#) on page 3-23, an Oracle Names server stores many kinds of information, including database locations, Oracle Names server locations, net service names, aliases, and global database links. If you would like to store any of the following kinds of information, you must register it using either Oracle Net Manager or the Oracle Names Control utility:

- [Database Services](#)
- [Net Service Names](#)
- [Global Database Links and Database Qualifiers](#)
- [Aliases](#)

Registration needs to occur with only one Oracle Names server in an administrative region. The other Oracle Names servers see the information if data is stored in a region database, or the information is propagated to other Oracle Names if you chose not to use a region database.

After registering these kinds of information, you should query them, as described in ["Testing Network Objects with Oracle Net Manager or Oracle Names Control Utility"](#) on page 16-24.

Database Services

To register database service names and addresses with an Oracle Names server, use either Oracle Net Manager or Oracle Names Control utility:

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none"> 1. Start Oracle Net Manager. <p>See Also: "Starting Oracle Net Manager" on page 5-3</p> <ol style="list-style-type: none"> 2. In the navigator pane, expand Oracle Names Servers. 3. Select an Oracle Names server. 4. From the list in the right pane, select Manage Data. 5. Choose the Adv. tab. 6. Choose Add. 7. Enter the database service name of the database in the Name field. The service name is automatically domain qualified if the domain specified with the <code>sqlnet.ora</code> file parameter <code>NAMES.DEFAULT_DOMAIN</code> is set to database domain name. 8. Enter A . SMD in the Type field. A . SMD specifies that the object you are registering is a network address. 9. Enter the connect descriptor of the database in the Value field with the following syntax <code>(DESCRIPTION=(ADDRESS=...)(CONNECT_DATA=(SERVICE_NAME SID=service_name SID)))</code> <p>Note: The connect descriptor cannot contain any spaces.</p> 10. Choose Execute. 11. Choose File > Save Network Configuration. 	<p>From the operating system command line, enter:</p> <pre>namesctl NAMESTL> REGISTER service -t oracle_database -d (DESCRIPTION=(ADDRESS= ...) (CONNECT_DATA=(SERVICE_NAME SID=service_ name SID)))</pre> <p>For example, to register a database service called <code>sales.us.acme.com</code> with a default TCP/IP protocol address, enter the following:</p> <pre>NAMESTL> REGISTER sales -t oracle_database -d (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sale s-server)(PORT=1521))(CONNECT_DATA=(SERVICE_ NAME=sales.us.acme.com)))</pre> <p>If the <code>sqlnet.ora</code> file parameter <code>NAMES.DEFAULT_DOMAIN</code> is set to the same domain name as the database domain, then the database service name does not need to be qualified with the domain.</p> <p>Note: The connect descriptor cannot contain any spaces.</p>

See Also:

- ["About Connect Descriptors"](#) on page 9-2 for more information about connect descriptors
- ["Configuring a Default Domain for Clients"](#) on page 11-3 for further information about the `NAMES.DEFAULT_DOMAIN` parameter
- [Oracle Net Services Reference Guide](#) for more information about protocol syntax needed for address information

Net Service Names

Oracle Corporation advises creating a net service name for each service. If a net service name is not created, a user trying to connect to a database service must pass the database service name to make a connection. Rather than provide this information to your end users, create a net service name for the service.

To create a net service name, use either Oracle Net Manager or Oracle Names Control utility:

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none">1. Start Oracle Net Manager. See Also: "Starting Oracle Net Manager" on page 5-32. In the navigator pane, expand Oracle Names Servers.3. Select an Oracle Names server.4. From the list in the right pane, select Manage Data.5. Choose the Net Service Names tab.6. Choose Add.7. Enter the net service name in the Service Name field. You can qualify the net service name with the client's domain. The net service name is automatically domain qualified if the <code>sqlnet.ora</code> file parameter <code>NAMES.DEFAULT_DOMAIN</code> is set.8. Select a protocol the listener is configured to listen on.9. Enter the appropriate protocol information for the selected protocol in the fields provided. Note: Additional addresses can be created by choosing plus (+) at the bottom of the Address tab.10. Enter a destination service. If the destination service is an Oracle9<i>i</i> or Oracle8<i>i</i> database, then enter the service name in the Service Name field. If destination service is prior to Oracle8<i>i</i>, then choose "Oracle8 Release 8.0 Compatible Identification", and enter its SID in the SID field. Note: If you want to configure advanced CONNECT_DATA options besides the destination service, choose Advanced.11. Choose Execute.12. Choose File > Save Network Configuration.	<p>From the operating system command prompt, enter:</p> <pre>namesctl NAMESTL> REGISTER net_service_name -d (description=[(address_list=](address= ...)(connect_data=(service_name sid=...)))[])</pre> <p>For example, to register a net service name of <code>sales</code> with a default TCP/IP protocol address, enter the following:</p> <pre>NAMESTL> REGISTER sales -d (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sale s-server)(PORT=1521))(CONNECT_DATA=(SERVICE_ NAME=sales.us.acme.com)))</pre> <p>You can qualify the net service name with the client's domain. The net service name is automatically domain qualified if the <code>sqlnet.ora</code> file parameter <code>NAMES.DEFAULT_DOMAIN</code> is set.</p> <p>Note: The connect descriptor cannot contain any spaces.</p>

See Also:

- ["About Connect Descriptors" on page 9-2](#) for more information about connect descriptors
- ["Configuring a Default Domain for Clients" on page 11-3](#) for information about the NAMES.DEFAULT_DOMAIN parameter
- *Oracle Net Services Reference Guide* for more information about protocol syntax needed for address information

If you have been using local naming as your configuration method and want to migrate to an Oracle Names configuration, you can upload the net service name information used in the `tnsnames.ora` file to an Oracle Names server. To load information from a local naming configuration file into an Oracle Names server, use either Oracle Net Manager or Oracle Names Control utility:

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none"> 1. Start Oracle Net Manager. See Also: "Starting Oracle Net Manager" on page 5-3 2. In the navigator pane, expand Oracle Names Servers. 3. Select an Oracle Names server. 4. From the list in the right pane, select Manage Data. 5. Choose the Net Service Names tab. 6. Choose Load. 7. Enter the path for your current master local naming configuration file in the File field. 8. Choose Execute. 9. Choose File > Save Network Configuration. 	From the operating system command line, enter: <pre>namesctl NAMESTL> LOAD_TNSNAMES tnsnames.ora</pre>

Global Database Links and Database Qualifiers

Oracle Corporation recommends having one **global database link** for each database that is named after the database's **global database name**. For example, if the global database name is `sales.us.acme.com`, you can create a global database link of `sales.us.acme.com` and register that name with an Oracle Names server.

You verify that a global database link is working correctly by performing a `SELECT` statement on a table, for example:

```
SQL> SELECT * FROM emp@sales.us.acme.com
```

Unless you want to specify user name and password credentials for a global database link, no additional configuration is required to use the global database link.

To define a user name and password for a global database link:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Oracle Names Servers.
3. Select an Oracle Names server.
4. From the list in the right pane, select Manage Data.
5. Choose the Links tab.
6. Select Add.
7. Enter the global database link in the DB Link Name field.

Note: If the `GLOBAL_NAMES` initialization parameter has been set to `true` in the initialization parameter file, the name entered must be the same as the global database name.

8. Enter valid user name and password credentials for the database in the User and Password fields.
9. Choose Execute.
10. If you want to use a link qualifier, see ["Adding Link Qualifiers" on page 9-47](#).
11. Choose File > Save Network Configuration.

Adding Link Qualifiers

While one global database link for each database is recommended, there is no limit on the number of **link qualifiers** that may be added. Link qualifiers are appended to global database links to provide alternate settings for the database user name and password credentials. For example, a link qualifier of `fieldrep` can be appended to `sales.us.acme.com`. A connection to a remote database with the global database link and link qualifier would be:

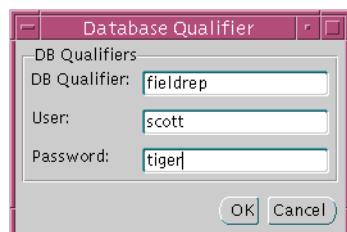
```
SQL> SELECT * FROM emp@sales.us.acme.com@fieldrep
```

Link qualifiers are primarily for older Oracle database environments where multiple database links are used to get to the different schemas in the database. Rather than use multiple database links, multiple link qualifiers can be created for a database link. Multiple link qualifiers enable the global database links to comply with the `GLOBAL_NAMES=true` setting in the initialization parameter file. This setting checks to ensure that the global database link (between the @ signs) is the same as the global database name.

To create a global link qualifier for a global database link, use Oracle Net Manager. The Oracle Names Control utility does not support creation of link qualifiers.

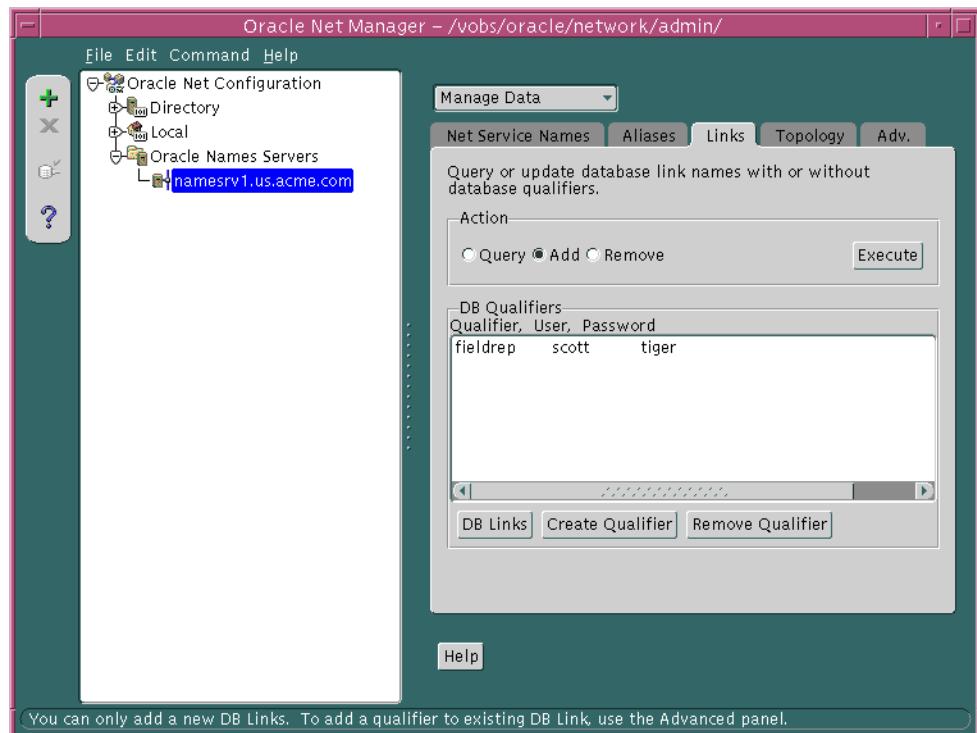
1. Follows Steps 1 through 8 in the procedure in "[Global Database Links and Database Qualifiers](#)" on page 9-46.
2. Choose DB Qualifiers.
3. Choose Create Qualifier.

The Database Qualifier dialog box appears:



4. Enter a name for the link qualifier, valid user name and password credentials, and then choose OK.

The link qualifier is added to the DB Qualifiers list box:



5. Repeat Steps 3 and 4 for each link qualifier you want to add.
6. Choose Execute.
7. Choose File > Save Network Configuration.

To verify that a global database link and link qualifier is working correctly, perform a SELECT statement on a table the user is authorized for:

```
SQL> SELECT * FROM table@db_link;
SQL> SELECT * FROM table@db_link@link_qualifier;
```

For example:

```
SQL> SELECT * FROM EMP@sales.us.acme.com;
SQL> SELECT * FROM EMP@sales.us.acme.com@fieldrep;
```

See Also: *Oracle9i Database Administrator's Guide*

Aliases

An alias can be used for any type of item stored in the Oracle Names server, such as an Oracle Names server, net service name, or global database link.

To create an alias, use Oracle Net Manager:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Oracle Names Servers.
3. Select an Oracle Names server.
4. From the list in the right pane, select Manage Data.
5. Choose the Aliases tab.
6. Select Add.
7. Enter the alias in the Alias field and the actual name of the object in the Canonical Name field.
8. Choose Execute.
9. Choose File > Save Network Configuration.

Task 8: Delegate Domains to Delegated Administrative Regions

Each Oracle Names server is responsible for being able to forward a client request to:

- At least one domain level down the domain hierarchy towards an object's authoritative administrative region
- The root administrative region if the requested object is another branch of the Oracle Names hierarchy

If delegated administrative regions are configured, the upper-level administrative region must delegate subdomains to the delegated administrative regions. If domain delegation is not done, then the Oracle Names servers in the upper-level administrative region assume authority over all subdomains.

To delegate a domain, use either Oracle Net Manager or Oracle Names Control utility:

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none">1. Start Oracle Net Manager. See Also: "Starting Oracle Net Manager" on page 5-32. In the navigator pane, expand Oracle Names Servers3. Select an Oracle Names server in the upper-level region.4. From the list in the right pane, select Manage Data.5. Choose the Topology tab.6. Choose Delegate Domain.7. Enter the name of the subdomain in the Domain Name field.8. Enter the Oracle Names server that serves the domain in the Names Server Name field, and its protocol address in the Names Server Address field. <p>See Also: <i>Oracle Net Services Reference Guide</i> for more information about protocol syntax needed for address information</p> <ol style="list-style-type: none">9. Choose Execute.10. Choose File > Save Network Configuration.	<p>Create and register an alias with the Oracle Names server:</p> <pre>namesctl NAMESTL> DELEGATE_DOMAIN domain onames_server (ADDRESS=...)</pre> <p>Note: The protocol address cannot contain any spaces.</p> <p>See Also: <i>Oracle Net Services Reference Guide</i> for more information about protocol syntax needed for address information</p> <p>For example, to delegate domain us.acme.com serviced by namesrv3.us.acme.com, enter the following:</p> <pre>NAMESTL> DELEGATE_DOMAIN us.acme.com namesrv3.us.acme.com (ADDRESS=(PROTOCOL=tcp)(HOST=namesrv3-server)(PORT=1575))</pre>

Task 9: Specify Domain Hints to Forward Requests to Remote Oracle Names Servers

A **domain hint** contains the name of the domain and at least one address of an Oracle Names server in that domain. This enables an Oracle Names server to forward the client requests to a specific address, reducing network traffic.

It is necessary for all Oracle Names servers in delegated administrative regions to be configured with a domain hint to an Oracle Names server in the root administration region. This enables an Oracle Names server to forward requests anywhere outside their own subtrees, because all name lookups can be found by forwarding through the root. If a domain hint to the root administrative region is not specified, then the local Oracle Names server will be unable to forward requests to other regions.

If you want to forward requests directly to other administrative regions, configure additional domain hints. Without additional hints, initial requests to remote regions will route through the root Oracle Names server in the region. Domain data will be cached for all regions in the forward chain by all Oracle Names servers in the forward chain. Subsequent requests may then be sent directly to the appropriate region.

To configure a domain hint, manually configure the `names.ora` file with the `NAMES.DOMAIN_HINTS` parameter. The syntax for this parameter follows:

```
NAMES.DOMAIN_HINTS=
  (HINT_DESC=
    (HINT_LIST=
      (HINT=(NAME=onames_server) (ADDRESS=...)))
    (DOMAIN_LIST=
      (DOMAIN=domain)))
```

Note: Specify the root domain with a dot (.) or a null value.

The `HINT_LIST` specifies the list of Oracle Names servers to forward the initial set of queries for the domains listed in the `DOMAIN_LIST`. A hint contains the name and address of an Oracle Names server in the remote administrative region. The Oracle Names server caches the results of those queries in its memory. If the queries fail, the Oracle Names servers listed in the `HINT_LIST` will not be cached and the local Oracle Names server continues to run without information about the root administrative region. The `HINT_LIST` should include enough Oracle Names servers to guarantee that the local Oracle Names server can resolve the query for the root or other Oracle Names servers.

Example: Domain Hint for the Root Administrative Region

In the following example, NAMES.DOMAIN_HINTS contains a domain hint for Oracle Names server `rootsvr.com` that is located in the root domain of the remote administrative region. The DOMAIN parameter is left null, meaning that the hint is for the root domain.

```
NAMES.DOMAIN_HINTS=
(HINT_DESC=
(HINT_LIST=
(HINT=
(NAME=rootsvr.com)
(ADDRESS=(PROTOCOL=tcp)(HOST=rootsvr)(PORT=1575)))
(DOMAIN_LIST=
(DOMAIN=)))
```

When the local Oracle Names server is started:

1. It reads the NAMES.DOMAIN_HINTS parameter. For each domain listed in the DOMAIN_LIST, it calls the Oracle Name server listed in the HINT_LIST and queries for all Oracle Names servers in the domain.
2. Based on the time-to-live (TTL) of the answer, the local Oracle Names server sets up queries that are automatically issued before the remote Oracle Names server data expires.

Example: Domain Hint for Multiple Oracle Names Servers

The following example shows a hint to query two domains, the root domain and the `us.acme.com` domain, for Oracle Names servers `rootsvr.com` and `namesrv.us.acme.com`.

```
NAMES.DOMAIN_HINTS=
(HINT_DESC=
(HINT_LIST=
(HINT=
(NAME=rootsvr.com)
(ADDRESS=(PROTOCOL=tcp) (HOST=rootsvr) (PORT=1575) ))
(HINT=
(NAME=namesrv.us.acme.com)
(ADDRESS=(PROTOCOL=tcp) (HOST=hr-server) (PORT=1575) )))
(DOMAIN_LIST=
(DOMAIN=
(DOMAIN=us.acme.com) ))
```

In this query, the local Oracle Names server:

1. Sends a query for the root administrative region, using the addresses given in the `HINT`.
2. Sends a query for the `us.acme.com` domain, using the same addresses from the `HINT` descriptor.

Task 10: Connect to the Database

Clients can connect to the database using the following syntax:

```
CONNECT username/password@net_service_name
CONNECT username/password@database_service
```

Removing an Oracle Names Server

To remove an Oracle Names server:

1. Undefine an Oracle Names server and its authoritative domain with the Oracle Names Control utility's UNREGISTER_NS command. From the command line, enter:

```
namesctl  
NAMESTL> UNREGISTER_NS {oname_server}{domain}
```

For example, to unregister a Oracle Names server named namesrv1 in domain us.acme.com, enter the following:

```
NAMESTL> UNREGISTER_NS namesrv1.us.acme.com us.acme.com
```

2. If it is running, stop the Oracle Names server. Use either Oracle Net Manager or the Oracle Names Control utility.

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none">1. Start Oracle Net Manager. See Also: "Starting Oracle Net Manager" on page 5-32. In the navigator pane, expand Oracle Names Servers.3. Select the Oracle Names server.4. From the list in the right pane, select Manage Server.5. Choose the Control tab.6. Choose Stop, and then choose Apply to stop the Oracle Names server. <ol style="list-style-type: none">3. Remove the names.ora file from the \$ORACLE_HOME/network/admin directory on UNIX, or the ORACLE_HOME\network\admin directory on Windows NT.4. Remove the ckp*.ora file. By default the checkpoint files are stored in the \$ORACLE_HOME/network/names directory on UNIX, and the ORACLE_HOME\network\names directory on Windows NT.5. Reconfigure clients and the database server with the correct list of Oracle Names servers to contact, as well as the preferred naming method.	<p>From the command line, enter:</p> <pre>namesctl NAMESTL> STOP</pre>

See Also:

- ["Task 4: Configure Clients and Database Servers To Use Oracle Names Servers" on page 9-34](#) for instructions on discovering Oracle Names server and selecting the preferred naming method
- ["Configuring Preferred Oracle Names Servers" on page 11-11](#) for instructions on working with the NAMES.REFERRED_SERVERS parameter in the `sqlnet.ora` file

6. Restart the Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

Configuring the Host Naming Method

In environments where simple connectivity is desired, host naming can eliminate the need for service name lookup in the `tnsnames.ora` files. However, for large or complex environments where advanced features such as connection pooling, external procedures, or Heterogeneous Services, which require additional connect information, are desired, host naming is not suitable. In these cases, another naming method is recommended.

The host naming method is available for TCP/IP network environments only. With host naming, clients use the server's global database name in their connect string to connect to the database. The global database name is equivalent to a host name or an alias in an existing name resolution service.

Clients can connect to a server using the alias if:

- You are connecting to an Oracle database service with Oracle Net Server/Oracle Net Client software installed
- Your client and server are connecting over a TCP/IP protocol
- All names are resolved through an IP address translation mechanism such as **Domain Name System (DNS)**, or a centrally maintained TCP/IP `hosts` file
- No advanced features like **Oracle Connection Manager** or security are requested or required

To configure the host naming method, perform these tasks:

[Task 1: Configure the Listener with the Global Database Name](#)

[Task 2: Configure HOSTNAME as the First Naming Method](#)

[Task 3: Set Up Host Name Resolution Environment](#)

[Task 4: Connect to the Database](#)

Task 1: Configure the Listener with the Global Database Name

The global database name must be registered with the listener. How this registration takes place depends upon the release of the database.

Oracle9i or Oracle8i

If the database can find the listener, information about the database is automatically registered with the listener, including the global database name. The listener is found if:

- The default listener named `LISTENER` running on TCP/IP on port 1521 is running
- The `LOCAL_LISTENER` parameter is set in the initialization file

If the database cannot find the listener, the `listener.ora` file must be configured with the `GLOBAL_DBNAME` parameter, as described in "[Oracle8 Release 8.0 or Previous](#)".

Oracle8 Release 8.0 or Previous

If the database is an Oracle8 release 8.0 database or Oracle7 database, database information is registered with the listener through the `listener.ora` file. For these database, statically configure the `SID_LIST_listener_name` section to include the `GLOBAL_DBNAME` parameter, as shown next:

```
SID_LIST_listener=
(SID_LIST=
  (SID_DESC=
    (GLOBAL_DBNAME=sales.us.acme.com)
    (SID_NAME=db1)
    (ORACLE_HOME=/u01/app/oracle/8.0.5))
```

See Also: ["Configuring Static Service Information" on page 12-10](#)

Task 2: Configure HOSTNAME as the First Naming Method

Configure host naming as the first method specified in the NAMES.DIRECTORY_PATH parameter in the `sqlnet.ora` file. This parameter specifies the order of naming methods Oracle Net can use to resolve connect identifiers to connect descriptors.

To specify host naming as the first naming method:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select Naming.
4. Choose the Methods tab.
5. Choose HOSTNAME from the Available Methods list, then choose the right-arrow button.
6. Select HOSTNAME in the Selected Methods list, then use the Promote button to move the selection to the top of the list.
7. Choose File > Save Network Configuration.

The `sqlnet.ora` file updates with the NAMES.DIRECTORY_PATH parameter, listing hostname first:

```
NAMES.DIRECTORY_PATH=(hostname, tnsnames)
```

Task 3: Set Up Host Name Resolution Environment

The global database name must be resolved through an IP address translation mechanism, such as DNS, NIS, or a centrally-maintained TCP/IP host file, `/etc/hosts`.

For example, if a global database name of `sales.us.acme.com` for a database exists on a computer named `sales-server`, the entry in the `/etc/hosts` file would look like the following:

```
#IP address of server      host name      alias
144.25.219.146            sales-server    sales.us.acme.com
```

Note that the domain section of the global database name must match the network domain.

Task 4: Connect to the Database

Clients can connect to the database using the alias. Using the example in ["Task 3: Set Up Host Name Resolution Environment"](#), the client can use `sales.us.acme.com` in the connect string:

```
CONNECT username/password@sales.us.acme.com
```

If the client and server are in the same domain of `us.acme.com`, the client needs to enter only `sales` in the connect string.

Configuring External Naming Methods

External naming refers to the method of resolving a net service name, stored in a non-Oracle naming service, to a network address. External naming services include:

- [Network Information Service \(NIS\)](#)
- [Cell Directory Services \(CDS\)](#)

See Also: *Oracle Advanced Security Administrator's Guide* for instructions on configuring CDS

Configuring NIS

Organizations and corporations already using Network Information Service (NIS) as part of their systems infrastructure have the option to store net service names and addresses in NIS, using NIS external naming.

When a user gives a command such as

```
sqlplus scott/tiger@payroll
```

(where payroll is an Oracle service) NIS external naming on the node running the client program (or database server acting as a client program) contacts an NIS server located somewhere in the network, and passes the net service name to the NIS server. The NIS server resolves the net service name into a Oracle Net address and returns this address to the client program (or server acting as a client program). The client program then uses this address to connect to the Oracle database.

A computer that acts as an NIS server runs a program called `ypserv`, which handles name requests. The `ypserv` program stores different types of data in special files called [maps](#). For example, passwords are stored in a map called `passwdbyname`. Oracle database service names are stored in a map called `tnsnames`.

When a user issues a connect string, NIS External Naming uses an RPC call to contact the `ypserv` program and passes the Oracle net service name `payroll` and the name of the map—`tnsnames`. The `ypserv` program looks in the `tnsnames` map for the name `payroll` and its corresponding value, which is the address for the net service name. The address is returned to the client, and the client program (or server acting as a client program) uses this address to contact the database server.

System Requirements

NIS External Naming requires SQL*Net 2.2 or greater.

Task 1: Configure NIS Servers to Support the NIS External Naming

Before configuring servers to support the NIS external naming, make sure that NIS is configured and running on the NIS servers that need to resolve Oracle database net service names. Consult your NIS documentation for specifics.

Adding the "tnsnames" Map to the Existing Set of NIS Maps To add the tnsnames map to the existing set of NIS maps:

1. Create a `tnsnames.ora` file, as specified in "[Configuring the Local Naming Method](#)" on page 9-5.

Note: Keep a copy of the `tnsnames.ora` file, preferably in `$ORACLE_HOME/network/admin` directory. You may need to use this file again later to load net service names into the NIS map.

2. Convert the contents of the `tnsnames.ora` file to a tnsnames map using the `tns2nis` program.

Note: The `tns2nis` program is supplied with NIS External Naming on the Oracle Universal Installer tape or disk.

For example, run `tns2nis` on the command line with one argument:

```
tns2nis tnsnames.ora
```

The `tns2nis` program reads the `tnsnames.ora` file from the current directory. (If `tnsnames.ora` file is not located in the current directory, you can use a full path name to specify its location—for example, `/etc/tnsnames.ora` or `$ORACLE_HOME/network/admin/tnsnames.ora`).

The "tnsnames" map is then written into the current working directory.

3. Copy `tnsnames` to the NIS server, if it is not already there.

4. Install the `tnsnames` map using `makedbm`, which is an NIS program. Refer to your NIS documentation for more information.

Note: This step should be performed by the person in charge of NIS administration.

The `makedbm` program converts the `tnsnames` map into two files that the NIS server can read. The location of these files is operating system specific.

See Also: Operating system-specific documentation for details

For example, to generate and install a `tnsnames` map on Sun Solaris, as root enter the following at the command line

```
# makedbm tnsnames /var/yp/'domainname'/tnsnames
```

Verifying that the "tnsnames" Map Has Been Properly Installed You can test the NIS server to see if the map has been installed properly by typing a command with the format:

```
ypmatch net_service_name tnsnames
```

For example, you might enter:

```
ypmatch payroll.com tnsnames
```

This returns the length of the address (in characters) followed by the address; for example:

```
99 (description=(address=(protocol=tcp)
(host=garlic)(port=1999)))
(connect_data=(service_name=dirprod))
```

Task 2: Configure Clients

To configure clients, configure NIS as the first method specified in the NAMES.DIRECTORY_PATH parameter in the `sqlnet.ora` file. This parameter specifies the order of naming methods Oracle Net can use to resolve connect identifiers to connect descriptors.

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select Naming.
4. Choose the Methods tab.
5. Select NIS from the Available Methods list, then choose the right-arrow button.
6. Select NIS in the Selected Methods list, then use the Promote button to move the selection to the top of the list.
7. Choose File > Save Network Configuration.

The `sqlnet.ora` file updates with the NAMES.DIRECTORY_PATH parameter, listing nis first:

```
NAMES.DIRECTORY_PATH=(nis, hostname, tnsnames)
```


Exporting Naming Data to a Directory Server

This chapter explains how to export data stored in a `tnsnames.ora` file or an Oracle Names server to an LDAP-compliant directory server.

This chapter contains these topics:

- [Exporting Net Service Names from a `tnsnames.ora` File](#)
- [Exporting Network Objects from an Oracle Names Server](#)

See Also: ["Directory Server Support"](#) on page 3-4 for an overview of directory server support of Oracle Net Services

Exporting Net Service Names from a tnsnames.ora File

If a `tnsnames.ora` file already exists, its **net service names** can be exported to a directory server. The export is performed for one domain at a time.

The tasks to export data from a `tnsnames.ora` file are as follows:

[Task 1: Create Structure in Directory Server](#)

[Task 2: Create Oracle Contexts](#)

[Task 3: Configure Directory Server Usage](#)

[Task 4: Export Objects To a Directory Server](#)

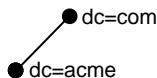
Note: These tasks assume that you have a directory server installed and running.

Task 1: Create Structure in Directory Server

In the directory server, create the **directory information tree (DIT)** with the structure in which you want to import net service names. Create the structure leading up to the top of the **Oracle Context**.

For example, if the `tnsnames.ora` file supports a domain structure `acme.com` and you want to replicate this domain in the directory, create domain component entries of `dc=com` and `dc=acme` in the directory, as depicted in [Figure 10–1](#).

Figure 10–1 acme.com in Directory Server



You can replicate the domain structure you currently use with `tnsnames.ora`, or you can develop an entirely different structure. Introducing an entirely different structure can change the way in which clients enter the net service name in the connect string. Therefore, Oracle Corporation recommends considering relative and absolute naming issues prior to changing the structure.

See Also:

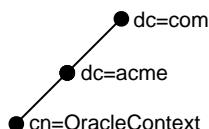
- [Directory server vendor documentation for directory entry configuration instructions](#)
- ["Client Connections Using Directory Naming" on page 3-10](#)

Task 2: Create Oracle Contexts

Create an Oracle Context under each DIT location that you created in Task 1. The Oracle Context has a **relative distinguished name (RDN)** of (cn=OracleContext). The Oracle Context stores network object entries, as well as other entries for other Oracle components. In [Figure 10–2](#), cn=OracleContext is created under dc=acme, dc=com.

To create the Oracle Context, you must use [Oracle Net Configuration Assistant](#).

Figure 10–2 Oracle Context



See Also:

- [Chapter 3, "Configuration Management Concepts"](#) for further information about the Oracle Context
- ["Configuring Directory Usage After Installation" on page 8-5](#) for instructions on creating an Oracle Context

Task 3: Configure Directory Server Usage

If not already done as a part of creating the Oracle Contexts, configure the Oracle home for directory server usage. The Oracle home you configure should be the one that will perform the export.

See Also: [Chapter 8, "Setting Up Directory Server Usage"](#) for further information about configuring directory server usage

Task 4: Export Objects To a Directory Server

To export net service names contained in a `tnsnames.ora` file to a directory:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

2. If the `tnsnames.ora` file you want to export is not the one loaded into Oracle Net Manager, then use File > Open Network Configuration to select the `tnsnames.ora` file to export.
3. Choose Command > Directory > Export Net Service Names.

The Directory Server Migration Wizard starts.

4. Choose Next.

If net service names with multiple domain were detected in the `tnsnames.ora` file, then the Select Domain page appears. Continue to Step 5.

If the net service names are not domain qualified, the Select Net Service Names page appears. Skip to Step 6.

5. Select the network domain whose net service names you want to export, and then choose Next.

The Select Net Service Names page appears.

6. Select the net service names from the list to export, and then choose Next.

The Select Destination Context page appears.

7. In the Select Destination Context page, perform the following:

- a. From the Directory Naming Context list, select the directory entry that contains the Oracle Context. The directory naming context is part of a directory subtree that contains one or more Oracle Contexts.
- b. From the Oracle Context list, select the Oracle Context to which you want to export the selected net service names.

- c. Choose Next.

The Directory Server Update page appears with the status of the export operation.

8. Choose Finish to dismiss the Directory Server Migration Wizard.

Exporting Network Objects from an Oracle Names Server

Note: Directory naming does not currently support the following Oracle Names object types:

- Aliases
- User names and passwords for **global database links** and **link qualifiers**

Therefore, you cannot export these entries. If you are using aliases, create duplicate net service names in the directory, as described in "[Configuring the Directory Naming Method](#)" on page 9-12. If you are using global database links and link qualifiers and require their use, continue to use Oracle Names for this release.

Database services and net service names stored in an Oracle Names server can be exported directly to a directory server or indirectly to an **LDAP Data Interchange Format (LDIF)** file, which can then be used to load the directory server. Data is exported from a specified domain. If the domain has authority for delegated domains, then the data from the delegated domains can also be exported.

Release 8.1.6 or Later Clients

New client installations that have been configured for directory server usage can take advantage of directory naming once data is exported to a directory server and directory naming is selected as a naming method. To export data for new client configurations, follow the procedure in "[Exporting Data to a Directory Server](#)" on page 10-7.

Release 8.1.5 or Previous Clients

In configurations with clients prior to release 8.1.6 which have not been configured for directory server usage, or clients that do not support directory naming, you have two choices. You can:

- Continue to use Oracle Names servers
 - If you continue to use Oracle Names servers, data stored in Oracle Names servers and directory server must be maintained and synchronized separately.
- Configure Oracle Names servers as **Oracle Names LDAP Proxy servers**
 - Oracle Names LDAP Proxy servers are Oracle Names servers that have been configured to proxy for directory servers. Upon startup, Oracle Names LDAP

Proxy servers obtain network object information from the directory server. This provides a single point of definition for all data in the directory server and does not require that data in both Oracle Names server and directory server be maintained separately and simultaneously.

To export data for older client configurations, follow the procedures in both "[Exporting Data to a Directory Server](#)" on page 10-7 and "[Configuring Oracle Names LDAP Proxy Servers](#)" on page 10-15.

Exporting Data to a Directory Server

The tasks to export data from an Oracle Names server are as follows:

[Task 1: Create Structure in Directory Server](#)

[Task 2: Create Oracle Contexts](#)

[Task 3: Configure Directory Server Usage](#)

[Task 4: Obtain List of Objects to Export](#)

[Task 5: Export Objects To a Directory Server](#)

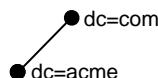
Note: These tasks assume that a directory server is installed and running and the Oracle Names Control utility (`namesctl`) executable is release 9.0. You can use a release 8.1.6 or release 8.1.7 `namesctl` executable to perform export tasks. However, the instructions for using the `DUMP_LDAP` command are different from release 8.1 and release 9.0. If you are using release 8.1, see the *Net8 Administrator's Guide* for that release.

Task 1: Create Structure in Directory Server

In the directory server, create the DIT with the structure in which you want to import Oracle Names objects. Create the structure leading up to the top of the Oracle Context.

For example, if Oracle Names has a domain structure `acme.com` and you want to replicate this domain in the directory, create domain component entries `dc=com` and `dc=acme` in the directory, as depicted in [Figure 10-3](#).

Figure 10-3 acme.com in Directory Server



You can replicate the domain structure you currently use with Oracle Names, or you can develop an entirely different structure. Introducing an entirely different structure will change the way in which clients enter the **connect identifier** in the connect string. Therefore, Oracle Corporation recommends that you consider the relative and absolute naming issues prior to changing the structure.

Important: If you plan to use Oracle Names LDAP Proxy servers that support multiple administrative regions, Oracle Corporation recommends mirroring the current Oracle Names domain structure in the directory DIT structure. Using a different structure may require modifying the topology defined for the Oracle Names LDAP Proxy servers. Support for topology modification is not currently provided. The following constitutes a topology change:

- Addition or removal of any Oracle Names servers, as well as modification of an Oracle Names server's name or protocol address
 - Modification of domain definitions for an administrative region or its delegated administrative regions
-

See Also:

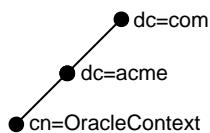
- Directory server vendor documentation for directory entry configuration instructions
- "[Client Connections Using Directory Naming](#)" on page 3-10

Task 2: Create Oracle Contexts

Create an Oracle Context under each DIT location that you created in Task 1. The Oracle Context has a RDN of (cn=OracleContext). Oracle Contexts may be required for the root domain and each of its delegated domains. In [Figure 10-4](#), cn=OracleContext is created under dc=acme,dc=com.

To create the Oracle Context, you must use Oracle Net Configuration Assistant.

Figure 10-4 Oracle Context



See Also:

- [Chapter 3, "Configuration Management Concepts"](#) for further information about the Oracle Context
- ["Configuring Directory Usage After Installation" on page 8-5](#) for instructions on creating an Oracle Context

Task 3: Configure Directory Server Usage

If not already done as a part of creating the Oracle Contexts, configure the Oracle home for directory server usage. The Oracle home you configure should be the one that will perform the export.

See Also: [Chapter 8, "Setting Up Directory Server Usage"](#) for further information about configuring directory server usage

Task 4: Obtain List of Objects to Export

Determine the Oracle Names domain structure and the objects within that structure. The Oracle Names Control utility offers commands described in [Table 10–1](#) to help you with this task.

Table 10–1 Oracle Names Control Utility Structural Commands

Command	Description
LIST_DOMAINS	Lists all of the authoritative domains See Also: LIST_DOMAINS command in <i>Oracle Net Services Reference Guide</i>
LIST_DELEGATED	Lists all of the delegated domains See Also: LIST_DELEGATE command in <i>Oracle Net Services Reference Guide</i>
LIST_OBJECTS	Lists all of the authoritative network objects See Also: LIST_OBJECTS command in <i>Oracle Net Services Reference Guide</i>

Task 5: Export Objects To a Directory Server

The Oracle Names Control utility enables you to export network objects from a domain into the directory server using the DUMP_LDAP command. You can use the DUMP_LDAP command to export objects directly to a directory server or indirectly to an LDIF file, which can then be used to load the exported objects to the directory server.

Run DUMP_LDAP from an authoritative Oracle Name server for the domain. The syntax to export data directly to a directory server is as follows:

```
namesctl  
NAMESTCL> DUMP_LDAP [source] [destination] [options] {-h host} {-p  
port} {-D user_dn} {-w password}
```

See Also: DUMP_LDAP command in *Oracle Net Services Reference Guide* for a description of the arguments

The syntax to export data to a LDIF file is as follows:

```
namesctl  
NAMESTCL> DUMP_LDAP [source] [destination] [options] {-f filename}
```

Note: The generated LDIF file can later be loaded into the directory server with the ldapadd command or the ldapmodify command with the add option. See your directory server's vendor documentation for syntax usage.

If the directory server's DIT structure has been designed to match the current Oracle Names structure, review the following examples to understand how to export data:

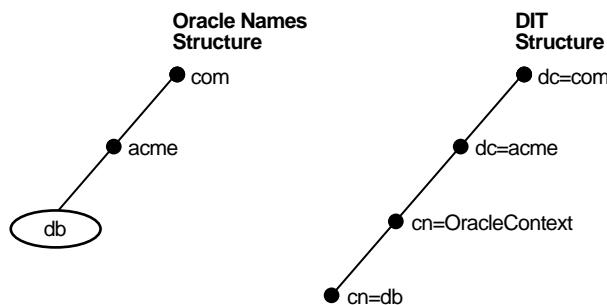
- [Example 1: Exporting Data from a Single Domain to the Same DIT Node](#)
- [Example 2: Exporting Data from a Domain Tree to a Similar DIT](#)

See Also: ["Configuring Oracle Names LDAP Proxy Servers"](#) on page 10-15 for examples of exporting data to a dissimilar DIT structure

Example 1: Exporting Data from a Single Domain to the Same DIT Node

Figure 10–5 shows an Oracle Names domain structure of acme.com. It contains a database service called db. The DIT has been designed with domain entries that match the Oracle Names domain structure. With this DIT structure, db.acme.com can be exported to cn=OracleContext,dc=acme,dc=com.

Figure 10–5 Single Domain Export to the Same DIT Node



Either of the following syntaxes can be used to export data from Oracle Names to the configured DIT structure:

```
NAMESCTL> DUMP_LDAP acme.com (dn:dc=acme,dc=com) -f sample.ldif  
NAMESCTL> DUMP_LDAP acme.com -f sample.ldif
```

In the first line of syntax, the destination **distinguished name (DN)**, excluding cn=OracleContext, is explicitly specified. It is not necessary to specify the destination DN, as shown in the second line of syntax, because the destination DIT structure is domain based and matches the domain model used in Oracle Names. Note that cn=OracleContext is automatically inserted.

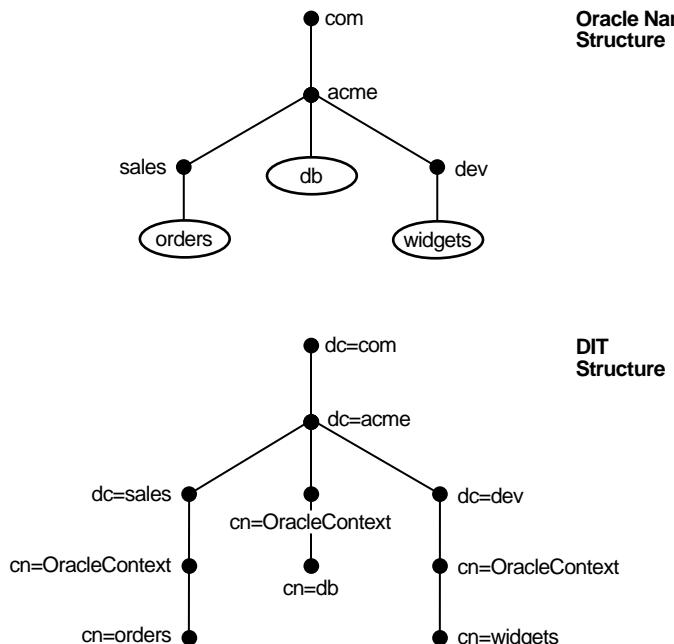
The database service db is exported to cn=OracleContext,dc=acme,dc=com and has a DN location of (dn: cn=db, cn=OracleContext, dc=acme, dc=com).

Example 2: Exporting Data from a Domain Tree to a Similar DIT

Data can be exported from a root domain and its delegated domains in Oracle Names to a directory server that uses a similar DIT in one step rather than one domain at a time.

[Figure 10–6](#) shows an Oracle Names structure that contains a root domain of acme.com and delegated domains of sales.acme.com and dev.acme.com. Database services db, orders, and widgets reside in acme.com, sales.acme.com and dev.acme.com, respectively. The directory server's DIT is similar to the Oracle Names domain structure.

Figure 10–6 Root and Subdomain Domain Export to Similar DIT



The following syntax is used to export data from the acme.com root domain and its delegated domains to the configured DIT structure:

```
NAMESCTL> DUMP_LDAP acme.com -R -f sample.ldif
```

The following table shows how database objects in acme.com, sales.acme.com, and dev.acme.com are mapped to DNs in the directory server.

Database Object in Oracle Names	New DN in Directory Server
db.acme.com	dn:cn=db,cn=OracleContext,dc=acme,dc=com
orders.sales.acme.com	dn:cn=orders,cn=OracleContext,dc=sales, dc=acme,dc=com
widgets.dev.acme.com	dn:cn=widgets,cn=OracleContext,dc=dev, dc=acme,dc=com

Configuring Oracle Names LDAP Proxy Servers

If you are not ready to upgrade clients to a version that supports directory naming, you must use Oracle Names LDAP Proxy servers. Generally, client configurations do not require modification. However, if the directory server's DIT structure does not match the Oracle Names domain structure, you must also reconfigure the NAMES.DEFAULT_DOMAIN parameter in the `sqlnet.ora` file to point to the new domain structure.

See Also: ["Configuring a Default Domain for Clients"](#) on page 11-3 for instructions

The tasks to create Oracle Names LDAP Proxy servers are as follows:

- [Task 1: Upgrade Oracle Names Servers to Release 9.0](#)
- [Task 2: Start Oracle Names Servers](#)
- [Task 3: Populate the Directory Server](#)
- [Task 4: Configure Oracle Names Servers as Proxies](#)
- [Task 5: Run the Oracle Names LDAP Proxy Servers](#)

Task 1: Upgrade Oracle Names Servers to Release 9.0

Upgrade all Oracle Names servers within a region to Oracle Names release 9.0. Releases prior to 9.0 do not support Oracle Names LDAP Proxy servers.

See Also: [Oracle9i Database Migration](#)

Task 2: Start Oracle Names Servers

Start each of the Oracle Names servers to generate the `ckptop.ora` file. Use either Oracle Net Manager or the Oracle Names Control utility.

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none">1. In the navigator pane, expand Oracle Names Servers.2. Select the Oracle Names server.3. From the list in the right pane, select Manage Server.4. Choose the Control tab.5. Choose Start, and then choose Apply to generate the topology data.	From the command line, enter: <code>namesctl</code> <code>NAMESCTL> START</code>

Starting release 9.0 Oracle Names servers creates a `ckptop.ora` file in `$ORACLE_HOME/network/names` on UNIX and `ORACLE_HOME\network\names` on Windows NT, or in the file specified by the `names.ora` file parameter `NAMES_TOPOLOGY_CHECKPOINT_FILE`. This file contains topology data that defines the domains in the administrative region and the Oracle Names servers that have authority for them. Specifically, topology data consists of definitions for all parent domains and Oracle Names servers in the region. The Oracle Names servers use this information to understand the structure of the domain tree.

Oracle Names LDAP Proxy servers require the generated `ckptop.ora` file. If the Oracle Names structure has multiple administrative regions, Oracle Corporation recommends mirroring the current Oracle Names domain structure in the directory DIT structure. Using a different structure may require modifying the topology defined for the Oracle Names LDAP Proxy servers. Support for topology modification is not currently supported. The following constitutes a topology change:

- Addition or removal of any Oracle Names servers, as well as modification of an Oracle Names server's name or protocol address
- Modification of domain definitions for an administrative region or its delegated administrative regions

Task 3: Populate the Directory Server

If you have not already done so, populate the directory:

1. Create the DIT in the directory structure.

See Also: ["Task 1: Create Structure in Directory Server" on page 10-8](#)

2. Create Oracle Contexts in the directory under the root domain and each of its delegated domains.

See Also: ["Configuring Directory Usage After Installation" on page 8-5](#) for instructions on creating an Oracle Context

3. Configure each Oracle Names server within a region with directory usage settings in the `ldap.ora` file.

To do this, select the "Select the directory server you want to use" option in Oracle Net Configuration Assistant.

See Also: [Chapter 8, "Setting Up Directory Server Usage"](#)

4. Export data from Oracle Names under the appropriate Oracle Contexts in the directory server. Use either the procedure described in ["Exporting Data to a Directory Server" on page 10-7](#) or an alternate external method.

Task 4: Configure Oracle Names Servers as Proxies

Configure each Oracle Names server within a region to load directory server information from a specific DN.

To configure Oracle Names Servers as proxy servers:

1. Stop each Oracle Names server. Use either Oracle Net Manager or the Oracle Names Control utility.

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none">1. In the navigator pane, expand Oracle Names Servers.2. Select the Oracle Names server.3. From the list in the right pane, select Manage Server.4. Choose the Control tab.5. Choose Stop, and then choose Apply to stop the Oracle Names server.	From the command line, enter: <code>namesctl NAMESTL> STOP</code>

2. Set the NAMES.ADMIN_REGION parameter in the names.ora file to the directory server's DN or read the directory server's DN from an LDIF input file.

The syntax for an Oracle Names LDAP Proxy server to load the data from a directory server is as follows:

```
NAMES.ADMIN_REGION=
  (REGION=
    (TYPE=ldap)
    [ (USERID=user_dn) ]
    [ (PASSWORD=password) ]
    [ (HOST=host) ]
    [ (PORT=port) ]
    [ (SUBTREE_LIST=
      [ (SUBTREE=(BASE=base_DN) [ (SCOPE=sub|one) ) ]
      [ (SUBTREE=(BASE=base_DN) [ (SCOPE=sub|one) ) ]
    [ ]) ])
```

Values from equivalent ldap.ora file parameters are used as defaults for the USER, HOST, and SUBTREE parameters.

Set the NAMES . ADMIN_REGION sub-parameter as follows:

Table 10–2 Sub-Parameters of the NAMES.ADMIN_REGION Parameter for LDAP

Sub-Parameter	Description
TYPE	Specify that the Oracle Names LDAP Proxy server is to load data directly from a directory server.
USERID	(Optional) This entry is necessary if data is restricted. Specify a directory user with read privileges in the form of a DN. For example, c=us, o=acme, ou=admin, cn=scott is the DN for an administrator named Scott. Note: Do not prefix the DN with dn:
PASSWORD	(Optional) This entry is required if data is restricted. Specify the password for the directory user.
HOST	(Optional) Specify the directory server host name.
PORT	(Optional) Specify the listening TCP/IP port for the directory server.
TIMEOUT	(Optional) Specify the time limit in seconds that the Oracle Names LDAP Proxy server can spend performing a search of directory objects. This time limit cannot be greater than the time limit set for searches in the directory server. By default, the time limit is set to 10 seconds, which is sufficient for most searches. See Also: "Increasing Search Size Limit" on page 17-16 for instructions for increasing the time limit of ldapsearch
SUBTREE_LIST SUBTREE	(Optional) Use the SUBTREE parameter to specify the DN. Use the SUBTREE_LIST parameter if there are multiple DNs. The SUBTREE parameter supports the following sub-parameters: BASE—Specify a subtree location in the DIT from which the Oracle Names Proxy server should load data. The location is specified as a DN of a LDAP directory entry. The Oracle Names Proxy loads directory naming data from one or more Oracle Contexts located in the subtree below the entry specified. Note: Do not prefix the DN with "dn:". For example, (BASE=dn:dc=com) is not acceptable. SCOPE—Specifies where the Oracle Names Proxy server looks for directory data to load. one (default) specifies that the Oracle Names Proxy server look only in one Oracle Context directly below the entry specified in the BASE parameter sub specifies the Oracle Names Proxy server look in all the Oracle Contexts in the subtree below the entry specified in the BASE parameter.

The following example shows an Oracle Names LDAP Proxy server configured to load the data from an Oracle Context that is directly under the DN (dn:dc=acme,dc=com) and all Oracle Contexts under the DN subtree (dn:dc=us,dc=acme,dc=com).

```
NAMES.ADMIN_REGION=
  (REGION=
    (TYPE=LDAP)
    (HOST=ldap_server)
    (PORT= 1389)
    (SUBTREE_LIST=
      (SUBTREE=(BASE=dc=acme,dc=com))
      (SUBTREE=(BASE=dc=us,dc=acme,dc=com) (scope=sub))
    )
  )
)
```

The syntax for an Oracle Names LDAP Proxy server to load the data from an LDIF file is as follows:

```
NAMES.ADMIN_REGION=
  (REGION=
    (TYPE=ldif)
    [(FILE=ldif_file)]))
```

Table 10–3 Sub-Parameters of the NAMES.ADMIN_REGION Parameter for LDIF

Sub-Parameter	Description
TYPE	Specify that the Oracle Names LDAP Proxy server is to use data from an LDIF file.
FILE	Specify the input LDIF file name. See Also: Directory server vendor documentation for LDIF file creation information

The following example shows an Oracle Names LDAP Proxy server configured to load data from LDIF file onames.ldif.

```
NAMES.ADMIN_REGION=
  (REGION=
    (TYPE=LDIF)
    (FILE=/private/eminer/nm/90/proxy/onames.ldif))
```

The following LDIF file excerpt shows a DN of
(dn: cn=sales,cn=OracleContext,dc=acme,dc=com) that contains net service names sales and hr.

```
dn: cn=sales,cn=OracleContext,dc=us,dc=acme,dc=com
objectclass: top
objectclass: orclNetService
cn: sales
orclNetDescString:
(DESCRIPTION=(ADDRESS_
LIST=(ADDRESS=(PROTOCOL=tcp)(Host=sales-pc)(Port=1521)))(CONNECT_DATA=(SERVICE_
NAME=sales.us.acme.com)))
dn: cn=hr,cn=OracleContext,dc=us,dc=acme,dc=com
objectclass: top
objectclass: orclNetService
cn: hr
orclNetDescString:
(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(COMMUNITY=TCP_
COMMUNITY)(PROTOCOL=tcp)(Host=hr-pc)(Port=1521)))(CONNECT_DATA=(SERVICE_
NAME=hr.us.acme.com)))
```

Task 5: Run the Oracle Names LDAP Proxy Servers

Run each of the Oracle Names servers with the new configuration. Use either Oracle Net Manager or the Oracle Names Control utility.

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none">1. In the navigator pane, expand Oracle Names Servers.2. Select the Oracle Names server.3. From the list in the right pane, select Manage Server.4. Choose the Control tab.5. Choose Start, and then choose Apply.	From the command line, enter: <code>namesctl NAMESCTL> START</code>

See Also: "Oracle Names LDAP Proxy Server Error Reporting" on page 17-16 region load operation troubleshooting advice

Considerations for Dissimilar DIT Structures

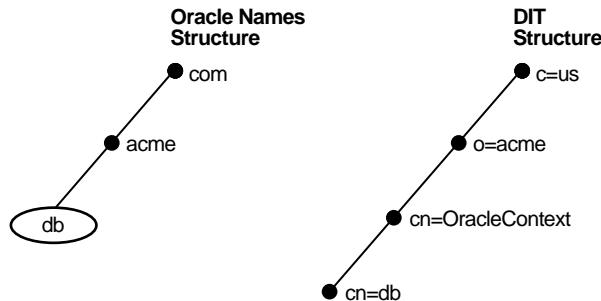
If your directory server's structure has been designed with a DIT that is dissimilar to the current Oracle Names domain structure, review the following examples to understand how to export data:

- [Example 1: Exporting Data from a Single Domain to a Non-DC Node](#)
- [Example 2: Exporting Data from a Domain Tree to a Non-DC DIT](#)
- [Example 3: Exporting Data from Multiple Domains to One DIT Node](#)
- [Example 4: Reorganizing a Tree Structure During an Export](#)

Example 1: Exporting Data from a Single Domain to a Non-DC Node

[Figure 10–7](#) shows an Oracle Names domain structure of acme.com. It contains a database service called db. The directory server has been designed with a DIT of cn=OracleContext, o=acme, c=us.

Figure 10–7 Single Domain Export to One DIT Node



The following syntax can be used to export data from the acme.com domain to the configured DIT structure:

```
NAMESTCL> DUMP_LDAP acme.com (dn:c=us,o=acme) -f sample.ldif
```

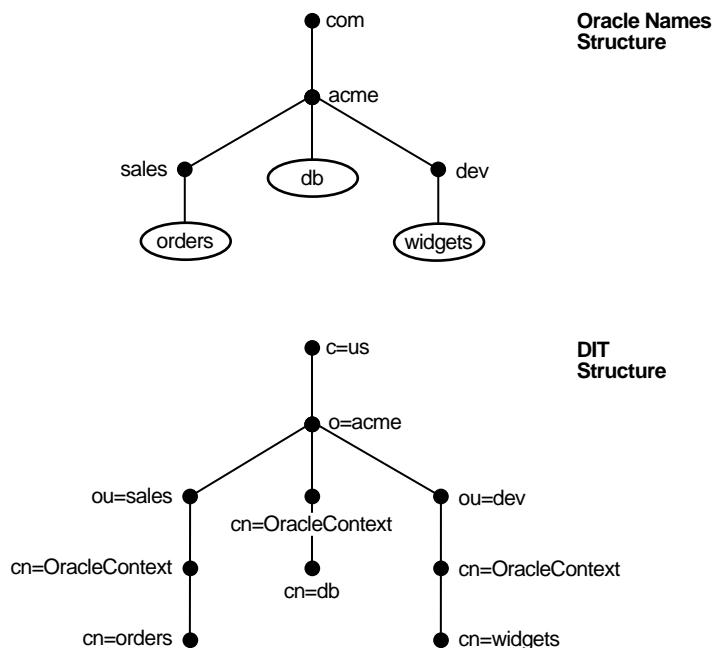
Because the DIT is different from the Oracle Names structure, the destination DN must be explicitly specified. cn=OracleContext is automatically pre-appended to the left of the destination DN. In other words, you do not need to explicitly specify cn=OracleContext in the DN.

Example 2: Exporting Data from a Domain Tree to a Non-DC DIT

Data can be exported from an Oracle Names administrative region to a directory server that uses a dissimilar DIT. RDNs in the destination DN must be specified without a value for the delegated domains.

[Figure 10–8](#) shows an Oracle Names structure that contains a root domain of acme.com and delegated domains of sales.acme.com and dev.acme.com. Database services of db, orders, and widgets reside in acme.com, sales.acme.com and dev.acme.com, respectively. The directory server's DIT has a top-level structure of o=acme, c=us that correlates to the acme.com in Oracle Names. The subtrees, ou=sales and ou=dev, correlate to the sales.acme.com and dev.acme.com delegated domains in Oracle Names.

Figure 10–8 Root and Subdomain Domain Export to Dissimilar DIT



The following syntax is used to export data from the acme.com root domain and its delegated domains to the configured DIT structure:

```
NAMESCTL> DUMP_LDAP acme.com (dn:ou,o=acme,c=us) -R -f sample.ldif
```

Note that organizationalUnitName (ou) contains no value, so that the sales and dev subdomain of acme.com in the source region can be mapped to an ou.

The following table shows how database objects in acme.com, sales.acme.com, and dev.acme.com are mapped to DNs in the directory server. All objects are created under cn=OracleContext RDNs in the directory server.

Database Object in Oracle Names	New DN in Directory Server
db.acme.com	dn:cn=db,cn=OracleContext,o=acme,c=us
orders.sales.acme.com	dn:cn=orders,cn=OracleContext,ou=sales,o=acme,c=us
widgets.dev.acme.com	dn:cn=widgets,cn=OracleContext,ou=dev,o=acme,c=us

If acme.com contained a subdomain of mktg.dept.acme.com, network objects in that subdomain would not be exported. This is because the destination DN (dn:ou,o=acme,c=us) permits only one level of delegated domains. In order to export objects from mktg.dept.acme.com, the following syntax would be required:

```
NAMESCTL> DUMP_LDAP acme.com (dn:ou,ou,o=acme,c=us) -f sample.ldif
```

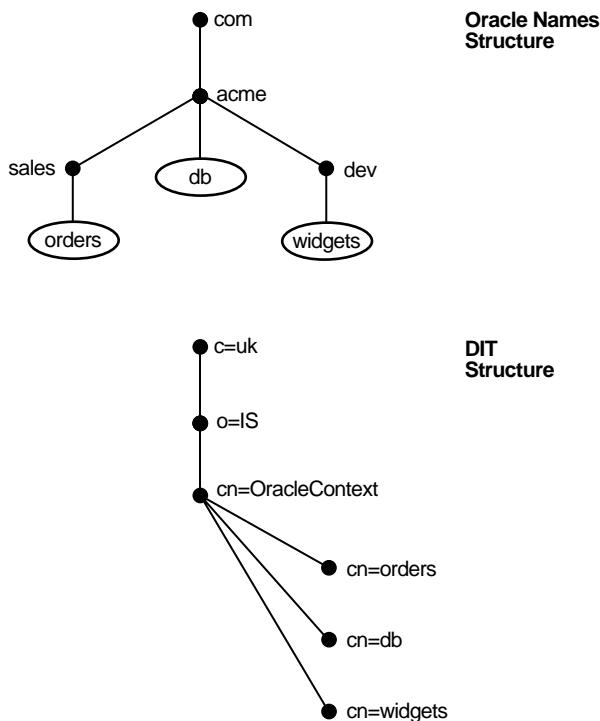
This syntax enables up to two levels of delegated domains to be exported. By adding additional wildcards, you can specify any level of depth.

Example 3: Exporting Data from Multiple Domains to One DIT Node

Data can be exported from multiple domains to one node in the destination DIT.

Figure 10–9 shows an Oracle Names structure that contains a root domain of acme.com and delegated domains of sales.acme.com and dev.acme.com. Database services of db, orders, and widgets reside in acme.com, sales.acme.com and dev.acme.com, respectively. The directory server's DIT has a structure of o=IS, c=uk that contains no subtrees that correlate to the Oracle Names delegated domains.

Figure 10–9 Root and Subdomain Domain Export to One DIT Node



All data can be exported from the root domain and the delegated domains to `cn=OracleContext, o=IS, c=uk` in the DIT with the following syntax:

```
NAMESCTL> DUMP_LDAP acme.com (dn:o=IS,c=uk) -R -f sample.ldif
```

The following table shows how database objects in acme.com, sales.acme.com, and dev.acme.com are mapped to DNs in the directory server. All objects are created under cn=OracleContext RDNs in the directory server.

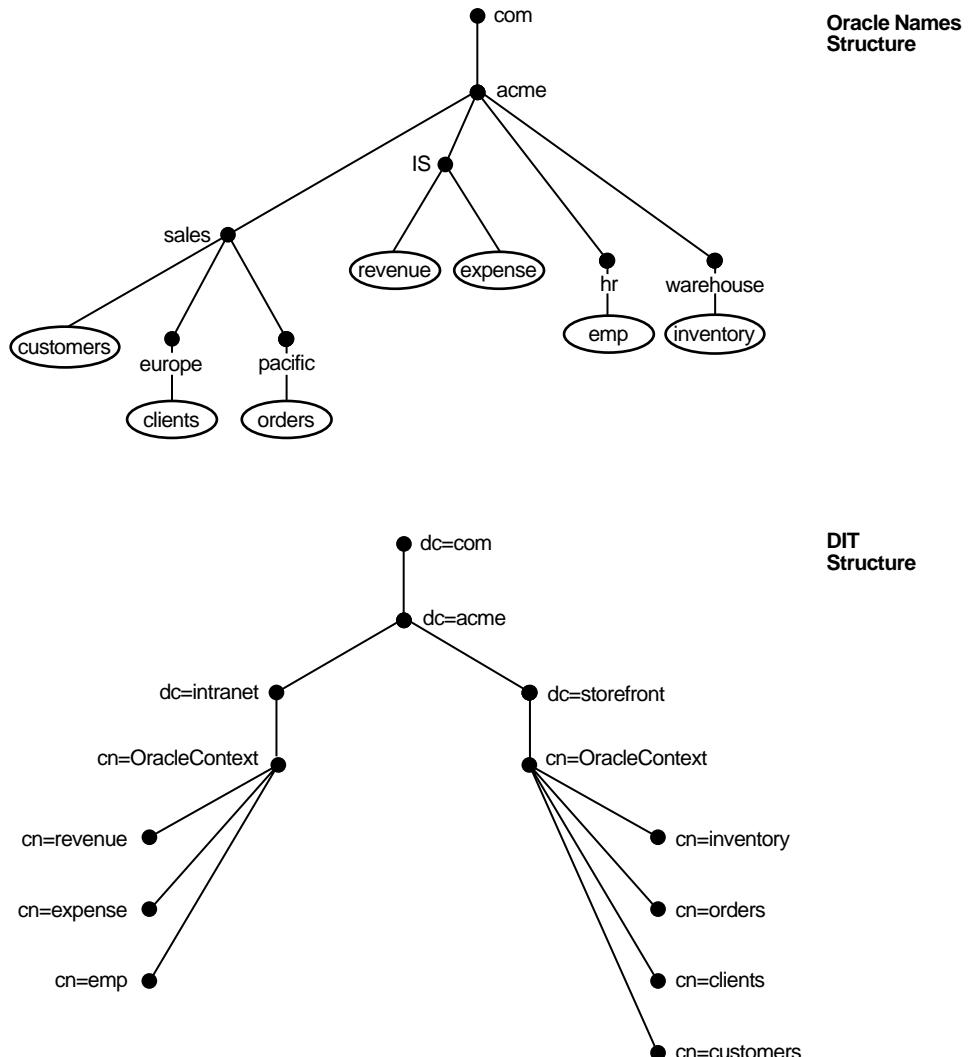
Database Object in Oracle Names	New DN in Directory Server
db.acme.com	dn:cn=db,cn=OracleContext,o=IS,c=uk
orders.sales.acme.com	dn:cn=orders,cn=OracleContext,o=IS,c=uk
widgets.dev.acme.com	dn:cn=widgets,cn=OracleContext,o=IS,c=uk

If one of the delegated domains contained a database object named db, it would not be exported. This is because the db database object's name would conflict with the db object exported from db.acme.com.

Example 4: Reorganizing a Tree Structure During an Export

In the previous examples, you saw how data can be exported to a non-DC DIT and how data can be exported from multiple domains to a one node in the DIT. This example combines these two types of exports to demonstrate how to export data to a DIT with a very different structure.

[Figure 10-10](#) on page 10-29 shows an Oracle Names structure that contains a root domain of acme.com and four delegated domains, each of which contains at least one database service object. The directory server's DIT has a top-level structure of dc=acme,dc=com that correlates to the acme.com domain in Oracle Names. The two subtrees, dc=intranet and dc=storefront, are unrelated to the delegated domains in Oracle Names.

Figure 10–10 Multiple Domain Export to Two Subtrees

In order to export data from the Oracle Names domain structure to the DIT, each domain must be exported separately:

```

DUMP_LDAP IS.acme.com (dn:dc=intranet,dc=com,dc=acme) -f sample.ldif
DUMP_LDAP hr.acme.com (dn:dc=intranet,dc=com,dc=acme) -f sample.ldif
DUMP_LDAP warehouse.acme.com (dn:dc=storefront,dc=com,dc=acme) -f sample.ldif
DUMP_LDAP sales.acme.com (dn:dc=storefront,dc=com,dc=acme) -R -f sample.ldif
  
```

The first two DUMP_LDAP commands export database objects to cn=Oraclecontext,dc=intranet,dc=acme,dc=com. The last two DUMP_LDAP commands export database objects to cn=Oraclecontext,dc=storefront,dc=acme,dc=com. The -R option in the DUMP_LDAP sales.acme.com command enables the database service objects to be exported from sales.acme.com, europe.sales.acme.com, and pacific.sales.acme.com.

The following table shows how database objects in the Oracle Names domains are mapped to DNs in the directory server. All objects are created under cn=OracleContext RDNs in the directory server.

Database Object in Oracle Names	New DN in Directory Server
revenue.is.acme.com	dn:cn=revenue,cn=OracleContext,dc=intra,dc=acme,dc=com
expense.is.acme.com	dn:cn=expense,cn=OracleContext,dc=intra,dc=acme,dc=com
emp.hr.acme.com	dn:cn=emp,cn=OracleContext,dc=intra,dc=acme,dc=com
inventory.warehouse.com	dn:cn=inventory,cn=OracleContext,dc=storefront,dc=acme,dc=com
customer.sales.acme.com	dn:cn=customer,cn=OracleContext,dc=storefront,dc=acme,dc=com
clients.europe.sales.acme.com	dn:cn=clients,cn=OracleContext,dc=storefront,dc=acme,dc=com
orders.pacific.sales.acme.com	dn:cn=orders,cn=OracleContext,dc=storefront,dc=acme,dc=com

11

Configuring Profiles

This chapter describes how to configure client and server configuration parameters in [profiles](#). A profile is a collection of parameters that specifies preferences for enabling and configuring Oracle Net features on the client or database server. A profile is stored and implemented through the `sqlnet.ora` file.

This chapter contains these topics:

- [Profile Configuration Overview](#)
- [Configuring Client Attributes for Names Resolution](#)
- [Configuring Database Access Control](#)
- [Configuring Advanced Profile Information](#)
- [Configuring Preferred Oracle Names Servers](#)
- [Configuring External Naming Methods](#)
- [Configuring Oracle Advanced Security](#)

Profile Configuration Overview

You can use a profile to:

- Specify the client domain to append to unqualified names
- Prioritize **naming methods**
- Enable logging and tracing features
- Route connections through specific processes
- Configure parameters for **external naming**
- Configure **Oracle Advanced Security**
- Use protocol-specific parameters to restrict access to the database

Profile Configuration During Installation

Oracle Universal Installer launches **Oracle Net Configuration Assistant** after software installation on the client and server. Oracle Net Configuration Assistant configures the following:

- Default network domain

The computer's network domain is automatically selected as the default domain. This domain is appended to any unqualified net service name given in the connect string, and then compared to net service names stored in a tnsnames.ora file or Oracle Names server. For example, if the client tnsnames.ora file contains a net service name of sales.us.acme.com, the user can enter the following connect string:

```
CONNECT scott/tiger@sales  
sales gets searched as sales.us.acme.com.
```

- The order of the naming methods the computer will use to resolve a **connect identifier** to a **connect descriptor**

Configuration with the Oracle Net Configuration Assistant during installation results in the following entries in the sqlnet.ora file:

```
NAMES.DEFAULT_DOMAIN=us.acme.com  
NAMES.DIRECTORY_PATH=(onames, tnsnames)
```

NAMES.DEFAULT_DOMAIN specifies the network domain, and NAMES.DIRECTORY_PATH specifies the priority order of the naming methods to use to resolve connect identifiers.

If the installed configuration is not adequate, you can use **Oracle Net Manager** to enhance the sqlnet.ora configuration.

Configuring Client Attributes for Names Resolution

The following sections describe available client configuration options:

- [Configuring a Default Domain for Clients](#)
- [Prioritizing Naming Methods](#)
- [Routing Connection Requests](#)

Configuring a Default Domain for Clients

In environments where the client often requests names from a specific domain, it is appropriate to set a default domain in the client `sqlnet.ora` file with the `NAMES.DEFAULT_DOMAIN` parameter.

When a default domain is set, it is automatically appended to any unqualified net service name given in the connect string, and then compared to net service names stored in a `tnsnames.ora` file or Oracle Names server.

For example, if the client `tnsnames.ora` file contains a net service name of `sales.us.acme.com`, the user can enter the following connect string:

```
CONNECT scott/tiger@sales
```

In this example, `sales` gets searched as `sales.us.acme.com`.

If the connect string includes the domain extension, such as in `CONNECT scott/tiger@sales.us.acme.com`, the domain is not appended. If a net service name in a `tnsnames.ora` file or Oracle Names server is not domain qualified and this parameter is set, the net service name must be entered with a dot ("."). For example, if domain is set to `us.acme.com` and the client `tnsnames.ora` file contains a net service name of `sales`, the user would enter the following connect string:

```
CONNECT scott/tiger@sales.
```

Note: During installation, Oracle Net Configuration Assistant sets the default domain to your client's network domain.

To specify a default domain:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select Naming.
4. Choose the Oracle Names tab.
5. In the Default Domain field, enter the domain.
6. Choose File > Save Network Configuration.

The `sqlnet.ora` file should contain an entry that looks like the following:

```
NAMES.DEFAULT_DOMAIN=us.acme.com
```

Prioritizing Naming Methods

After naming methods are configured, as described in [Chapter 9, "Configuring Naming Methods"](#) they must be prioritized. The naming method at the top of the list is used first to resolve a net service name. If it cannot resolve the net service name, the second method in the list is used.

To specify the order of naming methods:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select Naming.
4. Choose the Methods tab.

[Table 11-1](#) describes the naming method values listed in the Methods tab.

Table 11-1 Naming Method Values

Naming Method Value	Description
TNSNAMES	<p>Local Naming</p> <p>Net service names are resolved using the <code>tnsnames.ora</code> file which resides on the client.</p> <p>See Also: "Configuring the Local Naming Method" on page 9-5</p>
LDAP	<p>Directory Naming</p> <p>Net service names and database service names are resolved through a directory server.</p> <p>See Also: "Configuring the Directory Naming Method" on page 9-12</p>
ONAMES	<p>Oracle Names</p> <p>Net service names and database service names are resolved centrally through an Oracle Names server.</p> <p>See Also: "Configuring the Oracle Names Method" on page 9-19</p>
HOSTNAME	<p>Host Naming</p> <p>Net service names are resolved using the host naming method. Certain criteria must be met to use host naming.</p> <p>See Also: "Configuring the Host Naming Method" on page 9-56</p>

Naming Method Value	Description
CDS	<p>Cell Directory Services</p> <p>This naming method is available with Oracle Advanced Security.</p> <p>See Also: <i>Oracle Advanced Security Administrator's Guide</i></p>
NIS	<p>Network Information Service</p> <p>Service information is resolved through an existing NIS.</p> <p>See Also: "Configuring NIS" on page 9-60</p>

5. Select naming methods from the Available Methods list, and then choose the right-arrow button.
The selected naming methods move to the Selected Methods list.
6. Order the naming methods according to the order in which you want Oracle Net to try to resolve the net service name or database service name. Select a naming method in the Selected Methods list, and choose Promote or Demote to move the selection up or down in the list.
7. Choose File > Save Network Configuration.

The `sqlnet.ora` file updates with the `NAMES.DIRECTORY_PATH` parameter:

`NAMES.DIRECTORY_PATH=(ldap, tnsnames)`

Routing Connection Requests

Clients and servers acting as clients can be configured so connection requests are directed to a specific process at all times. [Table 11–2](#) describes the `sqlnet.ora` file settings that you can set to route connection requests.

Table 11–2 Routing Settings in `sqlnet.ora`

Oracle Net Manager Option	sqlnet.ora File Parameter	Description
Always Use Dedicated Server	<code>USE_DEDICATED_SERVER</code>	Use to force the listener to spawn a dedicated server for all network sessions from this client. It does this by adding <code>(SERVER=dedicated)</code> to the <code>CONNECT_DATA</code> section of connect descriptors. Connections from this client use dedicated servers, even if shared server is configured.
Prefer Connection Manager Routing	<code>USE_CMAN</code>	<p>Use to send the client to an Oracle Connection Manager whenever possible. If no Oracle Connection Manager protocol addresses are available, connections are routed through any available listener protocol address.</p> <p>Note: If you are using Oracle Connection Manager with Oracle Names, this option must be set on clients and Oracle Names servers.</p>

To route connection requests:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select General.
4. Choose the Routing tab.
5. Select the preferred way that you want connection requests routed.

See Also: [Table 11–2](#) for a description of the fields and options

6. Choose File > Save Network Configuration.

Configuring Database Access Control

You can configure the `sqlnet.ora` file to allow access to some clients and deny access to others. [Table 11–3](#) describes the available settings.

Table 11–3 Access Control Settings in `sqlnet.ora`

Oracle Net Manager Field/Option	<code>sqlnet.ora</code> File Parameter	Description
Clients Allowed to Access	<code>TCP.INVITED_NODES</code>	Use to specify which clients using the TCP/IP protocol are allowed access to the database.
Clients Excluded from Access	<code>TCP.EXCLUDED_NODES</code>	Use to specify which clients using the TCP/IP protocol are denied access to the database
Check TCP/IP Client Access Rights	<code>TCP.VALIDNODE_CHECKING</code>	<p>Use to specify whether to screen access to the database.</p> <p>If this field is selected, Oracle Net Manager checks the parameters <code>TCP.EXCLUDED_NODES</code> and <code>TCP.VALIDNODE_CHECKING</code> to determine which clients to allow access to the database. If this field is deselected, Oracle Net Manager does not screen clients.</p>

To configure database access control:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select General.
4. Choose the Access Rights tab.
5. Ensure that the checkbox Check TCP/IP Client Access Rights is selected.
6. In the two list boxes, enter either a host name or an IP address for a client that you wish to include or exclude, using commas to delimit entries placed on the same line.

Configuring Advanced Profile Information

[Table 11–4](#) describes the advanced `sqlnet.ora` file settings that you can set.

Table 11–4 Advanced Settings in `sqlnet.ora`

Oracle Net Manager Field/Option	sqlnet.ora File Parameter	Description
TNS Time Out Value	<code>SQLNET.EXPIRE_TIME</code>	<p>Use to specify a dead connection time-out value. Oracle Net sends a probe periodically to verify that a client/server connection is still active. This ensures that connections are not left open indefinitely, due to an abnormal client termination. If the probe finds a dead connection, or a connection that is no longer in use, it returns an error, causing the server process to exit.</p> <p>Limitations on using the dead connection detection feature are:</p> <ul style="list-style-type: none"> ▪ Dead connection detection is not allowed on bequeathed connections. ▪ Though very small, a probe packet generates additional traffic that may downgrade network performance. ▪ Depending on which operating system is in use, the server may need to perform additional processing to distinguish the connection probing event from other events that occur. This can also result in downgrading network performance. <p>Note: This option can be used only on the server.</p>
Client Registration ID	<code>SQLNET.CLIENT_REGISTRATION</code>	Use to specify a unique identifier for a client. This identifier is passed to the listener with any connection request. The identifier can be any string up to 128 characters long.

Oracle Net Manager Field/Option	sqlnet.ora File Parameter	Description
Turn Off UNIX Signal Handling	BEQUEATH_DETACH	<p>Use to turn on or off UNIX signal handling.</p> <p>Since the client application spawns a server process internally through the Bequeath protocol as a child process, the client application becomes responsible for cleaning up the child process when it completes. When the server process completes its connection responsibilities, it becomes a defunct process. Signal handlers are responsible for cleaning up these defunct processes. Setting this parameter configures the client profile to pass this process to the UNIX init process by disabling signal handlers.</p>
Disable Out-of-Band Break	DISABLE_OOB	<p>Use to turn on or off out of band breaks.</p> <p>If deselected or set to off, enables Oracle Net to send and receive "break" messages using urgent data provided by the underlying protocol.</p> <p>If selected or set to on, disables the ability to send and receive "break" messages using urgent data provided by the underlying protocol. Once enabled, this feature applies to all protocols used by this client.</p> <p>See Also: Operating system documentation to determine if the protocol you are using supports urgent data requests. TCP/IP is an example of a protocol that supports this feature.</p>

To set advanced features for clients:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select General.
4. Choose the Advanced tab.
5. Enter the values for the field(s)/option(s) you want to set.

See Also: [Table 11-4](#) for a description of the fields and options

6. Choose File > Save Network Configuration.

Configuring Preferred Oracle Names Servers

If you are using Oracle Names as a naming method, you can specify the name and address of **preferred Oracle Names servers** to take precedence over any other available Oracle Names server addresses. Oracle Net routes name resolution requests to each preferred Oracle Names Server until a response is received.

Preferred Oracle Names servers are an alternative to using the Discover Oracle Names Servers command in Oracle Net Manager or using the Oracle Names Control utility REORDER_NS command. These commands create a list of Oracle Names servers based on what is available throughout the network and ranks them in the order of fastest response time. Preferred Oracle Names servers override any other Oracle Names servers found during the discovery process. Once you have initially discovered an Oracle Names server, you may want to delete preferred Oracle Names servers.

See Also:

- ["Differences Between Versions of Oracle Names" on page 3-30](#) for a description of how Oracle Names server lists are created
- ["Task 4: Configure Clients and Database Servers To Use Oracle Names Servers" on page 9-34](#) for information about discovering Oracle Names servers

To specify a preferred Oracle Names server:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select Preferred Oracle Names Servers.
4. Choose New.

A Preferred Server tab appears.

5. Choose the protocol and enter the requested protocol address information for an Oracle Names server configured on that address.

See Also: *Oracle Net Services Reference Guide* for protocol parameter settings

6. Repeat Steps 4 and 5 for each additional Oracle Names server you want added to the list.

7. Choose File > Save Network Configuration.

The `sqlnet.ora` file updates with the `NAMES.PREFERRED_SERVERS` parameter:

```
NAMES.PREFERRED_SERVERS=
(ADDRESS=(PROTOCOL=tcp)(HOST=namesrv1)(PORT=1575))
```

Note: The preferred Oracle Names servers must match the Oracle Names listening protocol addresses configured in the `names.ora` file with the `NAMES.ADDRESSES` parameter.

Configuring External Naming Methods

Configure required client parameters needed for external naming methods, including **Cell Directory Services (CDS)** and **Network Information Service (NIS)**, in the profile. [Table 11–5](#) describes the `sqlnet.ora` file external naming settings that you can set.

Table 11–5 *External Naming Methods Settings in sqlnet.ora*

Oracle Net Manager Field	sqlnet.ora File Parameter	Description
Cell Name	<code>NAMES.DCE.PREFIX</code>	Enter a valid DCE cell name (prefix).
Meta Map	<code>NAMES.NIS.META_MAP</code>	Specify the map, a special file that contains the database service name.

To configure external naming method parameters:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select Naming.
4. Enter a value in the appropriate field for the external naming method you are using.
5. Choose File > Save Network Configuration.

Configuring Oracle Advanced Security

Oracle Advanced Security enables data encryption and integrity checking, enhanced authentication, single sign-on, and support for Distributed Computing Environment (DCE). Oracle Advanced Security also provides centralized user management on LDAP-compliant directory servers and certificate-based single sign-on; this functionality relies on the **Secure Sockets Layer (SSL)**.

To configure a client or server to use Oracle Advanced Security features:

1. Start Oracle Net Manager.

See Also: "Starting Oracle Net Manager" on page 5-3

2. In the navigator pane, expand Local > Profile.
3. From the list in the right pane, select Oracle Advanced Security.

Each Oracle Advanced Security tab page enables you to configure a separate set of parameters.

See Also:

- Choose the Help button on the particular tab page
- Oracle Advanced Security procedural topics in the Oracle Net Manager online help. To access these topics in the online help, choose Oracle Advanced Security > How To in the online help
- *Oracle Advanced Security Administrator's Guide* for further information about configuration

4. Choose or edit options as applicable.
5. Choose File > Save Network Configuration.

Configuring and Administering the Listener

The **listener** is a separate process that runs on the database server computer. It receives incoming client connection requests and manages the traffic of these requests to the database server. This chapter describes how to configure the listener to accept client connections.

This chapter contains these topics:

- [Listener Configuration Overview](#)
- [Listener Configuration During Installation](#)
- [Customizing Listener Configuration](#)
- [Listener Administration](#)

See Also:

- [Chapter 2, "Connectivity Concepts"](#) for a description of how the listener is used during an initial connection request
- [Chapter 4, "Architecture of Oracle Net Services"](#) for an architectural overview of the listener

Listener Configuration Overview

Note: A release 9.0 listener is required for an Oracle9*i* database. Previous versions of the listener are not supported for use with an Oracle9*i* database. However, it is possible to use a release 9.0 listener with previous versions of the Oracle database.

A listener is configured with one or more listening protocol addresses, information about supported services, and parameters that control its runtime behavior. The listener configuration is stored in a configuration file named `listener.ora`.

Because all of the configuration parameters have default values, it is possible to start and use a listener with no configuration. This default listener has a name of `LISTENER`, supports no services upon startup, and listens on the following TCP/IP protocol address:

```
(ADDRESS=(PROTOCOL=tcp)(HOST=host_name)(PORT=1521))
```

Supported services, that is, the services to which the listener forwards client requests, can be configured in the `listener.ora` file or this information can be dynamically registered with the listener. This dynamic registration feature is called **service registration** and is used by Oracle9*i* or Oracle8*i* instances. The registration is performed by the **PMON process**—an instance background process—of each database instance that has the necessary configuration in the database **initialization parameter file**. Dynamic service registration does not require any configuration in the `listener.ora` file.

See Also: ["Listener Architecture" on page 4-10](#)

Service registration offers the following benefits:

- Simplified configuration

Service registration reduces the need for the `SID_LIST_listener_name` parameter setting, which specifies information about the databases served by the listener, in the `listener.ora` file.

Note: The `SID_LIST_listener_name` parameter is still required if you are using Oracle Enterprise Manager to manage the database.

- Connect-time failover

Because the listener always knows the state of the instances, service registration facilitates automatic failover of the client connect request to a different instance if one instance is down.

In a static configuration model, a listener would start a dedicated server upon receiving a client request. The server would later find out that the instance is not up, causing an "Oracle not available" error message.

- Connection load balancing

Service registration enables the listener to forward client connect requests to the least loaded instance and **dispatcher** or **dedicated server**. This balances the load across the **service handlers** and nodes.

Note that Oracle Enterprise Manager tools require static service configuration in the `listener.ora` file.

See Also:

- "[Listener Architecture](#)" on page 4-10 for an architectural overview of service registration
- "[Configuring Service Registration](#)" on page 12-14
- "[Configuring Address List Parameters](#)" on page 15-4
- "[Configuring Connection Load Balancing](#)" on page 15-10

Listener Configuration During Installation

Oracle Universal Installer launches **Oracle Net Configuration Assistant** during software installation. Oracle Net Configuration Assistant enables you to configure the listening protocol address and service information for an Oracle database.

During an Enterprise Edition or Standard Edition installation on the database server, Oracle Net Configuration Assistant automatically configures a listener with a name of LISTENER that has a TCP/IP listening protocol address for the Oracle database. During a Custom installation, Oracle Net Configuration Assistant prompts you to configure a listener name and a protocol address of your choice.

Additionally, a listening IPC protocol address for **external procedures**—functions that can be called from PL/SQL code—is automatically configured, regardless of the installation type.

Oracle Net Configuration Assistant also automatically configures service information for the Oracle database and the external procedures in the `listener.ora` file.

Note: While service information is not required in the `listener.ora` file, Oracle Net Configuration Assistant assumes that you are using Oracle Enterprise Manager to monitor the database. Oracle Enterprise Manager requires this information to discover the database.

Figure 12–1 shows a `listener.ora` file sample. The `LISTENER` entry defines the listening protocol address for a listener named `LISTENER`, and the `SID_LIST_LISTENER` entry provides information about the services supported by the listener `LISTENER`.

Figure 12–1 Example `listener.ora` File

```
LISTENER=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sale-server)(PORT=1521))
      (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))))
SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (GLOBAL_DBNAME=sales.us.acme.com)
      (ORACLE_HOME=/oracle90)
      (SID_NAME=sales))
    (SID_DESC=
      (SID_NAME=plsextproc)
      (ORACLE_HOME=/oracle90)
      (PROGRAM=extproc)))
```

See Also: *Oracle Net Services Reference Guide* for further information about identifying listeners by unique names and creating multiple listener entries in the `listener.ora` file

Customizing Listener Configuration

If the default or installed configuration is not adequate for a particular environment, you can use **Oracle Net Manager** to customize the `listener.ora` configuration.

This section contains these configuration topics:

- [Configuring Listening Protocol Addresses](#)
- [Configuring Protocol Addresses for Oracle9i JVM](#)
- [Configuring Static Service Information](#)
- [Configuring Password Authentication for the Listener](#)

Configuring Listening Protocol Addresses

To configure additional protocol addresses for the listener:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Local > Listeners.
3. Select a listener.

Note: If a `listener.ora` file has been configured, but the listener has never been started with the Listener Control utility, the Listeners folder does not display a listener. If this occurs, exit Oracle Net Manager, start the listener, as described in ["Starting Oracle Net Services Components" on page 16-2](#), and start Oracle Net Manager again.

If a listener has never been created:

- a. Choose plus (+) from the toolbar or select Edit > Create.
 - b. Enter a unique listener name in the Choose Listener Name dialog box.
4. From the list in the right pane, select Listening Locations.
 5. Choose Add Address.

A new Address tab appears.

6. Select a protocol, and then enter the appropriate parameter information for the selected protocol in the fields provided.

See Also: *Oracle Net Services Reference Guide* for further information about protocol addresses and parameters

When configuring the listener to listen on TCP/IP, you should enter the default port of 1521. If you do not, you must configure the LOCAL_LISTENER parameter in the **initialization parameter file** and resolve the listener name through a naming method.

See Also: ["Registering Information with a Nondefault Listener"](#) on page 12-15

If the computer has more than one IP address and you want the listener to listen on all available IP addresses, configure TCP/IP or TCP/IP with SSL and enter the host name of the computer in the Host field.

7. Repeat Steps 5 and 6 for additional protocols.
8. Choose File > Save Network Configuration.

Configuring Protocol Addresses for Oracle9i JVM

Connections to **Oracle9i JVM** (formerly named Oracle JServer in release 8.1) require the TCP/IP or TCP/IP with SSL listening protocol addresses with presentation information.

If the database is release 8.1 or earlier, configure protocol addresses statically, using the following procedure, even if a release 9.0 listener is used. If both listener and database are release 9.0, this procedure is unnecessary because configuration occurs dynamically during service registration.

To configure a protocol address for Oracle JServer release 8.1:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Local > Listeners.
3. Select a listener.
4. From the list in the right pane, select Listening Locations.
5. Choose Add Address.
A new Address tab appears.
6. Select the TCP/IP or TCP/IP with SSL protocol from the Protocol list.
7. Enter the host name of the database in the Host field.
8. Enter port 2481 if the selected protocol is TCP/IP in the Port field, or enter port 2482 if the selected protocol is TCP/IP with SSL in the Port field.
9. Choose "Statically dedicate this address for JServer connections".
10. Choose File > Save Network Configuration.

The `listener.ora` file updates with the following:

```
listener=
  (DESCRIPTION_LIST=
    (DESCRIPTION=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=2481))
      (PROTOCOL_STACK=
        (PRESENTATION=giop)
        (SESSION=raw))))
```

See Also:

- *Oracle9i Enterprise JavaBeans Developer's Guide and Reference* for information on configuring client connections to EJB and CORBA applications
- *Oracle9i Java Stored Procedures Developer's Guide* for information on configuring client connections to Java stored procedures

Handling Large Volumes of Concurrent Connection Requests

If you expect the listener to handle large volumes of concurrent connection requests, you may need to specify a listen queue size for its listening endpoints.

To specify a queue size for a protocol address, enter a value to the QUEUESIZE parameter at the end of the protocol address. For example:

Figure 12–2 listener.ora File with Queue Size

```
LISTENER=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)(QUEUESIZE=20))
```

Note: Currently, you can configure only the queue size for listeners operating on TCP/IP. The default queue size is system specific. On Solaris, the default queue size is set to 5. The queue size is 5 for Windows NT 4.0 Workstation and 50 for Windows NT 4.0 Server.

Configuring Static Service Information

In order for the listener to accept client connect requests to an Oracle8 release 8.0 or Oracle7 database, the `listener.ora` file must be configured with information about the database. Static configuration is also required for other services, such as **external procedures**, **Heterogeneous Services**, and some management tools, including Oracle Enterprise Manager.

For Oracle9*i* and Oracle8*i* databases, the listener uses the dynamic information about the database and instance it has received through service registration before using statically configured information.

Table 12–1 describes static service parameters that you can set in the `listener.ora` file.

Table 12–1 Service Settings in *listener.ora*

Oracle Net Manager Field	listener.ora File Parameter	Description
SID	SID_NAME	The Oracle System Identifier (SID) identifies the instance. You can obtain the SID value from the INSTANCE_NAME parameter in the initialization parameter file. Note: This setting is required on UNIX and Windows NT.
Global Database Name	GLOBAL_DBNAME	The global database name is comprised of the database name and database domain name. You can obtain the GLOBAL_DBNAME value from the SERVICE_NAMES parameter, or from the DB_NAME and DB_DOMAIN parameters in the initialization parameter file.
Oracle Home Directory	ORACLE_HOME	The Oracle Home directory identifies the Oracle home location of the database that you are specifying. Note: This setting is required on UNIX.

Important: If you are using **connect-time failover** or **Transparent Application Failover (TAF)**, such as in an Oracle9*i* Real Application Cluster environment, Oracle Corporation recommends not setting the GLOBAL_DBNAME parameter.

See Also:

- [Chapter 15, "Enabling Advanced Features for Oracle Net Services"](#) for more information about statically configuring the listener for external procedures and heterogeneous services
- [Oracle Enterprise Manager Configuration Guide](#) for further information about Oracle Enterprise Manager

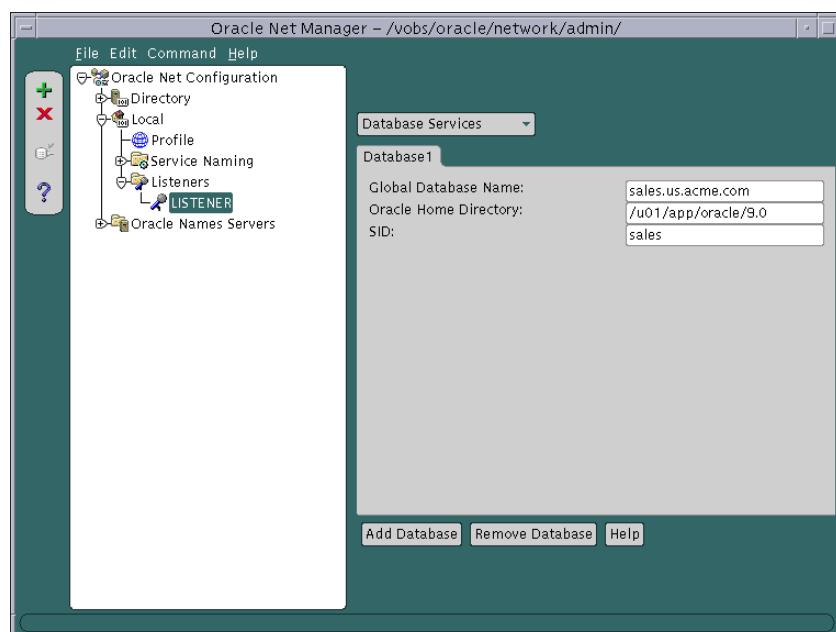
To statically configure the listener:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

2. In the navigator pane, expand Local > Listeners.
3. Select a listener.
4. From the list in the right pane, select Database Services.
5. Choose Add Database.

A new Database tab appears.



See Also: [Table 12-1](#) on page 12-10 for a description of the fields

6. Enter the global database name, location of the Oracle home directory, and the Oracle System Identifier (SID) of the instance in the appropriate fields.
7. Choose File > Save Network Configuration.

For example, an excerpt of a `listener.ora` file statically configured for a database service called `sales.us.acme.com` looks like this:

```
SID_LIST_listener=
(SID_LIST=
(SID_DESC=
(GLOBAL_DBNAME=sales.us.acme.com)
(SID_NAME=sales)
(ORACLE_HOME=/u01/app/oracle/9.0)))
```

Configuring Password Authentication for the Listener

Note: If you are administering the listener remotely over an insecure network and require maximum security, configure the listener with a secure protocol address that used the [TCP/IP with SSL protocol](#). If the listener has multiple protocol addresses, ensure that the TCP/IP with SSL protocol address is listed first in the `listener.ora` file.

It is important to provide security through a password for the listener. With a password, privileged operations, such as saving configuration changes or stopping the listener, used from the Listener Control utility will require a password.

Use the Listener Control utility's `CHANGE_PASSWORD` command or the Oracle Net Manager to set or modify an encrypted password in the `PASSWORDS_listener_name` parameter in the `listener.ora` file. If the `PASSWORDS_listener_name` parameter is set to an unencrypted password, you must manually remove it from the `listener.ora` file prior to modifying it. If the unencrypted password is not removed, you will be unable to successfully set an encrypted password.

To set or modify an encrypted password with Oracle Net Manager:

1. In the navigator pane, expand Oracle Net Configuration > Local > Listeners.
2. Select a listener.
3. From the list in the right pane, select General Parameters.
4. Select the Authentication tab.
5. Choose Require a Password for Listener Operations.
6. Enter the password in the Password field, and re-enter it in the Confirm Password field.
7. Choose File > Save Network Configuration.

To set a new encrypted password with the `CHANGE_PASSWORD` command, issues the following commands from the Listener Control utility:

```
LSNRCTL> CHANGE_PASSWORD
Old password: <enter>
New password: password
Reenter new password: password
LSNRCTL> SAVE_CONFIG
```

Bold denotes user input.

Note that the password is not displayed when entered.

To modify an encrypted password with the CHANGE_PASSWORD command:

```
LSNRCTL> CHANGE_PASSWORD  
Old password: password  
New password: password  
Reenter new password: password  
LSNRCTL> SAVE_CONFIG
```

See Also: *Oracle Net Services Reference Guide* for further information about the CHANGE_PASSWORD command

Configuring Service Registration

Dynamic service registration is configured in the database initialization file. It does not require any configuration in the `listener.ora` file. However, listener configuration must be synchronized with the information in the database initialization file.

This section contains the following configuration topics related to service registration:

- [Configuring Service Registration](#)
- [Registering Information with the Default, Local Listener](#)
- [Registering Information with a Nondefault Listener](#)
- [Registering Information with a Remote Listener](#)
- [Configuring a Naming Method](#)

Configuring Service Registration

To ensure service registration works properly, the initialization parameter file should contain the following parameters:

- `SERVICE_NAMES` for the database service name
- `INSTANCE_NAME` for the instance name

For example:

```
SERVICE_NAMES=sales.us.acme.com  
INSTANCE_NAME=sales
```

The `SERVICE_NAMES` defaults to the [global database name](#), which is comprised of the values from the `DB_NAME` and `DB_DOMAIN` parameters in the initialization parameter file.

See Also: *Oracle9i Database Reference* for further information about the `SERVICE_NAMES` and `INSTANCE_NAME` parameters

Registering Information with the Default, Local Listener

By default, the PMON process registers service information with its local listener on the default local address of TCP/IP, port 1521. As long as the listener configuration is synchronized with the database configuration, PMON can register service information with a nondefault local listener or a remote listener on another node. Synchronization is simply a matter of specifying the protocol address of the listener in the `listener.ora` file and the location of the listener in the initialization parameter file.

Registering Information with a Nondefault Listener

If you want PMON to register with a local listener that does not use TCP/IP, port 1521, configure the `LOCAL_LISTENER` parameter in the initialization parameter file to locate the local listener. If you are using shared server, you can also use the `LISTENER` attribute of the `DISPATCHERS` parameter in the initialization parameter file to register the dispatchers with a nondefault local listener.

Note: The `LISTENER` attribute overrides the `LOCAL_LISTENER` parameter.

Set the `LOCAL_LISTENER` parameter as follows:

```
LOCAL_LISTENER=listener_alias
```

Set the `LISTENER` attribute as follows:

```
DISPATCHERS="(PROTOCOL=tcp)(LISTENER=listener_alias)"
```

`listener_alias` is then resolved to the listener protocol addresses through a naming method, such as a `tnsnames.ora` file on the database server.

For example, if the listener is configured to listen on port 1421 rather than port 1521, you can set the `LOCAL_LISTENER` parameter in the initialization parameter file as follows:

```
LOCAL_LISTENER=listener1
```

Using the same listener example, you can set the `LISTENER` attribute as follows:

```
DISPATCHERS="(PROTOCOL=tcp)(LISTENER=listener1)"
```

You can then resolve `listener1` in the local `tnsnames.ora` as follows:

```
listener1=
(DESCRIPTION=
(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1421)))
```

To register information with another local listener:

1. Configure the `listener.ora` file with the protocol address of the local listener.

See Also: ["Configuring Listening Protocol Addresses" on page 12-6](#)
2. Configure the `LOCAL_LISTENER` parameter in the initialization parameter file to locate the local listener. If you are using shared server, you can also use the `LISTENER` attribute of the `DISPATCHERS` parameter in the initialization parameter file.
3. Resolve the listener name alias for the `LOCAL_LISTENER` or the `LISTENER` setting through a `tnsnames.ora` file or an Oracle Names server.

See Also: ["Configuring a Naming Method" on page 12-18](#)

Registering Information with a Remote Listener

Registration to remote listeners, such as in the case of Oracle9*i* Real Application Clusters, can be configured for shared server and dedicated server environments.

In a shared server environment, configure the `LISTENER` attribute of the `DISPATCHERS` parameter or the `REMOTE_LISTENER` parameter in the initialization parameter file. In a dedicated server environment, configure the `REMOTE_LISTENER` parameter in the database initialization parameter file.

Note: The `LISTENER` attribute overrides the `REMOTE_LISTENER` parameter.

Set the `REMOTE_LISTENER` parameter as follows:

```
REMOTE_LISTENER=listener_alias
```

Set the `LISTENER` attribute as follows:

```
DISPATCHERS="(PROTOCOL=tcp)(LISTENER=listener_alias)"
```

`listener_alias` is then resolved to the listener protocol addresses through a naming method, such as a `tnsnames.ora` file on the database server.

For example, if two remote listeners are configured on remote servers sales1-server and sales2-server to listen on port 1521, you can set the REMOTE_LISTENER parameter in the initialization file for the instance on host sales1-server as follows:

```
REMOTE_LISTENER=listener_sales2
```

You can set the REMOTE_LISTENER parameter in the initialization file for the instance on host sales2-server as follows:

```
REMOTE_LISTENER=listener_sales1
```

You can then resolve listener_sales2 in the local tnsnames.ora on sales1-server as follows:

```
listener_sales2=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
```

Likewise, you can resolve listener_sales1 in the local tnsnames.ora on sales2-server as follows:

```
listener_sales1=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521)))
```

If shared server is supported, you can set the DISPATCHERS parameter in the initialization parameter file as follows:

```
DISPATCHERS=" (PROTOCOL=tcp) (LISTENER=listeners_sales)"
```

You can then resolve listener_sales in the local tnsnames.ora as follows:

```
listeners_sales=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521))
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
```

To register information with remote listener:

1. Configure the `listener.ora` file with the protocol addresses of the remote listeners.

See Also: ["Configuring Listening Protocol Addresses"](#) on page 12-6

2. In a shared server environment, configure the `LISTENER` attribute of the `DISPATCHERS` parameter or the `REMOTE_LISTENER` parameter in the initialization parameter file. In a dedicated server environment, configure the `REMOTE_LISTENER` parameter in the database initialization parameter file.
3. Resolve the listener name alias for the `LISTENER` or the `REMOTE_LISTENER` setting through a `tnsnames.ora` file or an Oracle Names server.

See Also: ["Configuring a Naming Method"](#)

Configuring a Naming Method

The listener name alias specified for the `LOCAL_LISTENER` parameter, `REMOTE_LISTENER` parameter, or `LISTENER` attribute can be resolved through one of the following naming methods:

- [tnsnames.ora File](#)
- [Oracle Names Server](#)

tnsnames.ora File A net service name entry should be created for the protocol address without the `CONNECT_DATA` section of the connect descriptor. Oracle Net Manager does not enable you to configure a `tnsnames.ora` file without the `CONNECT_DATA` information. Therefore, Oracle Corporation recommends that you manually modify the `tnsnames.ora` file.

For example, if `LOCAL_LISTENER` is set to `listener1` and `listener1` uses TCP/IP on port 1421, the entry in the `tnsnames.ora` file would be:

```
listener1=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1421))
```

Note: Multiple addresses are supported, but connect-time failover and client load balancing features are not supported.

See Also: [Chapter 15, "Enabling Advanced Features for Oracle Net Services"](#) for further information about multiple address configuration

Oracle Names Server An entry for the listener alias name can be made to an Oracle Names server through Oracle Net Manager:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

2. In the navigator pane, expand Oracle Names Servers.
3. Select the Oracle Names server.
4. From the list in the right pane, select Manage Data.
5. Choose the Advanced tab.
6. Choose Add, enter the listener alias name in the Name field, A . SMD record type in the Type field, and enter the address in the Value field, for example:
`(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=2481))`
7. Choose File > Save Network Configuration.

Listener Administration

Once the listener is configured, the listener can be administered with the Listener Control utility. This section describes some of the common administrative tasks for the listener, including the following topics:

- [Starting and Stopping the Listener](#)
- [Monitoring Runtime Behavior](#)
- [Monitoring Log Files](#)

See Also: [Oracle Net Services Reference Guide](#) for a complete listing of all the Listener Control utility commands and the Oracle Net Manager online help contents

Starting and Stopping the Listener

STOP Command

To stop the listener from the command line, enter:

```
lsnrctl STOP [listener_name]
```

where *listener_name* is the name of the listener defined in the `listener.ora` file. It is not necessary to identify the listener if you are using the default listener, named LISTENER.

START Command

To start the listener from the command line, enter:

```
lsnrctl START [listener_name]
```

where *listener_name* is the name of the listener defined in the `listener.ora` file. It is not necessary to identify the listener if you are using the default listener, named LISTENER.

In addition to starting the listener, the Listener Control utility verifies connectivity to the listener.

Monitoring Runtime Behavior

The STATUS and SERVICES commands provide information about the listener. When entering these commands, follow the syntax as shown for the STOP and START commands.

STATUS Command

The STATUS command provides basic status information about a listener, including a summary of listener configuration settings, the listening protocol addresses, and a summary of services registered with the listener.

Note: You can also obtain the status of a listener through the Oracle Enterprise Manager Console. See the *Oracle Enterprise Manager Administrator's Guide* for further information.

The STATUS command generates output with the sections described in [Table 12–2](#).

Table 12–2 Listener Control Utility STATUS Command

Output Section	Description
STATUS of the LISTENER	<p>Specifies the following:</p> <ul style="list-style-type: none"> ■ Name of the listener ■ Version of listener ■ Start time and up time ■ Tracing level ■ Logging and tracing configuration settings ■ listener.ora file being used ■ Whether a password is set in listener.ora file ■ Whether the listener can respond to queries from an SNMP-based network management system
Listening Endpoints Summary	Lists the protocol addresses the listener is configured to listen on
Services Summary	Displays a summary of the services registered with the listener and the service handlers allocated to each service
Service	Identifies the registered service
Instance	<p>Specifies the name of the instance associated with the service along with its status and number of service handlers associated with the service</p> <p>Status can be one of the following:</p> <ul style="list-style-type: none"> ■ A READY status means that the instance can accept connections. ■ A BLOCKED status means that the instance cannot accept connections. ■ A READY/SECONDARY status means that this is a secondary instance in an Oracle9i Real Application Clusters primary/secondary configuration and is ready to accept connections. ■ An UNKNOWN status means that the instance is registered statically in the listener.ora file rather than dynamically with service registration. Therefore, the status is non known.

Figure 12–3 shows example output of the STATUS command.

Figure 12–3 Listener Control Utility's STATUS Command Output

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=net)))
STATUS of the LISTENER
-----
Alias                      LISTENER
Version                    TNSLSNR for Solaris: Version 9.0.1.0.0 -
Development
Start Date                 15-SEP-2000 20:22:00
Uptime                     0 days 0 hr. 5 min. 22 sec
Trace Level                support
Security                   OFF
SNMP                       OFF
Listener Parameter File   /oracle/network/admin/listener.ora
Listener Log File          /oracle/network/log/listener.log
Listener Trace File        /oracle/network/trace/listener.trc
Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=net)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=sales-server)(PORT=2484))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=2481))(PRESENTATION
=GIOP))
Services Summary...
Service "sales.us.acme.com" has 1 instance(s).
    Instance "sales", status READY, has 3 handler(s) for this service...
Service "hr.us.acme.com" has 1 instance(s).
    Instance "hr", status READY, has 2 handler(s) for this service...
The command completed successfully
```

SERVICES Command

The SERVICES command provides detailed information about the services and instances registered and the service handlers allocated to each instance.

The SERVICES command generates output with the sections described in [Table 12–3](#).

Table 12–3 Listener Control Utility SERVICES Command

Output Section	Description
Service	Identifies the registered service
Instance	<p>Specifies the name of the instance associated with the service</p> <p>The status field indicates if the instance is able to accept connections.</p> <ul style="list-style-type: none"> ▪ A READY status means that the instance can accept connections. ▪ A BLOCKED status means that the instance cannot accept connections. ▪ A READY/SECONDARY status means that this is a secondary instance in an Oracle9i Real Application Cluster primary/secondary configuration and is ready to accept connections. ▪ An UNKNOWN status means that the instance is registered statically in the listener.ora file rather than dynamically with service registration. Therefore, the status is non known.
Handlers	<p>Identifies the name of the service handler. Dispatchers are named D000 through D999. Dedicated servers have a name of DEDICATED.</p> <p>This section also identifies the following about the service handler:</p> <ul style="list-style-type: none"> ▪ established—The number of client connections this service handler has established ▪ refused—The number of client connections it has refused ▪ current—The number of client connections it is handling, that is, its current load ▪ max—The maximum number of connections for the service handler, that is, its maximum load ▪ state—The state of the handler: <ul style="list-style-type: none"> - A READY state means that the service handler can accept new connections. - A BLOCKED state means that the service handler cannot accept new connections. <p>Following this, additional information about the service handler displays, such as whether the service handler is a dispatcher, a local dedicated server, or a remote dedicated server on another node.</p>

Figure 12–4 shows example output of the SERVICES command.

Figure 12–4 Listener Control Utility's SERVICES Command Output

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=net) ))
Services Summary...
Service "sales.us.acme.com" has 1 instance(s).
  Instance "sales", status READY, has 3 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
        LOCAL SERVER
      "D000" established:0 refused:0 current:0 max:10000 state:ready
        DISPATCHER <machine: sales-server, pid: 1689>
          (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=52414))
      "D001" established:0 refused:0 current:0 max:10000 state:ready
        DISPATCHER <machine: sales-server, pid: 1691>
          (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=52415))
Service "hr.us.acme.com" has 1 instance(s).
  Instance "hr", status READY, has 2 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
        LOCAL SERVER
      "D000" established:0 refused:0 current:0 max:10000 state:ready
        DISPATCHER <machine: sales-server, pid: 11326>
          (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=58361))
The command completed successfully
```

This output shows that two database services, `sales.us.acme.com` and `hr.us.acme.com`, are registered with the listener.

Client connection requests to `sales.us.acme.com` are handled by two dispatchers named `D000` and `D001` and one dedicated server. All handlers have a status of `ready`, indicating that they are ready to receive connections.

Client connection requests to `hr.us.acme.com` are handled by one dispatcher named `D001` and one dedicated server.

Monitoring Log Files

When you notice any of the following conditions, review and monitor the listener log file for the following:

- Long connection establishment times
- Connectivity problems and refusals
- Unexpected shutdown of the listener that could indicate a denial-of-service attack

See Also: ["Analyzing Listener Log Files" on page 17-30](#)

Configuring Oracle Connection Manager

This chapter describes how to configure Oracle Connection Manager features.

This chapter contains these topics:

- [Oracle Connection Manager Configuration Overview](#)
- [Configuring Oracle Connection Manager](#)
- [Enabling Oracle Connection Manager Features](#)

Note: Oracle Connection Manager is available for installation with Oracle9*i* Enterprise Edition.

See Also:

- [Chapter 1, "Networking Challenges in the Internet Age"](#) for an introductory level overview of Oracle Connection Manager concepts
- [Chapter 4, "Architecture of Oracle Net Services"](#) for an architectural overview of Oracle Connection Manager

Oracle Connection Manager Configuration Overview

Oracle Connection Manager is a **proxy server**, an intermediate server that forwards connection requests to the database server or to other proxy servers. It has three primary functions:

- Session multiplexing
- Access control
- Protocol conversion

Session multiplexing—You can quickly configure Oracle Connection Manager to funnel multiple client sessions through a network connection to a **shared server** destination.

Access control—You can use rule-based configuration to filter out certain client requests and accept others.

Protocol conversion—Oracle Connection Manager accepts connection requests in any protocol that Oracle supports.

Configuring Oracle Connection Manager

Configuring Oracle Connection Manager is a four-part process:

1. Configure the `cman.ora` file on the Oracle Connection Manager computer. This file specifies listening endpoints for the server, the route path for Oracle Connection Manager, access control rules, and Oracle Connection Manager performance parameters.
2. Configure clients with the protocol addresses of Oracle Connection Manager and the listener.
3. (Optional) Configure the database server for session multiplexing.

This section contains these topics:

- [Configuring the Oracle Connection Manager Computer](#)
- [Configuring Clients for Oracle Connection Manager](#)
- [Configuring the Oracle Database Server for Oracle Connection Manager](#)

Configuring the Oracle Connection Manager Computer

Note: Oracle Net Manager does not support configuration of the cman.ora file, so changes must be made manually.

To configure the Oracle Connection Manager computer, you can define four types of parameters in the cman.ora file:

- Listening endpoint list for the CMGW gateway process
- Listening endpoint list for the CMADMIN administrative process
- Access control rule list
- Parameter list

The cman.ora file is located in the \$ORACLE_HOME/network/admin directory on UNIX, and in *ORACLE_HOME*\network\admin directory on Windows NT.

[Figure 13–1](#) shows an example cman.ora file.

Figure 13–1 Example cman.ora File

```
CMAN=
  (ADDRESS=(PROTOCOL=tcp)(HOST=proxysvr)(PORT=1630))
  (ADDRESS=(PROTOCOL=tcps)(HOST=144.25.22.217)(PORT=2484))
CMAN_ADMIN=
  (ADDRESS=(PROTOCOL=tcp)(HOST=proxysvr)(PORT=1830))
CMAN_RULES=
  (RULE=(SRC=206.62.226.32/27)(DST=dlsun1593)(SRV=*)(ACT=accept)))
CMAN_PROFILE=
  (PARAMETER_LIST=
    (LOG_LEVEL=2)
    (TRACING=on))
```

Listening Endpoint Lists (CMAN and CMAN_ADMIN)

The listening endpoint list specifies protocol addresses for the gateway and administrative processes. The gateway process, CMGW, uses the CMAN parameter and the administrative process, CMADMIN, uses the CMAN_ADMIN parameter.

Both processes can be configured with multiple protocol addresses. In [Figure 13-1](#) on page 13-3, the CMGW process has two protocol addresses. The first address shown is the default listening protocol address of TCP/IP, port 1630:

```
(ADDRESS=(PROTOCOL=tcp) (HOST=proxysvr) (PORT=1630))
```

The second is the protocol address of Oracle Connection Manager if TCP/IP with **Secure Sockets Layer (SSL)** is used:

```
(ADDRESS=(PROTOCOL=tcps) (HOST=144.25.22.217) (PORT=2484))
```

Access Control Rule List (CMAN_RULES)

The access control rule list specifies which connections are accepted, rejected, or dropped.

```
(RULE=(SRC=206.62.226.32/27) (DST=dlsun1593) (SRV=*) (ACT=accept))
```

In the previous example, SRC=206.62.226.32/27 designates the IP address of the client, or source. DST=dlsun1593 designates the destination host name. The abbreviation ACT stands for "action"—that is, accept, reject, or drop.

Note: Enter at least one rule; otherwise the default option "reject" is overridden and all connections are accepted.

Parameter List (CMAN_PROFILE)

The parameter list set attributes for an Oracle Connection Manager.

See Also: *Oracle Net Services Reference Guide* for a complete list of parameters and their default and allowed values

Configuring Clients for Oracle Connection Manager

To route clients to the database server through Oracle Connection Manager, configure either the `tnsnames.ora` file or the directory server with a **connect descriptor** that specifies protocol addresses of Oracle Connection Manager and the listener. Take note of the following:

- If using Oracle Names servers, Oracle Connection Manager automatically registers the protocol address for Oracle Connection Manager with the Oracle Names server. In order for Oracle Names servers to send requests to Oracle Connection Manager, `USE_CMAN=true` must be set in the `sqlnet.ora` file on computers where each Oracle Names server resides. To set this parameter, follow the procedure in ["Routing Connection Requests" on page 11-7](#).
- If using a `tnsnames.ora` file or a directory, the protocol addresses and the `SOURCE_ROUTE` parameter must be set. To configure the local or directory naming methods for Oracle Connection Manager, perform these tasks:
 - ["Task 1: Configure an Oracle Connection Manager Protocol Address" on page 13-5](#)
 - ["Task 2: Configure a Listener Protocol Address" on page 13-7](#)

Task 1: Configure an Oracle Connection Manager Protocol Address

To configure a protocol address for Oracle Connection Manager:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Directory or Local, and then select Service Naming.
3. Choose plus (+) from the toolbar, or choose Edit > Create.
The Welcome page of the Net Service Name Wizard appears.
4. Enter any name in the Net Service Name field.
5. Choose Next.
The Protocol Settings page appears.
6. Select the protocol on which Oracle Connection Manager is configured to listen on. By default this protocol is TCP/IP.
7. Choose Next.

The Protocol Settings page appears.

8. Enter the appropriate parameter information for the selected protocol in the fields provided. If you are using TCP/IP, the default port to use is 1630.

See Also: *Oracle Net Services Reference Guide* for protocol parameter settings

9. Choose Next.

The Service page appears.

10. Select a release and then enter the name of destination database service.

If the destination service is an Oracle9*i* or Oracle8*i* database, choose (Oracle8*i* or later), and enter a service name in the Service Name field. If destination service is an Oracle release 8.0 database, choose (Oracle8 or Previous), and enter an **Oracle System Identifier (SID)** for an instance in the Database SID field.

See Also: "[About Connect Descriptors](#)" on page 9-2 for further information about setting the service name string

11. Choose Next.

Note: Do not click Test, because a connection cannot be tested at this point.

12. Choose Finish to save your configuration and dismiss Net Service Name Wizard.

The new net service name and the Oracle Connection Manager protocol address is added to the Service Naming folder.

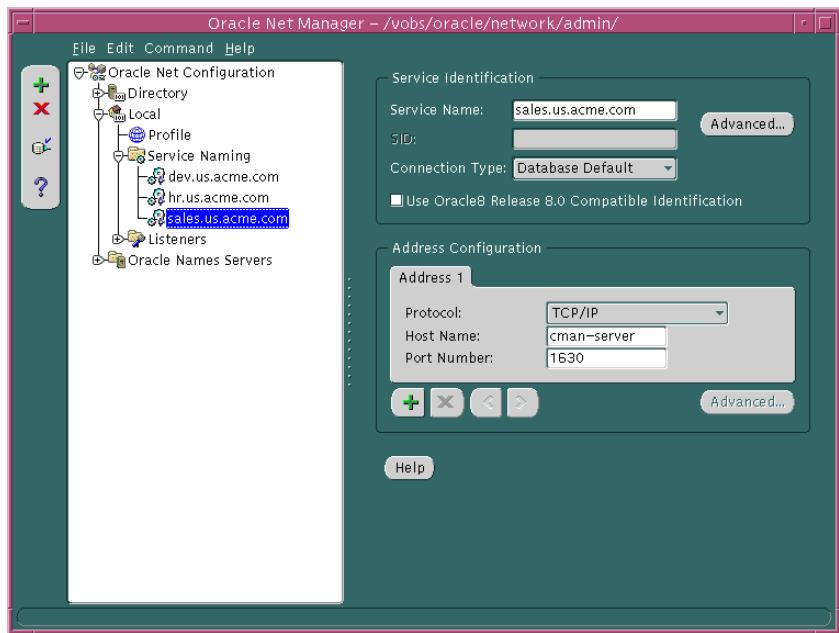
Task 2: Configure a Listener Protocol Address

After the Oracle Connection Manager protocol address is specified, create an address for the listener, so that the Oracle Connection Manager can connect to the server.

To configure the protocol address for the listener:

1. In the Oracle Net Manager, select the net service name that you created in ["Task 1: Configure an Oracle Connection Manager Protocol Address" on page 13-5](#).

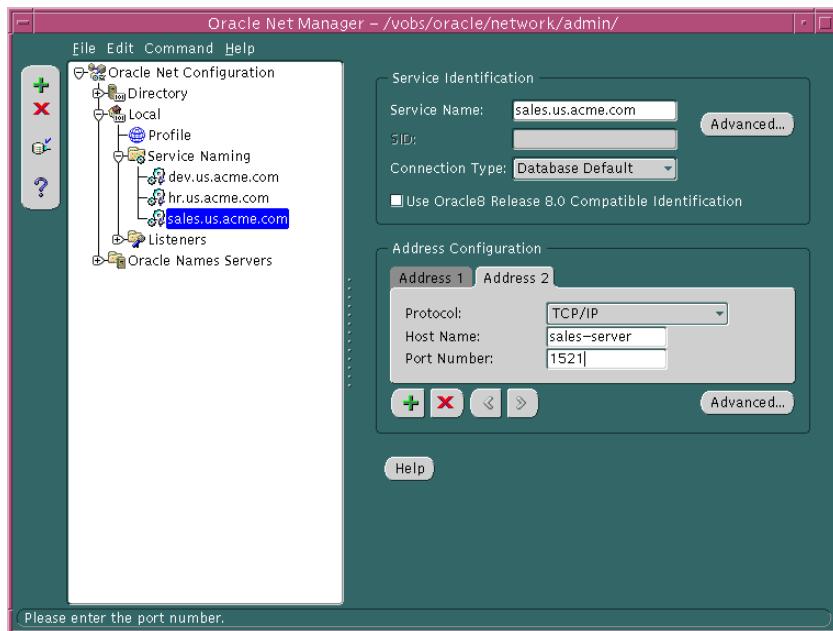
Notice that the Oracle Connection Manager protocol address is displayed in the Address 1 tab:



2. In the Address Configuration box, choose plus (+).

A new Address tab displays.

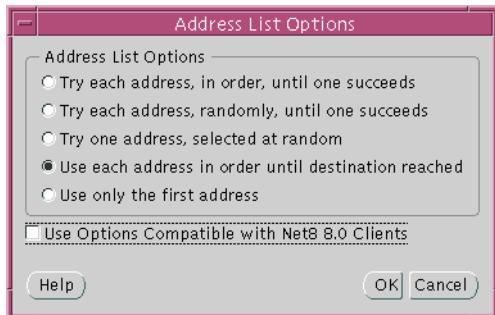
3. Select a protocol configured for the listener, and then enter the appropriate parameter information for that protocol. If you are using TCP/IP, the default port number to use is 1521:



See Also: *Oracle Net Services Reference Guide* for further information about protocol addresses and parameters

4. In the Address Configuration box, click Advanced.

The Address List Options dialog box appears:



5. Select "Use each address in order until destination reached," and then choose OK.

This option sets SOURCE_ROUTE=on. It tells the client to connect to the first protocol address, the Oracle Connection Manager, and from the first protocol address to the second protocol address, the listener.

6. Choose File > Save Network Configuration.

Note: If you specify the addresses of two or more Oracle Connection Managers in the connect descriptor, you can manually include parameters for load balancing and connect-time failover. See *Oracle Net Services Reference Guide* for examples of how to configure multiple listeners for load balancing and failover.

Figure 13–2 shows a comparison of a regular `tnsnames.ora` file and a `tnsnames.ora` file with an entry to use Oracle Connection Manager. Oracle Connection Manager entries are shown in boldface text.

Figure 13–2 tnsnames.ora File with and without Oracle Connection Manager

```
sales=                                         sales=
  (DESCRIPTION=                                (DESCRIPTION=
    (ADDRESS=                                    (SOURCE_ROUTE=yes)
      (PROTOCOL=tcp)                         (ADDRESS=
      (HOST=sales-server)                    (PROTOCOL=tcp)
      (PORT=1521))                          (HOST=cman-server)
    (CONNECT_DATA=                            (PORT=1630))
      (SERVICE_NAME=sales.com) ))           (ADDRESS=
                                              (PROTOCOL=tcp)
                                              (HOST=sales-server)
                                              (PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=sales.com) ))
```

tnsnames.ora File Element	Description
<code>SOURCE_ROUTE=yes</code>	This entry creates a source route of protocol addresses through an Oracle Connection Manager to the destination database.
<code>(ADDRESS=</code> <code>(PROTOCOL=tcp)</code> <code>(HOST=cman-server)</code> <code>(PORT=1630))</code> <code>(ADDRESS=</code> <code>(PROTOCOL=tcp)</code> <code>(HOST=sales-server)</code> <code>(PORT=1521))</code>	The first protocol address enables the client to connect to Oracle Connection Manager. From here, the Oracle Connection Manager connects to the database service through a second protocol address.

Figure 13–3 depicts a client using TCP/IP to connect to Oracle Connection Manager, and Oracle Connection Manager using TCP/IP with SSL to connect to a database server.

Figure 13–3 tnsnames.ora File Configured for Protocol Conversion Support

```
sales=
  (DESCRIPTION=
    (SOURCE_ROUTE=yes)
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=cman-server)
      (PORT=1521))
    (ADDRESS=
      (PROTOCOL=tcps)
      (HOST=sales-server)
      (PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=sales.com)))
```

Configuring the Oracle Database Server for Oracle Connection Manager

Configuring the database server involves optionally configuring the server for session multiplexing.

To enable Connection Manager to take advantage of session multiplexing, set the DISPATCHERS parameter in the [initialization parameter file](#) with the attributes PROTOCOL and MULTIPLEX.

```
DISPATCHERS= (PROTOCOL=tcp) (MULTIPLEX=on)
```

See Also:

- ["Enabling Session Multiplexing"](#) on page 13-10 for configuration details.
- [Chapter 14, "Configuring Shared Server"](#) for more information about configuring shared server.

Enabling Oracle Connection Manager Features

This section contains these topics:

- [Enabling Session Multiplexing](#)
- [Enabling Access Control](#)
- [Enabling Protocol Conversion Support](#)

The first feature is enabled using the parameter DISPATCHERS in the initialization parameter file, the second and third using the parameters RULE_LIST and CMAN parameters in the cman.ora file.

Enabling Session Multiplexing

Once the attributes PROTOCOL and MULTIPLEX have been added to the parameter DISPATCHERS in the initialization parameter file, enabling session multiplexing is simply a matter of ensuring that MULTIPLEX is set to on or to an equivalent value.

See Also: "Enabling Session Multiplexing" on page 13-13

You can set different levels of multiplexing, as the [Table 13-1](#) shows.

Table 13-1 Session Multiplexing Levels

Attribute	Description
PROTOCOL (PRO or PROT)	The network protocol—TCP in the example above—for which the dispatcher generates a listening endpoint.
MULTIPLEX (MUL or MULT)	Used to enable session multiplexing If 1, on, yes, true, or both is specified, then multiplexing is enabled for both incoming and outgoing network sessions. If in is specified, then multiplexing is enabled for incoming network sessions. If out is specified, then multiplexing is enabled for outgoing network sessions. If 0, no, off, or false is specified, then multiplexing is disabled for both incoming and outgoing network sessions.

Note: You can configure the DISPATCHERS parameter using the [Oracle Database Configuration Assistant](#).

Enabling Access Control

Oracle Connection Manager includes a feature that enables you to control client access to designated database servers in a TCP/IP environment. By specifying certain filtering rules, you can allow or restrict specific clients access to a database server.

To configure access control:

1. Manually create a `cman.ora` file, if one does not already exist.
2. Add the parameter `CMAN_RULES` with the following sub-parameters:

```
(CMAN_RULES=
  (RULE_LIST=
    (RULE=(SRC=source_host)
      (DST=destination_host)
      (SRV=service)
      (ACT=accept | reject | drop))))
```

3. Add the following parameters for each rule described in [Table 13–2](#) as needed.

Table 13–2 Per-Rule Parameters

Parameter	Description
SRC	Specify the source host name or IP address (client).
DST	Specify the destination host name or IP address (database server).
SRV	Specify the service name of the Oracle9 <i>i</i> or Oracle8 <i>i</i> database (obtained from the <code>SERVICE_NAME</code> parameter in the initialization parameter file) or SID value of the pre-release 8.1 database (obtained from the <code>ORACLE_SID</code> environment variable or registry value).
ACT	Specify to accept, reject, or drop incoming requests based on the above three parameters.

See Also: *Oracle Net Services Reference Guide* for default values and allowed values of Oracle Connection Manager parameters

You can define multiple rules in the `RULE_LIST`. The action (`ACT`) in the first matched `RULE` is applied to the connection request. If no rules are defined, all connections are accepted.

In the following example, client computer `client1-pc` is denied access to the service `sales.us.acme.com`, but client `144.25.23.45` is granted access to the service `db1`.

```
(RULE_LIST=
  (RULE=(SRC=client1-pc)(DST=sales-pc)(SRV=sales.us.acme.com)(ACT=reject))
  (RULE=(SRC=144.25.23.45)(DST=144.25.187.200)(SRV=db1)(ACT=accept)))
```

Enabling Protocol Conversion Support

Oracle Connection Manager provides support for protocol conversion, enabling clients and a database server configured with different network protocols to communicate with each other. An Oracle Connection Manager can listen on any protocol that Oracle supports.

Without this kind of support, a client that uses Named Pipes cannot connect to a database server that uses TCP/IP. If Oracle Connection Manager is configured for TCP/IP, the client can connect to Oracle Connection Manager using Named Pipes, and Oracle Connection Manager can connect to the database server using TCP/IP.

As the following example shows, the `cman.ora` must be configured with a protocol address for each protocol that the client uses. Sample addresses for three protocols are shown—TCP/IP, Named Pipes, and TCP/IP with SSL.

```
(CMAN=
  (ADDRESS=(PROTOCOL=tcp)(HOST=proxysvr1)(PORT=1630))
  (ADDRESS=(PROTOCOL=nmp)(SERVER=proxysvr_pc)(PIPE=cmanpipe))
  (ADDRESS=(PROTOCOL=tcps)(HOST=144.25.22.217)(PORT=2484)))
```

By default, Oracle Connection Manager accepts client connection requests at the following default protocol address:

```
CMAN=(ADDRESS=(PROTOCOL=tcp)(HOST=anyhost)(PORT=1630))
```

To configure access control:

1. Manually create a `cman.ora` file, if one does not already exist.
2. Add protocol addresses for each of the protocols that the client uses in the `CMAN` parameter:

```
CMAN=
  (ADDRESS=(protocol_address_information))
  (ADDRESS=(protocol_address_information))
```

See Also: *Oracle Net Services Reference Guide* for further information about protocol addresses and parameters

Configuring Shared Server

Shared server is an architecture that enables a database server to allow many user processes to share very few server processes, so the number of users that can be supported is increased. With shared server, many user processes connect to a **dispatcher**. The dispatcher directs multiple incoming network session requests to a common queue. An idle shared server process from a shared pool of server processes picks up a request from the queue. This means a small pool of server processes can serve a large number of clients.

This chapter describes how to configure **shared server**. It contains these topics:

- [Configuring Shared Server with the DISPATCHERS Parameter](#)
- [Enabling Connection Pooling](#)
- [Allocating Resources](#)
- [Using Shared Server on Clients](#)
- [Overriding Shared Server on Clients](#)

See Also:

- *Oracle9i Database Administrator's Guide* for further information about shared server configuration
- *Oracle9i Database Performance Guide and Reference* for further information about tuning shared server parameters

Configuring Shared Server with the DISPATCHERS Parameter

To enable a shared server configuration, set the DISPATCHERS parameter in the database [initialization parameter file](#).

Note: Oracle Database Configuration Assistant enables you to configure this parameter.

After setting this parameter, restart the instance to enable shared server configuration. Set DISPATCHERS as follows:

```
DISPATCHERS="(attribute=value)"
```

One of the following attributes is required to enable shared server:

- [ADDRESS \(ADD or ADDR\)](#)
- [DESCRIPTION \(DES or DESC\)](#)
- [PROTOCOL \(PRO or PROT\)](#)

Table 14–1 Required Attributes of the DISPATCHERS Parameter

Attribute	Description
ADDRESS (ADD or ADDR)	Specify the network protocol address of the endpoint on which the dispatchers listen.
DESCRIPTION (DES or DESC)	Specify the network description of the endpoint on which the dispatchers listen, including the network protocol address. The syntax is as follows: (DESCRIPTION=(ADDRESS=...))
PROTOCOL (PRO or PROT)	Specify the network protocol for which the dispatcher generates a listening endpoint. For example: (PROTOCOL=tcp)

See Also: *Oracle Net Services Reference Guide* for further information about protocol address syntax

The following attributes are optional:

- CONNECTIONS (CON or CONN)
- DISPATCHERS (DIS or DISP)
- LISTENER (LIS or LIST)
- MULTIPLEX (MUL or MULT)
- POOL (POO)
- SERVICE (SER or SERV)
- SESSIONS (SES or SESS)
- TICKS (TIC or TICK)

Table 14–2 Optional Attributes of the DISPATCHERS Parameter

Attribute	Description
CONNECTIONS (CON or CONN)	Specify the maximum number of network connections to allow for each dispatcher. The default is operating system specific. For example, 1024 is the default for Sun Solaris and Windows NT.
DISPATCHERS (DIS or DISP)	Specify the initial number of dispatchers to start. The default is 1.

Attribute	Description
LISTENER (LIS or LIST)	<p>Specify an alias name for the listener(s) with which the PMON process registers dispatcher information. Set the alias to a name which is resolved through a naming method.</p>
	<p>This attribute need only be specified if the:</p> <ul style="list-style-type: none"><li data-bbox="587 390 1254 494">■ Listener is a local listener that uses a nondefault address, that is, not TCP/IP, port 1521 and is not specified with the LOCAL_LISTENER or REMOTE_LISTENER initialization parameters<li data-bbox="587 511 951 537">■ Listener(s) are on other nodes <p>The LISTENER attribute overrides the REMOTE_LISTENER and the LOCAL_LISTENER initialization parameters.</p> <p>Important: Resolve the listener alias through a naming method, such as a tnsnames.ora file on the database server or an Oracle Names server.</p> <p>For example, if the listener alias name is <code>listener_sales</code> with two listening endpoints of port 1521, and the chosen naming method is the local naming method, the entry in the tnsnames.ora file would look like the following:</p> <pre data-bbox="587 831 1239 934">listener_sales= (description= (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521)) (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))</pre>

Attribute	Description
POOL (POO)	<p>Use to enable connection pooling.</p> <p>If a number is specified, enables connection pooling for both incoming and outgoing idle network connections. The number specified is the timeout in ticks for both incoming and outgoing idle network connections.</p> <p>If on, yes, true, or both is specified, enables connection pooling for both incoming and outgoing idle network connections. A default timeout of 10 ticks is used for both incoming and outgoing idle network connections.</p> <p>If in is specified, connection pooling is enabled for incoming idle network connections and the default timeout of 10 ticks is used for incoming idle network connections. in can also be assigned a timeout in ticks value, such as (in=20). If the numeric value of a specified timeout is 0 or 1, then the default value of 10 ticks is used.</p> <p>If out is specified, connection pooling is enabled for outgoing idle network connections and the default timeout of 10 ticks is used for outgoing network connections. out can also be assigned a timeout in ticks value, such as (out=20). If the numeric value of a specified timeout is 0 or 1, then the default value of 10 ticks is used.</p> <p>If no, off, or false is specified, then connection pooling is disabled for both incoming and outgoing network connections.</p>
	See Also:
	<ul style="list-style-type: none"> ▪ "Connection Pooling" on page 1-15 ▪ "Enabling Connection Pooling" on page 14-8 ▪ TICKS (TIC or TICK) attribute
SERVICE (SER or SERV)	Specify the service name(s) the dispatchers register with the listeners. If no values are specified, then service names specified with the SERVICE_NAMES initialization parameter are used.
SESSIONS (SES or SESS)	<p>Specify the maximum number of network sessions to allow for each dispatcher.</p> <p>The default is operating system specific. Most operating systems have a default of 16 KB.</p>

Attribute	Description
TICKS (TIC or TICK)	Specify the length of a network tick in seconds. A tick is the amount of time it takes for a message to be sent and processed from the client to the database server or from the database server to the client. The value set is multiplied with the POOL timeout value to get the total connection pool timeout. The default is 15 seconds. For a fast network, Oracle Corporation recommends a tick size of ~1 second. For a slow network, Oracle Corporation recommends a tick size of ~3-4 seconds.

Setting the Initial Number of Dispatchers

The number of dispatchers started at instance startup is controlled by the DISPATCHERS attribute.

Note: Unlike the number of shared server processes, the number of dispatchers does not change automatically. The number of dispatchers can be explicitly changed with the SQL statement ALTER SYSTEM. You can change the number of dispatchers in this manner up to a maximum limit specified by the MAX_DISPATCHERS parameter. See *Oracle9i SQL Reference* for further information about the ALTER SYSTEM statement and *Oracle9i Database Performance Guide and Reference* for further information about the DISPATCHERS parameter.

The appropriate number of dispatchers for each instance depends upon the performance you want from your database, the host operating system limit on the number of connections for each process, which is operating system dependent, and the number of connections required for each network protocol.

Calculating the Initial Number of Dispatchers

Once you know the number of possible connections for each process for the operating system, calculate the initial number of dispatchers to create during instance startup, for each network protocol, using the following formula.

$$\text{number of dispatchers} = \text{CEIL} \left(\frac{\text{maximum number of concurrent sessions}}{\text{connections for each dispatcher}} \right)$$

CEIL represents the number roundest to the next highest whole integer.

Example: Initial Number of Dispatchers

Assume a system that has:

- 4000 users concurrently connected through TCP/IP and supports 1,000 connections for each process
- 2,500 sessions concurrently connected through TCP/IP with SSL and supports 1,000 connections for each process

In this case, the DISPATCHERS attribute for TCP/IP should be set to a minimum of four dispatchers and TCP/IP with SSL should be set to a minimum of three dispatchers:

```
DISPATCHERS= " (PROTOCOL=tcp) (DISPATCHERS=4) (CONNECTIONS=1000) "
DISPATCHERS= " (PROTOCOL=tcps) (DISPATCHERS=4) (CONNECTIONS=1000) " "
```

Depending on performance, you may need to adjust the number of dispatchers.

Example: Dispatcher Address with IP Address

To force the IP address used for the dispatchers, set the following:

```
DISPATCHERS= " (ADDRESS=(PROTOCOL=tcp) (HOST=144.25.16.201))
(DISPATCHERS=2) "
```

This starts two dispatchers that listen on host 144.25.16.201. Note that Oracle Net dynamically selects the TCP/IP port for the dispatcher.

Example: Dispatcher Address with PORT

To force the exact location of the dispatchers, add the PORT as follows:

```
DISPATCHERS= " (ADDRESS=(PROTOCOL=tcp)
(HOST=144.25.16.201)(PORT=5000))(DISPATCHERS=1) "
DISPATCHERS= " (ADDRESS=(PROTOCOL=tcp)
(HOST=144.25.16.201)(PORT=5001))(DISPATCHERS=1) "
```

Note: You can specify multiple DISPATCHERS in the initialization file, but they must be adjacent to each other.

Enabling Connection Pooling

Connection pooling is a resource utilization feature that enables you to reduce the number of physical network connections to a dispatcher. This is achieved by sharing or pooling a set of connections among the client processes.

To configure connection pooling, set the DISPATCHERS parameter in the initialization parameter file with the POOL attribute and the following optional attributes:

- CONNECTIONS (CON orCONN)
- SESSIONS (SES orSESS)
- TICKS (TIC orTICK)

Refer to the example in "[Example: Initial Number of Dispatchers](#)" on page 14-7. Connection pooling can allow each dispatcher 1,000 connections and 4,000 sessions for TCP/IP and 2,500 sessions for TCP/IP with SSL. This reduces the configuration to one dispatcher for each protocol, as shown in the following:

```
DISPATCHERS="(PROTOCOL=tcp)(DISPATCHERS=1)(POOL=on)(TICK=1)  
(CONNECTIONS=1000)(SESSIONS=4000)"  
  
DISPATCHERS="(PROTOCOL=tcp)(DISPATCHERS=1)(POOL=on)(TICK=1)  
(CONNECTIONS=1000)(SESSIONS=2500)"
```

See Also: ["Connection Pooling"](#) on page 1-15

Allocating Resources

An Oracle database can be represented by multiple service names. Because of this, a pool of dispatchers can be allocated exclusively for clients requesting a particular service. This way, the mission critical requests may be given more resources and, thus, in effect increase their priority.

For example, the following initialization parameter file sample shows two dispatchers. The first dispatcher services requests for clients requesting sales.us.acme.com. The other dispatcher services requests only for clients requesting adminsales.us.acme.com.

```
SERVICE_NAMES=sales.us.acme.com
INSTANCE_NAME=sales
DISPATCHERS="(PROTOCOL=tcp)"
DISPATCHERS="(PROTOCOL=tcp)(SERVICE=adminsales.us.acme.com)"
```

Using Shared Server on Clients

If shared server is configured and a client connection request arrives when no dispatchers are registered, the requests can be handled by a dedicated server process (configured in the `listener.ora` file). If you want a particular client always to use a dispatcher, configure (`SERVER=shared`) in the connect data portion of the connect descriptor. For example:

```
sales=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)
    (server=shared)))
```

If a dispatcher is not available, the client connection request is rejected.

See Also: "Configuring Advanced Connect Data Parameters" on page 15-7 to set the SERVER parameter

Overriding Shared Server on Clients

If the database is configured for shared server and a particular client requires a dedicated server, you can configure the client to use a dedicated server in one of the following ways:

- A net service name can be configured with a connect descriptor that contains (SERVER=dedicated) in the CONNECT_DATA section. For example:

```
sales=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)
    (SERVER=dedicated)))
```

- The client profile (sqlnet.ora file) can be configured with USE_DEDICATED_SERVER=on. This adds (SERVER=dedicated) to the CONNECT_DATA section of the connect descriptor the client uses.

Note: If USE_DEDICATED_SERVER is set to on, then existing (SERVER=value) entries in connect descriptors are overwritten with (SERVER=dedicated).

See Also:

- ["Configuring Advanced Connect Data Parameters"](#) on page 15-7 to set the SERVER parameter
- ["Routing Connection Requests"](#) on page 11-7 to set the USE_DEDICATED_SERVER parameter

Enabling Advanced Features for Oracle Net Services

This chapter describes how to configure advanced features of Oracle Net Services, including advanced connect data parameters, **load balancing**, **failover**, and connections to non-database services.

This chapter contains these topics:

- [Configuring Advanced Network Address and Connect Data Information](#)
- [Configuring Connection Load Balancing](#)
- [Configuring Transparent Application Failover](#)
- [Configuring Session Data Unit](#)
- [Configuring the VI Protocol for Application Web Server and Oracle Database Server Communication](#)
- [Configuring Connections to Non-Oracle Database Services](#)

Configuring Advanced Network Address and Connect Data Information

This section contains the following advanced connect descriptor topics:

- [Creating a List of Listener Protocol Addresses](#)
- [Configuring Address List Parameters](#)
- [Configuring Advanced Connect Data Parameters](#)

Creating a List of Listener Protocol Addresses

A database service may be accessed by more than one network route, or protocol address. In the following example, sales.us.acme.com can connect to sales.us.acme.com using listeners on either sales1-server or sales2-server.

```
sales.us.acme.com=
(DESCRIPTION=
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521))
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com) ))
```

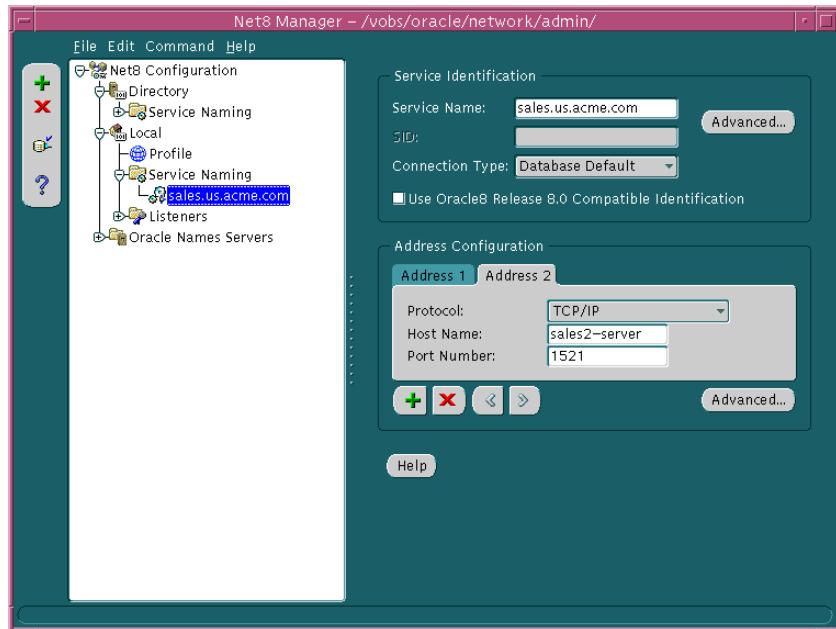
To add a network protocol address to an existing net service name or database service:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Directory or Local > Service Naming.
3. Select either a net service name or a database service.
The right pane displays the current destination service and address list.
4. In the Address Configuration box, choose plus (+).

A new Address tab appears:



5. Select the protocol and enter appropriate address information.

See Also: *Oracle Net Services Reference Guide* for details about protocol address parameters

6. Order the protocol addresses according to where they should be in the protocol address list with the left-arrow and right-arrow buttons. Unless multiple address options are configured, the first address in the list is contacted.

See Also: "[Configuring Address List Parameters](#)" on page 15-4 for address list options

7. Choose File > Save Network Configuration.

Configuring Address List Parameters

When a database service is accessible by multiple listener protocol addresses, specify the order in which the addresses are to be used. The addresses can be chosen randomly or tried sequentially.

When multiple protocol addresses have been configured for a net service name or database service, you can configure the parameters described in [Table 15–1](#).

Table 15–1 Address List Parameters

Parameter	Description
SOURCE_ROUTE (Source Routing)	<p>When set to <code>on</code>, instructs Oracle Net to use each address in the order presented until the destination reached. This parameter is required when reaching the destination requires more than one address stop. This parameter is used to enable connections to Oracle Connection Manager.</p> <p>See Also: "Configuring Clients for Oracle Connection Manager" on page 13-6</p>
FAILOVER (Connect-Time Failover)	<p>At connect time, instructs Oracle Net to fail over to a different listener if the first listener fails when set to <code>on</code>. The number of addresses in the list determines how many addresses are tried. When set to <code>off</code>, instructs Oracle Net to try one address.</p> <p>Connect-time failover is turned <code>on</code> by default for multiple address lists (<code>ADDRESS_LIST</code>), connect descriptors (<code>DESCRIPTION</code>), and multiple connect descriptors (<code>DESCRIPTION_LIST</code>).</p> <p>Important: Oracle Corporation recommends not setting the <code>GLOBAL_DBNAME</code> parameter in the <code>SID_LIST_listener_name</code> section of the <code>listener.ora</code> file. A statically configured global database name disables connect-time failover.</p>
LOAD_BALANCE (Client Load Balancing)	<p>When set to <code>on</code>, instructs Oracle Net to progress through the list of protocol addresses in a random sequence, balancing the load on the various listeners. When set to <code>off</code>, instructs Oracle Net to try the protocol addresses sequentially until one succeeds.</p> <p>Client load balancing is turned <code>on</code> by default for multiple connect descriptors (<code>DESCRIPTION_LIST</code>).</p>

Note: It is not possible to set client load balancing or connect-time failover with source routing. While connect-time failover and client load balancing choose an address from a list, source routing connects to each address in the list sequentially.

Implementation of these parameter depends on the naming method used. [Table 15–2](#) describes the tools to configure connect-time failover or client load balancing. Source routing involves other configuration that goes beyond the scope of this section.

See Also: ["Configuring Clients for Oracle Connection Manager"](#) on page 13-5 for more information about configuring clients for source routing

Table 15–2 Address List Configuration for Naming Method Types

Naming Method	Tool to Configure Address List Options
Local Naming	Net service names stored in a <code>tnsnames.ora</code> file can be configured for connect-time failover and client load balancing using Oracle Net Manager. See Also: The procedure that follows
Directory Naming	Net service names and database services stored in a directory can be configured for connect-time failover and client load balancing using Oracle Net Manager. See Also: The procedure that follows
Oracle Names	Net service names stored in an Oracle Names server can be configured for connect-time failover and client load balancing using the following tools: <ul style="list-style-type: none"> ■ Oracle Names Control utility's <code>REGISTER</code> command ■ Oracle Net Manager See Also: "Task 7: Register Data with the Oracle Names Server" on page 9-42

To configure connect-time failover or client load balancing:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

2. In the navigator pane, expand Directory or Local > Service Naming.
3. Select either a net service name or a database service.
The right pane displays the current destination service and address list.
4. In the Address Configuration box, choose Advanced.

The Address List Options dialog box appears:

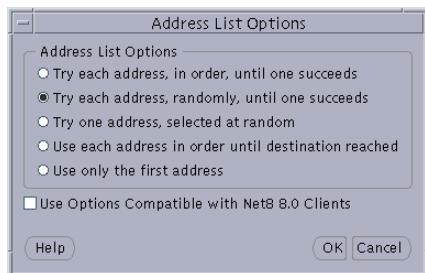


Table 15–3 describes the Address List Options dialog box options.

Table 15–3 Address List Options Dialog Box

Option	Parameter Setting
Try each address, in order, until one succeeds	For release 8.1 and 9.0 clients, FAILOVER=on For pre-release 8.1 clients, SOURCE_ROUTE=off
Try each address, randomly, until one succeeds	LOAD_BALANCE=on FAILOVER=on
Note: This option is not enabled if "Use Options Compatible with Net8 8.0 Clients" is selected.	
Try one address, selected at random	LOAD_BALANCE=on Note: This option is not enabled if "Use Options Compatible with Net8 8.0 Clients" is selected.
Use each address in order until destination reached	SOURCE_ROUTE=on
Use only the first address	LOAD_BALANCE=off Note: This option is not enabled if "Use Options Compatible with Net8 8.0 Clients" is selected.

5. Select one of the address list options, and then choose OK.
6. Choose File > Save Network Configuration.

The following example shows a `tnsnames.ora` file configured for client load balancing:

```
sales.us.acme.com=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
        (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521))
        (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
    (CONNECT_DATA=
      (SERVICE_NAME=sales.us.acme.com)) )
```

The following example shows a `tnsnames.ora` file configured for connect-time failover:

```
sales.us.acme.com=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (LOAD_BALANCE=off)
      (FAILOVER=ON)
        (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521))
        (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
    (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)) )
```

Configuring Advanced Connect Data Parameters

The `CONNECT_DATA` section of a connect descriptor defines the destination database service. In the following example, `SERVICE_NAME` defines a service called `sales.us.acme.com`:

```
sales.us.acme.com=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=sales.us.acme.com)) )
```

Besides the service name, connect data information can be optionally configured with the parameters described in [Table 15–4](#).

Table 15–4 Advanced Connect Data Settings

Oracle Net Manager Field/Option	tnsnames.ora File Parameter	Description
Instance Name	INSTANCE_NAME	<p>Use to identify the database instance to access.</p> <p>The instance name can be obtained from the INSTANCE_NAME parameter in the initialization parameter file.</p> <p>Note: This parameter is not enabled if "Use Oracle8i Release 8.0 Compatible Identification" is checked.</p> <p>See Also: "About Connect Descriptors" on page 9-2</p>
Session Data Unit Size	SDU	<p>To optimize the transfer rate of data packets being sent across the network, you can specify the session data unit (SDU) size to change the performance characteristics having to do with the packets sent across the network.</p> <p>See Also: "Configuring Session Data Unit" on page 15-28</p>
Use for Heterogeneous Services	HS	<p>If you want an Oracle server to access a non-Oracle system through Heterogeneous Services, turn this option on.</p> <p>See Also: "Configuring Oracle Net for Oracle Heterogeneous Services" on page 15-41</p>
Oracle Rdb Settings		
Oracle RDB Database	RDB_DATABASE	<p>Specify the file name of the Oracle RDB database.</p> <p>See Also: "Configuring Oracle Net for an Oracle Rdb Database" on page 15-44</p>
Type of Service	TYPE_OF_SERVICE	<p>Specify the type of service to use for the Oracle RDB database.</p> <p>See Also: "Configuring Oracle Net for an Oracle Rdb Database" on page 15-44</p>
Global Database Name	GLOBAL_NAME	<p>Use to identify an Oracle Rdb database.</p> <p>See Also: "Configuring Oracle Net for an Oracle Rdb Database" on page 15-44</p>

Implementation of these parameter depends on the naming method used. [Table 15–5](#) describes the tools to configure connect-time failover or client load balancing. Source routing involves other configuration that goes beyond the scope of this section.

See Also: ["Configuring Clients for Oracle Connection Manager"](#) on page 13-5 for more information about configuring clients for source routing

Table 15–5 Connect Data Configuration for Naming Method Types

Naming Method	Tool to Configure Multiple Address Options
Local Naming	Net service names stored in a <code>tnsnames.ora</code> file can be configured with advanced connect data information using Oracle Net Manager. See Also: The procedure that follows
Directory Naming	Net service names and database services stored in a directory can be configured with advanced connect data information using Oracle Net Manager. See Also: The procedure that follows
Oracle Names	Net service names and database services stored in an Oracle Names server can be configured with advanced connect data information using the following tools: <ul style="list-style-type: none"> ▪ Oracle Names Control utility's REGISTER command ▪ Oracle Net Manager See Also: "Task 7: Register Data with the Oracle Names Server" on page 9-42

To configure advanced `CONNECT_DATA` parameters for either a net service name or a database service:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

2. In the navigator pane, expand Directory or Local > Service Naming.
3. Select either a net service name or a database service.

The right pane displays the current destination service and address list.

4. In the Service Identification box, choose Advanced.

The Advanced Service Options dialog box appears:



See Also: [Table 15–4](#) on page 15–8 for a description of the fields and options

5. Enter fields or select options as appropriate, and then choose OK.
6. Choose File > Save Network Configuration.

Configuring Connection Load Balancing

Connection load balancing improves connection performance by balancing the number of active connections among multiple **dispatchers**. In an Oracle9*i* Real Application Clusters environment, connection load balancing also has the capability to balance the number of active connections among multiple instances.

Because the **PMON process** can register with remote listeners, a listener can always be aware of all instances and dispatchers, regardless of their location. Depending on the load information, a listener decides which instance and, if shared server is configured, which dispatcher to send the incoming client request to.

In a **shared server** configuration, a listener selects a dispatcher in the following order: 1) least loaded node, 2) least loaded instance, and 3) least loaded dispatcher for that instance. In a **dedicated server** configuration, a listener selects an instance in the following order: 1) least loaded node, and 2) least loaded instance.

If a database service has multiple instances on multiple nodes, the listener chooses the least loaded instance on the least loaded node. If shared server is configured, then the least loaded dispatcher of the selected instance is chosen.

To enable connection load balancing, the initialization parameter file must contain the following parameters:

- `SERVICE_NAMES` for the database service name
- `INSTANCE_NAME` for instance name
- `DISPATCHERS` to enable shared server configuration and specify the dispatchers used by the instance

An Oracle9*i* Real Application Clusters environment requires that the dispatchers on each instance be cross registered with the other listeners on the other nodes. This is achieved by the use of the `LISTENER` attribute of the `DISPATCHERS` parameter.

See Also:

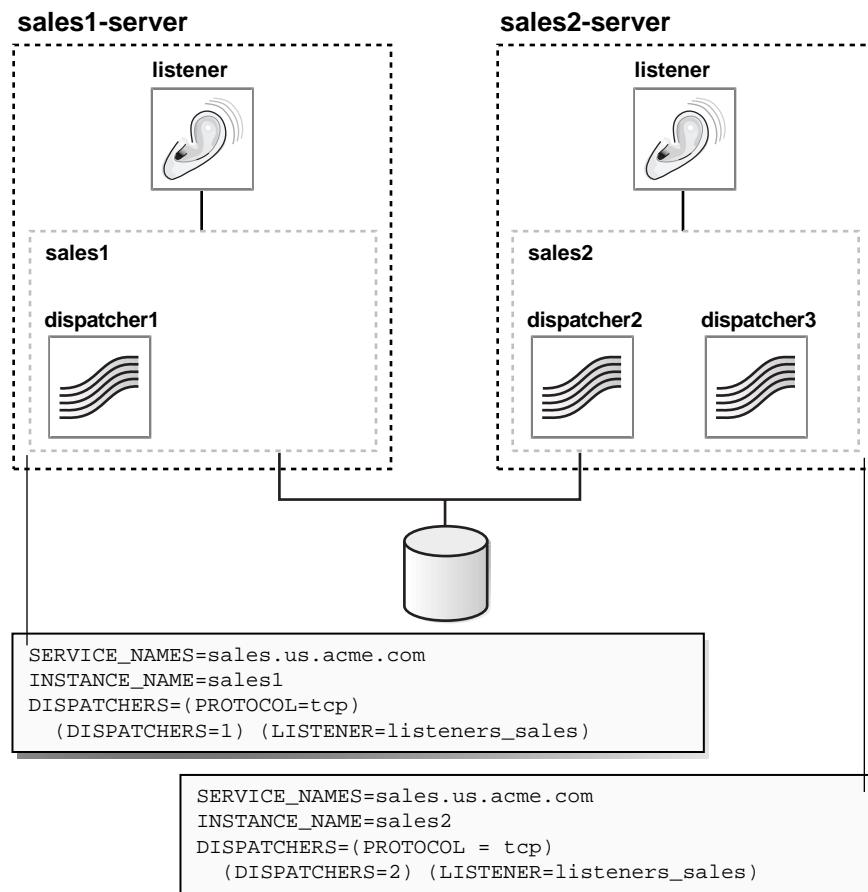
- ["Registering Information with a Remote Listener"](#) on page 12-16 for complete information about cross registration
- *Oracle9*i* Database Reference* for complete information about the `SERVICE_NAMES` and `INSTANCE_NAME` parameters
- [Chapter 14, "Configuring Shared Server"](#) for complete information about the `LISTENER` attribute

Note: For optimum connection load balancing results, the instances that belong to the same database service should be on equivalent hardware and software configurations.

Example: Connection Load Balancing for Shared Server Configuration

Figure 15–1 shows an Oracle9i Real Application Clusters shared server database with two instances, sales1 and sales2, of the same service, sales.us.acme.com. The instances sales1 and sales2 reside on computers sales1-server and sales2-server, respectively. sales1 has one dispatcher and sales2 has two dispatchers. Listeners named listener run on nodes 1 and 2, respectively. The listener attribute in the DISPATCHERS parameter has been configured to allow for service registration of information to both listeners.

Figure 15–1 Load Balancing Environment for a Shared Server Configuration



The `listeners_sales` value in (`LISTENER=listeners_sales`) can be then resolved through a local `tnsnames.ora` file on the both servers as follows:

```
listeners_sales=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521))
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
```

Based on the environment, the following actions occur. The numbered actions correspond to the arrows shown in [Figure 15–2](#) on page 15-14:

1. PMON processes for instances `sales1` and `sales2` register with both listeners. The listeners are updated on the load of the instances and dispatchers dynamically. The following load information is registered:

	sales1-server	sales2-server
1 Minute Node Load Average	600	400

	sales1	sales2
Number of Connections to Instance	200	300

	dispatcher1	dispatcher2	dispatcher 3
Number of Connections to Dispatchers	200	100	200

The load average on `sales2-server` (400) is less than the load average on `sales1-server` (600). This can happen if more processing is required on `sales1-server`. The number of connections to `sales1` (200) is the same as that of its only dispatcher, `dispatcher1`. The number of connections on `sales2` (300) is the sum of the connections on its two dispatchers, `dispatcher2` (100) and `dispatcher3` (200). Therefore, `sales2` has more connections than `sales1`. In this example, `sales2-server` is the least loaded node, `sales2` is the least loaded instance, and `dispatcher2` is the least loaded dispatcher.

2. The client sends a connect request.

A connect descriptor is configured to try each protocol address randomly until one succeeds:

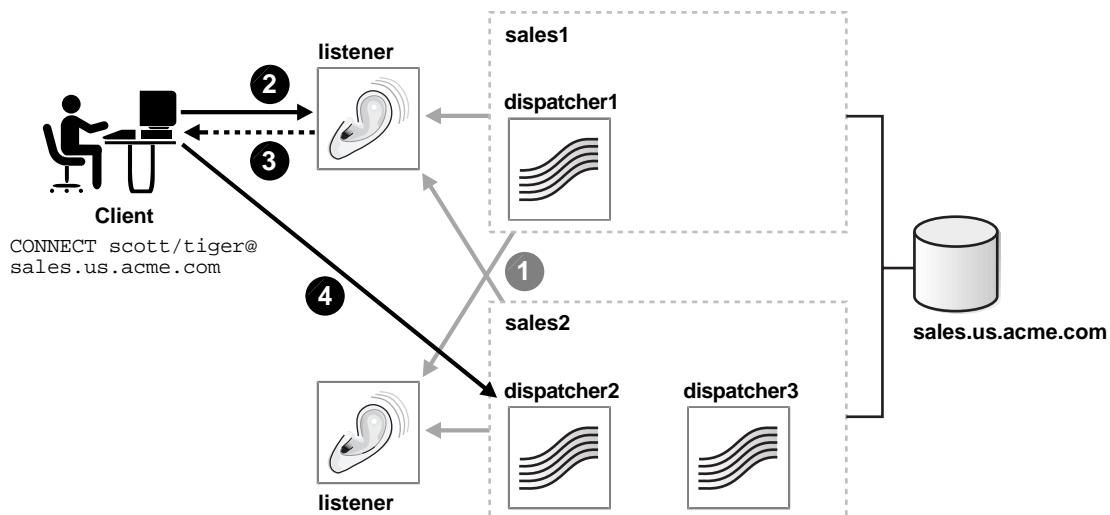
```
sales.us.acme.com=
  (DESCRIPTION=
    (LOAD_BALANCE=on)
    (FAILOVER=on)
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521))
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521))
    (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)))
```

The listener on sales1-server was randomly chosen to receive the client connect request.

The listener on sales1-server compares the load of the instances sales1 and sales2. The comparison takes into account the load on nodes sales1-server and sales2-server, respectively. Since sales2-server is less loaded than sales1-server, the listener chooses sales2-server over sales1-server.

3. The listener compares the load on dispatchers dispatcher2 and dispatcher3. Because dispatcher2 is less loaded than dispatcher3, the listener redirects the client connect request to dispatcher2.
4. The client connects directly to dispatcher2.

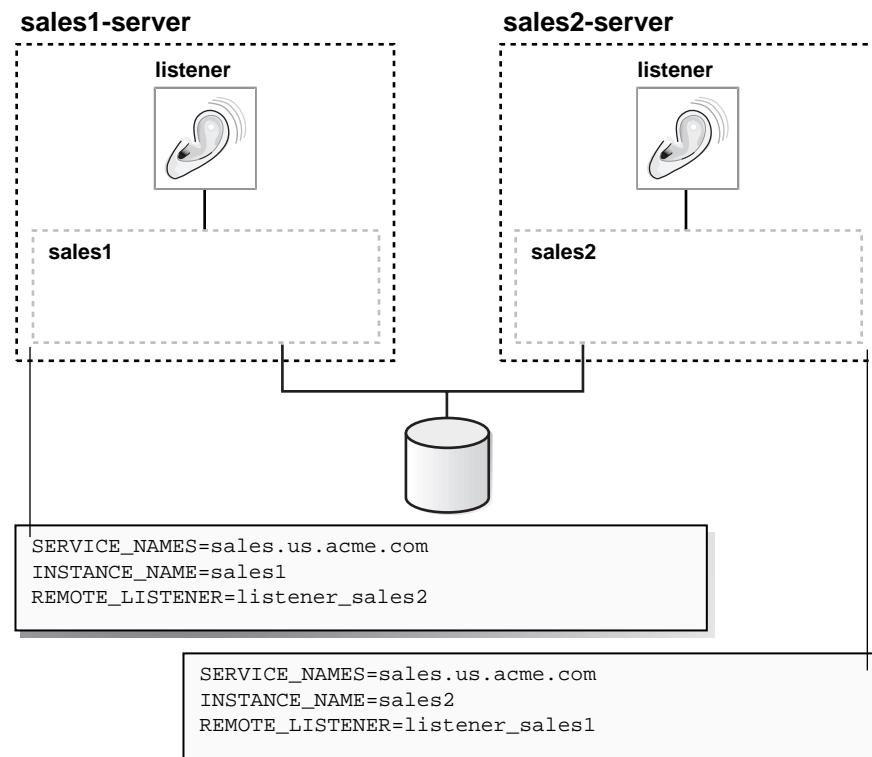
Figure 15–2 Load Balancing Example for a Shared Server Configuration



Example: Connection Load Balancing for Dedicated Server Configuration

Figure 15–3 shows an Oracle9i Real Application Clusters dedicated server database with two instances, sales1 and sales2, of the same service, sales.us.acme.com. The instances sales1 and sales2 reside on computers sales1-server and sales2-server, respectively. Listeners named listener run on nodes 1 and 2, respectively. The REMOTE_LISTENER parameter has been configured to allow for service registration of information to both listeners.

Figure 15–3 Load Balancing Environment for a Dedicated Server Configuration



The `listener_sales2` value in `(REMOTE_LISTENER=listener_sales2)` can be then resolved through a local `tnsnames.ora` file on the `sales1`-server as follows:

```
listener_sales2=
(DESCRIPTION=
 (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
```

The `listener_sales1` value in `(REMOTE_LISTENER=listener_sales1)` can be then resolved through a local `tnsnames.ora` file on the `sales2`-server as follows:

```
listener_sales1=
(DESCRIPTION=
 (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521)))
```

Based on the environment, the following actions occur. The numbered actions correspond to the arrows shown in [Figure 15–4](#) on page 15-17:

1. PMON processes for instances `sales1` and `sales2` register with both listeners. The listeners are updated on the load of the instances dynamically. The following load information is registered:

	<code>sales1-server</code>	<code>sales2-server</code>
1 Minute Node Load	450	200
Average		

	<code>sales1</code>	<code>sales2</code>
Number of Connections to Instance	200	150

In this example, `sales2-server` is the least loaded node and `sales2` is the least loaded instance.

2. The client sends a connect request.

A connect descriptor is configured to try each protocol address randomly until one succeeds:

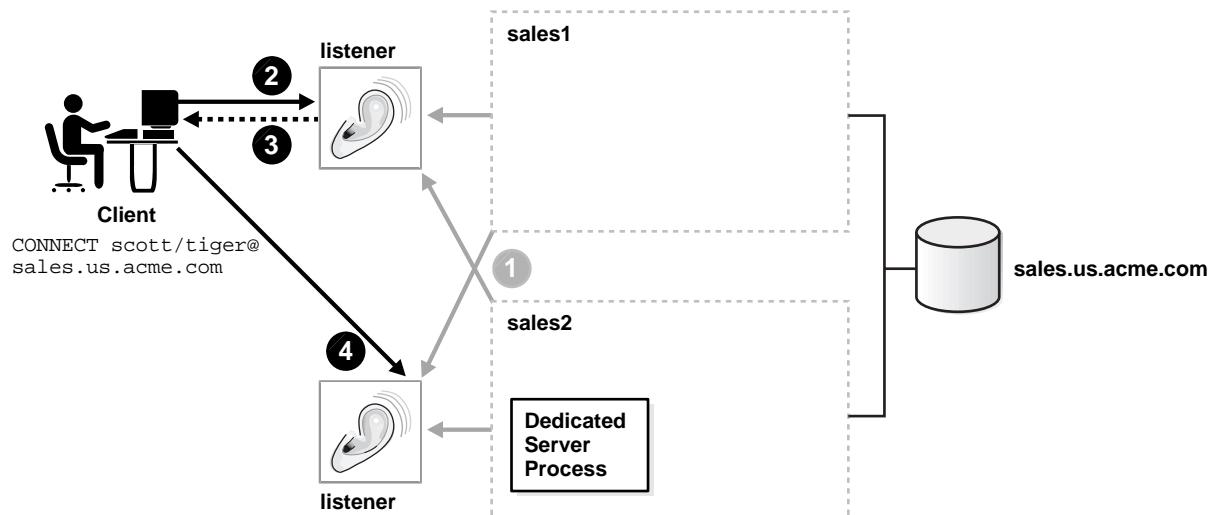
```
sales.us.acme.com=
(DESCRIPTION=
 (LOAD_BALANCE=on)
 (FAILOVER=on)
 (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521))
 (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521))
 (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)))
```

The listener on sales1-server was randomly chosen to receive the client connect request.

The listener on sales1-server compares the load of the instances sales1 and sales2. The comparison takes into account the load on nodes sales1-server and sales2-server, respectively. Since sales2-server is less loaded than sales1-server, the listener chooses sales2-server over sales1-server.

3. The listener on sales1-server redirects the client connect request to dispatcher2.
4. The client connects to the listener on sale2-server. The listener starts a dedicated server process, and the dedicated server process inherits the connection request from the listener.

Figure 15–4 Load Balancing Example for a Dedicated Server Configuration



Configuring Transparent Application Failover

Note: [Transparent Application Failover \(TAF\)](#) is available with Oracle9i Enterprise Edition.

TAF instructs Oracle Net to fail over a failed connection to a different listener. This enables the user to continue to work using the new connection as if the original connection had never failed.

TAF involves manual configuration of a net service name that includes the FAILOVER_MODE parameter included in the CONNECT_DATA section of the connect descriptor.

This sections covers the following topics:

- [What TAF Restores](#)
- [TAF Database Configurations](#)
- [FAILOVER_MODE Parameters](#)
- [TAF Implementation](#)
- [TAF Verification](#)

What TAF Restores

TAF automatically restores some or all of the following elements associated with active database connections. Other elements, however, may need to be embedded in the application code to enable TAF to recover the connection.

Client-Server Database Connections

TAF automatically reestablishes the connection using the same connect string or an alternate connect string that you specify when configuring failover.

Users' Database Sessions

TAF automatically logs a user in with the same user ID as was used prior to failure. If multiple users were using the connection, then TAF automatically logs them in as they attempt to process database commands. Unfortunately, TAF cannot automatically restore other session properties. These properties can, however, be restored by invoking a callback function.

Executed Commands

If a command was completely executed upon connection failure, and it changed the state of the database, TAF does not resend the command. If TAF reconnects in response to a command that may have changed the database, TAF issues an error message to the application.

Open Cursors Used for Fetching

TAF allows applications that began fetching rows from a cursor before failover to continue fetching rows after failover. This is called "select" failover. It is accomplished by re-executing a `SELECT` statement using the same snapshot, discarding those rows already fetched and retrieving those rows that were not fetched initially. TAF verifies that the discarded rows are those that were returned initially, or it returns an error message

Active Transactions

Any active transactions are rolled back at the time of failure because TAF cannot preserve active transactions after failover. The application instead receives an error message until a `ROLLBACK` is submitted.

Serverside Program Variables

Serverside program variables, such as PL/SQL package states, are lost during failures; TAF cannot recover them. They can be initialized by making a call from the failover callback.

See Also: *Oracle Call Interface Programmer's Guide*

TAF Database Configurations

TAF works with the following database configurations to effectively mask a database failure:

- Oracle9*i* Real Application Clusters
- Oracle Real Application Clusters Guard
- Replicated systems
- Standby databases
- Single instance Oracle database

See Also:

- *Oracle9*i* Real Application Clusters Installation and Configuration*
- *Oracle9*i* Data Guard Concepts and Administration*

FAILOVER_MODE Parameters

The FAILOVER_MODE parameter must be included in the CONNECT_DATA section of a connect descriptor. FAILOVER_MODE can contain the sub-parameters described in [Table 15–6](#).

Table 15–6 Sub-Parameters of the FAILOVER_MODE Parameter

FAILOVER_MODE Sub-Parameter	Description
BACKUP	Specify a different net service name for backup connections. A backup should be specified when using preconnect to pre-establish connections.
TYPE (Required)	Specify the type of failover. Three types of Oracle Net failover functionality are available by default to Oracle Call Interface (OCI) applications: <ul style="list-style-type: none"> ■ session: Set to failover the session. If a user's connection is lost, a new session is automatically created for the user on the backup. This type of failover does not attempt to recover selects. ■ select: Set to enable users with open cursors to continue fetching on them after failure. However, this mode involves overhead on the client side in normal select operations. ■ none: This is the default. No failover functionality is used. This can also be explicitly specified to prevent failover from happening.
METHOD	Determines how fast failover occurs from the primary node to the backup node: <ul style="list-style-type: none"> ■ basic: Set to establish connections at failover time. This option requires almost no work on the backup server until failover time. ■ preconnect: Set to pre-established connections. This provides faster failover but requires that the backup instance be able to support all connections from every supported instance.
RETRIES	Specify the number of times to attempt to connect after a failover. If DELAY is specified, RETRIES defaults to five retry attempts.
DELAY	Specify the amount of time in seconds to wait between connect attempts. If RETRIES is specified, DELAY defaults to one second.

Note: Oracle Net Manager does not provide support for TAF parameters. These parameters must be manually added.

TAF Implementation

Important: Do not set the GLOBAL_DBNAME parameter in the SID_LIST_listener_name section of the listener.ora. A statically configured global database name disables TAF.

Depending on the FAILOVER_MODE parameters, you can implement TAF in a number of ways. Oracle recommends the following methods:

- [Example: TAF with Connect-Time Failover and Client Load Balancing](#)
- [Example: TAF Retrying a Connection](#)
- [Example: TAF Pre-Establishing a Connection](#)

Example: TAF with Connect-Time Failover and Client Load Balancing

Implement TAF with connect-time failover and client load balancing for multiple addresses. In the following example, Oracle Net connects randomly to one of the protocol addresses on sales1-server or sales2-server. If the instance fails after the connection, the TAF application fails over to the other node's listener, reserving any SELECT statements in progress.

```
sales.us.acme.com=
  (DESCRIPTION=
    (LOAD_BALANCE=on)
    (FAILOVER=on)
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=sales1-server)
      (PORT=1521))
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=sales2-server)
      (PORT=1521)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)
    (FAILOVER_MODE=
      (TYPE=select)
      (METHOD=basic))))
```

Example: TAF Retrying a Connection

TAF also provides the ability to automatically retry connecting if the first connection attempt fails with the RETRIES and DELAY parameters. In the following example, Oracle Net tries to reconnect to the listener on sales1-server. If the failover connection fails, Oracle Net waits 15 seconds before trying to reconnect again. Oracle Net attempts to reconnect up to 20 times.

```
sales.us.acme.com=
(DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=tcp)
    (HOST=sales1-server)
    (PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)
    (FAILOVER_MODE=
      (TYPE=select)
      (METHOD=basic)
      (RETRIES=20)
      (DELAY=15))))
```

Example: TAF Pre-Establishing a Connection

A backup connection can be pre-established. The initial and backup connections must be explicitly specified. In the following example, clients that use net service name sales1.us.acme.com to connect to the listener on sales1-server are also preconnected to sales2-server. If sales1-server fails after the connection, Oracle Net fails over to sales2-server, preserving any SELECT statements in progress. Likewise, Oracle Net preconnects to sales1-server for those clients that use sales2.us.acme.com to connect to the listener on sales2-server.

```
sales1.us.acme.com=
(DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=tcp)
    (HOST=sales1-server)
    (PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)
    (INSTANCE_NAME=sales1)
    (FAILOVER_MODE=
      (BACKUP=sales2.us.acme.com)
      (TYPE=select)
      (METHOD=preconnect))))
```

```
sales2.us.acme.com=
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=sales2-server)
      (PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=sales.us.acme.com)
      (INSTANCE_NAME=sales2)
      (FAILOVER_MODE=
        (BACKUP=sales1.us.acme.com)
        (TYPE=select)
        (METHOD=preconnect))))
```

TAF Verification

You can query FAILOVER_TYPE, FAILOVER_METHOD, and FAILED_OVER columns in the V\$SESSION view to verify that TAF is correctly configured.

Use the V\$SESSION view to obtain information about the connected clients and their TAF status. For example, query the FAILOVER_TYPE, FAILOVER_METHOD, and FAILED_OVER columns to verify that you have correctly configured TAF as in the following SQL statement:

```
SELECT MACHINE, FAILOVER_TYPE, FAILOVER_METHOD, FAILED_OVER, COUNT(*)
FROM V$SESSION
GROUP BY MACHINE, FAILOVER_TYPE, FAILOVER_METHOD, FAILED_OVER;
```

The output before failover resembles the following:

MACHINE	FAILOVER_TYPE	FAILOVER_M	FAI	COUNT(*)
sales1	NONE	NONE	NO	11
sales2	SELECT	PRECONNECT	NO	1

The output after failover is:

MACHINE	FAILOVER_TYPE	FAILOVER_M	FAI	COUNT(*)
sales2	NONE	NONE	NO	10
sales2	SELECT	PRECONNECT	YES	1

Note: You can monitor each step of TAF using an appropriately configured OCI TAF CALLBACK function.

See Also:

- *Oracle Call Interface Programmer's Guide*
- *Oracle9i Database Reference* for more information about the V\$SESSION view

Specifying the Instance Role for Primary and Secondary Instance Configurations

The INSTANCE_ROLE parameter is an optional parameter for the CONNECT_DATA section of a connect descriptor. It enables you to specify a connection to the primary or secondary instance of Oracle9i Real Application Clusters and Oracle9i Real Application Clusters Guard configurations.

This parameter is useful when:

- You want to explicitly connect to a primary or secondary instance. The default is the primary instance.
- You want to use Transparent Application Failover (TAF) to preconnect to a secondary instance.

INSTANCE_ROLE supports the following values:

primary — Specifies a connection to the primary instance

secondary — Specifies a connection to the secondary instance

any — Specifies a connection to whichever instance has the lowest load, regardless of primary or secondary instance role

Example: Connection to Instance Role Type

In the following example, net service name sales_primary enables connections to the primary instance, and net service name sales_secondary enables connections to the secondary instance.

```
sales_primary=
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=sales1-server)
      (PORT=1521))
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=sales2-server)
      (PORT=1521))
```

```
(CONNECT_DATA=
  (SERVICE_NAME=sales.us.acme.com)
  (INSTANCE_ROLE=primary)))
sales_secondary=
(DESCRIPTION=
(ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales1-server)
  (PORT=1521))
(ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales2-server)
  (PORT=1521))
(CONNECT_DATA=
  (SERVICE_NAME=sales.us.acme.com)
  (INSTANCE_ROLE=secondary)))
```

Example: Connection To a Specific Instance

There are times when Oracle Enterprise Manager and other system management products need to connect to a specific instance regardless of its role to perform administrative tasks. For these types of connections, configure (**INSTANCE_NAME=instance_name**) and (**INSTANCE_ROLE=any**) to connect to the instance regardless of its role.

In the following example, net service name sales1 enables connections to the instance on sales1-server and sales2 enables connections to the instance on sales2-server. (**SERVER=dedicated**) is specified to force a dedicated server connection.

```
sales1=
(DESCRIPTION=
(ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales1-server)
  (PORT=1521))
(CONNECT_DATA=
  (SERVICE_NAME=sales.us.acme.com)
  (INSTANCE_ROLE=any)
  (INSTANCE_NAME=sales2)
  (SERVER=dedicated)))
sales2=
(DESCRIPTION=
(ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales2-server)
  (PORT=1521))
```

```
(CONNECT_DATA=
  (SERVICE_NAME=sales.us.acme.com)
  (INSTANCE_ROLE=any)
  (INSTANCE_NAME=sales2)
  (SERVER=dedicated)))
```

Example: TAF Pre-Establishing a Connection

If Transparent Application Failover (TAF) is configured, a backup connection can be pre-established to the secondary instance. The initial and backup connections must be explicitly specified. In the following example, Oracle Net connects to the listener on sales1-server and preconnects to sales2-server, the secondary instance. If sales1-server fails after the connection, the TAF application fails over to sales2-server, the secondary instance, preserving any SELECT statements in progress.

```
sales1.acme.com=
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=sales1-server)
      (PORT=1521)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)
    (INSTANCE_ROLE=primary)
    (FAILOVER_MODE=
      (BACKUP=sales2.acme.com)
      (TYPE=select)
      (METHOD=preconnect))))
sales2.acme.com=
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=sales2-server)
      (PORT=1521)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)
    (INSTANCE_ROLE=secondary)))
```

Configuring Session Data Unit

Before sending data across the network, Oracle Net buffers and encapsulates data into the SDU. Oracle Net sends the data stored in this buffer when the buffer is full, flushed, or when RDBMS tries to read data. When large amounts of data are being transmitted or when the message size is consistent, adjusting the size of the SDU buffers can improve performance, network utilization, or memory consumption.

The SDU size can range from 512 bytes to 32 KB. The default SDU for the client and the database is 2 KB.

Optimal SDU size depends on the maximum segment size (MSS) and message fragmentation. For TTC connections, configuring an SDU size larger than the 2 KB default requires configuring the SDU on both the client and server computers. When the configured values do not match, the lower of the two values will be used.

To minimize packet header overhead and message fragmentation, set the SDU size as a multiple of the MSS. When Oracle Advanced Security encryption is not used, increase the SDU size by one (1). For example, the TCP/IP version 4 MSS on Ethernet is 1460 bytes. Use a multiple of 1460 for the SDU size if encryption is used. If encryption is not used, increase the SDU size to 1461.

The packet header overhead and message fragmentation can be measured using a network sniffer or by analyzing Oracle Net trace files.

Client Configuration

To configure the client, set the SDU size with the SDU parameter in a connect descriptor as follows:

```
net_service_name=
  (DESCRIPTION=
    (SDU=2920)
    (ADDRESS=...)
    (ADDRESS=...)
    (CONNECT_DATA=
      (SERVER_NAME=sales.us.acme.com)))
```

Database Server Configuration

Database server configuration depends upon whether or not the database is configured to use shared server or dedicated server processes.

Shared Server Configuration

If using shared server processes, set the SDU size in the DISPATCHERS parameter as follows:

```
DISPATCHERS= " (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp))(SDU=2920))"
```

Ensure that the SDU size matches the value configured for the client.

Dedicated Server Configuration

If using dedicated server processes for a database that is dynamically registered with the listener through service registration, then the SDU size cannot be set. Instead, the 2 KB default is used.

If using dedicated server processes for a database that is registered with the listener through static configuration in the `listener.ora` file, then set the SDU size in the `SID_DESC` section of the `listener.ora` file as follows:

```
SID_LIST_listener_name=
  (SID_LIST=
    (SID_DESC=
      (SDU=2920)
      (SID_NAME=sales)))
```

Ensure that the SDU size matches the value configured for the client.

Configuring the VI Protocol for Application Web Server and Oracle Database Server Communication

In the release, Oracle Net provides support for the **Virtual Interface (VI) protocol**. The VI protocol is a standard communication protocol for clustered server environments. VI is an interface between a cluster Local Area Network (cLAN) Host Adapter, a type of network interface card, and the application. By using VI, applications place most of the messaging burden upon the host adapter, freeing the CPU for other tasks. As a result, VI is able to decrease network **latency** and CPU utilization.

VI is designed specifically for System Area Networks (SANs). A SAN is characterized by short-distance, high-performance communications between multiple server systems, such as application Web servers and database servers clustered into one switch.

This section describes how to set up Oracle Net support of VI for application Web server and database server communication. It includes the following topics:

- [Prerequisites to Using VI Protocol Support](#)
- [Configuring a VI Protocol Address](#)
- [Tuning VI](#)

Prerequisites to Using VI Protocol Support

Prior to configuring support for the VI protocol, install the required hardware:

1. Install a cLAN Host Adapter and the drivers provided by the Giganet VI package on both the application Web server and the database server.
2. Connect the cLAN Host Adapters to a cLAN Cluster Switch.

See Also: [Giganet installation documentation](#)

Note: Ensure that the application Web server and the database server are within 30 meters of one another.

VI hardware can be installed before or after an Oracle9*i* Server or Oracle9*i* Client installation that includes VI protocol support. If the hardware is installed after an Oracle installation, perform these steps:

1. Rerun the Oracle Universal Installer.
2. In the Available Products page, select Oracle9*i* Server or Oracle9*i* Client.
3. In the Installation Types page, select Custom.
4. In the Available Product Components page, select only Oracle Net Services.

Configuring a VI Protocol Address

A VI protocol address uses the parameters described in [Table 15–7](#).

Table 15–7 VI Protocol Settings in listener.ora

Oracle Net Manager Field	Address Parameter	Description
Protocol	PROTOCOL	Specify a value of vi.
Disc	DISC	<p>Specify the discriminator number. The DISC is similar to the TCP/IP PORT parameter. The number can be up to 16 digits.</p> <p>For simplicity, Oracle Corporation recommends using 1521 for the VI discriminator number. 1521 will not conflict with TCP/IP port 1521.</p> <p>Note: If the VI discriminator number is not specified, one is automatically generated by the VI protocol. Since you do not know what the number is generated, you cannot specify how to connect to it. For this reason, always set the VI discriminator number.</p>
Host	HOST	Specify the host name of the computer.

To configure a protocol address for VI:

1. Configure a VI protocol address in the `listener.ora` file on the database server:

- a. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

- b. In the navigator pane, expand Local > Listeners.
 - c. Select a listener.
 - d. From the list in the right pane, select Listening Locations.
 - e. Choose Add Address.
- A new Address tab appears.
- f. From the Protocol list, select VI, and enter the appropriate information in the fields supplied.
- g. Choose File > Save Network Configuration.

The `listener.ora` file updates with a VI protocol address. The following example shows a VI endpoint that uses discriminator number 1521 on the computer `sales-server`.

```
listener=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=vi)(HOST=sales-server)(DISC=1521)))
```

2. Create a net service name for the application Web server to use that matches the protocol address information configured in the `listener.ora` file.

The following example shows a net service name of `sales` that contains the same address information configured in the example output in Step 1.

```
sales=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=vi)(HOST=sales-server)(DISC=1521)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)))
```

See Also: [Chapter 9, "Configuring Naming Methods"](#) for more information about creating connect descriptors

Tuning VI

If you find that the performance between the application Web server and database server is not optimal, you can configure the number of buffers to use or use Oracle Connection Manager to reduce the demand on system resources. This section contains these topics:

- [Adjusting the Number of Buffers to Improve VI Performance](#)
- [Using Oracle Connection Manager to Improve VI Performance](#)

Adjusting the Number of Buffers to Improve VI Performance

The number of buffers is set in the `sqlnet.ora` file on the application Web server and database server with the `VI_RECEIVE_BUFFERS` and `VI_SEND_BUFFERS` parameters. The `VI_RECEIVE_BUFFERS` parameter specifies the number of buffers that have been posted on the receive queue for a VI endpoint, and the `VI_SEND_BUFFERS` parameter specifies the number of buffers that have been posted on the send queue for a VI endpoint. By default, these buffers are set to 15. The size of each buffer is the SDU size. The SDU has a default size of 2 KB.

In the following example, each connection will allocate 10 buffers for receiving data and five for sending data and lock the corresponding memory:

```
VI_RECEIVE_BUFFERS=10  
VI_SEND_BUFFERS=5
```

The number of buffers you configure affect the memory needed for the database server and application Web servers. The more buffers configured, the more memory required.

To calculate the memory requirements for the database server, use the following formula:

$$(number_connections * number_app_servers) * SDU_size * (VI_RECEIVE_BUFFERS + VI_SEND_BUFFERS)$$

where:

number_connections Specify the total number or expected connections from each application Web server to the database server.

number_app_servers Specify the expected number of application Web servers.

SDU_size Specify the SDU size configured.

VI_RECEIVE_BUFFERS Specify the number of VI_RECEIVE_BUFFERS configured in the `sqlnet.ora` file on the database server.

VI_SEND_BUFFERS Specify the number of VI_SEND_BUFFERS configured in the `sqlnet.ora` file on the database server.

To calculate the memory requirements for an application Web server, use the following formula:

$$number_connections * SDU_size * (VI_RECEIVE_BUFFERS + VI_SEND_BUFFERS)$$

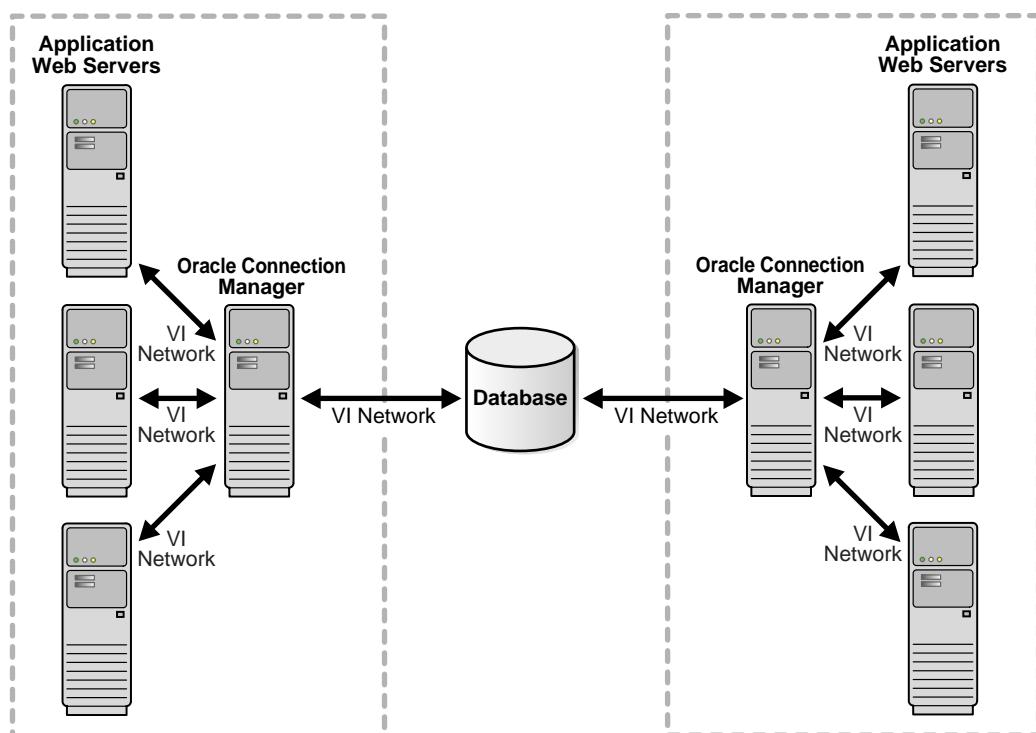
Use the `VI_RECEIVE_BUFFERS` and `VI_SEND_BUFFERS` parameters from the `sqlnet.ora` file on the application Web server.

Using Oracle Connection Manager to Improve VI Performance

To keep the number of connections up to an optimal level without burdening system resources, use Oracle Connection Manager's **session multiplexing** feature to enable the database server to use fewer connection endpoints for incoming requests.

Figure 15–5 shows how Oracle Connection Manager is used to offload some of the network I/O of the application Web servers.

Figure 15–5 Application Web Servers and Oracle Connection Manager



See Also: "Enabling Session Multiplexing" on page 13-13 for more information about configuring session multiplexing

Configuring Connections to Non-Oracle Database Services

The following topics describe how to configure connections to non-Oracle database services:

- [Configuring Oracle Net for External Procedures](#)
- [Configuring Oracle Net for Oracle Heterogeneous Services](#)
- [Configuring Oracle Net for an Oracle Rdb Database](#)

Configuring Oracle Net for External Procedures

External procedures are functions or procedures written in a third-generation language (3GL) that can be called from PL/SQL code. You can configure the listener to listen for external procedure calls. When a PL/SQL or SQL application calls an external procedure, the listener starts an external procedure agent, which by default is named `extproc`. Using the network connection established by the listener, the application passes the following information to the agent:

- Shared library name
- External procedure name
- Parameters (if necessary)

The agent then loads the shared library and invokes the external procedure.

The agent can reside on the same computer as the database server or on a remote computer with a listener.

By default, Oracle Net Configuration Assistant configures connections to external procedure during database server installation. For environments where the connection information does not exist, use the following steps to modify the `listener.ora` file and `tnsnames.ora` files, located in `$ORACLE_HOME/network/admin` on UNIX operating systems and `ORACLE_HOME\network\admin` on Windows:

1. If the agent resides on the database server, configure either a TCP/IP or IPC protocol address in the `listener.ora` file. If the external procedure agent resides on a remote computer, configure a TCP/IP protocol address in the `listener.ora` file. The following procedure describes creating an IPC address for the default external procedure agent `extproc`:

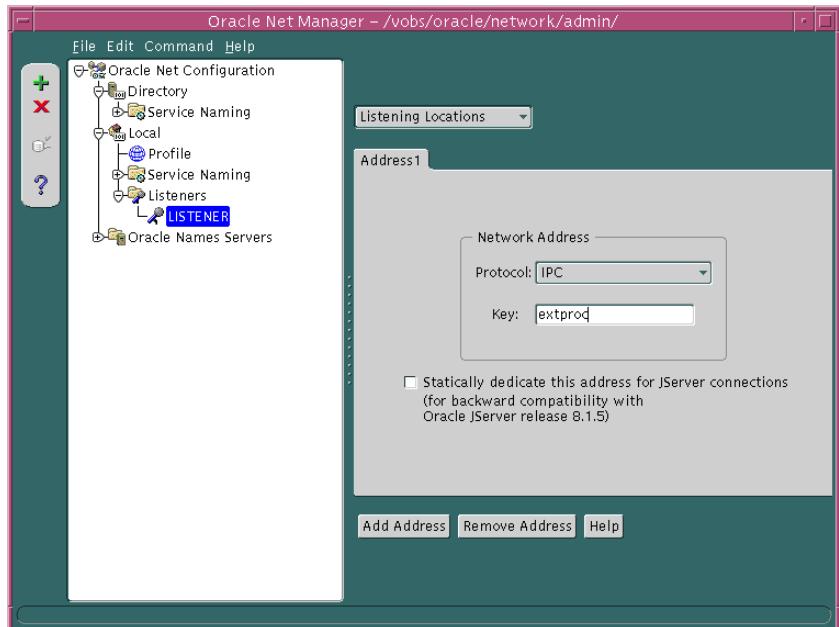
- a. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

- b. In the navigator pane, expand Local > Listeners.

- c. Select a listener.
- d. From the list in the right pane, select Listening Locations.
- e. Choose Add Address.

A new Address tab appears.



- f. From the Protocol list, select IPC, and enter a value for the KEY. Oracle Corporation recommends a KEY value of extproc.

Note: If the computer has more than one Oracle home or more than one listener, each listener must specify a unique KEY. For example, you can use extproc1 for the first listener, extproc2 for the second listener, and so on.

See Also: ["Configuring Listening Protocol Addresses"](#) on page 12-6 for more information about configuring listener protocol addresses

2. Add service information about extproc in the listener.ora file, including the parameters described in Table 15–8.

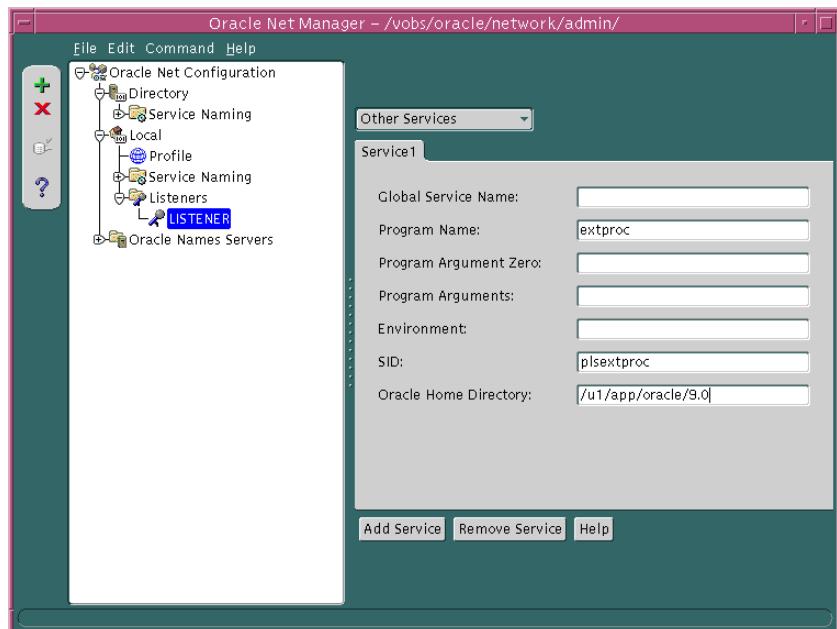
Table 15–8 External Procedures Settings in listener.ora

Oracle Net Manager Field	listener.ora Parameter	Description
SID	SID_NAME	Specify a system identifier for the external procedure agent with any name.
Oracle Home Directory	ORACLE_HOME	Specify the Oracle home location of the agent.
Program Name	PROGRAM	Specify the name of the external procedure agent executable. The executable must reside in the \$ORACLE_HOME/bin directory on UNIX, and the ORACLE_HOME\bin directory on Windows NT.

To configure the SID_NAME, ORACLE_HOME, and PROGRAM parameters:

- a. From the list in the right pane, select Other Services.
- b. Choose Add Service.

A new Service tab appears.



- c. Enter `extproc` in the Program Name field, a system identifier such as `plsextproc` in the SID field, and the Oracle home where the `extproc` executable resides in the Oracle Home field.

If the application requires that an environment variable be set before the agent is started, enter it the Environment field. For example, if the application requires environment variables `MYAPPDIR` for the binary location and `APPID` for the ID, you would enter the following in the Environment field:

```
'MYAPPDIR=/myappdir/bin', 'APPID=MYAPP'
```

See Also: *Oracle Net Services Reference Guide* for further about setting environment variables with `ENVS` parameter in the `listener.ora` file

- d. Choose File > Save Network Configuration.

The `listener.ora` file updates with external procedures, as shown in the following output:

```
LISTENER=
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=ipc)(KEY=extproc)))
SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=plsextproc)
      (ORACLE_HOME=/u1/app/oracle/9.0)
      (PROGRAM=extproc)))
```

3. Create a net service name in the database server `tnsnames.ora` file whose connect descriptor matches the information configured in the `listener.ora` file.

Based on the `listener.ora` file shown in Step 2, the `tnsnames.ora` file would be configured with the following entry:

```
EXTPROC_CONNECTION_DATA=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))
    (CONNECT_DATA=
      (SID=plsextproc)))
```

See Also: "Configuring the Local Naming Method" on page 9-5
for more information about creating a net service name

Configuring Oracle Net for Oracle Heterogeneous Services

Heterogeneous Services are an integrated component within the Oracle database server, and provides the generic technology for accessing non-Oracle systems from the Oracle database server. Heterogeneous Services enable you to:

- Use Oracle SQL to transparently access data stored in non-Oracle systems as if the data resides within an Oracle database server
- Use Oracle procedure calls to transparently access non-Oracle systems, services, or application programming interfaces (APIs), from your Oracle distributed environment

While Heterogeneous Services provides the generic technology in the Oracle database server, a Heterogeneous Service agent is required to access a particular non-Oracle system.

To initiate a connection to the non-Oracle system, the Oracle database server starts an agent process through the listener on the gateway. For the Oracle database server to be able to connect to the agent, perform the following steps:

1. Configure the listener on the gateway to listen for incoming requests from the Oracle database server and spawn Heterogeneous Services agents by configuring the parameters described in [Table 15–9](#) in the `listener.ora` file.

Table 15–9 Heterogeneous Services Settings in `listener.ora`

Oracle Net Manager Field	listener.ora Parameter	Description
SID	SID_NAME	Specify the Oracle System Identifier (SID).
Oracle Home	ORACLE_HOME	Specify the Oracle home location of the agent executable.
Program Name	PROGRAM	Specify the name of the agent executable.

To configure the SID, ORACLE_HOME, and PROGRAM parameters:

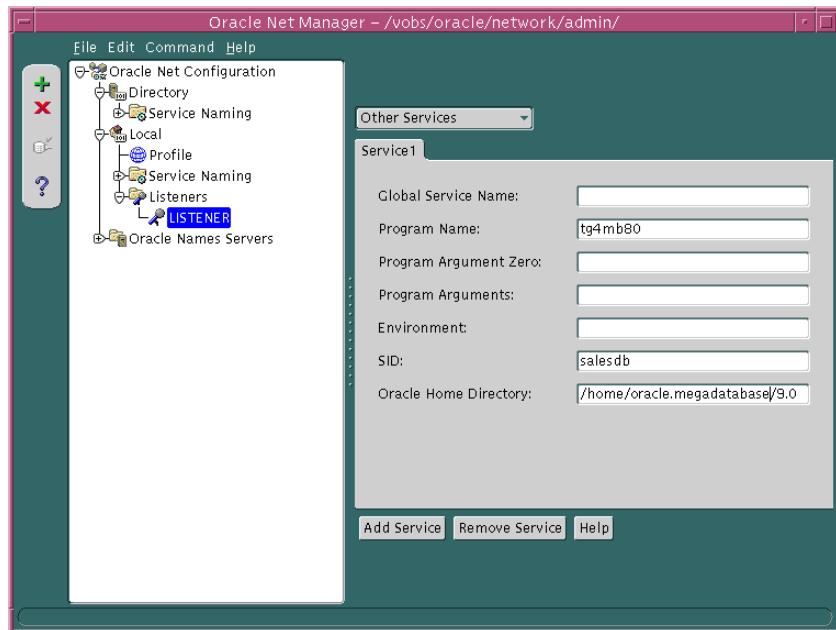
- a. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager"](#) on page 5-3

- b. In the navigator pane, expand Local > Listeners.
- c. Select a listener.
- d. From the list in the right-pane, select Other Services.

- e. Choose Add Service.

A new Service tab appears.



- f. Enter the program name in the Program Names field that gets executed to create a gateway, the Oracle home where the agent executable resides in the Oracle Home Directory field, and the SID or service name of the non-Oracle system in the SID field.

- g. Choose File > Save Network Configuration.

The `listener.ora` file updates information about the Heterogeneous Services, as shown in the following:

```
SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=salesdb)
      (ORACLE_HOME=/home/oracle/megabase/9.0)
      (PROGRAM=tg4mb80) ))
```

See Also: *Oracle9i Heterogeneous Connectivity Administrator's Guide*

2. On the computer where the Oracle database resides, set up a net service name to connect to the listener on the gateway. The connect descriptor must also include the HS=ok clause to make sure the connection uses Heterogeneous Services:

- a. Start Oracle Net Manager.

See Also: "[Starting Oracle Net Manager](#)" on page 5-3

- b. Create a net service name that can be used for connections from the Oracle database server to a non-Oracle system.

See Also: "[Task 1: Configure Net Service Names](#)" on page 9-6 for net service name creation for local naming

- c. In the navigator pane, expand Local > Service Naming.
 - d. Select the net service name.
 - e. Choose Advanced in the Service Identification group box.

The Advanced Service Options dialog box appears.



- f. Choose Use for Heterogeneous Services, and then choose OK.
 - g. Choose File > Save Network Configuration.

The `tnsnames.ora` file updates with the new net service name configured for Heterogeneous Services, as shown in the following:

```
megabase6_sales=
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=tcp) (HOST=dlsun206) (PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=sales6)
      (HS=ok)))
```

Configuring Oracle Net for an Oracle Rdb Database

Oracle Rdb is a database for Digital's 64-bit operating systems. Because Oracle Rdb has its own listener, the client interacts with Rdb in the same manner as it does with an Oracle database.

To initiate a connection to an Oracle Rdb, set up a net service name to connect to the Oracle Rdb database using the parameters described in [Table 15–10](#).

Table 15–10 Oracle RDB Database Settings in a Connect Descriptor

tnsnames.ora Parameter	Oracle Net Manager Field	Description
RDB_DATABASE	RDB Database	Specify the file name of an Oracle RDB database. Embed this parameter under the CONNECT_DATA parameter.
TYPE_OF_SERVICE	Type of Service	Specify the type of service to use for an Oracle RDB database. It is used by Rdb interface tools. This feature should only be used if the application supports both an Oracle Rdb and Oracle database, and you want the application to randomly choose (load balance). Embed this parameter under the DESCRIPTION parameter.
GLOBAL_NAME	Global Database Name	(Optional) Specify the Oracle Rdb database. Embed this parameter under the CONNECT_DATA parameter.

See Also: Oracle Rdb documentation

To configure a client for an Oracle Rdb database, use Oracle Net Manager:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. Create a net service name that can be used for connections from the Oracle server to a non-Oracle system.

See Also: ["Task 1: Configure Net Service Names" on page 9-6](#) for net service name creation for local naming

3. In the navigator pane, expand Directory or Local > Service Naming.
4. Select the net service name.
5. Choose Advanced in the Service Identification group box.

The Advanced Service Options dialog box appears.



See Also: [Table 15-10](#) on page 15-44 for a description of the Oracle Rdb fields

6. Enter the file name of an Oracle Rdb database in the Rdb Database field.
7. Optionally, enter the global database name in the Global Database Name field, and, if needed, specify the type of service in the Type of Service field, and then choose OK.

8. Choose File > Save Network Configuration.

The `tnsnames.ora` file updates with the new net service name configured for the Oracle Rdb database, as shown in the following:

```
alpha5=
  (DESCRIPTION=
    (ADDRESS=...)
    (CONNECT_DATA=
      (SERVICE_NAME=generic)
      (RBD_DATABASE=[.mf]mf_personnel.rdb)
      (GLOBAL_NAME=alpha5)))
```

In the following example, `TYPE_OF_SERVICE` is used to load balance between an Oracle Rdb database service and Oracle database service:

```
alpha5=
  (DESCRIPTION_LIST=
    (DESCRIPTION=
      (ADDRESS=...)
      (CONNECT_DATA=
        (SERVICE_NAME=generic)
        (RBD_DATABASE=[.mf]mf_personnel.rdb)
        (GLOBAL_NAME=alpha5)))
    (DESCRIPTION=
      (ADDRESS=...)
      (CONNECT_DATA=
        (SERVICE_NAME=sales.us.acme.com))
      (TYPE_OF_SERVICE=oracle9_database)))
```

See Also: Oracle Rdb documentation

Part III

Testing and Troubleshooting Oracle Net Services

Part III describes how to establish connections, and identify and diagnose problems with Oracle Net Services.

This part contains these chapters:

- [Chapter 16, "Establishing a Connection and Testing the Network"](#)
- [Chapter 17, "Troubleshooting Oracle Net Services"](#)

Establishing a Connection and Testing the Network

Once you have completed configuring the network, you should make a connection and test each component to ensure that the network is functioning properly. Oracle Net Services provide a variety of tools to help you start, test, and control an Oracle Names server, listener, and Oracle Connection Manager.

This chapter outlines procedures to make a connection and test network components. This chapter contains these topics:

- [Connecting to a Database](#)
- [Testing the Network](#)

Connecting to a Database

Connecting to a database involves starting network components and entering a connect string with a net service name, such as the following:

```
CONNECT username/password@connect_identifier
```

This section contains these topics:

- [Starting Oracle Net Services Components](#)
- [Entering a Connect String](#)
- [Connect Identifier and Connect Descriptor Syntax Characteristics](#)
- [Absolute Name Specification for Directory Naming](#)
- [Initiating Connections](#)

Starting Oracle Net Services Components

Client workstations and other servers connect to a listener with a net service name when logging onto an Oracle database server.

After installing and configuring all the network components, you need to start them to make the network functional. Following is an outline of the tasks you should perform to start the network components.

[Task 1: Start Oracle Names Servers](#)

[Task 2: Start Oracle Names Client Cache](#)

[Task 3: Start the Listener](#)

[Task 4: Start the Database](#)

[Task 5: Start Oracle Connection Manager](#)

Note: If the Oracle Names servers use a database to store the network information, you need to start the database and the listener prior to starting the Oracle Names servers.

Task 1: Start Oracle Names Servers

Start Oracle Names servers using either Oracle Net Manager or Oracle Names Control utility on the computers where Oracle Names server software is installed and configured. To start Oracle Names servers, use either Oracle Net Manager or Oracle Names Control utility:

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none">1. Start Oracle Net Manager. See Also: "Starting Oracle Net Manager" on page 5-32. In the navigator pane, expand the Oracle Names server folder.3. Select the Oracle Names server.4. From the list in the right pane, select Manage Server from the list box.5. Choose the Control tab.6. Choose the Start radio button from the Server Operations field.7. Choose Apply.	<p>From the command line, enter:</p> <pre>namesctl NAMESTL> START</pre> <p>The START command of the Oracle Names Control utility loads the Oracle Names server into memory and tells it to begin executing. At startup, the Oracle Names server loads its configuration and data.</p>

On Windows NT, an Oracle Names server can also be started through the Control Panel:

1. Double-click the Services icon in the Control Panel window.
2. Select the `Oraclehome_nameNamesonames_server` service.
3. Choose Start to start the service.
4. In the Services window, choose Close.

Task 2: Start Oracle Names Client Cache

To avoid clients looking up address information in an Oracle Names server each time, create a client cache of the information.

See Also: ["Task 5: \(Optional\) Configure Client Caches"](#) on page 9-38 for more information about client caches

To create a client cache:

1. Ensure discovery of Oracle Names servers has been performed, as described in ["Task 4: Configure Clients and Database Servers To Use Oracle Names Servers"](#) on page 9-34. The information gathered during discovery is used for a client's cache.
2. Create and start the client cache on the client:

```
namesctl  
NAMESTL> START_CLIENT_CACHE
```

On Windows NT, the client cache can also be started through the Control Panel:

1. Double-click the Services icon in the Control Panel window.
2. Select the `Oraclehome_nameClientCache` service.
3. Choose Start to start the service.
4. In the Services window, choose Close.

Task 3: Start the Listener

For Oracle Net to accept connections on the database server, start the listener with the Listener Control utility on the server:

1. Determine the status of the listener. From the command line, enter:

```
lsnrctl  
LSNRCTL> STATUS [listener_name]
```

where *listener_name* is the name of the listener defined in the *listener.ora* file. It is not necessary to identify the listener if you are using the default listener, named LISTENER.

If the STATUS command indicates that the listener is running, go to Step 2. If the listener is not running, go to Step 3.

Even if the listener is running, stop the listener, and start it again before proceeding to Step 2. To stop the listener, enter:

```
LSNRCTL> SET PASSWORD password  
LSNRCTL> STOP [listener_name]
```

SET PASSWORD is only required if the password is set in the *listener.ora* file. The password defaults to ORACLE.

2. Start the listener. Enter:

```
LSNRCTL> START [listener_name]
```

The Listener Control utility will display a status message indicating that the listener has started successfully. Check that all expected services for that listener are listed in the services summary in the status message.

3. Exit from the Listener Control utility. Enter:

```
LSNRCTL> EXIT
```

On Windows NT, the listener can also be started through the Control Panel:

1. Double-click the Services icon in the Control Panel window.
2. Select the *Oraclehome_nameTNSListener* service—the service name if you are using the default listener name LISTENER—or *Oraclehome_nameTNSListenerlsnr*, where *lsnr* is the nondefault listener name.
3. Choose Start to start the service.
4. In the Services window, choose Close.

Task 4: Start the Database

Use the tool of choice, such as SQL*Plus, to start the database:

1. Start SQL*Plus without connecting to the database:

```
sqlplus /nolog
```

2. Connect to Oracle as SYSDBA:

```
SQL> CONNECT username/password as sysdba
```

3. When you enter a STARTUP command, specify the database name and full path of the parameter file:

```
SQL> STARTUP database_name pfile=file
```

If you do not specify the PFILE option, the Oracle database uses the standard initialization parameter file located in the \$ORACLE_BASE/admin/*db_name/pfile/sid* directory on UNIX platforms, and *ORACLE_BASE\admin\db_name\pfile\sid* directory on Windows NT. If you do not specify a database name, then the database uses the value of the DB_NAME parameter specified in the initialization parameter file.

See Also: *Oracle9i Database Administrator's Guide* for further information about starting the database

Task 5: Start Oracle Connection Manager

If Oracle Connection Manager is installed and configured, start it with the Oracle Connection Manager Control utility, entering commands in the following order:

1. From the command line, enter:

```
CMCTL  
CMCTL> START cman
```

Oracle Connection Manager displays a status message indicating that Oracle Connection Manager has started successfully.

2. Exit from the Oracle Connection Manager Control utility. Enter:

```
CMCTL> EXIT
```

On Windows NT, the listener may also be started through the Control Panel:

1. Double-click the Services icon in the Control Panel window.
2. If you are using Oracle Names, select the `Oraclehome_nameCMAdminService` to acquire information about available Oracle Names Servers, then click Start. If you are not using Oracle Names, do not start this service.

The service starts.

3. Select the `Oraclehome_nameCManService` to start the Oracle Connection Manager, and click Start.

The service starts.

4. In the Services window, click Close.

Entering a Connect String

After the network components are started, as described in "[Connecting to a Database](#)" on page 16-2, you should be able to make a connection across the network. How you make a connection depends upon the naming method you configured in [Chapter 9, "Configuring Naming Methods"](#), and the tool used for the connection.

The connect strings takes the following basic form:

```
CONNECT username/password@connect_identifier
```

On most operating systems, you can define a default connect identifier that is resolved by a Oracle Net naming method. This way, a connect identifier does not need to be specified in the connect string. To define a default connect identifier, use the TWO_TASK environment variable on UNIX platforms and the LOCAL environment variable or registry entry on Windows NT.

For example, if the TWO_TASK environment variable is set to sales, you connect to a database from SQL*Plus with the following syntax:

```
CONNECT username/password
```

Oracle Net checks if TWO_TASK is set and uses the value sales as the connect identifier.

See Also: Operating system documentation for instructions on setting TWO_TASK and LOCAL

Further information about connect string format is provided in the following topics:

- [Connect Identifier and Connect Descriptor Syntax Characteristics](#)
- [Absolute Name Specification for Directory Naming](#)

Connect Identifier and Connect Descriptor Syntax Characteristics

Connect identifiers and connect descriptors used in a connect string cannot contain spaces, unless enclosed within single quotes ('') or double quotes (""). In the following examples, a connect identifier and a connect descriptor that contain spaces are enclosed within single quotes:

```
CONNECT scott/tiger@' (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)
(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)))'
```

```
CONNECT scott/tiger@'cn=sales, cn=OracleContext, dc=us, dc=acme, dc=com'
```

Single quotes ('') are required if a double quote (") is used in a connect identifier. For example:

```
CONNECT scott/tiger@'sales@Good"Fast"Food.com'
```

Likewise, double quotes (") are required if a single quote (') is used in a connect identifier. For example:

```
CONNECT scott/tiger@"cn=sales, cn=OracleContext, ou=Mary's Dept, o=acme"
```

Absolute Name Specification for Directory Naming

Note: JDBC OCI drivers support absolute naming. JDBC Thin drivers support absolute naming only when the complete DN is used. See the *Oracle9i JDBC Developer's Guide and Reference* for further information.

This section describes how to configure absolute names for the following namespaces:

- [Absolute Names for X.500 Namespaces](#)
- [Absolute Names for Domain Component Namespaces](#)

Absolute Names for X.500 Namespaces For X.500 namespaces, the default directory entry defined for the client must be in one of the following formats:

[ou],o

[ou],o,c

where [ou] represents an optional organizationalUnitName.

The absolute name the client uses as the connect identifier must be in one of the following formats:

cn[.ou].o

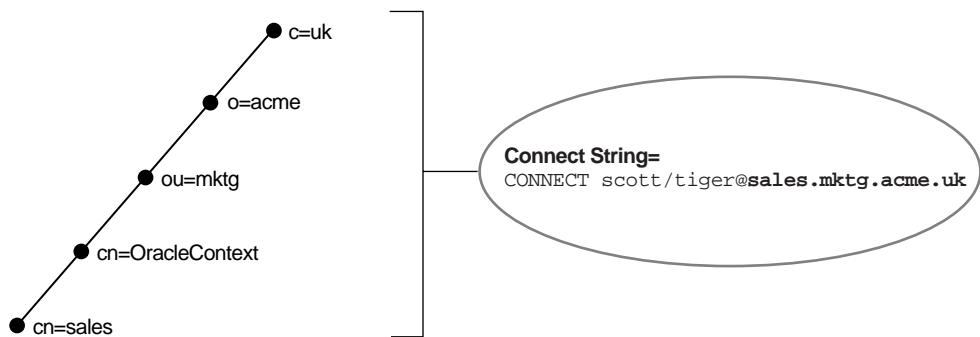
cn[.ou].o.c

where [cn] represents the Oracle Net entry.

For example, consider a client that has been configured with a default directory entry of ou=acctg,o=acme,c=us.

The directory contains an entry for a database called sales with a DN of (dn:cn=sales,cn=OracleContext,ou=mktg,o=acme,c=uk). In this scenario, the client requires a connect identifier of sales.mktg.acme.uk (cn.ou.o.c). [Figure 16-1](#) depicts this example.

Figure 16-1 Absolute Name for X.500 Namespaces



Absolute Names for Domain Component Namespaces For domain component namespaces, the default directory entry defined for the client must be in one of the following formats:

`dc[,dc][...]`

`ou,dc[,dc][...]`

where `[dc]` represents an optional domain component and `[...]` represents additional domain component entries.

The absolute name the client must use in the connect identifier must be in one of the following formats:

`cn.dc[.dc][...]`

`cn[.ou]@dc[.dc][...]`

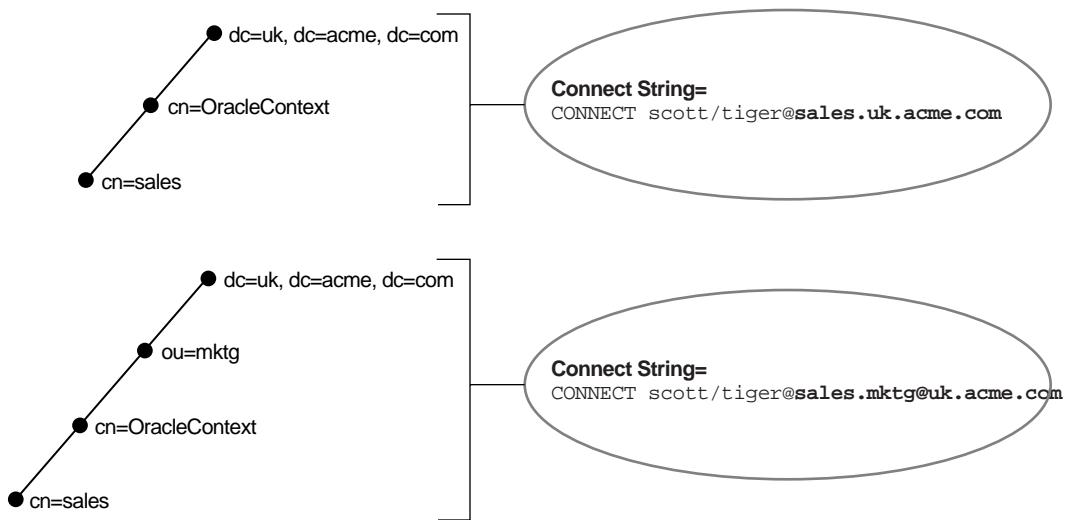
where `[cn]` represents the Oracle Net entry.

Example 1 Consider a client that has been configured with a default directory entry of `dc=us,dc=acme,dc=com`.

The directory server contains an entry for a database called `sales` with a DN of `(dn:cn=sales,cn=OracleContext,dc=uk,dc=acme,dc=com)`. In this scenario, the client requires a connect identifier of `sales.uk.acme.com` (`cn.dc.dc.dc`). [Figure 16-2](#) on page 16-12 depicts this example.

Example 2 Consider the same default directory entry as Example 1. The directory server contains an entry for a database called `sales` with a DN of `(dn:cn=sales,cn=OracleContext,ou=mktg,dc=uk,dc=acme,dc=com)`. Notice `ou=mktg`. Because domain components must be separated from organization units, the client requires a connect identifier of `sales.mktg@uk.acme.com` (`cn.ou@dc.dc.dc`). [Figure 16-2](#) on page 16-12 depicts this example.

Figure 16–2 Absolute Name for Domain Component Namespaces



Initiating Connections

There are a number of ways to initiate a connection to an Oracle server. Commonly used methods are described in these topics:

- [Connecting from the Operating System to Test a Client](#)
- [Connecting from the Tool Logon Screen to Test a Client](#)
- [Connecting from 3GL to Test a Client](#)
- [Connecting Using Special Commands Within Tools](#)

The specifics of use are slightly different in each case. Each of the general methods listed is briefly covered here. To identify the method used in a specific tool, refer to the user guide for the tool.

Connecting from the Operating System to Test a Client

The general form of connecting an application to a database server from the command line is:

```
tool username/password@net_service_name
```

For example:

```
SQLPLUS system/password@sales
```

To prevent the password from displaying during a logon, you can leave out the password parameter on the command line. For example:

```
SQLPLUS system@sales
```

You will be prompted to enter your password without it showing on screen.

Most Oracle tools can use the operating system command line to connect; some provide alternatives.

Connecting from the Tool Logon Screen to Test a Client

Some tools provide a logon screen as an alternative form of logon. A user can log on to a database server by identifying both the user name and net service name (*username@net_service_name*) in the user name field of the tool logon screen, and entering the password as usual in the password field.

Connecting from 3GL to Test a Client

In applications written using 3GL, the program must establish a connection to a server using the following syntax:

```
exec sql connect :username identified by :password
```

In this connection request, `:username` and `:password` are 3GL variables that can be set within the program either statically or by prompting the user. When connecting to a database server, the value of the `:username` variable is in the form:

`username@net_service_name`

The `:password` variable contains the password for the database account to which you are connecting.

Connecting Using Special Commands Within Tools

Some Oracle tools have commands for database connection, once the tool has been started, to allow an alternative username to be specified without leaving the tool. SQL*Plus allows the CONNECT command using the following syntax:

```
SQL> CONNECT username/password@net_service_name
```

For example:

```
SQL> CONNECT scott/tiger@serverx
```

This is very similar to the operating system command line method, except that it is entered in response to the tool prompt instead of the operating system prompt.

Other Oracle tools use slightly different methods specific to their function or interface. For example, Oracle CDE tools use logon buttons and a pop-up window with the username, password, and remote database ID field.

Testing the Network

The preferred sequence for testing the network is as follows:

1. Start and test each Oracle Names server (if included in your network layout).
2. Start and test each listener.
3. Start and test each Oracle Connection Manager (if included in your network layout).
4. Test the server with a loopback test.
5. Test client with a connection.

This section contains these topics:

- [Testing a Listener](#)
- [Testing Oracle Connection Manager](#)
- [Testing Configuration on the Database Server](#)
- [Testing Network Connectivity from the Client](#)
- [Testing an Oracle Names Server](#)
- [Testing Network Objects with Oracle Net Manager or Oracle Names Control Utility](#)

Testing a Listener

To test a listener, initiate a connection from a client to any active database controlled by that listener, as described in "[Testing Configuration on the Database Server](#)" on page 16-16.

Testing Oracle Connection Manager

To test Oracle Connection Manager, initiate a connection from a client to any active database for which a source route address has been created.

Testing Configuration on the Database Server

Once you have configured the network, test the configuration by performing a **loopback test** on the database server.

A loopback test uses Oracle Net to go from the database server back to itself, bypassing the Interprocess Communication (IPC). Performing a successful loopback verifies that Oracle Net is functioning on the database server.

To perform the loopback test, use Oracle Net Manager:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator, expand Directory or Local > Service Naming.
3. Select the net service name or database service.
4. Choose Command > Test Net Service.

Testing assumes the database and listener are running. If they are not, see ["Starting Oracle Net Services Components" on page 16-2](#) to start components.

During testing, a Connection Test dialog box appears, providing status and test results. A successful test results in the following message:

The connection test was successful.

If the test was successful, proceed to Step 5.

If the test was not successful:

- a. Ensure that the database and listener are running, and then choose Test.
 - b. Choose Change Login to change the user name and password for the connection, and then choose Test.
5. Choose Close to dismiss the Connect Test dialog box.

Testing Network Connectivity from the Client

To test several different clients in your network, initiate a connection to a database server from each of them by following the instructions in "[Entering a Connect String](#)" on page 16-8.

Oracle Net also provides the following tools to help evaluate network connectivity:

- [TNSPING Utility](#)
- [TRCROUTE Utility](#)
- [Oracle Net Manager](#)

TNSPING Utility

The TNSPING utility determines whether or not a service (for example, an Oracle database, an Oracle Names server, or any other Oracle service) on a Oracle Net network can be successfully reached.

If you can connect successfully from a client to a server (or a server to another server) using the TNSPING utility, it displays an estimate of the round trip time (in milliseconds) it takes to reach the Oracle Net service.

If it fails, it displays a message describing the error that occurred. This enables you to see the network error that is occurring without the overhead of a database connection.

Using TNSPING To invoke the TNSPING utility, enter the following:

```
tnsping net_service_name [count]
```

Note: Different platforms may have different interfaces, but the program accepts the same arguments. Invoke TNSPING for the display of the proper interface requirements.

- *net_service_name*: must exist in `tnsnames.ora` file or the name service in use, such as NIS or DCE's CDS.
- *count* (optional): determines how many times the program attempts to reach the server.

If the net service name specified is a database name, TNSPING attempts to contact the corresponding listener. It does not actually determine whether or not the database itself is running. Use SQL*Plus to attempt a connection to the database.

Following are some examples of TNSPING.

Example: Reaching a Database with TNSPING To connect to a database using a net service name of spotdb, the following is entered:

```
TNSPING spotdb
```

This produces the following message:

```
TNS Ping Utility for SunOS:  
Copyright (c) Oracle Corporation 1998. All rights reserved.  
Attempting to contact  
(ADDRESS=(PROTOCOL=TCP)(HOST=spot)(PORT=1521))  
OK (50msec)
```

To determine whether a connection can be made to the stprd database, and to specify that TNSPING try to connect 10 times and then give up, use the following command:

```
tnsping stprd 10
```

This command produces the following message:

```
TNS Ping Utility for SunOS:  
Copyright (c) Oracle Corporation 1998. All rights reserved.  
Attempting to contact (ADDRESS=(PROTOCOL=TCP)(HOST=spot)(PORT=1521))  
OK (290 msec)  
OK (100 msec)  
OK (70 msec)  
OK (70 msec)  
OK (60 msec)  
OK (70 msec)  
OK (70 msec)  
OK (80 msec)  
OK (180 msec  
OK (340 msec)
```

Example: Invalid Net Service Name with TNSPING Below is an example of TNSPING attempting to connect to an invalid net service name:

```
tnsping bad_db
```

This attempt produces the following message:

```
TNS Ping Utility for SunOS:  
Copyright (c) Oracle Corporation 1998. All rights reserved.  
TNS-03505: Failed to resolve name
```

Example: Valid Net Service Name with TNSPING Following is an example of using TNSPING to connect to a name that is valid, but that resolves to an address where no listener is located (for example, the listener may not be started):

```
tnsping testing
```

The following message is returned:

```
TNS Ping Utility for SunOS:  
Copyright (c) Oracle Corporation 1998. All rights reserved.  
Attempting to contact (ADDRESS=(PROTOCOL=tcp)(HOST=spot)(PORT=1521))  
TNS-12541: TNS:no listener
```

Example: Reaching an Oracle Names Server with TNSPING To check whether a Oracle Names server can be reached, use a command using the Oracle Net address as in the following:

```
tnsping (ADDRESS=(PROTOCOL=tcp)(HOST=fido)(PORT=1575))
```

A message similar to the following will be returned to the user:

```
TNS Ping Utility for SunOS:  
Copyright (c) Oracle Corporation 1998. All rights reserved.  
Attempting to contact (ADDRESS=(PROTOCOL=TCP)(HOST=fido)(PORT=1575))  
OK (70 msec)
```

TRCROUTE Utility

The Trace Route Utility (TRCROUTE) enables administrators to discover the path or route a connection is taking from a client to a server. If TRCROUTE encounters a problem, it returns an error stack to the client instead of a single error. These additional error messages make troubleshooting easier.

TRCROUTE is different from TNSPING in that it travels as a special type of connect packet, and is routed as such. As it travels toward its destination, the TRCROUTE connect packet collects the TNS addresses of every node it travels through. If an error occurs, TRCROUTE collects error information that shows where the error occurred. The Trace Route Utility displays the information collected on the client screen. You can redirect the TRCROUTE output to a file, and print it if you wish.

Effect on Performance The Trace Route Utility uses minimal resources. It gathers information in the connect data of a special connect packet; standard connect packets are not affected.

The server is not affected by TRCROUTE. The listener receives and processes the TRCROUTE connect packet. It returns the information to the client by putting it into a refuse packet. The server does not need to start up any new processes or deal with dummy connections.

Using TRCROUTE To invoke TRCROUTE, enter the following from the command line:

```
trcroute net_service_name
```

The following are two examples of trace route output.

Example: Successful Trace Route

The following example shows a successful Trace Route packet that traveled from a client to a listener.

```
trcroute tcp_direct
Trace Route Utility for Solaris:
Copyright (c) Oracle Corporation 1998. All rights reserved.

Route of TRCROUTE:-----
Node: Client           Time and address of entry into node:
-----
01-DEC-98 13:26:36 ADDRESS= PROTOCOL=TCP Host=shining-sun Port=1581
Node: Server           Time and address of entry into node:
-----
01-DEC-98 13:27:20 ADDRESS= PROTOCOL=TCP Host=setting-sun Port=1521
```

Example: Trace Route with Error

The following examples shows an unsuccessful Trace Route packet that could not reach the listener because the listener was not up.

```
trcroute tcp_direct
Trace Route Utility for SVR4:
Copyright (c) Oracle Corporation 1998. All rights reserved.

Route of TRCROUTE:-----
Node: Client           Time and address of entry into node:
-----
01-DEC-98 11:12:34 ADDRESS= PROTOCOL=TCP Host=shining-sun Port=1581
TNS-12224: TNS:no listener
TNS-12541: TNS:no listener
TNS-12560: TNS:protocol adapter error
TNS-03601: Failed in route information collection
```

Oracle Net Manager

To verify connectivity for a client computer, use Oracle Net Manager:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator, expand Directory or Local > Service Naming.
3. Select the net service name or database service.
4. Choose Command > Test Net Service.

Testing assumes that the database and listener are running. If they are not, see ["Starting Oracle Net Services Components" on page 16-2](#) to start components.

During testing, a Connection Test dialog box appears, providing status and test results. A successful test results in the following message:

The connection test was successful.

If the test was successful, proceed to Step 5.

If the test was not successful:

- a. Ensure that the database and listener are running, and then choose Test.
- b. Choose Change Login to change the user name and password for the connection, and then choose Test.

5. Choose Close to dismiss the Connect Test dialog box.

Testing an Oracle Names Server

To test an Oracle Names server, use the Oracle Names Control utility PING command. Following are two ways to PING the Oracle Names server labrador in the us.acme domain.

From the Oracle Names Control utility prompt, enter:

```
NAMESCTL> PING labrador.us.acme
```

You can test several Oracle Names servers with the same PING command. For example:

```
NAMESCTL> PING huey.uk.acme duey.uk.acme louie.uk.acme
```

PING responds with the time it takes to contact the Oracle Names server and return an acknowledgment. If PING fails, make sure the Oracle Names server is started or double-check the configured address of the Oracle Names server.

Testing Network Objects with Oracle Net Manager or Oracle Names Control Utility

The information stored in an Oracle Names server can be queried to verify registration with the `QUERY` command.

To use the `QUERY` command, use either Oracle Net Manager or Oracle Names Control utility:

Use Oracle Net Manager...	Use Oracle Names Control utility...
<ol style="list-style-type: none">1. Start Oracle Net Manager. <i>See Also: "Starting Oracle Net Manager" on page 5-3</i>2. Double-click the Oracle Names server folder.3. Select an Oracle Names server.4. Choose Manage Data from the drop-down list box.5. Choose the Advanced tab.6. Select Query.7. Enter the name of the object to query in the Name field.8. Optionally, enter the record type of the object in the Name field: A . SMD: Network address -CNAME . SMD: Alias -DL . RDBMS . OMD: Global database link -DLCR . RDBMS . OMD: Link qualifier -NS . SMD: Oracle Names server. System data used to communicate between Oracle Names servers -V1ADD . NPO . OMD: SQL*Net Version 1 connect descriptor9. Choose Execute.10. Choose File > Save Network Configuration.	<p>For a Oracle Names server in the administrative region, create and register an alias with the Oracle Names server:</p> <p>To display all data:</p> <pre>namesctl NAMESTL> QUERY name *</pre> <p>The record type of an object may also be requested. Common record types include:</p> <ul style="list-style-type: none">-A . SMD: Network address-CNAME . SMD: Alias-DL . RDBMS . OMD: Global database link-DLCR . RDBMS . OMD: Link qualifier-NS . SMD: Oracle Names server. System data used to communicate between Oracle Names servers-V1ADD . NPO . OMD: SQL*Net Version 1 connect descriptor <p>To display the database address for a net service name:</p> <pre>namesctl NAMESTL> QUERY name a.smd</pre> <p>The following example shows a query of the net service name <code>sales</code>:</p> <pre>namesctl NAMESTL> QUERY sales a.smd</pre> <p>The <code>QUERY</code> command returns the amount of time the transaction took and information about the network object.</p>

Troubleshooting Oracle Net Services

Oracle Net Services provide methods for understanding and resolving network problems through the use of log and trace files. These files keep track of the interaction between network components as errors occur. Evaluating this information will help you to diagnose and troubleshoot even the most complex network problems.

This chapter describes common network errors and outlines procedures for resolving them. It also describes methods for logging and tracing error information to diagnose and troubleshoot more complex network problems. This chapter contains these topics:

- [Diagnosing Oracle Net](#)
- [Resolving the Most Common Error Messages for Oracle Net Services](#)
- [Troubleshooting Tips from the Field for Oracle Net Services](#)
- [Troubleshooting Network Problems Using Log and Trace Files](#)
- [Logging Error Information for Oracle Net Services](#)
- [Oracle Net Tracing Error Information](#)
- [Contacting Oracle Support Services](#)

Diagnosing Oracle Net

If an attempt to make a basic peer-to-peer (single protocol network) connection returns an *ORA Error*, this section may help you diagnose the cause of the problem.

Any underlying fault, noticeable or not, is reported by Oracle Net with an error number or message that is not always indicative of the actual problem. This section helps you determine which parts of Oracle Net do function properly rather than the parts that do not work. It also helps you to decide in which of the following categories the fault belongs:

- Oracle software
- Operating system layer
- Other network layers

Testing the various network layers progressively should in most cases uncover any problem.

Server Diagnostics

Note: You may need assistance from your server administrator to follow the instructions in this section.

Answer the following questions:

- Is any other system (workstation/server) able to connect to the server using Oracle Net?
- Has the server, database, or listener configuration remained the same for some time?

If you answered YES to any of the above questions/statements, skip this section and continue to "[Client Diagnostics](#)" on page 17-4.

If you are unsure, or answered NO to any of the above questions, please continue.

Diagnosing Oracle Net on the server involves the following tasks:

- [Task 1: Verify the Database Is Running](#)
- [Task 2: Perform a Loopback Test](#)

Task 1: Verify the Database Is Running

To check that the database is up, login to the database and connect with a valid user name and password. For example:

```
SQLPLUS system/manager
```

A message appears, confirming that you are connected with the database. If you receive the following errors, ask your Database Administrator to assist you:

- ORA-1017: invalid U/P
- ORA-1034: Oracle not available

Task 2: Perform a Loopback Test

To perform a **loopback test** from the server to the database:

1. Ensure that the `listener.ora`, `tnsnames.ora`, and `sqlnet.ora` files exist in the `$ORACLE_HOME/network/admin` directory on UNIX, or the `ORACLE_HOME\network\admin` directory on Windows NT.

The search order for configuration files is as follows:

- a. For the `sqlnet.ora` file, current working directory from where an application is executed
- b. Directory specified by the `TNS_ADMIN` environment variable
If the `TNS_ADMIN` environment variable is not defined as a variable on Windows NT, it may be in the registry.
- c. The node's global configuration directory. For Sun Solaris, this directory is `/var/opt/oracle`. Windows NT does not have a central directory.

See Also: Operating system-specific documentation

- d. `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows operating systems
2. Follow the instructions in "[Testing Configuration on the Database Server](#)" on page 16-16 to perform a loopback test.
 - If the loopback test continues to fail, continue to the next step.
 - If the loopback test passes, skip to "[Client Diagnostics](#)".
 3. Check the **Problem/Solution Database** Web site at <http://support.oracle.com> for more specific information on the error received, or contact Oracle Worldwide Support.

Client Diagnostics

At this point, you know the serverside listener works properly, because you could verify at least one of the following statements:

- The server passed a loopback test, showing that the connection worked.
- Other computers connect also using Oracle Net to this same database.
- Connections from this workstation worked previous to making changes on this computer, such as the installation of a new product or a modification to the network configuration.

To perform diagnostics on the client:

1. Check that you have installed the same Oracle protocol(s) as were installed on the server.

On UNIX, you can run the `adapters` program to verify this. Run `adapters` at `$ORACLE_HOME/bin`.

Output similar to the following appears:

Installed Oracle Net Services Transport Protocols are:

```
IPC
TCP/IP
BEQueath
SSL
RAW
```

2. Check base connectivity for underlying network transport. Oracle Net technology depends on the underlying network for a successful connection.

Protocol	Verify that you can...
TCP/IP	Use terminal emulation or file transfer utilities, (PING, FTP, TELNET) from the client to the server.
Named Pipes	<ul style="list-style-type: none"> ▪ See other computers or servers on the Microsoft network. ▪ Ensure you are able to share drives within the network.

3. To ensure that both the Oracle Net foundation layer and the appropriate Oracle protocol support are present, verify that all Oracle Net Services software for the client has been installed.
4. Ensure that the client computer has the `tnsnames.ora` and the `sqlnet.ora` files in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows operating systems.

The search order for `sqlnet.ora` and `tnsnames.ora` follows:

- a. For the `sqlnet.ora` file, current working directory from where an application is executed
- b. Directory specified by the `TNS_ADMIN` environment variable

If the `TNS_ADMIN` environment variable is not defined as a variable on Windows NT, it may be in the registry.

- c. The node's global configuration directory. For Sun Solaris, this directory is `/var/opt/oracle`. Windows NT does not have a central directory.

See Also: Operating system-specific documentation

- d. `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows operating systems

If you have any other working client computers connecting to the selected Oracle database using Oracle Net, back up your existing files and copy both the working `tnsnames.ora` and `sqlnet.ora` files from the working computer onto the non-working client workstations. This eliminates the possibility of errors in the files.

5. Test the Oracle Net foundation layer as described in "[Testing Network Connectivity from the Client](#)" on page 16-17.

Note: Do *not* use the TNSPING utility. The TNSPING utility works like the TCP/IP PING utility and does *not* create and open a socket, nor does it connect with the listener. It ensures that the listener is present on the database server.

6. If the connection still fails:

- Use tracing as described in section "[Troubleshooting Network Problems Using Log and Trace Files](#)" on page 17-20
- Check the **Problem/Solution Database** Web site at <http://support.oracle.com> for a specific diagnostics bulletin on the error received
- Contact Oracle Support Services

Resolving the Most Common Error Messages for Oracle Net Services

Due to the complexity of network communications, network errors may originate from a variety of sources, for a variety of reasons. If an error occurs, applications such as SQL*Plus, that depend on network services from Oracle Net Services, will normally generate an error message.

A list of the most common network error messages follows:

- ORA-12154: TNS:could not resolve service name
- ORA-12198: TNS:could not find path to destination
- ORA-12203: TNS:unable to connect to destination
- ORA-12224: TNS:no listener
- ORA-12514: TNS:listener could not resolve SERVICE_NAME given in connect descriptor
- ORA-12520: TNS:listener could not find available handler for requested type of server
- ORA-12521: TNS:listener could not resolve INSTANCE_NAME given in connect descriptor
- ORA-12533: TNS:illegal ADDRESS parameters
- ORA-12545: TNS:name lookup failure
- ORA-12560: TNS:protocol adapter error
- ORA-3113: TNS:End of file on communication channel
- ORA-3121: No interface driver connection - function not performed

See Also: *Oracle9i Database Error Messages* for a complete listing of error messages

ORA-12154: TNS:could not resolve service name

Cause: Oracle Net could not locate the net service name specified in the `tnsnames.ora` configuration file.

Action: Perform these steps:

1. Verify that a `tnsnames.ora` file exists.

See Also: Step 4 on page 17-5 in "[Client Diagnostics](#)" on page 17-4 for configuration file location information

2. Verify that there are not multiple copies of the `tnsnames.ora` file.
3. In the `tnsnames.ora` file, verify that the net service name specified in your connect string is mapped to a connect descriptor.
4. Verify that there are no duplicate copies of the `sqlnet.ora` file.
5. If you are using domain names, verify that your `sqlnet.ora` file contains a `NAMES.DEFAULT_DOMAIN` parameter. If this parameter does not exist, you must specify the domain name in your connect string.
6. If you are not using domain names, and this parameter exists, delete it or disable it by commenting it out.
7. If you are connecting from a login dialog box, verify that you are not placing an "@" symbol before your connect net service name.
8. Activate client tracing and repeat the operation.

Cause: Oracle Net could not locate the database service name or net service name specified in the directory server.

Action: Perform these steps:

1. Verify that the database service or net service name entry exists in the directory that this computer was configured to use.

See Also: [Chapter 8, "Setting Up Directory Server Usage"](#) for directory setup instructions

2. Verify that the `sqlnet.ora` file includes the following entry:

`NAMES.DIRECTORY_PATH=(ldap, other_naming_methods)`

ORA-12198: TNS:could not find path to destination**ORA-12203: TNS:unable to connect to destination**

Cause: The client cannot find the desired database.

Action: Perform these steps:

1. Verify that you have entered the net service name you wish to reach correctly.
2. Verify that the net service name ADDRESS parameters in the connect descriptor.
3. If using local naming, verify that the `tnsnames.ora` file is stored in the correct directory.

See Also: Step 4 on page 17-5 in "[Client Diagnostics](#)" for configuration file location information

4. Verify that the listener on the remote node has started and is running. Enter:

```
lsnrctl  
LSNRCTL> STATUS [listener_name]
```

listener_name is the name of the listener defined in the `listener.ora` file. It is not necessary to identify the listener if you are using the default listener, named LISTENER.

If the output indicates the listener is not running, try starting it with the command:

```
LSNRCTL> START [listener_name]
```

5. If you are connecting from a login box, verify that you are not placing an "@" symbol before your connect net service name.

ORA-12203: TNS:unable to connect to destination

Cause: ORA-12203 error is a generic error that often shields secondary errors.

Action: Check the latest `sqlnet.log` file for secondary ORA messages.

Cause: An invalid net service name was supplied in the connect string.

Action: Verify that the net service name supplied in the connect string exists in the `tnsnames.ora` file or directory server and the ADDRESS information for that net service name is valid. Ask yourself the following questions:

- Is the `SERVICE_NAME` correct?
- Is the `HOST` correct?
- Is the `PORT` specified correct?

Cause: The `tnsnames.ora` file is not located in the proper directory.

Action: Make sure the `tnsnames.ora` file is in the proper location.

See Also: Step 4 on page 17-5 in "[Client Diagnostics](#)" for configuration file location information

Cause: The (`HOST=server_name`) parameter for TCP/IP addresses is not consistent on the client and server computers.

Action: Ensure that the values for these parameter are the same on the server and client.

For TCP/IP, make sure that the `HOST` parameter in `listener.ora` on the server and in the `tnsnames.ora` file on the client point to the same name, or at least to names that are then translated to the same IP address by each system. This is especially important for servers with multiple IP addresses assigned to the various network interfaces on the server.

Cause: The destination system's listener is not listening.

Action: Verify that the remote system's listener is running. Enter:

```
lsnrctl  
LSNRCTL> STATUS [listener_name]
```

`listener_name` is the name of the listener defined in the `listener.ora` file. It is not necessary to identify the listener if you are using the default listener, named `LISTENER`.

If the output indicates the listener is not running, try starting it with the command:

```
LSNRCTL> START [listener_name]
```

Cause: There are underlying network transport problems.

Action: Use the utilities supplied with the underlying networking protocol to verify that the protocol itself is functional. For example, with TCP/IP, try to ping the remote system.

Cause: The incorrect Oracle protocol for the selected networking protocol is installed. A missing protocol support driver usually produces the following errors in the `sqlnet.log` or any client trace file:

- ORA-12203
- ORA-12538
- ORA-00508

Action: Check that you have installed the appropriate Oracle protocol. On UNIX, you can run the `adapters` program to verify this. Run `adapters` at `$ORACLE_HOME/bin`.

Output similar to the following displays:

```
Installed Oracle Net Transport Protocols are:
```

```
IPC  
TCP/IP  
BEQueath  
SSL  
RAW
```

```
...
```

ORA-12224: TNS:no listener

Cause: The connection request could not be completed because the listener is not running.

Action: Perform these steps:

1. Ensure that the supplied destination address matches one of the addresses used by the listener.
2. Verify that this is not a version compatibility problem.

ORA-12533: TNS:illegal ADDRESS parameters

Cause: The protocol specific parameters in the ADDRESS section of the designated connect descriptor are incorrect.

Action: Correct the protocol address.

See Also: *Oracle Net Services Reference Guide* for correct protocol syntax

ORA-12514: TNS:listener could not resolve SERVICE_NAME given in connect descriptor

Cause: The service name specified in the connect descriptor is incorrect, or the database service is not registered with the listener.

Action: Perform these steps:

1. Check to make sure the SERVICE_NAME specified in the connect descriptor is correct.
2. Ensure the database instance is running. If the instance not running, start it so that it can register with the listener. You can use the Listener Control utility SERVICES command to see what services are currently registered with the listener.

See Also: ["SERVICES Command"](#) on page 12-23

ORA-12520: TNS:listener could not find available handler for requested type of server

Cause: The type of service handler requested by the client is incorrect or not registered for the requested SERVICE_NAME/INSTANCE_NAME, or the database instance is not registered with the listener.

Action: If you suspect the problem is the wrong type of service handler, perform these steps:

1. If (`SERVER=value`) is set in the connect descriptor, ensure that the value is set to the appropriate service handler type for the database, that is, `dedicated` for dedicated server or `shared` for dispatchers. You can use the Listener Control utility SERVICES command to see what service handlers are currently registered with the listener.

See Also: ["SERVICES Command"](#) on page 12-23

2. If USE_DEDICATED_SERVER is set to on in the sqlnet.ora file, then ensure the database is configured to use dedicated servers. If it is not, set this parameter to off.
3. Ensure that the database instance is running. If the instance not running, start it so that it can register with the listener.

ORA-12521: TNS:listener could not resolve INSTANCE_NAME given in connect descriptor

Cause: The INSTANCE_NAME in the connect descriptor is incorrect, or the database instance is not registered with the listener.

Action: Perform these steps:

1. Check to make sure the service name specified in the connect descriptor is correct.
2. Ensure the database instance is running. If the instance not running, start it so that it can register with the listener. You can use the Listener Control utility SERVICES command to see what instances are currently registered with the listener.

See Also: ["SERVICES Command" on page 12-23](#)

ORA-12545: TNS:name lookup failure

Cause: The listener on the remote node cannot be contacted.

Action: Perform these steps:

1. Verify that the ADDRESS in the tnsnames.ora file and the listener.ora file is correct.
2. Verify that the listener on the remote node has been started. Enter:

```
lsnrctl  
LSNRCTL> STATUS [listener_name]
```

listener_name is the name of the listener defined in the listener.ora file. It is not necessary to identify the listener if you are using the default listener, named LISTENER.

If the output indicates the listener is not running, try starting it with the command:

```
LSNRCTL> START [listener_name]
```

ORA-12560: TNS:protocol adapter error

Action: The listener was unable to start a process connecting the user to the database server.

Cause: Perform these steps:

1. Turn on tracing and repeat the operation.
2. Evaluate the contents of the trace file to diagnose the problem.

See Also: ["Oracle Net Tracing Error Information" on page 17-39](#)

ORA-3113: TNS:End of file on communication channel

Cause: An error has occurred on the database server.

Action: Check the `alert_sid.log` on the server. The location of `alert_sid.log` is specified by the `BACKGROUND_DUMP_DEST` initialization parameter.

Cause: An unexpected end of file was processed on the communication channel. This may be an indication that the communications link may have gone down at least temporarily; it may indicate that the server has gone down.

Action: You may need to modify your retransmission count.

See Also: Operating system-specific documentation for more information about troubleshooting this error

ORA-3121: No interface driver connection - function not performed

Cause: A SQL*Net version 1 prefix was erroneously used in the connect string.

Action: Do not use the following prefixes in the connect string.

- T:
- X:
- P:

Cause: Only the user name and password were specified from a client computer that had no local Oracle database installed.

Action: Specify a connect string.

Troubleshooting Directory Naming Errors

Directory naming issues associated with connectivity errors such as ORA-12154, ORA-12203, or ORA-12224 for database service or net service name entries in a directory server require analysis of the data. You can analyze the data contained within a directory server with the `ldifwrite` command line tool.

`ldifwrite` enables you to convert all or part of the information residing in a directory server to **LDAP Data Interchange Format (LDIF)**. The `ldifwrite` tool performs a subtree search, including all entries below the specified **distinguished name (DN)**, including the DN itself.

The `ldifwrite` tool syntax is as follows:

```
ldifwrite -c net_service_name/database_service -b base_DN -f ldif_file
```

Table 17–1 *Idapwrite Arguments*

Argument	Description
<code>-c net_service_name/database_service</code>	Specify the net service name or database service name that will connect you to the directory server.
<code>-b base_DN</code>	Specify the base of the subtree to be written out in LDIF format.
<code>-f ldif_file</code>	Specify the input file name.

The following example writes all the Oracle Net Services entries under `dc=us,dc=acme,dc=com` into the `output1.ldif` file:

```
ldifwrite -c ldap -b "dc=us,dc=acme,dc=com" -f output.ldif
```

Oracle Names LDAP Proxy Server Error Reporting

Errors in the region load operation will be reported in the Oracle Names server log file (`names.log`). These errors may range from failure to contact the directory server to errors with the query for all, some, or one of the records.

Some directories, such as Oracle Internet Directory, have limits on `ldapsearch` operations. There are settings in the directory server that limit the number of objects returned by the search and the amount of time spent performing a search.

Increasing Search Size Limit

The size limit specifies how many objects can be returned from a search. The default limit is 1000. If this limit is exceeded, you will see the following errors in the `names.log` file:

```
NNO-00062: cannot load domain data from configuration database  
NNO-00850: Error: LDAP query returns 4
```

You can also use the `ldapsearch` command line tool to mimic what the Oracle Names server will do when it loads its region. The following syntax shows loading data from DN (`dn:dc=acme,dc=com`):

```
ldapsearch -p 389 -h host -b "dc=acme,dc=com"  
"(objectclass=orclNetService)(objectclass=orclService)"
```

After returning the allowed number of object, `ldapsearch` returns the following error message:

```
ldapsearch: Sizelimit exceeded
```

You can modify the size limit using the following sample LDIF file output. Enter the appropriate DN. In addition, set `orclsizelimit` high enough to allow for the number of databases defined in the region in the directory server, with a little room for future expansion.

```
dn:  
changetype: modify  
replace: orclsizelimit  
orclsizelimit: 5000
```

Increasing the Search Time Limit

The time limit specifies the amount of time that can be spent performing a search. The default time limit is 10 seconds. Ten seconds is sufficient to query for roughly 1,000 object, which is sufficient for most searches. If the query exceeds the time limit, you will see the following errors in the names.log file:

```
NNO-00062: cannot load domain data from configuration database  
NNO-00850: Error: LDAP query returns 105
```

You can modify the time limit using the following sample LDIF file output. Enter the appropriate DN.

```
dn:  
changetype: modify  
replace: orcltimelimit  
orcltimelimit: 20
```

The time limit is applied at both the directory server and API levels. Therefore, in addition to resetting the directory server time limit, you will also need to set the TIMEOUT sub-parameter of NAMES.ADMIN_REGION. For example:

```
NAMES.ADMIN_REGION=  
(REGION=  
  (TIMEOUT=20)  
  (TYPE=ldap)  
  (HOST=dlsun1598)  
  (PORT=389)  
  (SUBTREE=(BASE=dc=acme,dc=com)))
```

Troubleshooting Tips from the Field for Oracle Net Services

Here are some tips you may find helpful when you are having difficulty diagnosing network problems:

- **Use the node or network address during configuration instead of the name of the server computer**

This eliminates any internal lookup problems and make the connection slightly faster.

- **If you are using TCP/IP addresses, use the IP address rather than host name**

For example, change the
(HOST=*server_name*) line in the `tnsnames.ora` file with the internet address, for example (HOST=198.32.3.5).

- **Perform a loopback test**

Perform a loopback test on the server as described in ["Testing Configuration on the Database Server"](#) on page 16-16. If the test passes, `ftp` the `tnsnames.ora` and `sqlnet.ora` files to the client.

- **Check what is between you and the server**

If it is a wide area network (WAN), identify any intermediate systems that may not work correctly. If all computers are fine, the problem may be a timing issue.

- **Verify Whether or Not There Is a Timing Issue**

Timing issues are associated with ORA-12203, ORA-12535, or ORA-12547 errors in the client log files.

To resolve this, try speeding up the connection by using exact addresses instead of names and increase the `CONNECT_TIMEOUT_Listener_name` parameter in the `listener.ora` file. The default value for this parameter is 10 seconds.

- **Determine which Oracle applications are failing**

`SQL*Plus` may work, but CASE tools may not. If you determine the problem is a data volume issue, try to transfer a large (5 MB) file with the base connectivity.

Questions to Ask When Troubleshooting Oracle Net Services

Here are some questions to ask yourself when diagnosing a problem:

- **Do all computers have a problem, or is it just one?**

If one computer works and another does not, and you are confident that the same software (Oracle and third-party products) is installed, on each computer, swap out the network cables, if they are close enough, to see if the problem moves. If it does move, it indicates that the problem has something to do with the client/server connection and is not local to the PC.

- **What kind of links exist between the client and the server, for example, X.25, ISDN, Token Ring, or leased line?**

Sniffers and LAN analyzers are useful for intermittent failing connections or detecting time-outs and re-sent packets. You can also see what side of the conversation is waiting for a response.

Troubleshooting Network Problems Using Log and Trace Files

Oracle Net Services provide detailed information about the source and context of problems as they arise. This information is generated and stored in log and trace files. The process of logging and tracing error information will help you to diagnose and resolve network problems.

Logging Error Information for Oracle Net Services

All errors encountered in Oracle Net Services are appended to a log file for evaluation by a network or database administrator. The log file provides additional information for an administrator when the error message on the screen is inadequate to understand the failure. The log file, by way of the error stack, shows the state of the software at various layers.

To ensure that all errors are recorded, logging cannot be disabled on clients or Names Servers. Furthermore, only an administrator may replace or erase log files. The log file for the listener also includes Audit Trail information about every client connection request, as well as most listener control commands.

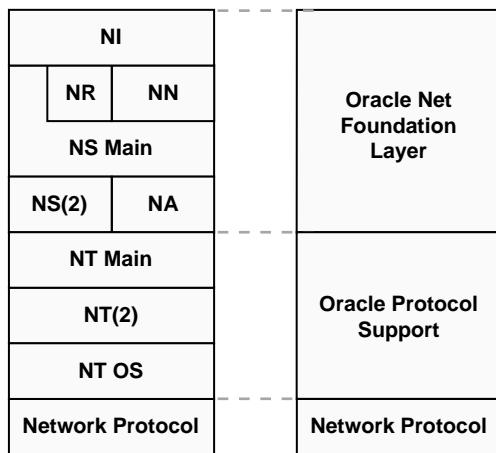
This section contains these topics:

- [Oracle Net Error Stacks](#)
- [Oracle Net Services Log File Names](#)
- [Setting Oracle Net Services Log Parameters](#)
- [Setting Oracle Net Log Settings During Runtime of Control Utilities](#)
- [Using Oracle Net Log Files](#)
- [Listener Log Audit Trail Information](#)
- [Analyzing Oracle Connection Manager Logs](#)

Oracle Net Error Stacks

Log files provide information contained in an error stack. An error stack refers to the information that is produced by each layer in an Oracle communications stack as the result of a network error. [Figure 17–1](#) depicts the relationship among error stack components and Oracle Net.

Figure 17–1 Error Stack Components Mapped to Oracle Net



The error stack components in [Figure 17–1](#) are described in [Table 17–2](#).

Table 17–2 Error Stack Components

Error Stack Component	Description
NI	Network Interface. This layer provides a generic interface for Oracle clients, servers, or external processes to access Oracle Net functions. The NI layer handles the "break" and "reset" requests for a connection. NI uses the Network Routing (NR) layer to obtain network route information for pre-Oracle9 <i>i</i> clients, and the Network Naming (NN) layer to resolve names to connect descriptors. For Oracle9 <i>i</i> clients, NI goes directly to the Network Session (NS) layer.
NN	Network Naming (Oracle Names). This layer resolves connect identifiers to connect descriptors.

Error Stack Component	Description
NS	Network Session (main and secondary layers). These layers receive requests from NI, and settle all generic computer-level connectivity issues, such as: the location of the server or destination (open, close functions); whether one or more protocols will be involved in the connection (open, close functions); and how to handle interrupts between client and server based on the capabilities of each (send, receive functions).
	NS uses NR to route the network session to the destination and Network Authentication (NA) to negotiate any authentication requirements with the destination.
NR	Network Routing. This layer routes the network session to the destination.
NA	Network Authentication. This layer negotiates authentication and encryption requirements.
NT	Network Transport (main, secondary, and operating system layers). These layer map Oracle Net foundation layer functionality to industry-standard protocols.

Example: Error Stack

As an example, suppose that a user of a client application tries to establish a connection with a database server using Oracle Net and TCP/IP, and the user enters:

```
sqlplus scott/tiger@hrserver.com
```

The following error displays:

```
ORA-12203: TNS:Unable to connect to destination
```

This message indicates that the connection to the server failed because the database could not be contacted. Although the application displays only a one-line error message, an error stack that is much more informative is recorded in the log file by the network layer.

On the client side, a `sqlnet.log` file ([Figure 17–2](#)) contains an error stack corresponding to the ORA-12203 error.

Figure 17–2 sqlnet.log File

```
*****
Fatal OSN connect error 12203, connecting to:
  (DESCRIPTION=(CONNECT_DATA=(SID=trace)(CID=(PROGRAM=
    (HOST=lala)(USER=sviavant)))(ADDRESS_LIST=(ADDRESS=
      (PROTOCOL=ipc)(KEY=trace))(ADDRESS=(PROTOCOL=tcp)
      (HOST=lala)(PORT=1521)))) 

VERSION INFORMATION:
TNS for SunOS:
Oracle Bequeath NT Protocol Adapter for SunOS:
Unix Domain Socket IPC NT Protocol Adaptor for SunOS:
TCP/IP NT Protocol Adapter for SunOS:
  Tracing to file: /home/sviavant/trace_admin.trc
  Tns error struct:
    nr err code: 12203
    TNS-12203: TNS:unable to connect to destination
    ns main err code: 12541
    TNS-12541: TNS:no listener
    ns secondary err code: 12560
    nt main err code: 511
    TNS-00511: No listener
    nt secondary err code: 61
    nt OS err code: 0
```

Oracle Net Services Log File Names

Each Oracle Net Services component produces its own log file. [Table 17–3](#) provides the default file names and a description of the information they contain.

Table 17–3 Oracle Net Log Files

Log File	Contains error information about the...
sqlnet.log	Client or database server
listener.log	Listener
names.log	Oracle Names Server
cman_pid.log on UNIX cmanpid.log on Windows NT	Oracle Connection Manager CMGW gateway process
cmadm_pid.log on UNIX cmadmpid.log on Windows NT	Oracle Connection Manager CMADMIN administrative process

Setting Oracle Net Services Log Parameters

Parameters that control logging, including the type and amount of information logged, as well as the location where the files are stored, are set in the configuration file of each network component as described in [Table 17–4](#).

Table 17–4 Oracle Net Log Parameters

Network Component	Configuration Files
Client	sqlnet.ora
Database Server	sqlnet.ora
Listener	listener.ora
Oracle Names Server	names.ora
Oracle Connection Manager	cman.ora

This section contains these topics:

- [sqlnet.ora Log Parameters](#)
- [listener.ora Log Parameters](#)
- [names.ora Log Parameters](#)
- [cman.ora Log Parameters](#)
- [Setting Oracle Net Log Parameters in Configuration Files](#)

See Also: *Oracle Net Services Reference Guide* for more information about these parameters

sqlnet.ora Log Parameters

Table 17–5 describes the log parameters settings that can be set in the `sqlnet.ora` file.

Table 17–5 sqlnet.ora Log Parameters

sqlnet.ora Parameter	Oracle Net Manager Field	Description
LOG_DIRECTORY_CLIENT	Client Information: Log Directory	Establishes the destination directory for the client log file. By default, the client directory is the current working directory.
LOG_FILE_CLIENT	Client Information: Log File	Sets the name of the log file for the client. By default the log name is <code>sqlnet.log</code> .
LOG_DIRECTORY_SERVER	Server Information: Log Directory	Establishes the destination directory for the database server log files. By default the server directory is <code>\$ORACLE_HOME/network/log</code> on UNIX and <code>ORACLE_HOME\network\log</code> on Windows NT.
LOG_FILE_SERVER	Not applicable	Sets the name of the log file for the database server. By default the log name is <code>sqlnet.log</code> .

listener.ora Log Parameters

Table 17–6 describes the log parameters settings that can be set in the `listener.ora` file.

Table 17–6 *listener.ora Log Parameters*

listener.ora Parameter	Oracle Net Manager Field	Description
<code>LOG_DIRECTORY_listener_name</code> and <code>LOG_FILE_listener_name</code>	Log File	Establishes the destination directory and file for the log file that is automatically generated for listener events. By default the directory is <code>\$ORACLE_HOME/network/log</code> on UNIX and <code>ORACLE_HOME\network\log</code> on Windows NT, and the file name is defaulted to <code>listener.log</code> .

names.ora Log Parameters

Table 17–7 describes the log parameters settings that can be set in the `names.ora` file.

Table 17–7 *names.ora Log Parameters*

names.ora Parameter	Oracle Net Manager Field	Description
<code>NAMES.LOG_DIRECTORY</code>	Log Directory	Establishes the destination directory for log files. By default, the directory is <code>\$ORACLE_HOME/network/log</code> on UNIX and <code>ORACLE_HOME\network\log</code> on Windows NT.
<code>NAMES.LOG_FILE</code>	Log File	Sets the name of the log file for the client. By default the log name is <code>names.log</code> .

cman.ora Log Parameters

Table 17–8 describes the log parameters settings that can be set in the `cman.ora` file.

Table 17–8 cman.ora Log Parameters

cman.ora Parameter	Description
LOG_LEVEL	<p>Establishes the level of logging. Five levels are supported:</p> <ul style="list-style-type: none"> ■ 0—No log output ■ 1—Basic reporting ■ 2—RULE_LIST matching lookup reporting ■ 3—Relay blocking reporting ■ 4—Relay I/O counts reporting <p>The CMGW gateway process creates a log file called <code>cman_pid.log</code> on UNIX and <code>cmanpid.log</code> on Windows NT. The CMADMIN administrative process creates a log file called <code>cmadm_pid.log</code> on UNIX and <code>cmadmpid.log</code> on Windows NT.</p>

Setting Oracle Net Log Parameters in Configuration Files

Logging parameters for the `sqlnet.ora` file, `listener.ora` files and `names.ora` file can be set with the Oracle Net Manager. The `cman.ora` file logging parameters must be set manually.

See Also: *Oracle Net Services Reference Guide*

To set logging parameters:

1. Start Oracle Net Manager.

See Also: "Starting Oracle Net Manager" on page 5-3

2. Specify the log parameters:

Log File	Set logging parameters here...
sqlnet.log	<ol style="list-style-type: none">1. In the navigator pane, expand Local > Profile.2. From the list in the right pane, select General.3. Choose the Logging tab.4. Specify the settings.
listener.log	<ol style="list-style-type: none">1. In the navigator pane, expand Local > Listeners.2. Select a listener.3. From the list in the right pane, select General.4. Choose the Logging and Tracing tab.5. Specify the settings.
names.log	<ol style="list-style-type: none">1. In the navigator pane, expand Oracle Names Servers.2. Select an Oracle Names server.3. From the list in the right pane, select Configure Server.4. Choose the Adv. tab.5. Specify the log directory and file name.

3. Choose File > Save Network Configuration.

Setting Oracle Net Log Settings During Runtime of Control Utilities

Logging can be set during runtime of control utilities. Note that setting logging with a control utility will not set parameters in the *.ORA files; the setting is only valid for the session of the control utility:

- For a listener, use the `SET LOG_FILE` and `SET LOG_DIRECTORY` commands from the Listener Control utility.
- For an Oracle Connection Manager, use the `SET LOG_LEVEL` command from the Oracle Connection Manager control utility.
- For an Oracle Names server, use the `SET LOG_FILE_NAME` command from the Oracle Names Control utility, or set logging settings through Oracle Net Manager.

See Also: *Oracle Net Services Reference Guide*

To set tracing for an Oracle Names server with Oracle Net Manager:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator, expand the Oracle Names Servers folder.
3. Select an Oracle Names server.
4. From the list in the right pane, select Manage Server.
5. Choose the Logging tab.
6. Specify the log directory and file name.
7. Choose File > Save Network Configuration.

Using Oracle Net Log Files

To use a log file to diagnose a network error:

1. Review the log file for the most recent error number you received from the application. Note that this is almost always the last entry in the log file.
2. Starting from the bottom of the file, locate the first nonzero entry in the error report. This is usually the actual cause.
3. If that error does not provide the desired information, review the next error in the stack until you locate the correct error information.
4. If the cause of the error is still not clear, turn on tracing and repeat the statement that produced the error message.

Analyzing Listener Log Files

This section describes what is recorded in the listener log file, including:

- [Listener Log Audit Trail Information](#)
- [Listener Service Registration Event Information](#)
- [Listener Direct Hand-Off Information](#)

Listener Log Audit Trail Information

The listener log file contains audit trail information that enables you to gather and analyze network usage statistics, as well as information indicating the following:

- A client connection request
- A RELOAD, START, STOP, STATUS, or SERVICES command issued by the Listener Control utility

Format of the Listener's Log Audit Trail

The Audit Trail formats text into the following fields:

Timestamp * Connect Data * [Protocol Info] * Event * [SID | Service] * Return Code

Properties of the Audit Trail are as follows:

- Each field is delimited by an asterisk (*).
- Protocol address information and SID or service name information appear only when a connection is attempted.
- A successful connection or command returns a code of zero.
- A failure produces a code that maps to an error message.

See Also: ["Resolving the Most Common Error Messages for Oracle Net Services"](#) on page 17-7 for the most common Oracle Net errors or [Oracle9i Database Error Messages](#) for a complete listing of error messages

Example: Listener Log Event for Successful Reload Request

Typical output to the log file upon a RELOAD request follows.

```
14-SEP-1999 00:29:54 *
(connect_
data=(cid=(program=) (host=dlsun1013) (user=jdoe) ) (command=stop) (arguments=64) (ser
vice=listener) (version=135290880))
* stop * 0
```

Example: Listener Log Event for Successful Connection Request

Typical output to the log file upon a connection request follows.

```
10-AUG-1999 15:28:58 *
(connect_data=(service_name=sales.us.acme.com) (cid=(program=) (host=dlsun1013)
(user=jdoe)))
* (address=(protocol=tcp) (host=144.25.185.246) (port=41349)) * establish
* sales.us.acme.com * 0
```

Using Audit Trail Information

You can use Audit Trail information to view trends and user activity by first storing it in a table and then collating it into a report format. To import the data into a table, use an import utility such as SQL*Loader.

Listener Service Registration Event Information

The listener records service registration events. During service registration, the instance background process **PMON process** provides the listener with information about the following:

- Service names for each running instance of the database
- Instance names of the database
- Service handlers (dispatchers or dedicated servers) available
- Dispatcher, instance, and node load information
- Dynamic listening endpoints

The following service registration-related events are recorded in the `listener.log` file:

Table 17–9 Service Registration Event Log Information

Event	Description
service_register	The listener received registration information for an instance.
service_update	The listener received updated registration information for a particular instance, such as dispatcher or instance load information.
service_died	The listener lost its connection to PMON. All registration information for the instance is discarded. Clients will be unable to connect to the instance until PMON registers it again.

Format of the Listener Service Registration Information

The service registration events are formatted into the following fields:

Timestamp * Event * Instance Name * Return Code

Properties of service registration fields are as follows:

- Each field is delimited by an asterisk (*).
- It is normal for the events to appear multiple times in a row for one instance.
- A successful registration returns a code of zero, meaning the client can connect to the instance.
- A failure produces a code that maps to an error message.

See Also:

- ["Resolving the Most Common Error Messages for Oracle Net Services" on page 17-7](#) for the most common Oracle Net errors
- *Oracle9i Database Error Messages* for a complete listing of error messages

Example: Listener Log with Service Registration Events

The following example shows a log file with service registration events. Notice how the listener is able to receive a client request after a successful `service_register` event, but is unable to receive client requests after a `service_died` event.

```
-----
10-AUG-1999 15:28:43 * service_register * sales * 0
10-AUG-1999 15:28:43 * service_register * sales * 0
10-AUG-1999 15:28:58 *
(connect_data=(service_name=sales.us.acme.com)(cid=(program=)(host=dlsun1013)
(user=jdoe)))
* (address=(protocol=tcp)(host=144.25.185.246)(port=41349)) * establish
* sales.us.acme.com * 0
10-AUG-1999 15:38:44 * service_update * sales * 0
10-AUG-1999 15:38:44 * service_update * sales * 0
10-AUG-1999 15:48:45 * service_update * sales * 0
10-AUG-1999 15:48:45 * service_update * sales * 0
10-AUG-1999 15:50:57 *
(connect_data=(service_name=sales.us.acme.com)(cid=(program=)(host=dlsun1013)(u
ser=jdoe)))
* (address=(protocol=tcp)(host=144.25.185.246)(port=41365)) * establish
* sales.us.acme.com * 0
10-AUG-1999 15:51:26 * service_died * sales * 12537
10-AUG-1999 15:51:26 * service_died * sales * 12537
10-AUG-1999 15:52:06 *
(connect_data=(service_name=sales.us.acme.com)(cid=(program=)(host=dlsun1013)(u
ser=jdoe))
* (address=(protocol=tcp)(host=144.25.185.246)(port=41406)) * establish
* sales.us.acme.com * 12514
TNS-12514: TNS:listener could not resolve SERVICE_NAME given in connect
descriptor
-----
```

Listener Direct Hand-Off Information

The listener records direct hand-off events to **dispatchers**. These events are formatted into the following fields:

Timestamp * Presentation * Handoff * Error Code

Properties of direct hand-off fields are as follows:

- Each field is delimited by an asterisk (*).
- A successful connection or command returns a code of zero.
- A failure produces a code that maps to an error message.

See Also: "Resolving the Most Common Error Messages for Oracle Net Services" on page 17-7 for the most common Oracle Net errors or *Oracle9i Database Error Messages* for a complete listing of error messages

Example: Listener Log Event for Direct Hand-Off

A direct hand-off event in the log file is shown in the following example.

```
21-MAY-1999 10:54:55 * oracle.aurora.net.SALESHttp2 * handoff * 0
```

Analyzing Oracle Connection Manager Logs

Oracle Connection Manager generates two types of log files: one for its CMGW gateway process (*cman_pid.log*) and one for its CMADMIN administrative process.

[Figure 17–3](#) and [Figure 17–4](#) show examples of the log files.

Figure 17–3 cman_pid.log

```
(TIMESTAMP=30-OCT-98 18:03:10) (EVENT=10) (VERSION=8.1.6.0.0)
(TIMESTAMP=30-OCT-98 18:03:10) (EVENT=36) (rule_list=
(rule=(src=spcstn)(dst=x)(srv=x)(act=accept)))
(TIMESTAMP=30-OCT-98 18:03:10) (EVENT=32) (PARAMETER_LIST=(MAXIMUM_
RELAYS=1024) (RELAY_STATISTICS=no) (AUTHENTICATION_LEVEL=0) (LOG_LEVEL=1) (SHOW_TNS_
INFO=no) (ANSWER_TIMEOUT=0) (MAXIMUM_CONNECT_DATA=1024) (USE_ASYNC_
CALL=yes) (TRACE_DIRECTORY=default) (MAX_FREELIST_BUFFERS=0))
(TIMESTAMP=30-OCT-98 18:03:10) (EVENT=34) (ADDRESS_LIST=
(ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1610)(QUEUESIZE=32)))
(TIMESTAMP=30-OCT-98 18:03:12) (EVENT=38) (COMMAND=2)
(TIMESTAMP=30-OCT-98
18:03:27) (EVENT=26) (RLYN0=0) (SRC=(ADDRESS=(PROTOCOL=tcp)(HOST=spcstn.us.oracle.c
om)(PORT=34758))) (DST=(ADDRESS=(PROTOCOL=tcp)(HOST=144.25.187.89)(PORT=1581)))
(TIMESTAMP=30-OCT-98 18:03:43) (EVENT=28) (RLYN0=0) (SINCE=30-OCT-98
18:03:27) (STATISTICS=(IN=(BYTES=0)(PACKETS=0)(DCDS=0)(OUT=(BYTES=0)(PACKETS=0)(D
CDS=0)))
```

Figure 17–4 cmadm_pid.log

```
(TIMESTAMP=30-OCT-98 18:03:09) (EVENT=Sent Admin Status to UI)
(TIMESTAMP=30-OCT-98 18:03:10) (EVENT=CMan Registration)
```

See Also: "Analyzing Listener Log Files" on page 17-26

cman_pid.log Event Codes

The `cman_pid.log` on UNIX and `cmanpid.log` on Windows NT reports events using event code numbers. The event code reported depends upon the log level set with the `LOG_LEVEL` parameter in the `cman.ora` file or with the Oracle Connection Manager Control utility command `SET LOG_LEVEL`. This section explains what each of these event codes represents.

Table 17-10 cman_pid.log Log Level 1 Event Codes

Code	Description
10	Gateway is starting up
12	Gateway is shutting down
14	Listening on TNS address(es)
18	Answer failed See Also: " Reasons for Event Code 18 " on page 17-37
20	Refusing in-coming call See Also: " Reasons for Event Code 20 " on page 17-37
26	Relay is now open
28	Relay is now closed
30	Statistics report
32	<PARAMETER_LIST>
34	<ADDRESS_LIST>
36	<RULE_LIST>
38	Oracle Connection Manager Control utility command
40	Oracle Connection Manager Control utility command refused because the gateway is busy
42	Dead connection detected
44	Relay has timed out
11	Bad <ADDRESS_LIST> argument
13	Bad <PARAMETER_LIST> argument
15	Bad <RULE_LIST> argument
23	Bad Oracle Connection Manager Control utility record

Code	Description
25	Command line argument is too long
27	Memory allocation failure
29	TNS error
31	TNS error while processing Oracle Connection Manager Control utility requests

Reasons for Event Code 18 The answer can fail due to the following:

Code	Description
1	Timed out
2	Connect data buffer is too small
3	Refused by TNS
4	TNS packet checksum error

Reasons for Event Code 20 The incoming call can be refused if:

Code	Description
1	Gateway is shutting down
1	Gateway is offline
3	No connect data on in-coming call
4	Bad connect data on in-coming call
5	All relays are in use
6	Unable to get relay buffers
7	Fatal TNS error
8	No available Oracle Advanced Security service
9	Reject from rule filtering
10	Out-going call failed
11	Refused by Oracle Net/TNS
12	Listener is not running

Code	Description
13	Listener is not reachable
14	Host name lookup failure
15	Protocol adapter (and probably the protocol stack) not loaded
16	No SOURCE_ROUTE set
17	Reject from rule or bad connect string data

Table 17-11 cman_pid.log Log Level 2 Event Codes

Code	Description
102	Answering in-coming call
104	Making out-going call
105	Accepting in-coming call
106	Rule match report

Table 17-12 cman_pid.log Log Level 3 Event Codes

Code	Description
202	Call will block (no asynchronous TNS support)
204	Relay blocked See Also: " Reasons for Event Code 204 "
206	Buffer contains leftover data

Reasons for Event Code 204 The relay can be blocked due to the following:

Code	Description
1	Waiting for writer to be ready
2	Waiting for writer to clear backlog
3	WOULDBLOCK error on receive
4	WOULDBLOCK or PARTIAL error on send
5	Repeated WOULDBLOCK or PARTIAL send error

Table 17–13 cman_pid.log Log Level 4 Event Codes

Code	Description
302	Read this many bytes
304	Wrote this many bytes
306	Wrote this many bytes on retry

Oracle Net Tracing Error Information

Tracing produces a detailed sequence of statements that describe network events as they are executed. Tracing an operation enables you to obtain more information on the internal operations of the components of Oracle Net than is provided in a log file. This information is output to files that can be evaluated to identify the events that led to an error.

CAUTION: Tracing uses a large amount of disk space and may have a significant impact upon system performance. Therefore, you should enable tracing only when necessary.

This section contains topics:

- [Oracle Net Trace File Names](#)
- [Setting Oracle Net Trace Parameters](#)
- [Setting Oracle Net Trace Settings During Runtime of Control Utilities](#)
- [Evaluating Oracle Net Traces](#)

Oracle Net Trace File Names

Each Oracle Net component produces its own trace file. [Table 17–14](#) provides the default file names and a description of the information they contain.

Table 17–14 Oracle Net Trace Files

Trace File	Component
sqlnet.trc	Client
svr_pid.trc	Database server
listener.trc	Listener
names.trc	Oracle Names Server
cman_pid.trc on UNIX cmanpid.trc on Windows NT	Oracle Connection Manager CMGW gateway process
cmadm_pid.trc on UNIX cmadmpid.trc on Windows NT	Oracle Connection Manager CMADMIN administrative process

Setting Oracle Net Trace Parameters

Parameters that control tracing, including the type and amount of information logged, as well as the location where the files are stored, are set in the configuration file of each network component as described in [Table 17–15](#).

Table 17–15 Oracle Net Trace Parameters

Trace Parameters	Configuration Files
Client	sqlnet.ora
Server	sqlnet.ora
Listener	listener.ora
Oracle Names Server	names.ora
Oracle Connection Manager processes	cman.ora

This sections contains these topics:

- [sqlnet.ora Trace Parameters](#)
- [listener.ora Trace Parameters](#)
- [names.ora Trace Parameters](#)
- [cman.ora Trace Parameters](#)
- [Setting Oracle Net Trace Parameters in Configuration Files](#)

See Also: *Oracle Net Services Reference Guide* for more information about these parameters

sqlnet.ora Trace Parameters

Table 17–16 describes the trace parameters settings that can be set in the `sqlnet.ora` file.

Table 17–16 sqlnet.ora Trace Parameters

sqlnet.ora Parameter	Oracle Net Manager Field	Description
TRACE_LEVEL_CLIENT	Client Information: Trace Level	<p>Specifies the level of detail the trace facility records for the client.</p> <p>The trace level value can either be a value within the range of 0 (zero) to 16 (where 0 is no tracing and 16 represents the maximum amount of tracing) or a value of <code>off</code>, <code>admin</code>, <code>user</code>, or <code>support</code>.</p> <ul style="list-style-type: none"> ■ <code>off</code> (equivalent to 0) provides no tracing ■ <code>user</code> (equivalent to 4) traces to identify user-induced error conditions ■ <code>admin</code> (equivalent to 6) traces to identify installation-specific problems ■ <code>support</code> (equivalent to 16) provides trace information for troubleshooting information for Oracle Support Services
TRACE_DIRECTORY_CLIENT	Client Information: Trace Directory	Establishes the destination directory for the client trace file. By default, the client directory is <code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME\Network\trace</code> on Windows NT.
TRACE_FILE_CLIENT	Client Information: Trace File	Sets the name of the log file for the client. By default the log name is <code>sqlnet.trc</code> .

sqlnet.ora Parameter	Oracle Net Manager Field	Description
TRACE_UNIQUE_CLIENT	Client Information: Unique Trace File Name	<p>Determines whether or not a unique trace file is created for each client trace file. If the value is set to <code>on</code>, a process identifier is appended to the name of each trace file generated, so that several files can coexist. If the value is set to <code>off</code>, when a new trace file is created for a client, it overwrites the existing file.</p>
		This parameter creates unique trace files named <code>sqlnetpid.trc</code>
TRACE_LEVEL_SERVER	Server Information: Trace Level	<p>Indicates the level of detail the trace facility records for the database server. The trace level value can either be a value within the range of 0 (zero) to 16 (where 0 is no tracing and 16 represents the maximum amount of tracing) or a value of <code>off</code>, <code>admin</code>, <code>user</code>, or <code>support</code>.</p>
		<ul style="list-style-type: none"> ■ <code>off</code> (equivalent to 0) provides no tracing ■ <code>user</code> (equivalent to 4) traces to identify user-induced error conditions ■ <code>admin</code> (equivalent to 6) traces to identify installation-specific problems ■ <code>support</code> (equivalent to 16) provides trace information for troubleshooting information for Oracle Support Services
TRACE_DIRECTORY_SERVER	Server Information: Trace Directory	<p>Establishes the destination directory for the database server trace file. By default, the server directory is <code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME\network\trace</code> on Windows NT.</p>
TRACE_FILE_SERVER	Server Information: Trace File	<p>Sets the name of the trace file for the client. By default the log name is <code>svr_pid.trc</code>.</p>

You can also manually add the following TNSPING utility tracing parameters described in [Table 17-17](#) to `sqlnet.ora`. The TNSPING utility determines whether or not a service (such as a database, an Oracle Names Server, or other TNS services) on a Oracle Net network can be successfully reached.

Table 17-17 TNSPING Trace Parameters

sqlnet.ora Parameter	Description
<code>TNSPING. TRACE_LEVEL</code>	<p>Indicates the level of detail the trace facility records for the TNSPING utility.</p> <p>The trace level value can either be a value within the range of 0 (zero) to 16 (where 0 is no tracing and 16 represents the maximum amount of tracing) or a value of <code>off</code>, <code>admin</code>, <code>user</code>, or <code>support</code>.</p> <ul style="list-style-type: none"> ■ <code>off</code> (equivalent to 0) provides no tracing ■ <code>user</code> (equivalent to 4) traces to identify user-induced error conditions ■ <code>admin</code> (equivalent to 6) traces to identify installation-specific problems ■ <code>support</code> (equivalent to 16) provides trace information for troubleshooting information for Oracle Support Services
<code>TNSPING. TRACE_DIRECTORY</code>	<p>Establishes the destination directory for TNSPING trace file. By default, the directory is <code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME\network\trace</code> on Windows NT.</p>

listener.ora Trace Parameters

Table 17-18 describes the trace parameters settings that can be set in the `listener.ora` file.

Table 17-18 listener.ora Trace Parameters

listener.ora Parameter	Oracle Net Manager Field	Description
<code>TRACE_LEVEL_ listener_name</code>	Trace Level	<p>Indicates the level of detail the trace facility records for the listener.</p> <p>The trace level value can either be a value within the range of 0 (zero) to 16 (where 0 is no tracing and 16 represents the maximum amount of tracing) or a value of off, admin, user, or support.</p> <ul style="list-style-type: none"> ■ off (equivalent to 0) provides no tracing ■ user (equivalent to 4) traces to identify user-induced error conditions ■ admin (equivalent to 6) traces to identify installation-specific problems ■ support (equivalent to 16) provides trace information for troubleshooting information for Oracle Support Services
<code>TRACE_DIRECTORY_ listener_name</code>	Trace File	Establishes the destination directory and file for the trace file. By default the directory is <code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME\Network\trace</code> on Windows NT, and the file name is <code>listener.trc</code> .
<code>TRACE_FILE_ listener_name</code>		

names.ora Trace Parameters

Table 17–19 describes the trace parameters settings that can be set in the names.ora file.

Table 17–19 names.ora Trace Parameters

names.ora Parameter	Oracle Net Manager Field	Description
NAMES. TRACE_DIRECTORY	Trace Directory	Establishes the destination directory for trace files. By default, the directory is \$ORACLE_HOME/network/trace on UNIX and ORACLE_HOME\network\trace on Windows NT.
NAMES. TRACE_FILE	Trace File	Sets the name of the trace file. By default the trace name is names.trc.
NAMES. TRACE_LEVEL	Not applicable	<p>Indicates the level of detail the trace facility records for the Oracle Names server.</p> <p>The trace level value can either be a value within the range of 0 (zero) to 16 (where 0 is no tracing and 16 represents the maximum amount of tracing) or a value of off, admin, user, or support.</p> <ul style="list-style-type: none"> ▪ off (equivalent to 0) provides no tracing ▪ user (equivalent to 4) traces to identify user-induced error conditions ▪ admin (equivalent to 6) traces to identify installation-specific problems ▪ support (equivalent to 16) provides trace information for troubleshooting information for Oracle Support Services
NAMES. TRACE_UNIQUE	Make Trace File Unique	<p>If set to on, creates a unique file name for each trace session, allowing multiple trace files to coexist. A process identifier is appended to the name of the trace file generated.</p> <p>This parameter creates unique trace files named namespid.trc.</p>

cman.ora Trace Parameters

Table 17-20 describes the trace parameters settings that can be set in the `cman.ora` file.

Table 17-20 cman.ora Trace Parameters

cman.ora Parameter	Description
TRACE_LEVEL	Determines whether or not tracing is enabled for the Oracle Connection Manager yes enables tracing for the Oracle Connection Manager. The proxy processes, the CMON process, and the listener process create trace files called <code>cman_pid.trc</code> , <code>cman_mon.trc</code> , and <code>cman_lsnr.trc</code> , respectively.
TRACE_DIRECTORY	Establishes the destination directory for trace files. By default, the directory is <code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME\Network\trace</code> on Windows NT.

Setting Oracle Net Trace Parameters in Configuration Files

`sqlnet.ora`, `listener.ora` and `names.ora` logging parameters can be set with the Oracle Net Manager. The `cman.ora` tracing parameters must be set manually.

See Also: *Oracle Net Services Reference Guide*

To set tracing parameters:

1. Start Oracle Net Manager.

See Also: "[Starting Oracle Net Manager](#)" on page 5-3

2. Specify the trace parameters:

Trace File	Instruction
sqlnet.trc (for the client) svr_pid.trc (for the server)	<ol style="list-style-type: none">1. In the navigator pane, expand Local > Profile.2. From the list in the right pane, select General.3. Choose the Tracing tab.4. Specify the settings.
listener.trc	<ol style="list-style-type: none">1. In the navigator pane, expand Local > Listeners.2. Select a listener.3. From the list in the right pane, select General.4. Choose the Logging and Tracing tab.5. Specify the settings:
names.trc	<ol style="list-style-type: none">1. In the navigator pane, expand the Oracle Names Servers folder.2. Select an Oracle Names server.3. From the list in the right pane, select Configure Server.4. Choose the Adv. tab.5. Specify the trace directory and file name.

3. Choose File > Save Network Configuration.

Setting Oracle Net Trace Settings During Runtime of Control Utilities

Tracing can be set during a runtime of a control utility. Note that setting tracing with a control utility will not set parameters in the *.ora files; the setting is valid only for the session of the control utility:

- For the listener, use the SET TRC_FILE, SET TRC_DIRECTORY and SET TRC_LEVEL commands from the Listener Control utility.
- For Oracle Connection Manager, use the SET TRACE_LEVEL and SET TRACE_DIRECTORY commands from the Oracle Connection Manager Control utility.
- For an Oracle Names server, use the SET TRACE_FILE_NAME and SET TRACE_LEVEL commands from the Oracle Names Control utility, or set tracing settings through Oracle Net Manager.

See Also: *Oracle Net Services Reference Guide*

To set tracing for an Oracle Names server with Oracle Net Manager:

1. Start Oracle Net Manager.

See Also: ["Starting Oracle Net Manager" on page 5-3](#)

2. In the navigator pane, expand Oracle Names Servers.
3. Select an Oracle Names server.
4. From the list in the right pane, select Manage Server.
5. Choose the Logging tab.
6. Specify the trace level, directory and file name.
7. Choose File > Save Network Configuration.

Evaluating Oracle Net Traces

Trace files can help Oracle Support Service diagnose and troubleshoot network problems.

This section explains how to do basic analysis of trace files. The topics discussed include:

- [Flow of Data Packets Between Network Nodes](#)
- [Oracle Net Data Packet Formats](#)
- [Pertinent Oracle Net Trace Error Output](#)

Flow of Data Packets Between Network Nodes

Oracle Net performs its functions by sending and receiving data packets. By specifying a trace level of `support`, you can view the actual contents of the Oracle Net packet in your trace file. The order of the packet types sent and received will help you to determine how your connection was established.

Oracle Net Data Packet Formats

Each line in the trace file begins with a procedure followed by a message. Following each procedure is a line of hexadecimal data representing actual data. The actual data that flows inside the packet is sometimes viewable to the right of the hexadecimal data.

Next is a list of the Oracle Net packet keywords and descriptions of the types of packets they represent:

Keyword	Packet Type
NSPTCN	Connect
NSPTAC	Accept
NSPTRF	Refuse
NSPTRS	Resend
NSPDA	Data
NSPCNL	Control
NSPTMK	Marker

For example, the following line describes a procedure called "nscon" sending a NSPTCN packet over the network:

```
nscon: sending NSPTCN packet
```

Each packet has a keyword that denotes the packet type. All packet types begin with the prefix "nsp". It is helpful to remember this when reviewing trace files for specific packet information

[Figure 17-5](#) provides typical packet information.

Figure 17-5 Packet Information

```
nscon: entry
nscon: doing connect handshake...
nscon: sending NSPTCN packet
nspsend: entry
nspsend: plen=187, type=1
nspsend: 187 bytes to transport
nspsend:packet dump
nspsend:00 BB 00 00 01 00 00 00 | .....
nspsend:01 33 01 2C 0C 01 08 00 | .3.,.....
nspsend:7F FF 7F 08 00 00 00 01 | .....
nspsend:00 99 00 22 00 00 08 00 | ...".....
nspsend:01 01 28 44 45 53 43 52 | ..(DESCR|
nspsend:49 50 54 49 4F 4E 3D 28 | IPTION=(|
nspsend:43 4F 4E 4E 45 43 54 5F | CONNECT_|
nspsend:44 41 54 41 3D 28 53 49 | DATA=(SI|
nspsend:44 3D 61 70 33 34 37 64 | D=ap347d|
nspsend:62 31 29 28 43 49 44 3D | b1)(CID=|
nspsend:28 50 52 4F 47 52 41 4D | (PROGRAM|
nspsend:3D 29 28 48 4F 53 54 3D | =)(HOST=|
nspsend:61 70 32 30 37 73 75 6E | ap207sun|
nspsend:29 28 55 53 45 52 3D 6D | )(USER=m|
nspsend:77 61 72 72 65 6E 29 29 | warren))|
nspsend:29 28 41 44 44 52 45 53 | )(ADDRES|
nspsend:53 5F 4C 49 53 54 3D 28 | S_LIST=(|
nspsend:41 44 44 52 45 53 53 3D | ADDRESS=|
nspsend:28 50 52 4F 54 4F 43 4F | (PROTOCO|
nspsend:4C 3D 74 63 70 29 28 48 | L=tcp)(H|
nspsend:4F 53 54 3D 61 70 33 34 | OST=ap34|
nspsend:37 73 75 6E 29 28 50 4F | 7sun)(PO|
nspsend:52 54 3D 31 35 32 31 29 | RT=1521)|
nspsend:29 29 00 00 00 00 00 00 | ))). ....|
nspsend: normal exit
nscon: exit (0)
```

Pertinent Oracle Net Trace Error Output

When there is a problem an Oracle Net connection, the error code is logged in the trace file. [Figure 17–6](#) depicts typical trace file output for a failed SQL*Plus connection to a database server.

Figure 17–6 Trace Example

```
[09-MAR-2001 13:34:07] nsprecv: entry
[09-MAR-2001 13:34:07] nsbal: entry
[09-MAR-2001 13:34:07] nsbgetfl: entry
[09-MAR-2001 13:34:07] nsbgetfl: normal exit
[09-MAR-2001 13:34:07] nsmal: entry
[09-MAR-2001 13:34:07] nsmal: 44 bytes at 0x132d90
[09-MAR-2001 13:34:07] nsmal: normal exit
[09-MAR-2001 13:34:07] nsbal: normal exit
[09-MAR-2001 13:34:07] nsprecv: reading from transport...
[09-MAR-2001 13:34:07] nttrd: entry
[09-MAR-2001 13:35:09] nttrd: exit
[09-MAR-2001 13:35:09] ntt2err: entry
[09-MAR-2001 13:35:09] ntt2err: Read unexpected EOF ERROR on 10
[09-MAR-2001 13:35:09] ntt2err: exit
[09-MAR-2001 13:35:09] nsprecv: transport read error
[09-MAR-2001 13:35:09] nsprecv: error exit
[09-MAR-2001 13:35:09] nserror: entry
[09-MAR-2001 13:35:09] nserror: nsres: id=0, op=68, ns=12537,
ns2=12560;
nt[0]=507, nt[1]=0, nt[2]=0; ora[0]=0, ora[1]=0, ora[2]=0
[09-MAR-2001 13:35:09] nscon: error exit
[09-MAR-2001 13:35:09] nsdo: nsctxrnmk=0
[09-MAR-2001 13:35:09] nsdo: error exit
[09-MAR-2001 13:35:09] nscall: unexpected response
[09-MAR-2001 13:35:09] nsclose: entry
[09-MAR-2001 13:35:09] nstimarmed: entry
[09-MAR-2001 13:35:09] nstimarmed: no timer allocated
[09-MAR-2001 13:35:09] nstimarmed: normal exit
[09-MAR-2001 13:35:09] nsdo: entry
[09-MAR-2001 13:35:09] nsdo: cid=0, opcode=98, *bl=0, *what=0,
uflgs=0x440, cflgs=0x2
[09-MAR-2001 13:35:09] nsdo: rank=64, nsctxrnmk=0
[09-MAR-2001 13:35:09] nsdo: nsctx: state=1, flg=0x4201, mvd=0
[09-MAR-2001 13:35:09] nsbfr: entry
[09-MAR-2001 13:35:09] nsbaddir: entry
[09-MAR-2001 13:35:09] nsbaddir: normal exit
[09-MAR-2001 13:35:09] nsbfr: normal exit
[09-MAR-2001 13:35:09] nsbfr: entry
[09-MAR-2001 13:35:09] nsbaddir: entry
```

```
[09-MAR-2001 13:35:09] nsbaddir: normal exit
[09-MAR-2001 13:35:09] nsbfrr: normal exit
[09-MAR-2001 13:35:09] nsdo: nsctxrnk=0
[09-MAR-2001 13:35:09] nsdo: normal exit
[09-MAR-2001 13:35:09] nsfclose: closing transport
[09-MAR-2001 13:35:09] nttdisc: entry
[09-MAR-2001 13:35:09] nttdisc: Closed socket 10
[09-MAR-2001 13:35:09] nttdisc: exit
[09-MAR-2001 13:35:09] nsfclose: global context check-out (from slot 0)
complete
[09-MAR-2001 13:35:09] nsnadisc: entry
[09-MAR-2001 13:35:09] nadisc: entry
[09-MAR-2001 13:35:09] nacomtm: entry
[09-MAR-2001 13:35:09] nacompd: entry
[09-MAR-2001 13:35:09] nacompd: exit
[09-MAR-2001 13:35:09] nas_dis: entry
[09-MAR-2001 13:35:09] nas_dis: exit
[09-MAR-2001 13:35:09] nau_dis: entry
[09-MAR-2001 13:35:09] nau_dis: exit
[09-MAR-2001 13:35:09] naeetrm: entry
[09-MAR-2001 13:35:09] naeetrm: exit
[09-MAR-2001 13:35:09] nagbltrm: entry
[09-MAR-2001 13:35:09] nau_gtm: entry
[09-MAR-2001 13:35:09] nau_gtm: exit
[09-MAR-2001 13:35:09] nagbltrm: exit
[09-MAR-2001 13:35:09] nadisc: exit
[09-MAR-2001 13:35:09] nsnadisc: normal exit
[09-MAR-2001 13:35:09] nsbfrr: entry
[09-MAR-2001 13:35:09] nsbaddir: entry
[09-MAR-2001 13:35:09] nsbaddir: normal exit
[09-MAR-2001 13:35:09] nsbfrr: normal exit
[09-MAR-2001 13:35:09] nsmfr: entry
[09-MAR-2001 13:35:09] nsmfr: 2256 bytes at 0x130508
[09-MAR-2001 13:35:09] nsmfr: normal exit
[09-MAR-2001 13:35:09] nsmfr: entry
[09-MAR-2001 13:35:09] nsmfr: 484 bytes at 0x1398a8
[09-MAR-2001 13:35:09] nsmfr: normal exit
[09-MAR-2001 13:35:09] nsfclose: normal exit
[09-MAR-2001 13:35:09] nsccall: connecting...
[09-MAR-2001 13:35:09] nsfclose: entry
[09-MAR-2001 13:35:09] nsfclose: normal exit
[09-MAR-2001 13:35:09] nladget: entry
[09-MAR-2001 13:35:09] nladget: exit
```

```
[09-MAR-2001 13:35:09] nsmfr: entry
[09-MAR-2001 13:35:09] nsmfr: 144 bytes at 0x132cf8
[09-MAR-2001 13:35:09] nsmfr: normal exit
[09-MAR-2001 13:35:09] nsmfr: entry
[09-MAR-2001 13:35:09] nsmfr: 156 bytes at 0x138e70
[09-MAR-2001 13:35:09] nsmfr: normal exit
[09-MAR-2001 13:35:09] nladtrm: entry
[09-MAR-2001 13:35:09] nladtrm: exit
[09-MAR-2001 13:35:09] nscall: error exit
[09-MAR-2001 13:35:09] nioqper: error from nscall
[09-MAR-2001 13:35:09] nioqper: nr err code: 0
[09-MAR-2001 13:35:09] nioqper: ns main err code: 12537
[09-MAR-2001 13:35:09] nioqper: ns (2) err code: 12560
[09-MAR-2001 13:35:09] nioqper: nt main err code: 507
[09-MAR-2001 13:35:09] nioqper: nt (2) err code: 0
[09-MAR-2001 13:35:09] nioqper: nt OS err code: 0
[09-MAR-2001 13:35:09] niomapnerror: entry
[09-MAR-2001 13:35:09] niqme: entry
[09-MAR-2001 13:35:09] niqme: reporting NS-12537 error as ORA-12537
[09-MAR-2001 13:35:09] niqme: exit
[09-MAR-2001 13:35:09] niomapnerror: returning error 12537
[09-MAR-2001 13:35:09] niomapnerror: exit
[09-MAR-2001 13:35:09] niotns: Couldn't connect, returning 12537
[09-MAR-2001 13:35:10] niotns: exit
[09-MAR-2001 13:35:10] nsbfrfl: entry
[09-MAR-2001 13:35:10] nsbrfr: entry
[09-MAR-2001 13:35:10] nsbrfr: nsbfs at 0x132d90, data at 0x132dc8.
[09-MAR-2001 13:35:10] nsbrfr: normal exit
[09-MAR-2001 13:35:10] nsbrfr: entry
[09-MAR-2001 13:35:10] nsbrfr: nsbfs at 0x1248d8, data at 0x132210.
[09-MAR-2001 13:35:10] nsbrfr: normal exit
[09-MAR-2001 13:35:10] nsbrfr: entry
[09-MAR-2001 13:35:10] nsbrfr: nsbfs at 0x12d820, data at 0x1319f0.
[09-MAR-2001 13:35:10] nsbrfr: normal exit
[09-MAR-2001 13:35:10] nsbfrfl: normal exit
[09-MAR-2001 13:35:10] nigtrm: Count in the NI global area is now 1
[09-MAR-2001 13:35:10] nigtrm: Count in the NL global area is now 1
```

The most efficient way to evaluate error codes is to find the most recent nserror entry logged, as the session layer controls the connection. The most important error messages are the ones at the bottom of the file. They are the most recent errors and the source of the problem with the connection.

For information about the specific return codes, use the Oracle UNIX error tool `oerr`, by entering the following at any command line:

```
oerr tns error_number
```

As an example, consider the following `nserror` entry logged in the trace file shown in [Figure 17-6](#) on page 17-51:

```
[09-MAR-2001 13:35:09] nserror: nsres: id=0, op=68, ns=12537,  
ns2=12560;  
nt[0]=507, nt[1]=0, nt[2]=0; ora[0]=0, ora[1]=0, ora[2]=0
```

Using `oserr`, you can find out more information about return codes 12537 and 507. (Bold denotes user input.)

```
oerr tns 12537  
12537, 00000, "TNS:connection closed"  
// *Cause: "End of file" condition has been reached; partner has  
disconnected.  
// *Action: None needed; this is an information message.
```

```
oerr tns 507  
00507, 00000, "Connection closed"  
// *Cause: Normal "end of file" condition has been reached; partner has  
// disconnected.  
// *Action: None needed; this is an information message.
```

Contacting Oracle Support Services

If you are still unable to resolve your problems, or if you are requested to contact Oracle Support Services to report the error, please have the following information at hand:

- The hardware and operating system release number on which the application is running
- The up-to-five-digit release number of all the Oracle networking products involved in the current problem
- The third-party vendor and version you are using
- If you encountered one or more error codes or messages, the exact code numbers and message texts in the order they appeared
- The kind of links that exist between the client and server

- A description of what does work
- The exact error message, if there is one
- An Oracle Net trace, if possible; if not, the log file is sufficient

Glossary

A

Access Control List (ACL)

The group of access directives that you define. The directives grant levels of access to specific data for specific clients and/or groups of clients.

ACL

See [Access Control List \(ACL\)](#).

access control

A feature of Oracle Connection Manager that sets rules for denying or allowing certain clients to access designated servers.

address

A unique network location used to identify a network object, such as a listener, Oracle Connection Manager, or Oracle Names server. Addresses have a specific format and must be unique.

administrative region

An organizational entity for administering Oracle Network Services' components. Each administrative region includes:

- One or more domains
- One or more Oracle Names servers
- One or more databases and listeners

alias

An alternative name for an existing network object. Once an alias is created, it is resolved to the same name as the initial network object. An Oracle Names server stores aliases for any defined net service name, database server, or database link.

application gateway

A host computer that runs the [Oracle Net Firewall Proxy](#). An application gateway looks and acts like a real server from the client's point of view, and a real client from the server's point of view. An application gateway sits between the Internet and company's internal network and provides middleman services (or proxy services) to users on either side.

ASCII character set

American Standard Code for Information Interchange character set, a convention for representing alphanumeric information using digital data. The collation sequence used by most computers with the exception of IBM and IBM-compatible computers.

attribute

A piece of information that describes some aspect of a directory entry. An entry comprises a set of attributes, each of which belongs to an [object class](#). Moreover, each attribute has both a type—which describes the kind of information in the attribute—and a value—which contains the actual data.

authentication method

A security method that enables you to have high confidence in the identity of users, clients, and servers in distributed environments. Network authentication methods can also provide the benefit of single sign-on for users. The following authentication methods are supported in Oracle9*i*, depending on whether or not [Oracle Advanced Security](#) is installed:

- CyberSafe
- RADIUS
- Kerberos
- [SSL](#)
- [Windows NT native authentication](#)

C

cache

Memory that stores recently-accessed data so that subsequent requests to access the same data can be processed quickly.

CDS

See [Cell Directory Services \(CDS\)](#).

Cell Directory Services (CDS)

An external naming method that enables users to use Oracle tools transparently and applications to access Oracle databases in a Distributed Computing Environment (DCE) environment.

central administration

An Oracle Names network where network management consists of one [administrative region](#) for the entire network. All Oracle Names servers know about one another and about all the services in the network. Contrast with [delegated administration](#).

client

A user, software application, or computer that requests the services, data, or processing of another application or computer. The client is the user process. In a network environment, the client is the local user process and the server may be local or remote.

client load balancing

Load balancing, whereby if more than one listener services a single database, a client can randomly choose between the listeners for its connect requests. This randomization enables all listeners to share the burden of servicing incoming connect requests.

client profile

The properties of a client, which may include the preferred order of [naming methods](#), client and server [logging](#) and [tracing](#), the domain from which to request names, and other client options for [Oracle Names](#) and [Oracle Advanced Security](#).

client/server architecture

Software architecture based on a separation of processing between two CPUs. One CPU acts as the client in the transaction, requesting and receiving services. The other acts as the server that provides the requests.

cman.ora file

A configuration file that specifies protocol addresses for incoming requests and administrative commands, as well as Oracle Connection Manager parameters and **access control** rules.

CMON process

An **Oracle Connection Manager** monitoring process that registers the location and load of the proxy process with the Oracle Connection Manager listener, enabling the listener to forward requests to the least loaded **proxy process**.

configuration files

Files that are used to identify and characterize the components of a network. Configuration is largely a process of naming network components and identifying relationships among those components.

connect data

A portion of the **connect descriptor** that defines the destination database **service name** or **Oracle System Identifier (SID)**. In the example below, SERVICE_NAME defines a database service called sales.us.acme.com:

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521)
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)))
```

connect descriptor

A specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information.

The destination service is indicated by using its **service name** for Oracle9i or Oracle8i databases or its **Oracle System Identifier (SID)** for Oracle release 8.0 databases. The network route provides, at a minimum, the location of the listener through use of a network address.

connect identifier

A **net service name** or **service name**, that resolves to a **connect descriptor**. Users initiate a connect request by passing a user name and password along with a connect identifier in a connect string for the service to which they wish to connect, for example:

```
CONNECT username/password@connect_identifier
```

connect string

Information the user passes to a service to connect, such as user name, password and **net service name**. For example:

```
CONNECT username/password@net_service_name
```

connect-time failover

A client connect request is forwarded to another listener if the first listener is not responding. Connect-time failover is enabled by **service registration**, because the listener knows if an instance is running to attempting a connection.

connection

An interaction between two processes on a network. Connections are originated by an initiator (client) that requests a connection with a destination (server).

connection load balancing

Load balancing, whereby the number of active connections among various instances and dispatchers for the same service are balanced. This enables listeners to make their routing decisions based on how many connections each dispatcher has and on how loaded the nodes that the instances run.

connection pooling

A resource utilization and user scalability feature that enables you to maximize the number of sessions over a limited number of protocol connections to a **shared server**.

connection request

A notification sent by an initiator and received by a listener that indicates that the initiator wants to start a connection.

D

data packet

See [packet](#).

database administrator (DBA)

(1) A person responsible for operating and maintaining an Oracle Server or a database application. (2) An Oracle username that has been given DBA privileges and can perform database administration functions. Usually the two meanings coincide. Many sites have multiple DBAs.

database link

A network object stored in the local database or in the network definition that identifies a remote database, a communication path to that database, and, optionally, a user name and password. Once defined, the database link is used to access the remote database.

A public or private database link from one database to another is created on the local database by a DBA or user.

A global database link is created automatically from each database to every other database in a network with [Oracle Names](#). Global database links are stored in the network definition.

See also [global database link](#), [private database link](#), and [public database link](#).

decentralized administration

See [delegated administration](#).

dedicated server

A server process that is dedicated to one client connection. Contrast with [shared server process](#).

default domain

The domain [domain](#) within which most client requests take place. It could be the domain where the client resides, or it could be a domain from which the client requests network services often. Default domain is also the client configuration parameter that determines what domain should be appended to unqualified network name requests. A name request is unqualified if it does not have a "." character within it.

delegated administration

A network where network management is delegated to one or more administrative regions below the **root administrative region**. Also referred to as distributed or decentralized administration. Contrast with **central administration**.

delegated administrative region

A region hierarchically below the **root administrative region**. Any region other than the root administrative region.

destination

The client that is the endpoint of a connection. The initiator of the connection requires some data or service of the destination.

directory information tree (DIT)

A hierarchical tree-like structure in a **directory server** of the **distinguished names (DNs)** of the entries.

directory naming

A **naming method** that resolves a database service or net service name to a connect descriptor, stored in a central directory server.

A directory server provides central administration of database services and net service names, reducing the work effort associated with adding or relocating services. Although net service names can be configured to alias a service, the directory server can refer to a database service directly without using a net service name. The database service is automatically added as an entry to the directory server during installation.

directory server

A directory server that is accessed with the Lightweight Directory Access Protocol (LDAP). An LDAP-compliant directory server can provide centralized storage and retrieval of database network components, user and corporate policies preferences, user authentication and security information, replacing clientside and serverside localized files.

dispatcher

A process that enables many clients to connect to the same server without the need for a dedicated server process for each client. A dispatcher handles and directs multiple incoming network session requests to shared server processes. See also **shared server**.

distinguished name (DN)

Name of entry in a [directory server](#). The DN specifies where the entry resides in the LDAP directory hierarchy, much the way a directory path specifies the exact location of a file.

distributed administration

See [delegated administration](#).

distributed processing

Division of front-end and back-end processing to different computers. Oracle Network Services support distributed processing by transparently connecting applications to remote databases.

domain

Any tree or subtree within the [Domain Name System \(DNS\)](#) namespace. Domain most commonly refers to a group of computers whose host names share a common suffix, the domain name.

domain hint

A NAMES.DOMAIN_HINTS parameter in the names.ora file that contains the name of the domain and at least one address of an Oracle Names server in that domain. This enables an Oracle Names server to forward the client requests to a specific address, reducing network traffic.

Domain Name System (DNS)

A system for naming computers and network services that is organized into a hierarchy of [domains](#). DNS is used in TCP/IP networks to locate computers through user-friendly names. DNS resolves a friendly name into an [IP address](#), which is understood by computers.

For Oracle Network Services, DNS translates the host name in a TCP/IP address into an IP address.

domestic domains

The set of domains that are managed within a given administrative region. Domains are domestic only in relation to a region; they are never domestic in any absolute sense. Also referred to as local domains.

DNS

Domain Name System. See [Domain Name System \(DNS\)](#).

E

error message

A message from a computer program informing you of a potential problem or condition preventing program or command execution.

enterprise role

An enterprise role is analogous to a regular database role, except that it spans authorization on multiple databases. An enterprise role is a category of roles that define privileges on a particular database. An enterprise role is created by the database administrator of a particular database. An enterprise role can be granted to or revoked to one or more enterprise users. The information for granting and revoking these roles is stored in the directory server.

enterprise user

A user that has a unique identity across an enterprise. Enterprise users connect to individual databases through a schema. Enterprise users are assigned enterprise roles that determine their access privileges on databases.

entry

The building block of a directory server, it contains information about an object of interest to directory users.

external naming

A [naming method](#) that uses a supported third-party naming service, such as [NIS](#) or [CDS](#).

external procedures

Functions or procedures written in a third-generation language (3GL) that can be called from PL/SQL code. Only C is supported for external procedures.

F

failover

See [connect-time failover](#).

firewall support

See [access control](#).

G

flat naming model

An Oracle Names infrastructure in which there is only one domain. All names must be unique within that domain.

foreign domains

The set of domains not managed within a given administrative region. Domains are foreign only in relation to a region; they are not foreign in any absolute sense. A network administrator typically defines foreign domains relative to a particular region to optimize Names Server caching performance.

General Inter-ORB Protocol (GIOP)

A **presentation layer** type that is used in **Oracle9i JVM** connections to provide messaging.

global database link

A **database link** that links each database in a network to all other databases. This enables any user of any database in the network to specify a global object name in a SQL statement or object definition. A global database link that is the same as the **global database name** is registered with the **Oracle Names server**.

global database name

The full name of the database which uniquely identifies it from any other database. The global database name is of the form "*database_name*.*database_domain*," for example, `sales.us.acme.com`.

The database name portion, `sales`, is a simple name you wish to call your database. The database domain portion, `us.acme.com`, specifies the database domain in which the database is located, making the global database name unique. When possible, Oracle Corporation recommends that your database domain mirror the network domain.

The global database name is the default service name of the database, as specified by the `SERVICE_NAMES` parameter in the **initialization parameter file**.

H

Heterogeneous Services

An integrated component that provides the generic technology for accessing non-Oracle systems from the Oracle database server. Heterogeneous Services enables you to:

- Use Oracle SQL to transparently access data stored in non-Oracle systems as if the data resides within an Oracle server.
- Use Oracle procedure calls to transparently access non-Oracle systems, services, or application programming interfaces (APIs), from your Oracle distributed environment.

hierarchical naming model

An infrastructure in which names are divided into multiple hierarchically-related domains. For [Oracle Names](#), hierarchical naming model can be used with either central or delegated administration.

host naming

A [naming method](#) resolution that enables users in a TCP/IP environment to resolve names through their existing name resolution service. This name resolution service might be [Domain Name System \(DNS\)](#), [Network Information Service \(NIS\)](#), or simply a centrally-maintained set of /etc/hosts files. Host Naming enables users to connect to an Oracle database server by simply providing the server computer's host name or host name alias. No client configuration is required to take advantage of this feature. This method is recommended for simple TCP/IP environments.

HTTP protocol

Hypertext transfer protocol. A protocol that provides the language that enables Web browsers and application Web servers to communicate.

initialization parameter file

File that contains information to initialize the database and instances.

instance

The combination of the **System Global Area (SGA)** and the Oracle background processes. When a database is started on a database server (regardless of the type of computer), Oracle allocates a memory area called the SGA and starts one or more Oracle processes. The memory and processes of an instance efficiently manage the associated database's data and serve the database users. You can connect to any instance to access information within a cluster database.

instance name

A name of an Oracle database instance. The instance name is identified by the `INSTANCE_NAME` parameter in the database initialization parameter file.

`INSTANCE_NAME` corresponds to the **Oracle System Identifier (SID)** of the instance. Clients can connect to a specific instance by specifying the `INSTANCE_NAME` parameter in the connect descriptor.

The instance name is included in the **connect data** part of the **connect descriptor**.

Internet Inter-ORB Protocol (IIOP)

An implementation of **General Inter-ORB Protocol (GIOP)** over TCP/IP or TCP/IP with SSL for connections to **Oracle9i JVM**.

Interprocess Communication

A protocol used by client applications that resides on the same node as the listener to communicate with the database. IPC can provide a faster local connection than TCP/IP.

IIOP Clients

Clients that use the **General Inter-ORB Protocol (GIOP)** presentation to access the Java option. These clients include:

- Enterprise JavaBeans (EJBs)
- CORBA Servers
- Java Stored Procedures

IP address

Used to identify a node on a network. Each computer on the network is assigned a unique IP address, which is made up of the network ID, and a unique host ID. This address is typically represented in dotted-decimal notation, with the decimal value of each octet separated by a period, for example 144.45.9.22.

IPC

See [Interprocess Communication](#).

J

Java Database Connectivity (JDBC) Driver

A driver that provides Java applications and applets access to an Oracle database.

K

keyword-value pair

The combination of a keyword and a value, used as the standard unit of information in connect descriptors and many configuration files. Keyword-value pairs may be nested; that is, a keyword may have another keyword-value pair as its value.

L

latency

Networking round-trip time.

LDAP

Lightweight Directory Access Protocol. The framework of design conventions supporting industry-standard [directory servers](#).

LDAP Data Interchange Format (LDIF)

The set of standards for formatting an input file for any of the LDAP command line utilities.

ldap.ora file

A file created by Oracle Net Configuration Assistant that contains the following directory server access information:

- Type of directory server
- Location of the directory server
- Default directory entry that the client or server will use to look up or configure connect identifiers for connections to database services

The `ldap.ora` file resides in `$ORACLE_HOME/network/admin` on UNIX platforms and `ORACLE_HOME\network\admin` on Windows operating systems.

link qualifier

A qualifier appended to a global database link to provide alternate settings for the database user name and password credentials. For example, a link qualifier of `fieldrep` can be appended to a global database link of `sales.us.acme.com`.

```
SQL> SELECT * FROM emp@sales.us.acme.com@fieldrep
```

listener

A process that resides on the server whose responsibility is to listen for incoming client connection requests and manage the traffic to the server.

Every time a client requests a network session with a server, a listener receives the actual request. If the client information matches the listener information, then the listener grants a connection to the server.

listener.ora file

A configuration file for the listener that identifies the following for a **listener**:

- Unique name
- Protocol addresses that it is accepting connection requests on
- Services it is listening for

The `listener.ora` file typically resides in `$ORACLE_HOME/network/admin` on UNIX platforms and `ORACLE_HOME\network\admin` on Windows NT.

Oracle9*i* does not require identification of the database service because of **service registration**. However, static service configuration is required for if you plan to use Oracle Enterprise Manager.

Listener Control utility

A utility included with Oracle Network Services to control various listener functions, such as to starting, stopping, and getting the status of the listener.

load balancing

A feature by which client connections are distributed evenly among multiple listeners, dispatchers, instances, and nodes so that no single component is overloaded.

Oracle Network Services support **client load balancing** and **connection load balancing**.

local naming

A **naming method** that locates network addresses by using information configured and stored on each individual client's **tnsnames.ora file**. Local naming is most appropriate for simple distributed networks with a small number of services that change infrequently.

location transparency

A distributed database characteristic that enables applications to access data tables without knowing where they reside. All data tables appear to be in a single database, and the system determines the actual data location based on the table name. The user can reference data on multiple nodes in a single statement, and the system automatically and transparently routes (parts of) SQL statements to remote nodes for execution if needed. The data can move among nodes with no impact on the user or application.

logging

A feature in which errors, service activity, and statistics are written to a log file. The log file provides additional information for an administrator when the error message on the screen is inadequate to understand the failure. The log file, by way of the error stack, shows the state of the software at various layers.

See also [tracing](#).

loopback test

A connection from the server back to itself. Performing a successful loopback verifies that Oracle Net is functioning on the database server.

LU6.2 protocol

Logical Unit Type 6.2. A protocol that enables an Oracle application on a PC to communicate with an Oracle database. This communication occurs over the System Network Architecture (SNA) network with the Oracle database on a host system that supports Advanced Program-to-Program Communication (APPC) architecture.

M

map

Files used by the **Network Information Service (NIS)** `ypserv` program to handle name requests.

Microsoft Active Directory

An LDAP-compliant directory server included with the Windows 2000 Server. It stores information about objects on the network, and makes this information available to users and network administrators. Active Directory also provides access to resources on the network using a single logon process.

Active Directory can be configured as a directory naming method to store service information that clients can access.

N

names.ora file

A configuration file that contains parameter settings for an **Oracle Names server**.

Named Pipes protocol

A high-level interface protocol providing interprocess communications between clients and servers using distributed applications. Named Pipes enables client/server conversation over a network using Named Pipes.

naming context

A subtree that resides entirely on one directory server. It is a contiguous subtree, that is, it must begin at an entry that serves as the top of the subtree, and extend downward to either leaf entries or references to subordinate naming contexts. It can range in size from a single entry to the entire **directory information tree (DIT)**.

An **Oracle Context** can be created under a naming context.

naming method

The resolution method used by a client application to resolve a **connect identifier** to a **connect descriptor** when attempting to connect to a database service. Oracle Net provides five naming methods:

- [local naming](#)
- [directory naming](#)
- [Oracle Names](#)
- [host naming](#)
- [external naming](#)

naming model

The set and structure of domains within which names can be allocated.

In a flat naming model, there is a single domain.

In a hierarchical naming model, the highest level is the root domain, and all other domains are hierarchically related.

net service name

A simple name for a service that resolves to a **connect descriptor**. Users initiate a connect request by passing a user name and password along with a net service name in a connect string for the service to which they wish to connect:

```
CONNECT username/password@net_service_name
```

Depending on your needs, net service names can be stored in a variety of places, including:

- Local configuration file, `tnsnames.ora`, on each client
- Directory server
- Oracle Names server
- External naming service, such as [NIS](#) or [CDS](#)

network

A group of two or more computers linked together through hardware and software to allow the sharing of data and/or peripherals.

network administrator

The person who performs network management tasks such as installing, configuring, and testing network components. The administrator typically maintains the configuration files, connect descriptors and service names, aliases, and public and global database links.

network character set

As defined by Oracle, the set of characters acceptable for use as values in keyword-value pairs (that is, in connect descriptors and configuration files). The set includes alphanumeric upper- and lowercase, and some special characters.

Network Information Service (NIS)

Sun Microsystems' Yellow Pages (yp) client/server protocol for distributing system configuration data such as user and host names between computers on a network.

Network Interface (NI)

A network layer that provides a generic interface for Oracle clients, servers, or external processes to access Oracle Net functions. The NI layer handles the "break" and "reset" requests for a connection.

network listener

See [listener](#).

network object

Any service that can be directly addressed on a network; for example, a listener or a Oracle Names server.

network protocol

See [Oracle protocol support](#).

Network Program Interface (NPI)

An interface for server-to-server interactions that performs all of the functions that the [OCI](#) does for clients, allowing a coordinating server to construct SQL requests for additional servers.

Network Session (NS)

A [session layer](#) that is used in typical Oracle Net connections to establish and maintain the connection between a client application and a database server.

NI

Network Interface

NIS

See [Network Information Service \(NIS\)](#).

NN

Network Naming (Oracle Names)

node

A computer or terminal that is part of a network

NPI

See [Network Program Interface \(NPI\)](#).

NR

Network Routing

NS

Network Session. See [Network Session \(NS\)](#).

NT

Network Transport. See [transport](#).

O**object class**

In a directory server, a named group of attributes. When you want to assign attributes to an entry, you do so by assigning to that entry the object classes that hold those attributes.

All objects associated with the same object class share the attributes of that object class.

OCI

Oracle Call Interface. See [Oracle Call Interface \(OCI\)](#).

OPI

See [Oracle Program Interface \(OPI\)](#).

Open Systems Interconnection (OSI)

A model of network architecture developed by ISO as a framework for international standards in heterogeneous computer network architecture.

The OSI architecture is split between seven layers, from lowest to highest:

1. Physical layer
2. Data link layer
3. Network layer
4. Transport layer
5. Session layer
6. Presentation layer
7. Application layer

Each layer uses the layer immediately below it and provides a service to the layer above.

Oracle Advanced Security

A product that provides a comprehensive suite of security features to protect enterprise networks and securely extend corporate networks to the Internet. Oracle Advanced Security provides a single source of integration with network encryption and authentication solutions, single sign-on services, and security protocols. By integrating industry standards, it delivers unparalleled security to the network.

Oracle Call Interface (OCI)

An application programming interface (API) that enables you to create applications that use the native procedures or function calls of a third-generation language to access an Oracle database server and control all phases of SQL statement execution. OCI supports the datatypes, calling conventions, syntax, and semantics of a number of third-generation languages including C, C++, COBOL and FORTRAN.

Oracle Connection Manager

A router through which a client connection request may be sent either to its next hop or directly to the database server. Clients who route their connection requests through an Oracle Connection Manager can then take advantage of the **session multiplexing, access control, or protocol conversion** features configured on that Oracle Connection Manager.

Oracle Connection Manager Control utility

A utility included with Oracle Network Services to control various functions, such as starting, stopping, and getting the status of the Oracle Connection Manager.

Oracle Context

A **RDN** of `cn=OracleContext` in a **directory information tree (DIT)** that is located under a **naming context** or an unpublished directory entry. An Oracle Context contains entries for use with Oracle features, such as Oracle Net **directory naming** and **Oracle Advanced Security enterprise user** security. There can be one or more Oracle Contexts in a directory server.

Oracle Database Configuration Assistant

A tool that enables you to create, delete, and modify a database.

Oracle Internet Directory

A directory server implemented as an application on the Oracle database. It enables retrieval of information about dispersed users and network resources. It combines Lightweight Directory Access Protocol (LDAP) Version 3, the open Internet standard directory server access protocol, with the high performance, scalability, robustness, and availability of the Oracle database.

Oracle Names

An Oracle directory service made up of a system of Oracle Names servers that provide name-to-address resolution for each service on the network.

Oracle Names Control utility

A utility included with Oracle Names to control various functions for Oracle Names servers, such as starting, stopping, and getting the status of an Oracle Names server.

Oracle Names LDAP Proxy server

Oracle Names servers that have been configured to proxy for **directory servers**. Upon startup, Oracle Names LDAP Proxy servers obtain network object information from a directory server. This provides a single point of definition for all data in a directory server and does not require that both Oracle Names servers and directory servers be maintained separately and simultaneously.

Oracle Names server

A computer that uses Oracle Names software to store network address information for a service along with its simple name so that client applications can request connections with simple names rather than lengthy addresses.

Oracle Net

A component of **Oracle Net Services** that enables a network session from a client application to an Oracle database server. Once a network session is established, Oracle Net acts as a data courier for the client application and the database server. It is responsible for establishing and maintaining the connection between the client application and database server, as well as exchanging messages between them. Oracle Net is able to perform these jobs because it is located on each computer in the network.

Oracle Net Configuration Assistant

A post-installation tool that configures basic network components after installation, including:

- Listener names and protocol addresses
- Naming methods the client will use to resolve **connect identifiers**
- Net service names in a `tnsnames.ora` file
- Directory server usage

Oracle Net Firewall Proxy

Product offered by some firewall vendors that supplies **Oracle Connection Manager** functionality.

Oracle Net foundation layer

A networking communication layer that is responsible for establishing and maintaining the connection between the client application and server, as well as exchanging messages between them.

Oracle Net Manager

A graphical user interface tool that combines configuration abilities with component control to provide an integrated environment for configuring and managing Oracle Net Services.

You can use Oracle Net Manager to configure the following network components:

- **Naming**

Define **connect identifiers** and map them to **connect descriptors** to identify the network location and identification of a service. Oracle Net Manager supports configuration of connect descriptors in a local `tnsnames.ora` file, centralized LDAP-compliant directory server, or an Oracle Names server.

- **Naming Methods**

Configure the different ways in which connect identifiers are resolved into connect descriptors.

- **Listeners**

Create and configure listeners to receive client connections.

Oracle Net Services

A suite of networking components that provide enterprise-wide connectivity solutions in distributed, heterogeneous computing environments. Oracle Net Services are comprised of the **Oracle Net**, **listener**, **Oracle Connection Manager**, **Oracle Net Configuration Assistant**, and **Oracle Net Manager**.

Oracle Program Interface (OPI)

A networking layer responsible for responding to each of the possible messages sent by **OCI**. For example, an OCI request to fetch 25 rows would have an OPI response to return the 25 rows once they have been fetched.

Oracle protocol support

A software layer responsible for mapping **Transparent Network Substrate (TNS)** functionality to industry-standard protocols used in the client/server connection.

Oracle Rdb

A database for Digital's 64-bit platforms. Because Oracle Rdb has its own listener, the client interacts with Rdb in the same manner as it does with an Oracle database.

Oracle schema

A set of rules that determine what can be stored in a [directory server](#). Oracle has its own schema that is applied to many types of Oracle entries, including Oracle Net Services' entries. The Oracle schema for Oracle Net Services' entries includes the attributes the entries may contain.

Oracle System Identifier (SID)

A name that identifies a specific instance of a running pre-release 8.1 Oracle database. For any database, there is at least one instance referencing the database.

For pre-release 8.1 databases, SID is used to identify the database. The SID is included in the connect descriptor of a [tnsnames.ora file](#) and in the definition of the listener in the [listener.ora file](#).

ORACLE_HOME

An alternate name for the top directory in the Oracle directory hierarchy on some directory-based operating systems.

OSI

See [Open Systems Interconnection \(OSI\)](#).

Oracle9i JVM

Provides support for Java stored procedures, JDBC, SQLJ, Common Object Request Broker Architecture (CORBA), and Enterprise JavaBeans (EJBs) in the Oracle database.

P

packet

A block of information sent over the network each time a connection or data transfer is requested. The information contained in packets depends on the type of packet: connect, accept, redirect, data, and so on. Packet information can be useful in troubleshooting.

parameter

Information passed to a program, command, or function, such as a file specification, a keyword, or a constant value.

password

A string (word or phrase) used for data security and known only to its owner. Passwords are entered in conjunction with an operating system login ID, Oracle username, or account name, in order to connect to an operating system or software application (such as the Oracle database). Whereas the username or ID is public, the secret password ensures that only the owner of the username can use that name, or access that data.

PMON process

A process monitor database process that performs process recovery when a user process fails. PMON is responsible for cleaning up the cache and freeing resources that the process was using. PMON also checks on dispatcher and server processes and restarts them if they have failed. As a part of [service registration](#), PMON registers instance information with the listener.

preferred Oracle Names server

The Oracle Names server(s) preferred by a client for names resolution; usually the Oracle Names Server that is physically closest to the client, or available over the least expensive network link.

presentation layer

A networking communication layer that manages the representation of information that application layer entities either communicate or reference in their communication. Example of session layers are [Two-Task Common \(TTC\)](#) and [General Inter-ORB Protocol \(GIOP\)](#).

private database link

A database link created by one user for his or her exclusive use.

See also [database link](#), [global database link](#), and [public database link](#).

profile

A collection of parameters that specifies preferences for enabling and configuring Oracle Net Services' features on the client or server. A profile is stored and implemented through the `sqlnet.ora` file.

protocol

A set of rules that defines how data is transported across the network.

protocol address

An address that identifies the network address of a network object.

When a connection is made, the client and the receiver of the request, such as the **listener**, **Oracle Names server**, or **Oracle Connection Manager**, are configured with identical protocol addresses. The client uses this address to send the connection request to a particular network object location, and the recipient "listens" for requests on this address. It is important to install the same protocols for the client and the connection recipient, as well as to configure the same addresses.

protocol conversion

A feature of Oracle Connection Manager that enables a client and server with different networking protocols to communicate with each other. This feature replaces functionality previously provided by the Oracle Multi-Protocol Interchange with SQL*Net version 2.

protocol stack

Designates a particular **presentation layer** and **session layer** combination.

proxy process

An **Oracle Connection Manager** worker process. Proxy processes receive the connections and evaluate against a set of rules whether to deny or allow access. If access is allowed, the proxy processes then forward the requests to the next hop, typically the database server. In addition to allowing or denying access, the proxy processes can also multiplex or funnel multiple client connections through a single protocol connection.

proxy server

A server that substitutes for the real server, forwarding client connection requests to the real server or to other proxy servers. Proxy servers provide access control, data and system security, monitoring, and caching.

public database link

A database link created by a DBA on a local database that is accessible to all users on that database.

See also **database link**, **global database link**, and **private database link**.

R

RDBMS

Relational Database Management System

RDN

See [relative distinguished name \(RDN\)](#).

relative distinguished name (RDN)

The local, most granular level entry name. It has no other qualifying entry names that would serve to address the entry uniquely. In the example, `cn=sales,dc=us,dc=acme,dc=com, cn=sales, cn=sales` is the RDN.

region

See [administrative region](#).

region database

Tables in an Oracle database that store Oracle Names information.

root administrative region

The highest level administrative region in a distributed installation. The root administrative region contains the root domain.

root domain

The highest level domain in a hierarchical naming model.

RPC

Remote Procedure Call

S

Secure Sockets Layer (SSL)

An industry standard protocol designed by Netscape Communications Corporation for securing network connections. SSL provides authentication, encryption, and data integrity using public key infrastructure (PKI).

server process

Database processes that handle a client request on behalf of a database.

service

Work done for others. The database is a service that stores and retrieves data for clients.

service handler

A process that acts a connection point from the listener to the database server. A service handler can be a **dispatcher**, or **dedicated server**.

service name

A logical representation of a database, which is the way a database is presented to clients. A database can be presented as multiple services and a service can be implemented as multiple database instances. The service name is a string that is the **global database name**, that is, a name comprised of the database name and domain name, entered during installation or database creation. If you are not sure what the global database name is, you can obtain it from the value of the `SERVICE_NAMES` parameter in the initialization parameter file.

The service name is included in the **connect data** part of the **connect descriptor**.

service registration

A feature by which the **PMON process** automatically registers information with a listener. Because this information is registered with the listener, the `listener.ora` file does not need to be configured with this static information.

Service registration provides the listener with information about:

- Service names for each running instance of the database
- Instance names of the database
- Service handlers (dispatchers and dedicated servers) available for each instance
 - These enable the listener to direct a client request appropriately.
- Dispatcher, instance, and node load information

This load information enables the listener to determine which dispatcher can best handle a client connection request. If all dispatchers are blocked, the listener can spawn a dedicated server for the connection.

service replication

A process that fully replicates a directory system on the network. New services need to register with only one **Oracle Names server**. The service replication process automatically distributes the new registration to all other active Oracle Names servers on the network.

session data unit (SDU)

A buffer that Oracle Net uses to place data before transmitting it across the network. Oracle Net sends the data in the buffer either when requested or when it is full.

session layer

A network layer that provides the services needed by the **protocol address** entities that enable them to organize and synchronize their dialogue and manage their data exchange. This layer establishes, manages, and terminates network sessions between the client and server. An example of a session layer is **Network Session (NS)**.

session multiplexing

Combining multiple sessions for transmission over a single network connection in order to conserve the operating system's resources.

shared server

A database server that is configured to allow many user processes to share very few server processes, so the number of users that can be supported is increased. With shared server configuration, many user processes connect to a **dispatcher**. The dispatcher directs multiple incoming network session requests to a common queue. An idle shared server process from a shared pool of server processes picks up a request from the queue. This means that a small pool of server processes can serve a large number of clients. Contrast with **dedicated server**.

shared server process

A process type used with **shared server** configuration.

SID

Oracle System Identifier. See **Oracle System Identifier (SID)**.

SID_LIST_/*listener_name*

A section of the `listener.ora` file that defines the **Oracle System Identifier (SID)** of the database served by the listener. This section is valid only for version 8.0 Oracle databases, as Oracle9*i* and Oracle8*i* instance information is automatically registered with the listener. Static configuration is also required for other services, such as external procedures and heterogeneous services, and some management tools, including Oracle Enterprise Manager.

single sign-on

The ability for a user to log in to different servers using a single password. This permits the user to authenticate to all servers the user is authorized to access.

sqlnet.ora file

A configuration file for the client or server that specifies:

- Client domain to append to unqualified service names or net service names
- Order of naming methods the client should use when resolving a name
- Logging and tracing features to use
- Route of connections
- Preferred Oracle Names servers
- External naming parameters
- Oracle Advanced Security parameters

The `sqlnet.ora` file typically resides in `$ORACLE_HOME/network/admin` on UNIX platforms and `ORACLE_HOME\Network\admin` on Windows operating systems.

SSL

See **Secure Sockets Layer (SSL)**.

System Global Area (SGA)

A group of shared memory structures that contain data and control information for an Oracle **instance**.

system or topology data

Data used by the Oracle Names server to control regular functioning or communicate with other Oracle Names servers. Includes interchanges, Oracle Names servers in a root region, and Oracle Names servers in any delegated region.

T

TCP/IP protocol

Transmission Control Protocol/Internet Protocol. The de facto standard communication protocol used for client/server conversation over a network.

TCP/IP with SSL protocol

A protocol that enables an Oracle application on a client to communicate with remote Oracle databases through the **TCP/IP protocol** and **Secure Sockets Layer (SSL)**.

tick

The amount of time it takes for a message to be sent and processed from the client to the server or from the server to the client

Thin JDBC Driver

Thin JDBC driver is Oracle's Type 4 driver designed for Java applet and Java application developers. The JDBC driver establishes a direct connection to the Oracle database server over Java sockets. Access to the database is assisted with a lightweight implementation of Oracle Net and **Two-Task Common (TTC)**.

TNS

See [Transparent Network Substrate \(TNS\)](#).

tnsnames.ora file

A configuration file that contains one or more **domains** mapped to **connect descriptors**. This file is used for the **LU6.2 protocol** method. The `tnsnames.ora` file typically resides in `$ORACLE_HOME/network/admin` on UNIX platforms and `ORACLE_HOME\Network\Admin`.

tracing

A facility that writes detailed information about an operation to an output file. The trace facility produces a detailed sequence of statements that describe the events of an operation as they are executed. Administrators use the trace facility for diagnosing an abnormal condition; it is not normally turned on.

See also [logging](#).

Transparent Application Failover (TAF)

A runtime failover for high-availability environments, such as Oracle9*i* Real Application Clusters and Oracle Fail Safe, that refers to the failover and re-establishment of application-to-service connections. It enables client applications to automatically reconnect to the database if the connection fails, and, optionally, resume a `SELECT` statement that was in progress. This reconnect happens automatically from within the Oracle Call Interface (OCI) library.

Transparent Network Substrate (TNS)

A foundation technology, built into the **Oracle Net foundation layer** that works with any standard network transport protocol.

transport

A networking layer that maintains end-to-end reliability through data flow control and error recovery methods. The **Oracle Net foundation layer** uses **Oracle protocol support** for the transport layer.

TTC

See **Two-Task Common (TTC)**.

Two-Task Common (TTC)

A **presentation layer** type that is used in a typical Oracle Net connection to provide character set and data type conversion between different character sets or formats on the client and server.

U

UPI

User Program Interface

V

virtual circuit

A piece of shared memory used by the **dispatcher** for client database connection requests and replies. The dispatcher places a virtual circuit on a common queue when a request arrives. An idle shared server picks up the virtual circuit from the common queue, services the request, and relinquishes the virtual circuit before attempting to retrieve another virtual circuit from the common queue.

VI protocol

Virtual Interface Protocol. A protocol used to improve latency and CPU utilization between application Web servers and database servers.

W

well-known Oracle Names server

Addresses for one or more Oracle Names servers hard coded into both the Oracle Names server and its clients. Oracle Names servers then become available at these well known addresses, so that clients do not need to be told, by way of configuration files, where to find the server.

Windows NT native authentication

An **authentication method** that enables a client single login access to a Windows NT server and a database running on the server.

Index

Numerics

- 1521 port, 9-24, 12-7
 - if not using, 12-15
- 1575 port, 3-33, 9-29
- 1630 port, 4-18, 13-15

A

- absolute naming
 - Java Database Connectivity (JDBC)
 - OCI drivers, 3-11, 16-9
 - Thin drivers, 16-9
 - with directory naming, 3-12
- Access Control Lists (ACLs)
 - described, 3-14
- access control to database
 - with Oracle Connection Manager, 1-21, 13-14
 - with sqlnet.ora parameters, 11-8
- ADDRESS (ADD or ADDR) attribute, 14-2
- Address List Options dialog box, 15-6
- administrative regions
 - defined, 9-20
 - delegated, 3-28
 - delegated, below root, 3-30
 - delegating domains, 9-50
 - described, 3-21
 - domain hints to remote, 9-51
 - in Oracle Names, 3-21
 - root, 3-28
- Advanced Program-to-Program Communication (APP), 4-6
- alert log
 - ORA-3113 error message, 17-14

- Always Use Dedicated Server option in Oracle Net Manager, 11-7
- APPC architecture, 4-6
- application Web servers, 1-6 to 1-8
 - architecture
 - listener, 4-10
 - Oracle Connection Manager, 4-17
 - Oracle Names, 3-18
 - Oracle Net stack communication layers, 4-2
- A.SMD Oracle Names record, 12-19
- Attempts Per Names Server field in Oracle Net Manager, 9-37
- attributes
 - c, 3-7
 - cn, 3-7
 - commonName, 3-7
 - country, 3-7
 - defined, 3-6
 - domain component, 3-7
 - o, 3-7
 - organization, 3-7
 - organizationUnitName, 3-7
 - ou, 3-7
- audit trail
 - described, 17-30
 - script for using information, 17-31
- authentication
 - with Oracle Advanced Security, 1-24, 1-25

B

- BACKGROUND_DUMP_DEST initialization parameter, 17-14
- BACKUP networking parameter, 15-21

BASE networking parameter, 10-19
Bequeath protocol, 2-12
BEQUEATH_DETACH networking parameter, 11-10
buffer flushing, described, 7-13

C

c attribute, 3-7
C:\ORANT, defined, xxx
C:\ORAWIN95, defined, xxx
CDS. See Cell Directory Service (CDS)
Cell Directory Services (CDS) external naming, 2-17, 11-6
configuring, 9-60
Cell Name field in Oracle Net Manager, 11-12
centralized naming, 3-2
advantages and disadvantages, 7-11
described, 2-16
establishing a connection with, 3-2
recommended for, 7-11
CHANGE_PASSWORD command, 12-13
Check TCP/IP Client Access Rights option in Oracle Net Manager, 11-8
ckpcch.ora file, 9-21
ckpcf.ora file, 9-21
ckpdom.ora file, 9-21
ckptop.ora file, 9-21, 10-16
client configuration
connect-time failover, 2-8
default domains, 11-3
discovering Oracle Names servers, 9-34
load balancing requests among listeners, 2-8, 15-4
local naming, 9-11
log files, 17-25
net service names, 6-5 to 6-7
Oracle Connection Manager address, 13-5
Oracle Rdb connections, 15-45
overriding shared server configuration, 14-10
preferred Oracle Names servers, 11-11
shared server, 14-9
sqlnet.log file, 17-25
trace files, 17-41
Transparent Application Failover (TAF), 2-8

client connections
methods, 16-13
syntax, 16-9
client load balancing, 2-8
configuring, 15-4
Client Registration ID field in Oracle Net Manager, 11-9
client testing
connectivity, 16-17
using application, 16-13
using operating system, 16-13
using special commands, 16-14
with 3GL, 16-14
with Oracle Net Manager, 16-22
client troubleshooting, 17-4
Clients Allowed to Access field in Oracle Net Manager, 11-8
Clients Excluded from Access field in Oracle Net Manager, 11-8
cmadm_pid.log file, 17-24
cmadm_pid.trc file, 17-40
CMADMIN process, 4-18
cmadmpid.log file, 17-24
cmadmpid.trc file, 17-40
CMAN networking parameter, 13-4
CMAN_ADMIN networking parameter, 13-4
cman_pid.log file, 17-24, 17-35
cman_pid.trc file, 17-40
CMAN_PROFILE networking parameter, 13-4
CMAN_RULES networking parameter, 13-4
cman.ora file
parameters
CMAN, 13-4, 13-15
CMAN_ADMIN, 13-4
CMAN_PROFILE, 13-4
CMAN_RULES, 13-4, 13-14
LOG_LEVEL, 17-27
TRACE_DIRECTORY, 17-46
TRACE_LEVEL, 17-46
setting up, 13-3
cmanpid.log file, 17-24
cmanpid.trc file, 17-40
CMGW process, 4-18
cn attribute, 3-7

configuring
access control, 11-8, 13-14
clients
 connect-time failover, 2-8, 15-4
 default domains, 11-3
 local naming, 9-11
 naming methods in profile, 11-5
 net service names, 6-5 to 6-7
 Oracle Names, 9-34
 session data unit, 15-28
 Transparent Application Failover (TAF), 2-8
connection pooling, 14-8
connection requests in a profile, 11-7
database servers
 local naming, 9-6
 Oracle Names, 9-34
 session data unit (SDU), 15-29
database service names
 Oracle Names, 9-43
default network domains, 11-3
directory naming, 9-12 to 9-19
domains, default, 11-11
external naming, 9-60 to 9-63
external procedures, 15-36
Heterogeneous Services connections, 15-41
host naming, 9-56 to 9-59
instance role, 15-25
listeners
 directory naming, 9-17
 host naming, 9-57
 local naming, 9-11
 Oracle Names, 9-41
 session data unit, 15-29
local naming, 9-5 to 9-11
localized management, 3-2
multiple addresses, 15-2
multiple domains in Oracle Names, 9-27, 9-32
naming methods, 9-1 to 9-63
net service names
 directory naming, 9-13
 external naming, 9-60
 local naming, 9-6
 Oracle Names, 9-44
Oracle Advanced Security, 11-13
Oracle Names, 9-19 to 9-53

Oracle Names servers, 9-24
Oracle Rdb database connections, 15-44
primary and secondary instances, 15-25
protocol addresses, 12-6
protocol conversion support, 13-15
quick reference, 5-19 to 5-22
region database for Oracle Names, 9-22, 9-28
replicated data among Oracle Names
 servers, 9-29
session data unit (SDU), 15-28
session multiplexing, 13-13
VI protocol, 15-31 to 15-34
connect identifiers, 16-9
connect strings, 16-2
connection load balancing, 2-9, 15-10
 dedicated server configurations, 15-15
service registration, 12-3
shared server configurations, 15-12
connection pooling
 advantages relative to session multiplexing, 7-4
 configuring, 14-8
 defined, 1-15
 using with shared server, 14-8
connections, 2-7
 adjusting listener queue size to avoid
 errors, 7-7, 7-12, 12-9
 bequeath, 2-12
 concurrent, increasing number of, 12-9
 connect strings, entering, 16-2, 16-8
 dedicated servers, 11-7
 directory naming, 9-17
 external procedures, 15-36
 Heterogeneous Services, 15-41
 host naming, 9-59
 local naming, 9-11
 methods, 16-13
 Network Information Service (NIS) external
 naming, 9-60
 Network Information Service external
 naming, 9-60
 Oracle Connection Manager, 11-7
 Oracle Names, 9-53
 Oracle Rdb databases, 15-44
 startup of components, 16-2
CONNECTIONS (CON or CONN) attribute, 14-3

connect-time failover
configuring, 15-4
GLOBAL_DBNAME networking parameter in
 listener.ora, 12-10, 15-4
service registration, 12-3
Create additional or upgrade existing Oracle
 Context option in Oracle Net Configuration
 Assistant, 8-8
Create or upgrade the Oracle Schema option in
 Oracle Net Configuration Assistant, 8-8

D

data transfer, maximizing, 7-13
database links, 3-24, 9-46
 global database link, 3-24
 private database, 3-24
 public database, 3-24
database server configuration
 allocating resources with shared server, 14-9
 discovering Oracle Names servers, 9-34
 external procedures, 15-32, 15-40
 local naming, 9-6
 log files, 17-25
 region database for Oracle Names, 9-22
 shared server, 14-2
database server connections
 methods, 16-13
 syntax, 16-9
database server testing, 16-16
database servers
 connecting, 6-7
 loopback tests, 17-4
 repository for Oracle Names, 3-22
 starting, 6-4, 16-6
 trace files, 17-41
database service names
 Oracle Names, 9-43
Database Services tab in Oracle Net
 Manager, 12-11
dc attribute, 3-7
dead connection detection
 limitations, 11-9

dedicated servers
 architecture, 1-13
 bequeathed sessions, 2-12
 connect descriptor configuration, 2-7
 defined, 1-12
 described, 2-12 to 2-14
 difference with shared server, 1-12
 routing connections, 11-7
Default Domain field in Oracle Net Manager, 9-37
default network domains
 configuring, 11-3
 in Oracle Names, 3-26
DELAY networking parameter, 15-21
DELEGATE_DOMAIN command, 9-50
delegated administrative regions
 below root, 3-30
 diagram, 3-29
 domains to, 9-50
 in Oracle Names, 3-28
DESCRIPTION parameter, 9-5
diagnosing
 client problems, 17-4
 errors. See troubleshooting
direct hand-off
 described, 2-10
 events in listener.log, 17-34
directory configuration
 adding users to the OracleNetAdmins
 group, 8-9
 after installation, 8-5
 choosing a directory server, 8-6
 default configuration, 8-2
 exporting
 net service names from a tnsnames.ora
 file, 10-2
 objects from Oracle Names, 10-6
 for clients and servers, 9-13
 migrating
 net service names from a tnsnames.ora
 file, 10-2
 objects from Oracle Names, 10-6
 Oracle Context creation, 8-8
 Oracle schema creation, 8-8
 setting up for first-time usage, 8-7
directory information tree (DIT), defined, 3-6

directory naming, 17-15
absolute naming, 3-12
advantages and disadvantages, 7-10
architecture, 3-6
authentication methods
 native, 3-14
 simple, 3-14
 SSL, 3-14
 strong, 3-14
configuring, 9-12 to 9-19
connecting to database services, 3-10
connecting with, 9-17
connections
 using an entry's absolute name, 3-12
 using an entry's relative name, 3-11
default configuration, 8-2
directory configuration, 9-13
establishing a connection with, 9-17
exporting
 net service names from a tnsnames.ora file, 10-2
objects from Oracle Names, 10-6 to 10-14
Java Database Connectivity (JDBC)
 OCI drivers, 3-11, 16-9
 Thin drivers, 3-11, 16-9
Java Database Connectivity (JDBC) OCI drivers, 3-5
Java Database Connectivity (JDBC) Thin drivers, 3-5, 3-12
ldap_search limitations, 17-16
ldapmodify tool, 8-9
ldap.ora file
 creating, 8-3
 described, 3-10
ldapwrite tool, 17-15
listener configuration for, 9-17
migrating
 net service names from a tnsnames.ora file, 10-2
 objects from Oracle Names, 10-6 to 10-14
object classes, 3-16
Oracle Context, 3-7
Oracle Names LDAP Proxy servers, 10-15
Oracle schema, 3-16
OracleDBCreators group, 3-10, 3-16, 8-7, 8-8

OracleNetAdmins group, 3-10, 3-15, 8-7, 8-8, 8-9, 9-13, 9-18
overview, 3-4
recommended for, 7-10
security, 3-13
troubleshooting, 17-15
with relative naming, 3-11
Directory Server Migration Wizard, 5-12, 10-5
directory servers
 attributes, 3-6
 directory information tree (DIT), 3-6
 distinguished name (DN), 3-6
 entry, 3-6
 ldap_search tool, 17-16
 ldapmodify tool, 8-9
 ldapwrite tool, 17-15
 Microsoft Active Directory, 3-5
 Oracle Context, 3-7
 Oracle Internet Directory, 3-5
 overview, 1-11
 performance, 3-13
 relative distinguished name (RDN), 3-6
 security, 3-13
Directory Usage configuration option in Oracle Net Configuration Assistant, 5-15
Directory Usage Configuration page in Oracle Net Configuration Assistant, 8-6
Disable Out-of-Band Break option in Oracle Net Manager, 11-10
DISABLE_OOB networking parameter, 11-10
Disc field for VI protocol address in Oracle Net Manager, 15-31
DISC parameter, 15-31
discovering Oracle Names servers
 described, 3-33
 procedure, 9-33
dispatchers, 1-12, 4-14
 described, 1-12, 2-10 to 2-11, 4-14
 number to start, 14-6
 setting the number of, 14-6
DISPATCHERS (DIS or DISP) attribute, 14-3

DISPATCHERS initialization parameter, 14-2
 ADDRESS attribute, 14-2
 configuring connection pooling, 14-8
 CONNECTIONS attribute, 14-3
 DISPATCHERS attribute, 14-3
 LISTENER attribute, 12-15, 12-16, 14-4
 MULTIPLEX attribute, 14-4
 POOL attribute, 14-5
 PROTOCOL attribute, 14-2
 SERVICE attribute, 14-5
 SESSIONS attribute, 14-5
 setting initially, 14-6
 TICKS attribute, 14-6
DISPATCHERS parameter
 MULTIPLEX attribute, 13-13
 PROTOCOL attribute, 13-13
distinguished name (DN), defined, 3-6
DLCR.RDBMS.OMD Oracle Names record, 16-24
domain hints, 9-51
domains
 default network, 3-26
 delegating, 9-50
 described, 3-21, 9-20
 multiple hierarchically, 9-27, 9-32
 multiple hierarchically related, 3-27
 required by root administrative regions, 3-28
 with a single domain naming structure,
 diagram, 3-25
Dynamic Discovery Option, 3-31

E

enterprise user security, 3-7
entry, defined, 3-6
error messages
 contacting Oracle Support Services, 17-54
 in trace file, 17-51
 NNL-00018, 9-28
 NNL-00024, 9-28
 NNO-00062, 17-16, 17-17
 NNO-00850, 17-16, 17-17
 ORA-00508, 17-11
 ORA-1017, 17-3
 ORA-1034, 17-3
 ORA-12154, 17-8

ORA-12198, 17-9
ORA-12203, 17-9, 17-10, 17-11, 17-18
 sample error stack, 17-22
ORA-12224, 17-11
ORA-12514, 17-12
ORA-12520, 17-12
ORA-12521, 17-13
ORA-12533, 17-12
ORA-12535, 17-18
ORA-12538, 17-11
ORA-12545, 17-13
ORA-12547, 17-18
ORA-12560, 17-14
ORA-3113, 17-14
ORA-3121, 17-14
resolving, 17-1 to 17-55
 using log file to track, 17-29
error stack
 described, 17-21
 entries in log files, 17-29
 sample, 17-22
 typical layers in, 17-21
establishing a session
 using centralized naming, 2-16
external naming
 advantages and disadvantages, 7-12
 Cell Directory Services (CDS), 2-17, 9-60, 11-6
 configuring, 9-60 to 9-63
 Network Information Service (NIS), 2-17, 9-60,
 11-6
 recommended for, 7-12
external procedures
 configuring, 15-36
 described, 15-36
 extproc program, 15-36
 extproc program, 15-36

F

failover
 connect-time, 2-8, 15-4
 Transparent Application Failover, 2-8
 Transparent Application Failover (TAF), 15-18
FAILOVER networking parameter, 15-4, 15-14,
 15-16

FAILOVER_MODE networking parameter, 15-18, 15-21
features, new
 client load balancing, xxxix
 connection load balancing, xl
 connection load balancing for dedicated server
 configurations, xxv
 connect-time failover, xxxix
 creating multiple Oracle Contexts, xxxv
 direct hand-off, xxxix
 directory naming, xxxix
 instance role specification for primary/secondary
 configurations, xxxix
name changes
 CIRCUITS initialization parameter, xxxiv
 DISPATCHERS initialization
 parameter, xxxiv
 MAX_DISPATCHERS initialization
 parameter, xxxiv
 MAX_SHARED_SERVERS initialization
 parameter, xxxiv
 Oracle Net, xxxiv
 Oracle Net Configuration Assistant, xxxiv
 Oracle Net foundation layer, xxxiv
 Oracle Net Manager, xxxiv
 shared server, xxxiv
 SHARED_SERVER_SESSIONS initialization
 parameter, xxxiv
 SHARED_SERVERS initialization
 parameter, xxxiv
 V\$SHARED_SERVER_MONITOR
 view, xxxiv
Oracle Names LDAP Proxy servers, xxxv
Oracle Net Configuration Assistant, xl
service naming, xxxix
service registration, xl
SERVICE_NAME parameter, xl
TCP/IP with SSL, xl
VI protocol, xxxv
FILE networking parameter, 10-20

G

global database link qualifiers, 9-46
global database links, 3-24, 9-46
 adding link qualifiers, 9-47
global database name
 configuring on the listener, 12-10, 12-14
 defined, 9-3
 described, 12-10, 12-14
Global Database Name field in Oracle Net Manager, 12-10, 15-8, 15-44
GLOBAL_DBNAME networking parameter, 12-10
GLOBAL_NAME networking parameter, 15-8, 15-44

H

Heterogeneous Services
 configuring connections to, 15-41
 described, 15-41
hierarchical naming model
 described, 3-26
 diagram, 3-26
host naming
 advantages and disadvantages, 7-11
 configuring, 9-56 to 9-59
 connecting with, 9-59
 establishing a connection with, 9-59
 listener configuration for, 9-57
 recommended for, 7-11
 requirements, 9-56
HOST networking parameter, 10-19
HOST parameter, 15-31
hosts file, 3-31
HS networking parameter, 15-8

I

Identix authentication, xxxvi
Initial Preallocated Requests field in Oracle Net Manager, 9-37
initialization parameter file
 BACKGROUND_DUMP_DEST
 parameter, 17-14
 DISPATCHERS parameter, 14-2
 INSTANCE_NAME parameter, 2-4

LOCAL_LISTENER parameter, 12-7, 12-15
REMOTE_LISTENER parameter, 12-16
SERVICE_NAMES parameter, 2-2, 9-3, 14-5
installation
 default configuration
 directory naming, 8-2
 listeners, 12-4
 local naming, 9-6
 profiles, 11-2
 Oracle Names, 9-20
Instance Name field in Oracle Net Manager, 15-8
instance role configuration, 15-25
 connections in TAF, 15-27
 connections to primary and secondary
 instances, 15-25
 connections to specific instances, 15-26
INSTANCE_NAME networking parameter, 2-4,
 9-3, 15-8, 15-25

J

Java Database Connectivity (JDBC)
 described, 1-5
 drivers, 1-5, 4-8
 OCI drivers, 4-8
 absolute naming support, 3-11, 16-9
 directory naming support, 3-5
 relative naming support, 3-11
 Thin drivers, 4-8
 absolute naming support, 16-9
 directory naming support, 3-5, 3-12
 relative naming support, 3-11
JDBC. See Java Database Connectivity (JDBC)

L

ldap_search tool limitations, 17-16
ldapmodify tool, 8-9
ldap.ora file
 creating, 3-10, 8-3
 described, 3-2
ldapwrite tool, 17-15
listener
 defined, 1-29

LISTENER (LIS or LIST) attribute, 12-15, 12-16,
 14-4
Listener configuration option in Oracle Net
 Configuration Assistant, 5-15
Listener Control utility
 commands
 SERVICES, 6-5, 12-23
 SET PASSWORD, 16-5
 START, 6-4, 16-5
 STATUS, 12-20, 16-5
 STOP, 16-5
 starting a listener, 12-20
 stopping a listener, 12-20
 using, 5-16
listener.log file, 17-24
listener.ora file
 described, 3-2
parameters
 GLOBAL_DBNAME, 12-10
 LOG_DIRECTORY_*listener_name*, 17-26
 LOG_FILE_*listener_name*, 17-26
 ORACLE_HOME, 12-10, 15-38, 15-41
 PASSWORDS_*listener_name*, 12-13
 PROGRAM, 15-38, 15-41
 SID_NAME, 12-10, 15-38, 15-41
 STARTUP_WAITTIME_*listener_name*, 17-44
 TRACE_FILE_*listener_name*, 17-44
 TRACE_LEVEL_*listener_name*, 17-44
listeners, 2-6, 15-41
 adjusting queue size for, 12-9
 client load balancing, 2-8
 configuration file, 9-2
 configuring
 address for Oracle Connection
 Manager, 13-7
 address list, 15-2
 directory naming method, 9-17
 external procedures connections, 12-10,
 15-36
 global database name, 12-10, 12-14
 Heterogeneous Services connections, 12-10,
 15-41
 host naming method, 9-57
 local naming method, 9-11
 multihomed hosts, 12-7

nondefault address, 12-15
Oracle Enterprise Manager, 12-10
Oracle home directory, 12-10
Oracle Names, 9-41
Oracle System Identifier, 12-10
Oracle9i JVM, 4-11, 12-8
protocol addresses, 12-6
SID, 12-10
connection load balancing, 2-9, 15-10
connect-time failover, 2-8
default address, 12-15
default configuration, 12-4
direct hand-off, 2-10
handling concurrent connections, 12-9
increasing queue size, 12-9
log files, 12-25, 17-26, 17-30
 audit trail, 17-30
 direct hand-off event information, 17-34
 service registration event information, 17-32
monitoring, 12-20, 12-23, 12-25
multihomed hosts, 12-7
multiple addresses, 15-2
passwords, setting, 12-13
queue size, 7-7, 7-12
redirect connections, 2-10
security, 12-13
starting, 6-4, 16-6
 from the command line, 16-5
 from the Control Panel, 16-5
testing configuration, 16-16
trace files, 17-44
 Transparent Application Failover (TAF), 2-8
listener.trc file, 17-40
load balancing
 client, 2-8
 connection, 2-9, 15-10
LOAD_BALANCE networking parameter, 15-4
LOAD_TNSNAMES command, 9-45
LOCAL environment variable, 16-8
local naming
 advantages and disadvantages, 7-10
 client configuration, 9-11
 configuring, 9-5 to 9-11
 connecting with, 9-11
 database server configuration, 9-6
default configuration, 9-6
establishing a connection with, 9-11
listener configuration for, 9-11
recommended for, 7-10
Local Net Service Name configuration option in
 Oracle Net Configuration Assistant, 5-15
LOCAL registry entry, 16-8
LOCAL_LISTENER initialization parameter, 12-7,
 12-15, 14-4
log files, 12-25, 17-25, 17-30
 cmadm_pid.log, 17-24, 17-27
 cmadmpid.log, 17-24, 17-27
 cman_pid.log, 17-24, 17-27, 17-35
 cmanpid.log, 17-24, 17-27
 default names for, 17-24
 listener.log, 17-24, 17-26
 names.log, 17-24, 17-26
 sqlnet.log, 17-24
 sqlnet.log for clients and database servers, 17-25
 using to track errors, 17-29
LOG_DIRECTORY_CLIENT networking
 parameter, 17-25
LOG_DIRECTORY_listener_name networking
 parameter, 17-26
LOG_DIRECTORY_SERVER networking
 parameter, 17-25
LOG_FILE_CLIENT networking parameter, 17-25
LOG_FILE_listener_name networking
 parameter, 17-26
LOG_FILE_SERVER networking parameter, 17-25
LOG_LEVEL networking parameter, 17-27
loopback test, 17-4
LU6.2 protocol
 described, 4-6
 supporting networks, 4-7
 supporting topologies, 4-7

M

maximizing data transfer, by adjusting SDU
size, 7-13
Maximum Open Connections field in Oracle Net
Manager, 9-37
Maximum Wait Each Attempt field in Oracle Net
Manager, 9-37

Meta Map field in Oracle Net Manager, 11-12
METHOD networking parameter, 15-21
Microsoft Active Directory, 3-5
multihomed hosts, 12-7
multiple addresses, 15-2, 15-4
 configuring client load balancing, 15-6
 configuring connect-time failover, 15-6
multiple listeners
 listeners
 multiple, 12-5
MULTIPLEX (MUL or MULT) attribute, 13-13, 14-4

N

Named Pipes protocol
 described, 4-6
Names Wizard, 5-13, 9-24, 9-29
NAMES.ADDRESSES networking parameter, 9-25, 9-30
NAMES.ADMIN_REGION networking
 parameter, 10-18
NAMES.ADMIN_REGION parameter, 9-25
NAMESTCL utility
 discovering Oracle Names servers, 9-35
NAMES.DCE.PREFIX networking
 parameter, 11-12
NAMES.DEFAULT_DOMAIN networking
 parameter, 9-37, 11-2, 11-4
NAMES DIRECTORY PATH networking
 parameter, 11-6
 cds, 11-6
 hostname, 11-5
 ldap, 11-5
 nis, 11-6
 onames, 11-5
 tnsnames, 11-5
NAMES.DOMAIN_HINTS networking
 parameter, 9-25, 9-30, 9-51
NAMES.DOMAINS networking parameter, 9-25, 9-27, 9-30, 9-32
NAMES.DOMAINS parameter, 9-27, 9-32
namesini.sql script, 9-23
NAMES.INITIAL_RETRY_TIMEOUT networking
 parameter, 9-37

names.log file, 17-24
NAMES.LOG_DIRECTORY networking
 parameter, 17-26
NAMES.LOG_FILE networking parameter, 17-26
NAMES.MAX_OPEN_CONNECTIONS
 networking parameter, 9-37
NAMES.MESSAGE_POOL_START_SIZE
 networking parameter, 9-37
NAMES.NIS.META_MAP networking
 parameter, 11-12
names.ora file
 described, 3-2
parameters
 BASE, 10-19
 FILE, 10-20
 HOST, 10-19
 NAMES.ADMIN_REGION, 10-18
 NAMES.LOG_DIRECTORY, 17-26
 NAMES.LOG_FILE, 17-26
 NAMES TRACE_DIRECTORY, 17-45
 NAMES TRACE_FILE, 17-45
 NAMES TRACE_LEVEL, 17-45
 NAMES TRACE_UNIQUE, 17-45
 PASSWORD, 10-19
 PORT, 10-19
 SCOPE, 10-19
 SUBTREE, 10-19
 SUBTREE_LIST, 10-19
 TIMEOUT, 10-19
 TYPE, 10-19, 10-20
 USERID, 10-19
namespid.trc file, 17-45
NAMES.PREFERRED_SERVERS networking
 parameter, 11-11, 11-12
NAMES.REQUEST_RETRIES networking
 parameter, 9-37
NAMES.SERVER_NAME networking
 parameter, 9-25, 9-30
NAMES TRACE_DIRECTORY networking
 parameter, 17-45
NAMES TRACE_FILE networking
 parameter, 17-45
NAMES TRACE_LEVEL networking
 parameter, 17-45

NAMES TRACE_UNIQUE networking
 parameter, 17-45
names.trc file, 17-40
namesupg.sql script, 9-23
naming methods
 Cell Directory Services (CDS), 2-17, 9-60
 centralized, 3-2
 choosing, 7-6, 7-10
 described, 2-16
 directory naming, 9-12 to 9-19
 external naming, 2-17, 9-60 to 9-63
 host naming, 9-56 to 9-59
 local naming, 2-16, 9-5 to 9-11
 localized, 3-2
 Network Information Service (NIS), 2-17, 9-60
 Oracle Names, 2-16, 9-19 to 9-53
 overview, 1-11
 prioritizing order, 11-5
Naming Methods configuration option in Oracle
 Net Configuration Assistant, 5-15
naming model
 hierarchical, 3-26
 single domain, 3-25
native authentication for directory naming, 3-14
NDS. See Novell Directory Services (NDS)
Net Service Name Wizard, 5-11, 9-7, 9-14, 13-5
net service names
 adding an address, 15-2
 configure an Oracle Connection Manager
 address, 13-5
 configuring, 6-5 to 6-7
 creating in Oracle Names, 9-44
 directory naming, 9-13
 external naming, 2-17, 9-60
 external procedures, 15-32, 15-40
 local naming, 2-16, 9-6
 multiple addresses, 15-2, 15-4
 Oracle Names, 9-44
 prioritizing naming methods, 11-5
 testing with TNSPING, 16-19
Net8 OPEN, xxxvii
Network Authentication (NA)
 layer in error stacks, 17-22
network availability, determining, 6-3

network configuration
 centralized management, 3-2
 localized management, 3-2
Network Information Service (NIS) external
 naming, 2-17, 11-6
configuring, 9-60
connecting with, 9-60
establishing a connection with, 9-60
maps, 9-61
Network Interface (NI)
 layer in error stacks, 17-21
Network Naming (NN)
 layer in error stacks, 17-21
network performance, improving
 by adjusting SDU size, 7-13
 client load balancing, 2-8
 listener queue size, 7-7, 7-12
network planning
 session data unit (SDU) size, 7-13
Network Routing (NR)
 layer in error stacks, 17-22
Network Session (NS), layer in error stacks, 17-22
Network Transport (NT), layer in error
 stacks, 17-22
networking configuration files
 ldap.ora file, 3-2
 listener.ora file, 3-2
 names.ora file, 3-2
 sqlnet.ora file, 3-3
 tnsnames.ora file, 3-3
networking planning
 internal networks
 availability, 7-5
 client load balancing, 7-5
 connection pooling, 7-3
 connect-time failover, 7-5
 JDBC drivers, 7-6
 listener queue size, 7-7
 naming methods, 7-6
 protocol conversion, 7-7
 scalability, 7-3
 security, 7-6
 session data unit (SDU) size, 7-7
 session multiplexing, 7-3
 tuning and performance, 7-7
 VI protocol, 7-7

- Internet networks**
- access control, 7-9
 - availability, 7-9
 - JDBC drivers, 7-9
 - naming methods, 7-9
 - scalability, 7-9
 - security, 7-9
 - tuning and performance, 7-10
 - VI protocol, 7-10
- new features**
- client load balancing, xxxix
 - connection load balancing, xl
 - connection load balancing for dedicated server configurations, xxxv
 - connect-time failover, xxxix
 - creating multiple Oracle Contexts, xxxx
 - direct hand-off, xxxix
 - directory naming, xxxix
 - instance role specification for primary/secondary configurations, xxxix
 - name changes
 - CIRCUITS initialization parameter, xxxiv
 - DISPATCHERS initialization
 - parameter, xxxiv
 - MAX_DISPATCHERS initialization
 - parameter, xxxiv
 - MAX_SHARED_SERVERS initialization
 - parameter, xxxiv
 - Oracle Net, xxxiv
 - Oracle Net Configuration Assistant, xxxiv
 - Oracle Net foundation layer, xxxiv
 - Oracle Net Manager, xxxiv
 - shared server, xxxiv
 - SHARED_SERVER_SESSIONS initialization
 - parameter, xxxiv
 - SHARED_SERVERS initialization
 - parameter, xxxiv
 - V\$SHARED_SERVER_MONITOR view, xxxiv
 - Oracle Names LDAP Proxy servers, xxxx
 - Oracle Net Configuration Assistant, xl
 - service naming, xxxix
 - service registration, xl
 - SERVICE_NAME parameter, xl
- TCP/IP with SSL, xl**
- VI protocol, xxxv**
- NIS. See Network Information Service (NIS)**
- NNL-00018 error message, 9-28**
- NNL-00024 error message, 9-28**
- NNO-00062 error message, 17-16, 17-17**
- NNO-00850 error message, 17-16, 17-17**
- nodes, described, 4-4**
- Novell Directory Services (NDS)**
- authentication, xxxvi
- Novell Directory Services (NDS) external naming, xxxvi**
- NS.SMD Oracle Names record, 16-24**
-
- O**
-
- o attribute, 3-7**
- object classes**
- described, 3-16
 - orclDBServer, 3-16
 - orclNetAddress, 3-17
 - orclNetAddressList, 3-17
 - orclNetDescription, 3-17
 - orclNetDescriptionList, 3-17
 - orclNetService, 3-16
- OCI, described, 4-4**
- ONRS_CONFIG table, 9-21**
- ORA-00508 error message, 17-11**
- ORA-1017 error messages, 17-3**
- ORA-1034 error messages, 17-3**
- ORA-12154 error message, 17-8**
- ORA-12198 error message, 17-9**
- ORA-12203 error message, 17-9, 17-10, 17-11, 17-18**
- sample error stack, 17-22
- ORA-12224 error message, 17-11**
- ORA-12514 error message, 17-12**
- ORA-12520 error message, 17-12**
- ORA-12521 error message, 17-13**
- ORA-12533 error message, 17-12**
- ORA-12535 error message, 17-18**
- ORA-12538 error message, 17-11**
- ORA-12545 error message, 17-13**
- ORA-12547 error message, 17-18**
- ORA-12560 error message, 17-14**
- ORA-3113 error message, 17-14**

ORA-3121 error message, 17-14
Oracle Advanced Security
 authentication, 1-24, 1-25
 configuring with Oracle Net Manager, 11-13
 overview, 1-32
 single sign-on, 1-24
Oracle Call Interface (OCI) layer, described, 4-4
Oracle Connection Manager
 and VI protocol performance, 15-35
 CMADMIN process, 4-18
 CMGW process, 4-18
 configuring
 access control, 13-14
 clients, 13-5, 13-6
 database server, 13-12
 Oracle Connection Manager
 computer, 13-3 to 13-4
 protocol address for listener, 13-7
 protocol address for Oracle Connection Manager, 13-5
 protocol conversion support, 13-15
 session multiplexing, 13-13
log files, 17-27
 cmadm_pid.log, 17-24
 cmadmpid.log, 17-24
 cman_pid.log, 17-24
 cmanpid.log, 17-24
 understanding, 17-35
overview, 1-30
processes, 4-17
routing connections, 11-7
shared server, 13-2
starting, 16-7
testing, 16-16
trace files, 17-46
 cmadm_pid.trc, 17-40
 cmadmpid.trc, 17-40
 cman_pid.trc, 17-40
 cmanpid.trc, 17-40
 configuring, 17-46
Oracle Connection Manager Control utility
 commands
 START, 16-7
 using, 5-17

Oracle Context
 creating, 8-3, 8-8
 defined, 3-7
Oracle Database Configuration Assistant
 directory naming, 3-9
Oracle Enterprise Manager
 SID_LIST_listener_name networking parameter in listener.ora, 12-10
Oracle home directory
 configuring on the listener, 12-10
Oracle Home Directory field in Oracle Net Manager, 12-10, 15-38, 15-41
Oracle Internet Directory, 3-5
Oracle Names, 2-16
 administrative regions, 3-21
 advantages and disadvantages, 7-11
 architecture, 3-18
 client cache, 9-38, 16-4
 configuring, 9-19 to 9-53
 clients and servers, 9-34
 region database, 9-22
 connecting with, 9-53
 creating Oracle Names servers, 9-21
 database as a repository for information, 3-22
 default network domains, 3-26
 delegated
 administrative regions, 3-30
 described, 3-19
 differences between versions, 3-30
 discovering Oracle Names servers, 9-33, 9-34
 discovery, 3-33
 domains, 3-21
 administering multiple, 9-27, 9-32
 delegated, 9-50
 hints to remote, 9-51
 Dynamic Discovery Option, 3-31
 establishing a connection with, 9-53
 exporting objects to directory
 server, 10-6 to 10-14
 hierarchical naming model, 3-26
 installing, 9-20
 listener configuration for, 9-41
 log files, 17-26
 multiple domains, 3-27
 multiple regions, 3-27

namesini.sql script, 9-23
namesupg.sql script, 9-23
naming network components, 3-25
querying objects, 16-24
recommended for, 7-11
region database, 3-22
regions, 3-21
registering
 database service names, 9-43
 global database link qualifiers, 9-46
 global database links, 9-46
 net service names, 9-44
 objects, 9-42
replicated data, 3-22, 9-29
root administrative region, 3-28
.sdns.ora file, 3-33
sdns.ora file, 3-33
single domain naming mode, 3-25
starting
 from Oracle Net Manager, 16-24
 from the command line, 16-24
 from the Control Panel, 16-7
starting from the command line, 16-24
storing
 aliases, 3-24
 database global database names and
 addresses, 3-23
 database links, 3-24
 net service names, 3-23
 other Oracle Names server names and
 addresses, 3-23
trace files, 17-45
version 1, 3-30
version 2, 3-31
version 8, 3-32

Oracle Names Control utility
 commands
 QUERY, 9-33, 16-24
 START, 9-33
 START_CLIENT_CACHE, 9-39, 16-4
 delegating domains, 9-50
 discovering Oracle Names servers, 9-33, 9-35
 loading tnsnames.ora file into Oracle
 Names, 9-45
 net service names, 9-44

querying objects, 16-24
REORDER_NS, 9-33
starting an Oracle Names server, 16-3
testing network objects with, 16-24
using, 5-18

Oracle Names LDAP Proxy servers
 configuring, 10-15
 described, 10-6
 error reporting, 17-16
 generating topology for, 10-16
 NAMES.ADMIN_REGION networking
 parameter, 10-18
 starting, 9-54, 10-18

Oracle Names servers
 ckpcch.ora file, 9-21
 ckpcfg.ora file, 9-21
 ckpdom.ora file, 9-21
 ckptop.ora file, 9-21, 10-16
 client cache, 9-38, 16-4
 configuring
 multiple domains, 9-27, 9-32
 creating, 9-21
 region database, 9-24
 replicated data, 9-29
 data stored in, 3-23
 default configuration, 9-22
 delegated administrative regions, 3-28
 delegated domains, 9-50
 deleting, 9-54
 described, 3-19
 discovery, 9-33
 domain hints, 9-51
 log files, 17-26
 preferred, 3-30, 11-11
 registering
 database service names, 9-43
 global database link qualifiers, 9-46, 9-47
 global database links, 9-46
 net service names, 9-44
 objects, 9-42
 removing, 9-54
 replicated data, 9-29
 root administrative regions, 3-28

starting, 9-28, 16-3
client cache, 9-38, 16-4
from Oracle Net Manager, 16-24
from the command line, 16-24
from the Control Panel, 16-7

storing
aliases, 3-24
database global database names and addresses, 3-23
database links, 3-24
net service names, 3-23
other Oracle Names server names and addresses, 3-23

testing, 16-23
TNSPING utility, 16-19
trace files, 17-45
well-known, 3-31

Oracle Net
buffers, 7-13
components, 1-29
defined, 1-27, 1-29
Java Database Connectivity (JDBC) drivers, 4-8
Oracle Net foundation layer, 1-27
overview, 1-3 to 1-32
scalability features, 1-12
understanding, 1-3 to 1-32

Oracle Net Configuration Assistant
described, 5-14
directory usage configuration
Choose a directory server and complete setup for first time usage of that directory option, 8-7
Create additional or upgrade existing Oracle Context option, 8-8
Create or upgrade Oracle Schema option, 8-8
Select the directory server you want to use option, 8-6

Directory Usage configuration option, 5-15
Directory Usage Configuration page, 8-6
listener configuration, 12-4
Listener configuration option, 5-15
local naming method, 9-10
Local Net Service Name configuration option, 5-15
Naming Methods configuration option, 5-15

net service names, 9-10
OracleDBCreators group, 3-10, 3-16, 8-7, 8-8
OracleNetAdmins group, 3-15, 8-7, 8-8
servers
listener configuration, 5-15
Oracle Context creation, 3-9
starting, 5-14
Welcome page, 5-14

Oracle Net foundation layer, 1-27, 4-4
Oracle Net Manager
adding addresses, 15-2
Address List Options dialog box, 15-6
clients
client load balancing, 15-4
connect-time failover, 15-4
default network domains, 11-3
local naming method, 9-7
Oracle Connection Manager, 13-5
Default Domain field, 9-37
delegating domains to administrative regions, 9-50, 9-54
described, 5-2
directory naming, 3-9
Directory Server Migration Wizard, 10-5
discovering Oracle Names servers, 9-35
external procedure connections, 15-36
Heterogeneous Services connections, 15-41
Instance Name field, 15-8
listeners
Database Services tab, 12-11
Global Database Name field, 12-10, 15-8
non-default through Oracle Names, 12-19
Oracle Home Directory field, 15-38, 15-41
Oracle Home Directory field for listeners, 12-10
Program Name field, 15-38, 15-41
protocol addresses, 12-6
SID field, 12-10, 15-38, 15-41
static service information, 12-10
local naming method, 9-7
menus
Command, 5-9
Edit, 5-8
File, 5-8
Help, 5-10

multiple address options, 15-4
Names Wizard, 9-24, 9-29
navigating, 5-4 to 5-10
navigator pane, 5-5
Net Service Name Wizard, 9-7, 9-14, 13-5
net service names, 9-7
Oracle Names
 Attempts Per Names Server field, 9-37
 database service names, 9-43, 16-3
 Default Domain field, 9-37
 discovering, 9-35
 global database link qualifiers, 9-47
 global database links, 9-46
 Initial Preallocated Requests field, 9-37
 Maximum Open Connections field, 9-37
 Maximum Wait Each Attempt field, 9-37
 net service names, 9-44
 non-default listener, 12-19
 preferred Oracle Names servers, 11-11
 querying network objects, 16-24
 region database, 9-24
Oracle Rdb Database, 15-45
Oracle RDB Database field, 15-8
Oracle RDB databases
 Global Database Name field, 15-44
 RDB Database field, 15-44
 Type of Service field, 15-44
profiles, 11-13
 advanced options, 11-9
 Always Use Dedicated Server option, 11-7
 Cell Name field, 11-12
 Check TCP/IP Client Access Rights
 option, 11-8
 Client Registration ID field, 11-9
 Clients Allowed to Access field, 11-8
 Clients Excluded from Access field, 11-8
 Disable Out-of-Band Break option, 11-10
 Meta Map field, 11-12
 Prefer Connection Manager Routing
 option, 11-7
 preferred Oracle Names servers, 11-11
 TNS Time Out Value option, 11-9
 Turn Off UNIX Signal Handling
 option, 11-10
right pane, 5-6
routing connection requests, 11-7

Session Data Unit (SDU) field in Oracle Net Manager, 15-8
specifying naming methods, 11-5
starting, 5-3
starting an Oracle Names server, 9-33
starting Oracle Names servers, 16-24
testing
 client configuration, 16-22
 server configuration, 16-16
toolbar, 5-7
Type of Service field, 15-8
Use for Heterogeneous Services option, 15-8, 15-43
Use Oracle8i Release 8.0 Compatible Identification option, 15-8
VI protocol addresses
 Disc field, 15-31
 Host field, 15-31
 Protocol field, 15-31
wizards, 5-10 to 5-13
Oracle Net Services
components
 listener, 1-29
 Oracle Connection Manager, 1-30
 Oracle Net, 1-27
 described, 1-27
Oracle protocols
described, 4-5
LU6.2, 4-6
Named Pipes, 4-6
TCP/IP with SSL, 4-6
Oracle Rdb database
configuring for connection to, 15-44
described, 15-44
Oracle RDB Database field in Oracle Net Manager, 15-8
Oracle schema
creating, 8-8
described, 3-16
Oracle Support Services, contacting, 17-54
Oracle System Identifier, configuring on the listener, 12-10
ORACLE_BASE
explained, xxx
ORACLE_HOME
explained, xxx

ORACLE_HOME networking parameter, 12-10, 15-38, 15-41
Oracle9i JVM
 connections, 4-11, 12-8
 listener configuration, 12-8
Oracle9i Real Application Clusters
 connect-time failover, 2-8, 15-4
 FAILOVER networking parameter, 15-4
 FAILOVER_MODE networking
 parameter, 15-21
 Transparent Application Failover (TAF), 15-18
OracleDBCreators group, 3-10, 3-16, 8-7, 8-8
OracleNetAdmins group, 3-10, 3-15, 8-7, 8-8, 8-9, 9-13, 9-18
orclDBServer object class, 3-16
orclNetAddress object class, 3-17
orclNetAddressList object class, 3-17
orclNetDescription object class, 3-17
orclNetDescriptionList object class, 3-17
orclNetService object class, 3-16
ou attribute, 3-7

P

packets, types of, 17-50
PASSWORD networking parameter, 10-19
PASSWORDS_listener_name parameter, 12-13
PING utility, 16-23
 testing Oracle Names servers with, 16-23
planning
 internal networks
 availability, 7-5
 connection pooling, 7-3
 connect-time failover, 7-5
 JDBC drivers, 7-6
 listener queue size, 7-7
 naming methods, 7-6
 protocol conversion, 7-7
 scalability, 7-3
 security, 7-6
 session data unit (SDU) size, 7-7
 session multiplexing, 7-3
 tuning and performance, 7-7
 VI protocol, 7-7

Internet networks
 access control, 7-9
 availability, 7-9
 JDBC drivers, 7-9
 naming methods, 7-9
 scalability, 7-9
 security, 7-9
 tuning and performance, 7-10
 VI protocol, 7-10
 session data unit (SDU) size, 7-13
PMON process, 12-2
POOL (POO) attribute, 14-5
port 1521
 if not using, 12-15
port 1575, 3-33, 9-29
port 1630, 4-18, 13-15
PORT networking parameter, 10-19
Prefer Connection Manager Routing option in
 Oracle Net Manager, 11-7
preferred Oracle Names servers, 3-30
prespawned dedicated servers, xxxvii
primary and secondary instances, 15-25
private database links, 3-24
profiles (sqlnet.ora)
 configuring
 advanced options, 11-9
 default domains, 11-3
 configuring preferred Oracle Names
 servers, 11-11
 default configuration, 11-2
 naming methods, specifying, 11-5
 prioritizing naming methods, 11-5
 routing connection requests, 11-7
Program Name field in Oracle Net Manager, 15-38, 15-41
PROGRAM networking parameter, 15-38, 15-41
PROTOCOL (PRO or PROT) attribute, 13-13, 14-2
protocol address, 2-6
protocol conversion support
 configuring, 13-15
protocol.ora file, xxxvii
protocols
 LU6.2, 4-6
 Named Pipes, 4-6
 TCP/IP with SSL, 4-6

proxy server, 13-2
public database links, 3-24

Q

QUERY command, 9-33, 16-24
A.SMD Oracle Names record, 12-19
DLCR.TDBMS.OMD Oracle Names record, 16-24
NS.SMD Oracle Names record, 16-24
V1ADD.NPO.OMD Oracle Names record, 16-24
queue size, 7-7, 7-12, 12-9
QUEUESIZE parameter, 12-9
for adjusting listener queue size, 7-7, 7-12, 12-9

R

randomizing requests among listeners, 2-8
RDB Database field, 15-44
RDB_DATABASE networking parameter, 15-8, 15-44
redirect connection, 2-10
region database
configure database server, 9-22, 9-28
create Oracle Names server, 9-24
defined, 3-22
regions
delegated administrative, 3-28
delegated administrative below root, 3-30
delegated administrative, diagram, 3-29
in Oracle Names, 3-21, 3-27
organizing multiple administrative network, 3-27
root administrative, 3-28
REGISTER command, 9-43, 9-44
registering
aliases with Oracle Names, 9-49
database service names with Oracle Names, 9-43
global database link qualifiers with Oracle Names, 9-46
global database links with Oracle Names, 9-46
net service names with Oracle Names, 9-44
objects with Oracle Names, 9-42
relative distinguished name (RDN), 3-6

relative naming
directory naming, 3-11
Java Database Connectivity (JDBC)
OCI drivers, 3-11
Thin drivers, 3-11
REMOTE_LISTENER initialization
parameter, 12-16
REORDER_NS command, 9-33, 9-35, 11-11
replicated data in Oracle Names, 3-22
replication of data in Oracle Names, 9-29
resolving
errors. See troubleshooting
RETRIES networking parameter, 15-21
root administrative regions
described, 3-28
domain requirements, 3-28
requirements, 3-28
root administrative regions, described, 3-28
root domains, described, 3-28
routing connections, 11-7

S

SAVE_CONFIG command, 12-13
scalability, of networks, 7-3
SCOPE networking parameter, 10-19
.sdns.ora file, 3-33
sdns.ora file, 3-33
SDU networking parameter, 15-8, 15-28
SDU. See session data unit (SDU)
SecurID authentication, xxvi
security
internal networks, 7-6
Internet networks, 7-9
listeners, 12-13
Select the directory server you want to use option in Oracle Net Configuration Assistant, 8-6
Select the directory server you want to use, and configure the directory server for Oracle usage option in Oracle Net Configuration Assistant, 8-7
server configuration
allocating resources with shared server, 14-9
discovering Oracle Names servers, 9-34
external procedures, 15-32, 15-40

local naming, 9-6
log files, 17-25
region database for Oracle Names, 9-22
shared server, 14-2
trace files, 17-41
server connections
 methods, 16-13
 syntax, 16-9
SERVER networking parameter, 2-7
server testing, 16-16
server troubleshooting, 17-3
servers
 loopback tests, 17-4
 starting, 6-4, 16-6
SERVICE (SER or SERV) attribute, 14-5
service handlers
 dedicated servers, 2-12 to 2-14
 dispatchers, 2-10 to 2-11
service name
 described, 2-2, 2-3, 9-2
service registration
 benefits, 12-3
 configuring, 12-3
 connection load balancing, 2-9, 12-3, 15-10
 connect-time failover, 12-3
 defined, 2-6
 events in listener.log, 17-32
 service_died listener log event, 17-32
 service_register listener log event, 17-32
 service_update listener log event, 17-32
service_died listener log event, 17-32
SERVICE_NAME networking parameter, 9-3
SERVICE_NAMES initialization parameter, 2-2, 9-3, 14-5
service_register listener log event, 17-32
service_update listener log event, 17-32
SERVICES command, 12-23
 of Listener Control utility, 6-5
session data unit (SDU), 7-13
 adjusting to improve network performance, 7-13
 configuring, 15-28
Session Data Unit (SDU) Size field in Oracle Net Manager, 15-8
session multiplexing, 1-16, 13-13
 advantages relative to connection pooling, 7-4
 overview, 1-17
SESSIONS (SES or SESS) attribute, 14-5
SET PASSWORD command
 of Listener Control utility, 16-5
shared server
 allocating resources, 14-9
 architecture, 1-13
 compared with dedicated server, 1-12
 configuring dispatchers, 14-6
 connect descriptor configuration parameters, 2-7
 connection load balancing, 2-9, 15-10
 defined, 1-12
 described, 4-14
 dispatchers, 1-12, 2-10 to 2-11, 4-14
 initialization parameters, 14-2
 starting, 14-2
 using with connection pooling, 14-8
 using with Oracle Connection Manager, 13-2
 virtual circuits, 4-14
SID field in Oracle Net Manager, 12-10, 15-38, 15-41
SID, configuring on the listener, 12-10
SID_LIST_*listener_name* parameter
 external procedures, 15-38
Heterogeneous Services, 15-41
Oracle Enterprise Manager
 listener.ora requirements, 12-10
SID_NAME networking parameter, 12-10, 15-38, 15-41
simple authentication for directory naming, 3-14
single domain naming model, 3-25
single sign-on
 Oracle Advanced Security, 1-24
SNA, 4-7
SOURCE_ROUTE networking parameter, 13-5, 13-10, 15-4, 15-6
SPX protocol, xxxviii
SQLNET.CLIENT_REGISTRATION networking parameter, 11-9
SQLNET.EXPIRE_TIME networking parameter, 11-9
sqlnet.log file, 17-24

sqlnet.ora file
described, 3-3
parameters
LOG_DIRECTORY_CLIENT, 17-25
LOG_DIRECTORY_SERVER, 17-25
LOG_FILE_CLIENT, 17-25
LOG_FILE_SERVER, 17-25
NAMES.DCE.PREFIX, 11-12
NAMES.DEFAULT_DOMAIN, 11-3
NAMES DIRECTORY PATH, 11-5
NAMES.NIS.META_MAP, 11-12
NAMES.PREFERRED_SERVERS, 11-11
TCP.EXCLUDED_NODES, 11-8
TCP.INVITED_NODES, 11-8
TCP.VALIDNODE_CHECKING, 11-8
TNSPING TRACE_LEVEL, 17-43
TRACE_DIRECTORY_CLIENT, 17-41
TRACE_DIRECTORY_SERVER, 17-42
TRACE_FILE_CLIENT, 17-41
TRACE_FILE_SERVER, 17-42
TRACE_LEVEL_CLIENT, 17-41
TRACE_UNIQUE_CLIENT, 17-42
USE_CMAN, 11-7, 13-5
USE_DEDICATED_SERVER, 11-7
parameters TNSPING TRACE_DIRECTORY, 17-43
sqlnet.trc file, 17-40
SSL authentication for directory naming, 3-14
stack communications
OCI, 4-4
Oracle Call Interface, 4-4
Oracle protocols, 4-5
START CMAN command, 16-7
START command
of Listener Control utility, 6-4, 12-20, 16-5
of Oracle Connection Manager Control utility, 16-7
of Oracle Names Control utility, 9-28, 9-33, 10-16, 10-22, 16-3
START_CLIENT_CACHE command, 9-39, 16-4
starting
an instance with dispatchers, 14-6
database server, 6-4
database servers, 16-6
databases, 6-4, 16-6

listeners, 6-4, 16-5, 16-6
Oracle Connection Manager, 16-7
Oracle Names client cache, 9-38, 16-4
Oracle Names LDAP Proxy servers, 9-54, 10-18
Oracle Names servers, 9-28, 16-3
Oracle Net Configuration Assistant, 5-14
Oracle Net Manager, 5-3
Oracle Net Services components, 16-2
STARTUP_WAITTIME *listener_name* networking parameter, 17-44
STATUS command
of Listener Control utility, 12-20, 16-5
STOP command
of Listener Control utility, 12-20, 16-5
of Oracle Names Control utility, 10-18
strong authentication for directory naming, 3-14
SUBTREE networking parameter, 10-19
SUBTREE_LIST networking parameter, 10-19
svr_pid.trc file, 17-40
syntax
for connect identifiers, 16-9
for Listener Control utility, 5-16
for Oracle Connection Manager Control utility, 5-17
for Oracle Names Control utility, 5-18
System Network Architecture (SNA), 4-7

T

TAF. See **Transparent Application Failover (TAF)**
TCP.EXCLUDED_NODES networking
parameter, 11-8
TCP.INVITED_NODES networking
parameter, 11-8
TCP/IP protocol
performance, 1-18
TCP/IP with SSL protocol
described, 4-6
TCP.VALIDNODE_CHECKING networking
parameter, 11-8

testing
client configuration
 using applications, 16-13
 using operating system, 16-13
 using special commands, 16-14
 with 3GL, 16-14
 with Oracle Net Manager, 16-22
 with TCROUTE, 16-20
 with TNSPING, 16-17
listener configuration, 16-16
network connectivity, 16-17
network objects, 16-24
Oracle Connection Manager, 16-16
Oracle Names servers, 16-23
server configuration, 16-16
 with control utilities, 5-16
TICKS (TIC or TICK) attribute, 14-6
TIMEOUT networking parameter, 10-19
TNS. See Transparent Network Substrate (TNS)
TNS Time Out Value option in Oracle Net Manager, 11-9
TNS_ADMIN environment variable, 17-4, 17-5
TNS_ADMIN registry entry, 17-4, 17-5
tnsnames.ora file
 configuring, 9-5
 described, 3-3
 exporting entries to directory server, 10-2
 migrating entries to directory server, 10-2
parameters
 BACKUP parameter, 15-21
 DELAY, 15-21
 FAILOVER, 15-4
 FAILOVER_MODE, 15-21
 GLOBAL_NAME, 15-8
 HS, 15-8
 INSTANCE_NAME, 15-8, 15-25
 LOAD_BALANCE, 15-4
 METHOD, 15-21
 RDB_DATABASE, 15-8
 RETRIES, 15-21
 SDU, 15-8, 15-28
 SOURCE_ROUTE, 13-5, 13-10, 15-4, 15-6
 TYPE, 15-21
 TYPE_OF_SERVICE, 15-8
TNSPING utility, 16-17
compared to TRCROUTE utility, 16-20
TNSPING TRACE_DIRECTORY networking
 parameter, 17-43
TNSPING TRACE_LEVEL networking
 parameter, 17-43
Trace Assistant, xxxviii
trace files
 cmadm_pid.trc, 17-40
 cmadmpid.trc, 17-40
 cman_pid.trc, 17-40
 cmanpid.trc, 17-40
 default names for, 17-40
 error message information, 17-51
 listener.trc, 17-40, 17-44
 namespid.trc, 17-45
 names.trc, 17-40, 17-45
 sqlnet.trc, 17-40
 sqlnet.trc for clients, 17-41
 svr_pid.trc, 17-40
 svr_pid.trc for servers, 17-41
TRACE_DIRECTORY networking
 parameter, 17-46
TRACE_DIRECTORY_CLIENT networking
 parameter, 17-41
TRACE_DIRECTORY_SERVER networking
 parameter, 17-42
TRACE_FILE_CLIENT networking
 parameter, 17-41
TRACE_FILE_listener_name networking
 parameter, 17-44
TRACE_FILE_SERVER networking
 parameter, 17-42
TRACE_LEVEL networking parameter, 17-46
TRACE_LEVEL_CLIENT networking
 parameter, 17-41
TRACE_LEVEL_listener_name networking
 parameter, 17-44
TRACE_LEVEL_SERVER networking
 parameter, 17-42

T

Transparent Application Failover (TAF)
configuring, 15-18
GLOBAL_DBNAME networking parameter in
 listener.ora, 12-10, 15-22
overview, 2-8
with instance role, 15-27

Transparent Network Substrate (TNS)
 benefits, 4-4
 described, 4-4

TRCROUTE utility
 described, 16-20
 performance effects, 16-20

troubleshooting, 17-1 to 17-55
 client, 17-4
 contacting Oracle Support Services, 17-54
 log files, 17-20
 loopback tests, 17-4
 questions, 17-19
 server, 17-3
 trace files, 17-20
 using log file, 17-29

Turn Off UNIX Signal Handling option in Oracle
 Net Manager, 11-10

TWO_TASK environment variable, 16-8

Two-Task Common (TTC), 4-4

TYPE networking parameter, 10-19, 10-20, 15-21

Type of Service field in Oracle Net Manager, 15-8,
 15-44

TYPE_OF_SERVICE networking parameter, 15-8,
 15-44

U

UNREGISTER command, 9-54

Use for Heterogeneous Services option in Oracle Net
 Manager, 15-8, 15-43

Use Options Compatible with Net8 8.0 Clients
 option, 15-6

Use Oracle8i Release 8.0 Compatible Identification
 option, 15-8

USE_CMAN networking parameter, 11-7, 13-5

USE_DEDICATED_SERVER networking
 parameter, 11-7

USERID networking parameter, 10-19

V

V\$SESSION table, 15-25

V1ADD.NPO.OMD Oracle Names record, 16-24

VI protocol
 and Oracle Connection Manager, 15-35
 configuring, 15-31 to 15-34

virtual circuits, 4-14

W

Welcome page in Oracle Net Configuration
 Assistant, 5-14

well-known Oracle Names servers, 3-31

wizards
 Directory Server Migration, 5-12
 Names, 5-13
 Net Service Name, 5-11
 Oracle Net Manager, 5-10 to 5-13

Y

ypserv program, 9-60