Oracle® Database

Net Services Reference

10*g* Release 2 (10.2)

B14213-01

June 2005

ORACLE®

Oracle Database Net Services Reference, 10*g* Release 2 (10.2)

B14213-01

Primary Author: Jennifer Polk

Contributors: Robert Achacoso, Gaurav Agarwal, Omar Bellal, David Chen, Howard Cohen, Toby Close, Pramodini Gattu, Harvey Eneman, Stella Kister, Shuvayu Kanjila, Vincent Lee, Catherine Luu, Scot McKinley, Ed Miner, Peter Povinec, Jose Wong

Contents

2 Oracle Connection Manager Control Utility

6 Local Naming Parameters (tnsnames.ora)

7 Listener Parameters (listener.ora)

8 Oracle Connection Manager Parameters (cman.ora)

C LDAP Schema for Oracle Net Services

Glossary

Index

Send Us Your Comments

Oracle Database Net Services Reference 10*g* Release 2 (10.2)

B14213-01

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

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

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation 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: Server Technologies Documentation Manager
- Postal service:

  Oracle Corporation
  Server Technologies Documentation
  500 Oracle Parkway, Mailstop 4op11
  Redwood Shores, CA 94065
  USA

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

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

Preface

The *Oracle Database Net Services Reference* contains a complete listing and description of the control utility commands and configuration file parameters available for managing components of Oracle Net Services.

This document describes the features of Oracle Database 10*g* software that apply to the Microsoft Windows and UNIX operating systems.

This preface contains these topics:

Audience

Oracle Database Net Services Reference is intended for network administrators who are responsible for configuring and administering network components.

To use this document, you need to be familiar with the networking concepts and configuration tasks described in *Oracle Database Net Services Administrator's Guide*.

Documentation Accessibility

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

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an

otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Organization

This document contains:

Part I, "Control Utilities"

Chapter 1, "Listener Control Utility"

This chapter describes the **Listener Control utility** commands.

Chapter 2, "Oracle Connection Manager Control Utility"

This chapter describes the **Oracle Connection Manager Control utility** commands.

Part II, "Configuration Parameters"

Chapter 3, "Syntax Rules for Configuration Files"

This chapter describes the syntax rules for networking configuration files.

Chapter 4, "Protocol Address Configuration"

This chapter describes how to configure a **protocol address**.

Chapter 5, "Profile Parameters (sqlnet.ora)"

This chapter describes the `sqlnet.ora` file parameters.

Chapter 6, "Local Naming Parameters (tnsnames.ora)"

This chapter describes the `tnsnames.ora` file parameters.

Chapter 7, "Listener Parameters (listener.ora)"

This chapter describes the `listener.ora` file parameters.

Chapter 8, "Oracle Connection Manager Parameters (cman.ora)"

This chapter describes the `cman.ora` file parameters.

Chapter 9, "Directory Usage Parameters (ldap.ora)"

This chapter describes the `ldap.ora` file parameters.

Part III, "Appendixes"

Appendix A, "Features Not Supported in this Release"
This appendix describes the control utility commands and parameters no longer supported by Oracle Net Services.

Appendix B, "Upgrade Considerations for Oracle Net Services"
This appendix describes coexistence and upgrade issues for Oracle Net Services.

Appendix C, "LDAP Schema for Oracle Net Services"
This appendix describes the **Oracle schema** for Oracle Net Services.

Glossary

Related Documentation

For more information, see these Oracle resources:

- *Oracle Database Net Services Administrator's Guide*
- Online Help for Net Services tools and utilities
- Oracle Database documentation set

A glossary of Net Services terms is available in the Oracle Net Services Administrator's Guide.

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 *Oracle Database Sample Schemas* for information on how these schemas were created and how you can use them yourself.

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

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://otn.oracle.com/membership/`

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://www.oracle.com/technology/documentation/`

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**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle Database Concepts* Ensure that the recovery catalog and target database do *not* 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.
`lowercase monospace (fixed-width) font`	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. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter `sqlplus` to open SQL*Plus. The password is specified in the `orapwd` file. Back up the datafiles and control files in the `/disk1/oracle/dbs` directory. The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table. Set the `QUERY_REWRITE_ENABLED` initialization parameter to `true`. Connect as `oe` user. The `JRepUtil` class implements these methods.
`lowercase italic monospace (fixed-width) font`	Lowercase italic monospace font represents placeholders or variables.	You can specify the *`parallel_clause`*. Run `U`*`old_release`*`.SQL` where *`old_release`* 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
`[ ]`	Brackets enclose one or more optional items. Do not enter the brackets.	`DECIMAL (`*`digits`*` [ , `*`precision`*` ])`
`{ }`	Braces enclose two or more items, one of which is required. Do not enter the braces.	`{ENABLE | DISABLE}`

Convention	Meaning	Example
|	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.	{ENABLE | DISABLE} [COMPRESS | NOCOMPRESS]
...	Horizontal ellipsis points indicate either: ■ 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	CREATE TABLE ... AS *subquery*; SELECT *col1*, *col2*, ... , *coln* FROM employees;
. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V$DATAFILE; NAME ---------- /fsl/dbs/tbs_01.dbf /fs1/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/*system_password* DB_NAME = *database_name*
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.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
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.	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;

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.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.

Convention	Meaning	Example
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (|), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	`c:\winnt"\"system32 is the same as C:\WINNT\SYSTEM32`
`C:\>`	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual.	`C:\oracle\oradata>`
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	`C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\"` `C:\>imp SYSTEM/`*`password`*` FROMUSER=scott TABLES=(emp, dept)`
HOME_NAME	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`*`HOME_NAME`*`TNSListener`

Convention	Meaning	Example
ORACLE_HOME and *ORACLE_BASE*	In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory. For Windows NT, the default location was `C:\orant`. This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle`. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\ora`*nn*, where *nn* is the latest release number. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. Refer to Oracle Database Platform Guide for Microsoft Windows (32-Bit) for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE*`\`*ORACLE_HOME*`\rdbms\admin` directory.

Part I

Control Utilities

Oracle Net Services provides control utilities to administer listeners, Oracle Names servers, and Oracle Connection Managers. Part 1 lists the commands that are available with each utility, including any applicable prerequisites, passwords, syntax or argument rules, and usage notes or examples to help you use them.

This part contains the following chapters:

1

Listener Control Utility

This chapter describes the commands and associated syntax of the **Listener Control utility**.

This chapter contains these topics:

Listener Control Utility Overview

The **Listener Control utility** enables you to administer **listener**s.You can use its commands to perform basic management functions on one or more listeners. Additionally, you can view and change parameter settings.

The basic syntax of Listener Control utility commands is as follows:

```
lsnrctl command [listener_name]
```

where *listener_name* is the name of the listener to be administered. If no name is specified, then the default name, LISTENER, is assumed.

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. The basic syntax of issuing commands from LSNRCTL> program prompt is as follows:

```
lsnrctl
LSNRCTL> command [listener_name]
```

> **Note:** You can combine commands in a standard text file, and then run them as a sequence of commands. To execute in batch mode, use the format:
>
> `lsnrctl @file_name`
>
> You can use either `REM` or `#` to identify comments in the batch script; all other lines are considered commands. Any commands that would typically require confirmation do not require confirmation during batch execution.

For a majority of commands, the Listener Control utility establishes an Oracle Net connection with the listener that is used to transmit the command. To initiate an Oracle Net connection to the listener, the Listener Control utility needs to obtain the **protocol addresses** for the named listener or a listener named `LISTENER`. This is done by resolving the listener name with one of the following mechanisms:

- `listener.ora` file in the directory specified by the `TNS_ADMIN` environment variable
- `listener.ora` file in the `$ORACLE_HOME/network/admin` directory on UNIX operating systems and the *`%ORACLE_HOME%`*`\network\admin` directory on Windows operating systems
- Naming method, for example, a `tnsnames.ora` file

If the listener name is `LISTENER` and it cannot be resolved, a protocol address of TCP/IP, port 1521 is assumed.

The Listener Control utility supports several types of commands:

- Operational commands, such as START, STOP, and so forth.
- Modifier commands, such as SET TRC_LEVEL
- Informational commands, such as STATUS and `SHOW LOG_FILE`
- Operational commands, such as EXIT, RELOAD, and HELP

SET and SHOW Commands of the Listener Control Utility

You can use the SET command to alter parameter values for a specified listener. You set the name of the listener you want to administer with the SET CURRENT_LISTENER command. Parameter values remain in effect until the listener is shut down. If you want these settings to persist, use the SAVE_CONFIG command to save changes to the `listener.ora`.

You can use the SHOW command to display the current value of a configuration setting.

Distributed Operations

The Listener Control utility can perform operations on a local or a remote listener.

To set up a computer to remotely administer a listener:

1. Ensure that the Listener Control utility (`lsnrctl`) executable is installed.

2. Ensure that the name of the listener you want to administer can be resolved through a `listener.ora` file or a naming method, as described in "Listener Control Utility Overview" on page 1-1.

All commands except START can be issued when a listener is administered remotely. The Listener Control utility can only start the listener on the same computer from where the utility is running.

When issuing commands, specify the listener name as an argument. For example:

```
LSNRCTL> SERVICES lsnr
```

If the name is omitted, then listener name set with the SET CURRENT_LISTENER command is used, or the default name, `LISTENER` is assumed.

Listener Security

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 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.

If the `PASSWORDS_`*`listener_name`* parameter is set in the `listener.ora` file or the `CHANGE_PASSWORD` command has been used to create a new, encrypted password, then the Listener Control utility will require a SET PASSWORD command prior to any protected command, such as STOP.

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 uses 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.

Listener Control Utility Commands

This section lists and describes the Listener Control utility commands.

CHANGE_PASSWORD

Purpose

Use the `CHANGE_PASSWORD` command to establish an encrypted password or change an encrypted password set with the `PASSWORDS_`*`listener_name`* parameter in the `listener.ora` file.

Prerequisites

None

Password Required If One Has Been Set:

Yes. If a password is set, then issue then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl CHANGE_PASSWORD [listener_name]
```

From the Listener Control utility:

```
LSNRCTL> CHANGE_PASSWORD [listener_name]
```

Arguments

[*listener_name*]: Specify the listener name, if the default name of LISTENER is not used.

Usage Notes

The Listener Control utility prompts you for the old password and then for the new one. It asks you to re-enter the new one, and then changes it. Neither the old nor the new password displays during this procedure. CHANGE_PASSWORD is usually followed by the SAVE_CONFIG command to save the new password in the listener.ora file. If a SAVE_CONFIG command is not issued, then the new password will be in effect only until the listener is shut down.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for further information about password security of the listener

Example

The following shows a new password of takd01 being set:

```
LSNRCTL> CHANGE_PASSWORD
Old password:
New password: takd01
Reenter new password: takd01
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=tpc)(HOST=sales-server)(PORT=1521)))
Password changed for LISTENER
The command completed successfully
LSNRCTL> SAVE_CONFIG
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521)))
Saved LISTENER configuration parameters.
Listener Parameter File   /oracle/network/admin/listener.ora
Old Parameter File   /oracle/network/admin/listener.bak
The command completed successfully
```

The following shows the password being changed from takd01 to smd01:

```
LSNRCTL> SET PASSWORD
Password: takd01
The command completed successfully
LSNRCTL> CHANGE_PASSWORD
Old password: takd01
New password: smd01
Reenter new password: smd01
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=tpc)(HOST=sales-server)(PORT=1521)))
Password changed for LISTENER
The command completed successfully
LSNRCTL> SAVE_CONFIG
```

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521)))
Saved LISTENER configuration parameters.
Listener Parameter File   /oracle/network/admin/listener.ora
Old Parameter File   /oracle/network/admin/listener.bak
The command completed successfully
```

EXIT

Purpose

Use the `EXIT` command to exit from the Listener Control utility.

Prerequisites

None

Password Required If One Has Been Set

No. If a password is set, then the SET PASSWORD command does not need to be issued prior to this command.

Syntax

From the Listener Control utility:

```
LSNRCTL> EXIT
```

Arguments

None

Usage Notes

This command is identical to the RELOAD command.

Example

```
LSNRCTL> EXIT
```

HELP

Purpose

Use the command `HELP` to provide a list of all the Listener Control utility commands or provide syntax help for a particular Listener Control utility command.

Prerequisites

None

Password Required If One Has Been Set

No. If a password is set, then the SET PASSWORD command does not need to be issued prior to this command.

Syntax

From the operating system:

```
lsnrctl HELP [command]
```

From the Listener Control utility:

```
LSNRCTL> HELP [command]
```

Arguments

[*command*]: Specify a HELP command. Commands are shown in the following example output.

When you enter a command as an argument to HELP, the Listener Control utility displays information about how to use the command. When you enter HELP without an argument, the Listener Control utility displays a list of all the commands.

Example

```
LSNRCTL> HELP
The following operations are available
An asterisk (*) denotes a modifier or extended command:
change_password
exit
quit
reload
services
set*
show*
spawn
start
status
stop
trace
version
```

QUIT

Purpose

Use the QUIT command to exit the Listener Control utility and return to the operating system prompt.

Prerequisites

None

Password Required If One Has Been Set

No. If a password is set, then the SET PASSWORD command does not need to be issued prior to this command.

Syntax

From the Listener Control utility

```
LSNRCTL> QUIT
```

Arguments

None

Usage Notes

This command is identical to the EXIT command.

Example

```
LSNRCTL> QUIT
```

RELOAD

Purpose

Use the RELOAD command to reread the listener.ora file. This command enables you to add or change statically configured services without actually stopping the listener.

In addition, the database services, instances, service handlers, and listening endpoints that were dynamically registered with the listener will be unregistered and subsequently registered again.

Prerequisites

None

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl RELOAD [listener_name]
```

From the Listener Control utility:

```
LSNRCTL> RELOAD [listener_name]
```

Arguments

[*listener_name*]: Specify the listener name, if the default name of LISTENER is not used.

Example

```
LSNRCTL> RELOAD
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521)))
The command completed successfully
```

SAVE_CONFIG

Purpose

Use the SAVE_CONFIG command to compare the current configuration state of the listener, including trace level, trace file, trace directory, and logging to the listener.ora file. Any changes are stored in listener.ora, preserving formatting, comments, and case as much as possible. Prior to modification of the listener.ora file, a backup of the file, called listener.bak, is created.

Password Required If One Has Been Set

Yes. If a password is set, then issue then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

From the operating system:

```
lsnrctl SAVE_CONFIG [listener_name]
```

From the Listener Control utility:

```
LSNRCTL> SAVE_CONFIG [listener_name]
```

Arguments

[*listener_name*]: Specify the listener name, if the default name of LISTENER is not used.

Usage Notes

This command enables you to save all runtime configuration changes to the listener.ora file, which can be especially useful for saving changed encrypted passwords.

Example

```
LSNRCTL> SAVE_CONFIG listener
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521)))
Saved LISTENER configuration parameters.
Listener Parameter File   /oracle/network/admin/listener.ora
Old Parameter File   /oracle/network/admin/listener.bak
The command completed successfully
```

SERVICES

Purpose

Use the SERVICES command to obtain detailed information about the database services, instances, and service handlers (dispatchers and dedicated servers) to which the listener forwards client connection requests.

Prerequisites

None

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl SERVICES [listener_name]
```

From the Listener Control utility:

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

Arguments

[*listener_name*]: Specify the listener name, if the default name of LISTENER is not used.

Usage Notes

See Also: *Oracle Database Net Services Administrator's Guide* for a complete description of SERVICES output

The SET DISPLAYMODE command changes the format and the detail level of the output.

Example

This example shows SERVICES output in the default display mode. The output shows the following:

- An instance named sales belonging to two services, sales1.us.acme.com and sales2.us.acme.com, with a total of three service handlers.
- Service sales1.us.acme.com is handled by one dispatcher only.
- Service sales2.us.acme.com is handled by one dispatcher and one dedicated server, as specified by in the following output.

```
LSNRCTL> SERVICES
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=net)))
Services Summary...
Service "sales1.us.acme.com" has 1 instance(s).
  Instance "sales", status READY, has 1 handler(s) for this service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:10000 state:ready
         DISPATCHER <machine: sales-server, pid: 5696>
         (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=53411))
Service "sales2.us.acme.com" has 1 instance(s).
  Instance "sales", status READY, has 2 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
         LOCAL SERVER
      "D001" established:0 refused:0 current:0 max:10000 state:ready
         DISPATCHER <machine: sales-server, pid: 5698>
         (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=52618))
The command completed successfully
```

SET

Purpose

Use the SET command to alter the parameter values for the listener. Parameter values changes remain in effect until the listener is shut down. To make the changes permanent, use the SAVE_CONFIG command to save changes to the listener.ora file.

Prerequisites

None

Password Required If One Has Been Set

No. If a password is set, then the SET PASSWORD command does not need to be issued prior to this command.

Syntax

From the operating system:

```
lsnrctl SET [parameter]
```

From the Listener Control utility:

```
LSNRCTL> SET [parameter]
```

Arguments

[*parameter*]: Specify a SET parameter to modify its configuration setting. Parameters are shown in the example output.

When you enter SET without an argument, the Listener Control utility displays a list of all the parameters.

Usage Notes

If you are using the SET commands to alter the configuration of a listener other than the default LISTENER listener, use the SET CURRENT_LISTENER command to set the name of the listener you want to administer.

Example

```
LSNRCTL> SET
The following operations are available with set.
An asterick (*) denotes a modifier or extended command.
current_listener
displaymode
inbound_connect_timeout
log_file
log_directory
log_status
password
raw_mode
save_config_on_stop
startup_waittime
trc_file
trc_directory
trc_level
```

SET CURRENT_LISTENER

Purpose

Use the SET CURRENT_LISTENER command to set the name of the listener to administer. Subsequent commands that would normally require *listener_name* can be issued without it.

Password Required If One Has Been Set

No. If a password is set, then the SET PASSWORD command does not need to be issued prior to this command.

Syntax

From the Listener Control utility

```
LSNRCTL> SET CURRENT_LISTENER [listener_name]
```

Arguments

[*listener_name*]: Specify the listener name, if the default name of LISTENER is not used.

Usage Notes

When SET CURRENT_LISTENER is set, the Listener Control utility commands act on the listener you set. You do not have to specify the name of the listener.

Example

```
LSNRCTL> SET CURRENT_LISTENER lsnr
Current Listener is lsnr
```

SET DISPLAYMODE

Purpose

Use the SET DISPLAYMODE command to change the format and level of detail for the SERVICES and STATUS commands.

Password Required If One Has Been Set

No. If a password is set, then the SET PASSWORD command does not need to be issued prior to this command.

Syntax

From the Listener Control utility:

```
LSNRCTL> SET DISPLAYMODE {compat | normal | verbose | raw}
```

Arguments

[compat]: Specify to display output that is compatible with older versions of the listener.

[normal]: Specify to display output in a formatted and descriptive output. Oracle Corporation recommends this mode.

[verbose]: Specify to display all data received from the listener in a formatted and descriptive output.

[raw]: Specify to display all data received from the listener without any formatting. This output should be used only if recommended by Oracle Support Services.

Example

```
LSNRCTL> SET DISPLAYMODE normal
Service display mode is NORMAL
```

SET INBOUND_CONNECT_TIMEOUT

Purpose

Use the SET INBUND_CONNECT_TIMEOUT command to specify the time, in seconds, for the client to complete its connect request to the listener after the network connection had been established.

If the listener does not receive the client request in the time specified, then it terminates the connection. In addition, the listener logs the IP address of the client and an ORA-12525:TNS: listener has not received client's request in time allowed error message to the listener.log file.

> **See Also:** *Oracle Database Net Services Administrator's Guide*for information about specifying the time-out for client connections

Syntax

From the Listener Control utility:

```
LSNRCTL> SET INCOUND_CONNECT_TIMEOUT
```

Arguments

{*time*}: Specify the time, in seconds. Default setting is 60 seconds.

Example

```
LSNRCTL> SET INBOUND_CONNECT_TIMEOUT 2
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
LISTENER parameter "inbound_connect_timeout" set to 2
The command completed successfully.
```

SET LOG_DIRECTORY

Purpose

Use the command SET LOG_DIRECTORY to set destination directory where the listener log file is written. By default, the log file is written to the $ORACLE_HOME/network/log directory on UNIX operating systems and the *ORACLE_HOME*\network\log directory on Windows.

Prerequisites

None

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl SET LOG_DIRECTORY {directory}
```

From the Listener Control utility:

```
LSNRCTL> SET LOG_DIRECTORY {directory}
```

Arguments

{*directory*}: Specify the directory path of the listener log file.

Example

```
LSNRCTL> SET LOG_DIRECTORY /usr/oracle/admin
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
LISTENER parameter "log_directory" set to /usr/oracle/admin
The command completed successfully
```

SET LOG_FILE

Purpose

Use the command SET LOG_FILE to set the name for the listener log file. By default, the log file name is listener.log.

Prerequisites

None

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl SET LOG_FILE {file_name}
```

From the Listener Control utility:

```
LSNRCTL> SET LOG_FILE {file_name}
```

Arguments

{*file_name*}: Specify file name of the listener log.

Example

```
LSNRCTL> SET LOG_FILE list.log
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
LISTENER parameter "log_file" set to list.log
The command completed successfully
```

SET LOG_STATUS

Purpose

Use the command SET LOG_STATUS to turn listener logging on or off

Prerequisites

None

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl SET LOG_STATUS {on | off}
```

From the Listener Control utility:

```
LSNRCTL> SET LOG_STATUS {on | off}
```

Arguments

[on]: Specify to turn logging on.

[off]: Specify to turn logging off.

Example

```
LSNRCTL> SET LOG_STATUS on
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
LISTENER parameter "log_status" set to ON
```

```
The command completed successfully
```

SET PASSWORD

Purpose

Use the command SET PASSWORD prior to privileged Listener Control utility commands, such as SAVE_CONFIG and STOP.

The password entered should match the one established for the PASSWORDS_*listener_name* parameter in the listener.ora file or set by the CHANGE_PASSWORD command.

Syntax

From the Listener Control utility:

```
LSNRCTL> SET PASSWORD
Password: password
```

Arguments

None

Usage Notes

You can enter this command when you start up the Listener Control utility or at any time during the session as needed.

See Also: "Listener Security" on page 1-3

Example

```
LSNRCTL> SET PASSWORD
Password: lnrc10g
The command completed successfully
```

SET SAVE_CONFIG_ON_STOP

Purpose

Use the command SET SAVE_CONFIG_ON_STOP to specify whether or not changes made to the parameter values for the listener by the SET commands are to be saved to the listener.ora file at the time the listener is stopped with the STOP command.

When changes are saved, the Listener Control utility tries to preserve formatting, comments, and letter case. Prior to modification of the listener.ora file, a back up of the file, called listener.bak, is created.

To have all parameters saved right away, use the SAVE_CONFIG command.

Password Required If One Has Been Set

No. If a password is set, then the SET PASSWORD command does not need to be issued prior to this command.

Syntax

From the operating system:

```
lsnrctl SET SAVE_CONFIG_ON_STOP  {on | off}
```

From the Listener Control utility:

```
LSNRCTL> SET SAVE_CONFIG_ON_STOP  {on | off}
```

Arguments

`[on]`: Specify to save configuration to `listener.ora`.

`[off]`: Specify to not save configuration to `listener.ora`.

Example

```
LSNRCTL> SET SAVE_CONFIG_ON_STOP on
LISTENER parameter "save_config_on_stop" set to ON
The command completed successfully
```

SET STARTUP_WAITTIME

> **Note:** This command is deprecated in Oracle9*i* and will be desupported in a future release. If you require this command to run the listener, please notify Oracle Support Services.

Purpose

Use the command `SET STARTUP_WAITTIME` to specify the amount of time for the listener to wait before responding to a START command.

Prerequisites

None

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl SET STARTUP_WAITTIME {time}
```

From the Listener Control utility:

```
LSNRCTL> SET STARTUP_WAITTIME {time}
```

Arguments

{*time*}: Specify the time, in seconds.

Example

```
LSNRCTL> SET STARTUP_WAITTIME 10
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
LISTENER parameter "startup_waittime" set to 10
The command completed successfully
```

SET TRC_DIRECTORY

Purpose

Use the command SET TRC_DIRECTORY to set the destination directory where the listener trace files are written. By default, the trace file are written to the $ORACLE_HOME/network/trace directory on UNIX operating systems and the *ORACLE_HOME*\network\trace directory on Windows.

Prerequisites

None

Password Required If One Has Been Set

No. If a password is set, then the SET PASSWORD command does not need to be issued prior to this command.

Syntax

From the operating system:

```
lsnrctl SET TRC_DIRECTORY {directory}
```

From the Listener Control utility:

```
LSNRCTL> SET TRC_DIRECTORY {directory}
```

Arguments

{*directory*}: Specify the directory path of the listener trace files.

Example

```
LSNRCTL> SET TRC_DIRECTORY /usr/oracle/admin
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
LISTENER parameter "trc_directory" set to /usr/oracle/admin
The command completed successfully
```

SET TRC_FILE

Purpose

Use the command SET TRC_FILE to set the name of the listener trace file. By default, the trace file name is listener.trc.

Prerequisites

None

Password Required If One Has Been Set

No. If a password is set, then the SET PASSWORD command does not need to be issued prior to this command.

Syntax

From the operating system:

```
lsnrctl SET TRC_FILE {file_name}
```

From the Listener Control utility:

```
LSNRCTL> SET TRC_FILE {file_name}
```

Arguments

{*file_name*}: Specify the file name of the listener trace.

Example

```
LSNRCTL> SET TRC_FILE list.trc
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
LISTENER parameter "trc_file" set to list.trc
The command completed successfully
```

SET TRC_LEVEL

Purpose

Use the command SET TRC_LEVEL to set a specific level of tracing for the listener.

Prerequisites

None

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl SET TRC_LEVEL {level}
```

From the Listener Control utility:

```
LSNRCTL> SET TRC_LEVEL {level}
```

Arguments

{*level*}: Specify one of the following trace levels:

- off for no trace output
- user for user trace information
- admin for administration trace information
- support for Oracle Support Services trace information

Usage Notes

This command has the same functionality as the TRACE command.

Example

```
LSNRCTL> SET TRC_LEVEL admin
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
LISTENER parameter "trc_level" set to admin
The command completed successfully
```

SHOW

Purpose

Use the command SHOW to view the current parameter values for the listener.

All of the SET parameters, except SET PASSWORD, have equivalent SHOW parameters.

Prerequisites

None

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to using this command.

Syntax

From the operating system:

```
lsnrctl SHOW [parameter]
```

From the Listener Control utility:

```
LSNRCTL> SHOW [parameter]
```

Arguments

[*parameter*]: Specify a SHOW parameter to view its configuration settings. Parameters are shown in the example output.

When you enter SET without an argument, the Listener Control utility displays a list of all the parameters.

Example

```
LSNRCTL> SHOW
The following properties are available with SHOW:
An asterisk (*) denotes a modifier or extended command:
current_listener
displaymode
inbound_connect_timeout
log_file
log_directory
log_status
rawmode
save_config_on_stop
startup_waittime
trc_file
trc_directory
trc_level
```

Note: SHOW STARTUP_WAITTIME is deprecated in Oracle9*i* and will be desupported in a future release. If you require this command to run the listener, please notify Oracle Support Services.

SPAWN

Purpose

Use the command SPAWN to start a program stored on the computer on which the listener is running, and which is listed with an alias in the listener.ora file.

Prerequisites

None

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl SPAWN [listener_name] {alias} [(arguments='arg1,arg2,...')]
```

From the Listener Control utility

```
LSNRCTL> SPAWN [listener_name] {alias} [(arguments='arg1,arg2,...')]
```

Arguments

[*listener_name*]: Specify the listener name, if the default name of LISTENER is not used.

{alias}: Specify the alias name of the program specified by the PROGRAM parameter in the listener.ora file.

[(arguments='*arg1*,*arg2*,...')]: Specify the arguments sent to by program that is to be spawned.

Example

```
LSNRCTL> SPAWN nstest_alias (ARGUMENTS='')
```

START

Purpose

Use the command START to start the named listener.

Prerequisites

Listener must not already be running.

Password Required If One Has Been Set

No. If a password is set, then the SET PASSWORD command does not need to be issued prior to this command.

Syntax

From the operating system:

```
lsnrctl START [listener_name]
```

From the Listener Control utility:

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

Arguments

[*listener_name*]: Specify the listener name, if the default name of LISTENER is not used.

Usage Notes

To start a listener configured in the listener.ora file with a name other than LISTENER, include that name.

For example, if the listener name is `tcp_lsnr`, enter:

```
lsnrctl START tcp_lsnr
```

From the Listener Control utility:

```
LSNRCTL> START tcp_lsnr
```

Example

```
LSNRCTL> START

Starting /private/dsteiner/sales/bin/tnslsnr: please wait...

TNSLSNR for Solaris: Version 9.0.1.0.0
System parameter file is /oracle/network/admin/listener.ora
Log messages written to /oracle/network/log/listener.log
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Solaris: Version 9.0.1.0.0
Start Date                15-NOV-2003 18:02:25
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  OFF
SNMP                      OFF
Listener Parameter File   /oracle/network/admin/listener.ora
Listener Log File         /oracle/network/log/listener.log
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)))
The listener supports no services
The command completed successfully
```

STATUS

> **Note:** You can also obtain the status of the listener through the Oracle Enterprise Manager Console. See the *Oracle Enterprise Manager Administrator's Guide* for further information.

Purpose

Use the command `STATUS` to display basic status information about a listener, including a summary of listener configuration settings, listening protocol addresses, and a summary of services registered with the listener.

Prerequisites

None

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl STATUS [listener_name]
```

From the Listener Control utility:

```
LSNRCTL> STATUS [listener_name]
```

Arguments

[*listener_name*]: Specify the listener name, if the default name of LISTENER is not used.

Usage Notes

> **See Also:** *Oracle Database Net Services Administrator's Guide* or a complete description of STATUS output

The SET DISPLAYMODE command changes the format and level of the detail of the output.

Example

The following example shows STATUS output in the default display mode. The output contains:

- Listener configuration settings
- Listening endpoints summary
- Services summary, which is an abbreviated version of the SERVICES command output

```
LSNRCTL> STATUS
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=net)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Solaris: Version 10.0.0.0.0 -
Beta
Start Date                15-JAN-2003 12:02:00
Uptime                    0 days 0 hr. 5 min. 29 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)))

Services Summary...
Service "sales1.us.acme.com" has 1 instance(s).
  Instance "sales", status READY, has 1 handler(s) for this service...
Service "sales2.us.acme.com" has 1 instance(s).
  Instance "sales", status READY, has 2 handler(s) for this service...
The command completed successfully
```

STOP

Purpose

Use the command STOP to stop the named listener.

Prerequisites

The listener must be running.

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl STOP [listener_name]
```

From the Listener Control utility:

```
LSNRCTL> STOP [listener_name]
```

Arguments

[*listener_name*]: Specify the listener name, if the default name of LISTENER is not used.

Example

```
LSNRCTL> STOP
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
The command completed successfully
```

TRACE

Purpose

Use the command TRACE to turn on tracing for the listener.

Password Required If One Has Been Set

Yes. If a password is set, then issue the SET PASSWORD command prior to this command.

Syntax

From the operating system:

```
lsnrctl trace {level}[listener_name]
```

From the Listener Control utility:

```
LSNRCTL> trace {level} [listener_name]
```

Arguments

{*level*}: Specify one of the following trace levels:

- off for no trace output
- user for user trace information

- admin for administration trace information
- support for Oracle Support Services trace information

[*listener_name*]: Specify the listener name, if the default name of LISTENER is not used.

Usage Notes

This command has the same functionality as SET TRC_LEVEL command.

Example

```
LSNRCTL> TRACE ADMIN lsnr
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
Opened trace file: /oracle/network/trace/listener.trc
The command completed successfully
```

VERSION

Purpose

Use the command VERSION to display the current version of Listener Control utility.

Prerequisites

None

Password Required If One Has Been Set

No. If a password is set, then the SET PASSWORD command does not need to be issued prior to this command.

Syntax

From the operating system:

```
lsnrctl VERSION [listener_name]
```

From the Listener Control utility:

```
LSNRCTL> VERSION [listener_name]
```

Arguments

[*listener_name*]: Specify the listener name, if the default name of LISTENER is not used.

Example

```
LSNRCTL> VERSION listener1
Connecting to ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
TNSLSNR for Solaris: Version 10.0.0.0.0
        TNS for Solaris: Version 10.0.0.0.0
        Oracle Bequeath NT Protocol Adapter for Solaris: Version 10.0.0.0.0
        Unix Domain Socket IPC NT Protocol Adaptor for Solaris: Version 10.0.0.0.0
        TCP/IP NT Protocol Adapter for Solaris: Version 10.0.0.0.0
The command completed successfully
```

2

Oracle Connection Manager Control Utility

This chapter describes the commands and associated syntax of the **Oracle Connection Manager Control utility**.

This chapter contains these topics:

Oracle Connection Manager Control Utility Overview

The Oracle Connection Manager Control utility enables you to administer **Oracle Connection Manager**s. You can use its commands to perform basic management functions on one or more Oracle Connection Managers. Additionally, you can view and change parameter settings.

Command Modes and Syntax

The basic syntax of the Oracle Connection Manager Control utility is as follows:

```
cmctl command [argument]
```

You can choose between two command modes:

- Interactive:

  Enter `cmctl` at the command line to obtain the program prompt; then issue the command:

  ```
  cmctl
  CMCTL> command
  ```

- One shot:

  Enter the entire command from the operating system:

  ```
  cmctl {command} [argument1 . . . argumentN] [-c instance_name] [-p password]
  ```

  Each command issued in this way can have the name of an Oracle Connection Manager and a password appended as arguments. If an Oracle Connection Manager name is not provided, the default instance name is assumed. A password is necessary only if one was set in a previous CMCTL session. Note that an

interactive session of Oracle Connection Manager requires that a password be entered only once, at the outset, if one has been set at all.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for an overview of the Oracle Connection Manager processes

- Batch mode:

  You can combine commands in a standard text file and then run them as a sequence of commands. To execute in batch mode, use the following format:

  ```
  cmctl @input_file
  ```

The Oracle Connection Manager Control utility supports four types of commands:

- Initialization and termination commands such as STARTUP and SHUTDOWN
- Alter commands such as SET LOG_LEVEL and SET EVENT
- Display commands, such as SHOW STATUS and SHOW RULES
- Gateway commands such as SHOW GATEWAYS and RESUME GATEWAYS

Note that while you can use SET commands to dynamically alter configuration parameters, these changes only remain in effect until the Oracle Connection Manager is shut down. You cannot save them to the `cman.ora` file. The one exception is the Oracle Connection Manager password, which you can save by issuing the command SAVE_PASSWORD.

Distributed Operations

The Oracle Connection Manager Control utility can perform operations on a local or a remote Oracle Connection Manager. Note, however, that an instance must be started locally—that is, on the computer where the instance is located.

To set up one instance of Oracle Connection Manager to remotely administer another:

1. Configure the `tnsnames.ora` file on the local computer to include the remote listening address. Assume, for instance, that the local Oracle Connection Manager is called cman1 and that it resides on proxysvr1. Assume, too, that the remote Oracle Connection Manager is called cman2 and that it resides on proxysvr2. The `tnsnames.ora` file on proxysvr1 would be configured this way:

   ```
   CMAN2=
   ((ADDRESS=(PROTOCOL=tcp)(HOST=proxysvr2)(PORT=1521)))
   ```

1. In the `cman.ora` file on the remote computer, set the value of the parameter `REMOTE_ADMIN` to `yes` as in the following example:

   ```
   CMAN2=
     (CONFIGURATION=
       (ADDRESS=(PROTOCOL=tcp)(host=proxysvr2)(port=1521))
       (PARAMETER_LIST=
   (REMOTE_ADMIN=YES)))
   ```

Oracle Connection Manager Control Utility Commands

This section lists and describes commands for the Oracle Connection Manager Control utility.

ADMINISTER

Purpose

Use the ADMINISTER command to choose an instance of Oracle Connection Manager.

Prerequisites

None.

Syntax

From the Oracle Connection Manager Control utility:

```
CMCTL> ADMINISTER [-c instance_name] using [password]
```

Arguments

[instance_name]: Specify the instance of Oracle Connection Manager that you would like to administer. Instances are defined in the cman.ora file.

[password]: Specify the password, if any, for this instance of Oracle Connection Manager.

Usage Notes

Issue ADMINISTER only in interactive mode. You cannot issue the command from the operating system.

ADMINISTER enables you to choose an Oracle Connection Manager to administer. To start this Oracle Connection Manager, you must issue .STARTUP

When you omit the instance name from the command, the instance administered defaults to the local instance.

A password is required only if one was provided at install time or during a previous session of the Oracle Connection Manager.

Example

```
CMCTL> ADMINISTER
Current instance is CMAN_user-sun.us.oracle.com (instance not yet started).
Connections refer to
(address=(protocol=tcp)(host=user-sun.us.oracle.com)(port=1630)).
The command completed successfully
```

CLOSE CONNECTIONS

Purpose

Use the CLOSE CONNECTIONS command to terminate connections, using specific qualifiers to select connections.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl CLOSE CONNECTIONS [in state] [gt time] [from source] [to destination]
[for service] [using gateway_process_id] [connect_identifier_list]
{-c cman_name}{-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> CLOSE CONNECTIONS [in state] [gt time] [from source] [to destination] [for
service] [using gateway_process_id] [connect_identifier_list]
```

Arguments

`[in state]`: Use one of the following values to specify the connection state:

- `idle`—Connections that are inactive in the established state
- `connecting`—Connections that are in the process of connecting
- `established`—Connections that are connected and are transferring data
- `terminating`—Connections that are disconnecting

If no state is specified, `CLOSE CONNECTIONS` defaults to all possible states. If the time qualifier is included under these conditions, the time specified is the amount of time that has elapsed since a client initiated a connection.

`[gt time]`: Use the following format to specify connections greater than the time indicated:

`gt[hh:mm:]ss`

`[from source]`: Use one of the following formats to specify the source address:

- `from IP`
- `from hostname`
- `from subnet`

`[to destination]`: Use one of the following formats to specify the destination address:

- `from IP`
- `from hostname`
- `from subnet`

`[for service]`: Use the following format to request a service:

`for service_name`

`[using gateway_process_id]`: Use this format to specify connections that are proxied by the gateway process indicated.

`[connect_identifier_list]`: Space between multiple connection identifiers in a list.

Usage Notes

Because the `CLOSE CONNECTIONS` command aborts connections, it might generate error messages on both client and server sides.

The `IDLE` state qualifier always requires a time qualifier.

Issuing `CLOSE CONNECTIONS` without an argument closes all connections.

Examples

The following shuts down connections in any state. The elapsed time of the connection must be greater than 1 hour and 30 minutes. The connection source is the specified subnet; the destination, the specified host name.

```
CMCTL:CMAN_user-sun.us.oracle.com>
CLOSE CONNECTIONS gt 1:30:00 from 206.62.226.32/27 to host1
```

The following shuts down those connections proxied by gateway process 0 that have been in the idle state more than 30 minutes:

```
CMCTL:CMAN_user-sun.us.oracle.com> CLOSE idle CONNECTIONS gt 30:00 using 0
```

The following shuts down connections that are connected to the service sales.us.acme.com:

```
CMCTL:CMAN_user-sun.us.oracle.com> CLOSE established CONNECTIONS for
sales.us.acme.com
```

EXIT

Purpose

Use the EXIT command to exit from the Oracle Connection Manager Control utility.

Prerequisites

None

Syntax

From the operating system:

```
cmctl EXIT {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> EXIT
```

Usage Notes

This command is identical to the QUIT command.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> EXIT
```

HELP

Purpose

Use the HELP command to provide a list of all commands for the Oracle Connection Manager Control utility or to provide help with the syntax of a particular command.

Prerequisites

None

Syntax

From the operating system:

```
cmctl HELP [command] {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> HELP [command]
```

Arguments

[*command*]: Specify a HELP command. Commands are shown in the following example output.

When you enter a command as an argument to HELP, the Oracle Connection Manager Control utility displays information about how to use the command. When you enter HELP without an argument, the Oracle Connection Manager Control utility displays a list of all the commands.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> HELP
The following operations are available
An asterisk (*) denotes a modifier or extended command:

administer      close*          exit            reload
resume*         save_password   set*            show*
shutdown        sleep           startup         suspend*
show_version    quit
```

QUIT

Purpose

Use the QUIT command to exit the Oracle Connection Manager Control utility and return to the operating system prompt.

Prerequisites

None

Syntax

From the operating system:

```
cmctl QUIT
```

From the Oracle Connection Manager Control utility:

```
CMCTL> QUIT
```

Usage Notes

This command is identical to the command.EXIT

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> QUIT
```

RELOAD

Purpose

Use the RELOAD command to dynamically reread parameters and rules.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl RELOAD {-c instance_name}{-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> RELOAD
```

Usage Notes

Configuration information modified using the command applies only to new connections. Existing connections are unaffected. SETRELOAD, on the other hand, restores configurations set in cman.ora, thereby overriding the SET command.

RELOAD reregisters gateways with the Oracle Connection Manager listener, in the course of which some new connections might be refused.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> RELOAD
The command completed successfully
```

RESUME GATEWAYS

Purpose

Use the RESUME GATEWAYS command to resume gateway processes that have been suspended.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl RESUME GATEWAYS [gateway_process_id]{cman_name}{-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> RESUME GATEWAYS [gateway_process_id]
```

Arguments

[*gateway_process_id*]: Specify one or more gateway processes to reopen. Space once between entries to specify multiple gateway processes.

Usage Notes

Issuing RESUME GATEWAYS without an argument reopens all gateway processes that have been closed.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> RESUME GATEWAYS 1
The command completed successfully
```

SAVE_PASSWORD

Purpose

Use the SAVE_PASSWORD command to save the current password to cman.ora, the configuration file for Oracle Connection Manager.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SAVE_PASSWORD {-c instance_name}{-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SAVE_PASSWORD
```

Usage Notes

If you execute this command, the next session of Oracle Connection Manager will start with this password intact.

Example

```
CMCTL> SAVE_PASSWORD
```

SET

Purpose

Use the `SET` command to display a list of parameters that can be modified using this command.

Prerequisites

None

Syntax

From the operating system:

```
cmctl SET
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET
```

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SET
The following operations are available after set
An asterisk (*) denotes a modifier or extended command:

aso_authentication_filter     outbound_connect_timeout
connection_statistics         password
event                         session_timeout
idle_timeout                  trace_directory
inbound_connect_timeout      trace_level
log_directory
log_level
```

SET ASO_AUTHENTICATION_FILTER

Purpose

Use the `SET ASO_AUTHENTICATION_FILTER` command to indicate whether the client must use Oracle Advanced Security to authenticate.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET ASO_AUTHENTICATION_FILTER {on | off}{-c instance_name}{-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET ASO_AUTHENTICATION_FILTER {on | off}
```

Arguments

`[on]`: Specify to reject connections that are not using Secure Network Service (SNS) to perform client authentication. SNS is part of Oracle Advanced Security.

`[off]` (default): Specify so that no authentication is required for client connections.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> set aso_authentication_filter ON
CMAN_user-sun.us.oracle.com parameter aso_authentication_filter set to ON
The command completed successfully
```

SET CONNECTION_STATISTICS

Purpose

Use the `SET CONNECTION_STATISTICS` command to specify whether gateway processes collect connection statistics.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET CONNECTION_STATISTICS {yes | no}{-c instance_name}{-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET CONNECTION_STATISTICS {yes | no}
```

Arguments

`[yes]`: Specify to have gateway processes collect connection statistics

`[no]`: (Default) Specify that gateway processes not collect connection statistics

Usage Notes

If `SET CONNECTION_STATISTICS` is set to `yes`, you can obtain statistics by issuing the command SHOW CONNECTIONS.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> set connection_statistics ON
CMAN_user-sun.us.oracle.com parameter connection_statistics set to ON
The command completed successfully
```

SET EVENT

Purpose

Use the `SET EVENT` command to log information for a particular event.

Syntax

From the operating system:

```
cmctl SET EVENT event_group [-c instance_name][-p password]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET EVENT event_group {on | off}
```

Arguments

`[event_group]`: Specify one of the following event groups:

- `init_and_term`—initialization and termination
- `memory_ops`—memory operations
- `conn_hdlg`—connection handling
- `proc_mgmt`—process management
- `reg_and_load`—Registration and load update
- `wake_up`—events related to CMADMIN wakeup queue
- `timer`—gateway timeouts
- `cmd_proc`—command processing
- `relay`—events associated with connection control blocks

`[on | off]`: Specify whether to turn an event group on or off.

Usage Notes

The `SET EVENT` command accepts only one argument. To log multiple events, you must reissue the command.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> set event memory_ops on
show connections [detail | count] {[in <state>][gt <[[hh:]mm:]ss>]
from <source>][to <destination>][for <service>][using <gw_id>]}
| [<id_list>] - Shows statistics of selected connections
The command completed successfully
```

SET IDLE_TIMEOUT

Purpose

Use the `SET IDLE_TIMEOUT` command to specify the amount of time a client can be idle without transmitting data.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET IDLE_TIMEOUT [time] {-c instance_name} {-p password}
```

From the From the Oracle Connection Manager Control utility:

```
CMCTL> SET IDLE_TIMEOUT [time]
```

Arguments

`[time]`: Specify the idle timeout in seconds. The default is `0`, which disables this feature.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SET IDLE_TIMEOUT 30
CMAN_user-sun.us.oracle.com parameter idle_timeout set to 30
The command completed successfully
```

SET INBOUND_CONNECT_TIMEOUT

Purpose

Use the `SET INBOUND_CONNECT_TIMEOUT` command to specify the maximum amount of time the Oracle Connection Manager listener will wait for a valid connection request from the client before timing out.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET INBOUND_CONNECT_TIMEOUT {time} {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET INBOUND_CONNECT_TIMEOUT {time}
```

Arguments

`[time]`: Specify the inbound connect timeout in seconds. The default is `60`, which disables this feature.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SET INBOUND_CONNECT_TIMEOUT 30
CMAN_user-sun.us.oracle.com parameter inbound_connect_timeout set to 30
The command completed successfully
```

SET LOG_DIRECTORY

Purpose

Use the SET LOG_DIRECTORY command to designate where the log files for an Oracle Connection Manager are written.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET LOG_DIRECTORY {directory_path} {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET LOG_DIRECTORY {directory_path}
```

Arguments

[*directory_path*]: Specify to indicate the location of the log directory. The default path is as follows:

- UNIX:

  ```
  $ORACLE_HOME/network/log directory
  ```

- Windows 2000/NT:

  ```
  ORACLE_HOME\network\log directory
  ```

Usage Notes

Issue the SHOW PARAMETERS command to determine the location of the log files.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com>
SET LOG_DIRECTORY /ade/user_cman_test/oracle/network/admin

CMAN_user-sun.us.oracle.com parameter log_directory set to /ade/user
_cman_test/oracle/network/admin

The command completed successfully
```

SET LOG_LEVEL

Purpose

Use the SET LOG_LEVEL command to set the log level for an Oracle Connection Manager.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET LOG_LEVEL {level} {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET LOG_LEVEL {level}
```

Arguments

[*level*]: Specify one of the following log levels:

- `off` for no logging
- `user` for user log information
- `admin` for administrative log information
- `support` (default) for Oracle Support Services log information

Usage Notes

Choose `off` to capture a minimum amount of log information. Choose `support` to capture a maximum amount.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SET LOG_LEVEL SUPPORT
CMAN_user-sun.us.oracle.com parameter log_level set to support
The command completed successfully
```

SET OUTBOUND_CONNECT_TIMEOUT

Example

Use the `SET OUTBOUND_CONNECT_TIMEOUT` command to specify the maximum amount of time the Oracle Connection Manager instance will wait for a valid connection with the server before timing out.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET OUTBOUND_CONNECT_TIMEOUT {time} {-c instance_name} {-p password}
```

From the From the Oracle Connection Manager Control utility:

```
CMCTL> SET OUTBOUND_CONNECT_TIMEOUT {time}
```

Arguments

[*time*]: Specify the outbound connect timeout in seconds. The default is 0.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SET OUTBOUND_CONNECT_TIMEOUT 30
CMAN_user-sun.us.oracle.com parameter outbound_connect_timeout set to 30
The command completed successfully
```

SET PASSWORD

Purpose

Use the `SET PASSWORD` command to assign a password to the Oracle Connection Manager instance.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET PASSWORD
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET PASSWORD
```

Arguments

None.

Usage Notes

This command may be used either to set a password for the first time or to change an existing one.

This command does not save the password to `cman.ora`. As a result the password is valid only for the current session. To save the password once you have set it, execute the SAVE_PASSWORD command.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SET PASSWORD

Enter Old password:
Enter New password:
Reenter New password:

The command completed successfully
```

SET SESSION_TIMEOUT

Purpose

Use the `SET SESSION_TIMEOUT` command to specify the maximum amount of time for a session of Oracle Connection Manager.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET SESSION_TIMEOUT {time} {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET SESSION_TIMEOUT {time}
```

Arguments

{*time*}: Specify the session timeout in seconds. The default is 0, which disables this feature.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SET SESSION_TIMEOUT 60
CMAN_user-sun.us.oracle.com parameter session_timeout set to 60
The command completed successfully
```

SET TRACE_DIRECTORY

Purpose

Use the SET TRACE_DIRECTORY command to designate where the trace files for an Oracle Connection Manager are written.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET TRACE_DIRECTORY {directory_path} {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET TRACE_DIRECTORY {directory_path}
```

Arguments

{*directory_path*}: Specify to indicate the location of the trace directory. The default path is as follows:

- UNIX:

  ```
  $ORACLE_HOME/network/trace
  ```

- Windows 2000/NT:

  ```
  ORACLE_HOME\network\trace
  ```

Usage Notes

Issue the SHOW PARAMETERS command to determine the location of the trace files.

Example

```
CMCTL:cman1>SET TRACE_DIRECTORY /ade/mpurayat_newtest/oracle/network/trace
cman1 parameter trace_directory set to /ade/mpurayat_newtest/oracle/network
/trace
The command completed successfully
```

SET TRACE_LEVEL

Purpose

Use the SET TRACE_LEVEL command to set the trace level for an Oracle Connection Manager.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET TRACE_LEVEL {level} {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET TRACE_LEVEL {level}
```

Arguments

{level}: Specify one of the following log levels:

- `off` (default) for no tracing
- `user` for user trace information
- `admin` for administrative trace information
- `support` for Oracle Support Services trace information

Usage Notes

Choose `off` to capture a minimum amount of trace information. Choose `support` to capture a maximum amount.

Issue the SHOW PARAMETERS command to determine the current trace level.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SET TRACE_LEVEL SUPPORT
CMAN_user-sun.us.oracle.com parameter trace_level set to user
The command completed successfully
```

SHOW

Purpose

Use the `SHOW` command to display a list of parameters that may be used as arguments for this command. Entering one of these parameters with the command displays the parameter value or values.

Prerequisites

None

Syntax

From the operating system:

```
cmctl SHOW {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW
```

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SHOW
The following operations are available after show
```

```
An asterisk (*) denotes a modifier or extended command:

all             gateways        status
connections     parameters      version
defaults        rules
events          services
```

SHOW ALL

Purpose

Use the SHOW ALL command to combine and display output from the SHOW PARAMETERS and SHOW RULES commands.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW ALL {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW ALL
```

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SHOW ALL
listener_address          |
(address=(protocol=tcp)(host=user-sun.us.oracle.com)(port=1630))
aso_authentication_filter |   OFF
connection_statistics     |   OFF
event_group               |   OFF
log_directory             | /ade/user_cman_test/oracle/network/log/
log_level                 | SUPPORT
max_connections           |   256
idle_timeout              |     0
inbound_connect_timeout   |     0
session_timeout           |     0
outbound_connect_timeout  |     0
max_gateway_processes     |    16
min_gateway_processes     |     2
max_cmctl_sessions        |     4
password                  |   OFF
remote_admin              |   OFF
trace_directory           | /ade/user_cman_test/oracle/network/trace/
trace_level               |   OFF
trace_timestamp           |   OFF
trace_filelen             |     0
trace_fileno              |     0
(rule_list=
 (rule=
  (src=*)
  (dst=*)
  (srv=*)
  (act=accept)
 )
)
The command completed successfully
```

SHOW CONNECTIONS

Purpose

Use the SHOW CONNECTIONS command to display information about specific connections or all connections.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW CONNECTIONS [information] [in state] [gt time] [from source]
to destination] [for service] [using gateway_process_id] [connect_identifier_list]
{-c instance_name}{-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW CONNECTIONS [information][in state] [gt time] [from source]
[to destination] [for service] [using gateway_process_id] [connect_identifier_
list]
```

Arguments

[*information*]: Use one of the following two values to display information about connections. Information categories include connection ID, source, destination, service, current state, total idle time, and total elapsed time.

- count: (default) Displays the total number of connections that meet the criteria specified by the other qualifiers.
- detail: Displays all information about connections specified by the other qualifiers.

[in *state*]: Use one of the following values to specify the connection state:

- idle—Connections that are inactive in the established state
- connecting—Connections that are in the process of connecting
- established—Connections that are connected and are transferring data
- terminating—Connections that are disconnecting

If no state is specified, SHOW CONNECTIONS defaults to all possible states. If the time qualifier is included under these conditions, the time specified is the amount of time that has elapsed since a client initiated a connection.

[gt *time*]: Use the following format to specify connections greater than the time indicated:

gt[*hh*:*mm*:]*ss*

[from *source*]: Use one of the following formats to specify the source address:

- from *IP*
- from *hostname*
- from *subnet*

[to *destination*]: Use one of the following formats to specify the destination address:

- from *IP*
- from *hostname*
- from *subnet*

[for *service*]: Use the following format to request a service:

for *service_name*

[using *gateway_process_id*]: Use this format to specify connections that are proxied by the gateway process indicated

using *gateway_process_id*

[*connect_identifier_list*]: Space between multiple connection identifiers in a list

Usage Notes

Connections are sorted by gateway process ID and connection identifier, in ascending order.

Issuing SHOW CONNECTIONS without an argument displays all connections.

Examples

The following displays a detailed description of connections in any state. The elapsed time of the connection must be greater than 1 hour and 30 minutes. The connection source is the specified subnet, and the destination the specified host name.

```
CMCTL> SHOW CONNECTIONS gt 1:30:00 from 206.62.226.32/27 to host1
```

The following displays the number of connections proxied by cman 0 that have been in the idle state more than 30 minutes:

```
CMCTL> SHOW idle CONNECTIONS count gt 30:00 using 0
```

The following displays a detailed description of connections that are connected to the service sales.us.acme.com:

```
CMCTL> SHOW established CONNECTIONS detail for sales.us.acme.com
```

SHOW DEFAULTS

Purpose

Use the SHOW DEFAULTS command to display default parameter settings.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

cmctl SHOW DEFAULTS {-c *instance_name*} {-p *password*}

From the Oracle Connection Manager Control utility:

CMCTL> SHOW DEFAULTS

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SHOW DEFAULTS
listener_address          |
(ADDRESS=(PROTOCOL=TCP)(HOST=user-sun.us.oracle.com)(PORT=1521))
aso_authentication_filter |   OFF
connection_statistics     |   OFF
event_group               |   OFF
log_directory             | /ade/user_cman_test/oracle/network/log/
log_level                 | SUPPORT
max_connections           |   256
idle_timeout              |     0
inbound_connect_timeout   |     0
session_timeout           |     0
outbound_connect_timeout  |     0
max_gateway_processes     |    16
min_gateway_processes     |     2
max_cmctl_sessions        |     4
password                  |   OFF
remote_admin              |   OFF
trace_directory           | /ade/user_cman_test/oracle/network/trace/
trace_level               |   OFF
trace_timestamp           |   OFF
trace_filelen             |     0
trace_fileno              |     0
The command completed successfully
```

SHOW EVENTS

Purpose

Use the `SHOW EVENTS` command to display the events that are in operation.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW EVENTS {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW EVENTS
```

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SHOW EVENTS
Event Groups:
memory_ops
The command completed successfully
```

SHOW GATEWAYS

Purpose

Use the `SHOW GATEWAYS` command to display the current status of a specific gateway process or processes. Statistics displayed include number of active connections, number of peak active connections, total number of connections handled, and number of connections refused.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW GATEWAYS [  gateway] {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW GATEWAYS [  gateway]
```

Arguments

[*gateway*]: Enter the ID of the gateway or gateways whose status you want to display

Issuing `SHOW GATEWAYS` without an argument displays the status of all gateway processes.

Usage Notes

If you want to display multiple gateways, use a space to separate the ID numbers when entering the command.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SHOW GATEWAYS 1
Gateway ID                    1
Gateway state                 READY
Number of active connections  0
Peak active connections       0
Total connections             0
Total connections refused     0
The command completed successfully
```

SHOW PARAMETERS

Purpose

Use the `SHOW PARAMETERS` command to display current parameter settings for an instance.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW PARAMETERS {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW PARAMETERS
```

Usage Notes

Several configuration parameters can be dynamically modified using the SET command; therefore, the information that SHOW PARAMETERS displays might be different from what appears in the `cman.ora` file.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SHOW PARAMETERS
listener_address          |
(address=(protocol=tcp)(host=user-sun.us.oracle.com)(port=1630))
aso_authentication_filter |    ON
connection_statistics     |    ON
event_group               | (memory_ops)
log_directory             | /ade/user_cman_test/oracle/network/log/
log_level                 | SUPPORT
max_connections           |   256
idle_timeout              |     0
inbound_connect_timeout   |     0
session_timeout           |     0
outbound_connect_timeout  |     0
max_gateway_processes     |    16
min_gateway_processes     |     2
max_cmctl_sessions        |     4
password                  |   OFF
remote_admin              |   OFF
trace_directory           | /ade/user_cman_test/oracle/network/trace/
trace_level               | SUPPORT
trace_timestamp           |   OFF
trace_filelen             |     0
trace_fileno              |     0
The command completed successfully
```

SHOW RULES

Purpose

Use the `SHOW RULES` command to display the access control list currently used by the instance.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW_RULES {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW_RULES
```

Usage Notes

You can update the rules list by issuing the RELOAD command.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SHOW RULES
Number of filtering rules currently in effect: 4
(rule_list=
  (rule=
    (src=usunnae12)
    (dst=usunnae13)
    (srv=*)
    (act=accept)
```

```
    (action_list=(mit=120)(mct=1800)(conn_stats=on)(aut=off))
  )
  (rule=
    (src=usunnae12)
    (dst=usunnae14)
    (srv=service2)
    (act=accept)
  )
  (rule=
    (src=*)
    (dst=usunnae15)
    (srv=*)
    (act=accept)
    (action_list=(mit=120)(mct=3000)(moct=200)(aut=on))
  )

  (rule=
    (src=*)
    (dst=usunnae16)
    (srv=*)
    (act=reject)
    (action_list=(moct=20)(aut=on))
  )

  (rule=
    (src=user-sun.us.oracle.com)
    (dst=user-sun.us.oracle.com)
    (srv=cmon)
    (act=accept)
    (action_list=(mit=100)(mct=1130)(moct=200)(aut=on))
  )
)
```

SHOW SERVICES

Purpose

Use the `SHOW SERVICES` command to display comprehensive information about the Oracle Connection Manager instance. The information displayed includes number of handlers for gateway and CMADMIN processes, listening ports of handlers, and number of connections—refused and current.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW SERVICES {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW SERVICES
```

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SHOW SERVICES
Services Summary...
Proxy service "cmgw" has 1 instance(s).
```

```
  Instance "cman", status READY, has 2 handler(s) for this service...
    Handler(s):
      "cmgw001" established:0 refused:0 current:0 max:256 state:ready
         <machine: user-sun, pid: 29190>
         (ADDRESS=(PROTOCOL=tcp)(HOST=user-sun)(PORT=33175))
      "cmgw000" established:0 refused:0 current:0 max:256 state:ready
         <machine: user-sun, pid: 29188>
         (ADDRESS=(PROTOCOL=tcp)(HOST=user-sun)(PORT=33174))
Service "cmon" has 1 instance(s).
  Instance "cman", status READY, has 1 handler(s) for this service...
    Handler(s):
      "cmon" established:0 refused:0 current:0 max:4 state:ready
         <machine: user-sun, pid: 29184>
         (ADDRESS=(PROTOCOL=tcp)(HOST=user-sun)(PORT=33168))
The command completed successfully
```

SHOW STATUS

Purpose

Use the `SHOW STATUS` command to display basic information about the instance, including version, start time, and current statistics.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW STATUS
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW STATUS
```

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SHOW STATUS
Status of the Instance
----------------------
Instance name             CMAN_user-sun.us.oracle.com
Version                   CMAN for Solaris: Version 10.1.0.0.0 - Beta
Start date                20-JAN-2003 14:50:35
Uptime                    0 days 1 hr. 25 min. 24 sec
Num of gateways started   2
Average Load level        0
Log Level                 SUPPORT
Trace Level               OFF
Instance Config file      /ade/user_cman_test/oracle/network/admin/cman.ora
Instance Log directory    /ade/user_cman_test/oracle/network/log/
Instance Trace directory  /ade/user_cman_test/oracle/network/trace/
The command completed successfully
```

SHOW VERSION

Purpose

Use the `SHOW VERSION` command to display the current version and name of the Oracle Connection Manager Control utility.

Prerequisites

None

Syntax

From the operating system:

```
cmctl SHOW VERSION {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW VERSION
```

Examples

```
CMCTL:CMAN_user-sun.us.oracle.com> SHOW VERSION
CMAN for Solaris: Version 10.1.0.0.0 - Beta
The command completed successfully
```

SHUTDOWN

Purpose

Use the SHUTDOWN command to shut down specific gateway processes or the entire Oracle Connection Manager instance.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHUTDOWN [gateways {gateway}] [normal | abort] {-c instance_name} {-p
password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHUTDOWN [gateways {gateway}] [normal | abort]
```

Arguments

[normal]: (default): Specify to reject new connections and terminate after existing connections close.

[abort]: Specify to shut down Oracle Connection Manager immediately, closing down all open connections.

[gateways]: Specify to shut down a specific gateways.

You can specify more than one gateway by inserting a space between them in the command line.

Usage Notes

Issuing SHUTDOWN without an argument shuts down all gateways.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SHUTDOWN GATEWAYS 0
The command completed successfully
```

STARTUP

Purpose

Use the STARTUP command to start an Oracle Connection Manager.

Prerequisites

An Oracle Connection Manager configured with the same protocol address must not be running.

Syntax

From the operating system:

```
cmctl STARTUP {-c instance_name} {-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> STARTUP
```

Usage Notes

Before issuing this command, you must issue the ADMINISTER command to choose an instance to start.

```
Issuing this command starts all three instance components—
the listener, CMADMIN, and the gateway processes.
The command fails if any one of these components is already running.
```

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> STARTUP
Starting CMAN instance: CMAN_user-sun.us.oracle.com, please wait...
TNS-04090: *** CMCTL WARNING: No password set in the CMAN instance ***
CMAN for Solaris: Version 10.1.0.0.0 - Beta
Status of the Instance
----------------------
Instance name             CMAN_user-sun.us.oracle.com
Version                   CMAN for Solaris: Version 10.1.0.0.0 - Beta
Start date                20-JAN-2003 19:04:25
Uptime                    0 days 0 hr. 0 min. 3 sec
Num of gateways started   2
Average Load level        0
Log Level                 SUPPORT
Trace Level               OFF
Instance Config file      /ade/user_cman_test/oracle/network/admin/cman.ora
Instance Log directory    /ade/user_cman_test/oracle/network/log/
Instance Trace directory  /ade/user_cman_test/oracle/network/trace/
The command completed successfully
```

SUSPEND GATEWAY

Purpose

Use the SUSPEND GATEWAY command to choose gateway processes that will no longer accept new client connections.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SUSPEND GATEWAY [  gateway_process_id]{-c instance_name}{-p password}
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SUSPEND GATEWAY [ gateway_process_id]
```

Arguments

[*gateway_process_id*]: Specify the gateway process that will no longer accept new connections. Specify multiple gateway processes by spacing once between entries.

Issuing `SUSPEND GATEWAY` without an argument suspends all gateway processes.

Usage Notes

Use the RESUME GATEWAYS command to enable gateway processes to accept new connections.

Example

```
CMCTL:CMAN_user-sun.us.oracle.com> SUSPEND GATEWAY 1
The command completed successfully
```

Part II

Configuration Parameters

Part II describes how to configure listening protocol addresses and Oracle Net Services configuration parameters.

This part contains the following chapters:

3

Syntax Rules for Configuration Files

This chapter describes the syntax rules for Oracle Net Services configuration files.

This chapter contains these topics:

Configuration File Syntax Overview

The Oracle Net Services configuration files consist of parameters which include keyword-value pairs. Keyword-value pairs are surrounded by parentheses:

```
parameter=(keyword=value)
```

Some keywords have other keyword-value pairs as their values:

```
(keyword=
    (keyword=value)
    (keyword=value))
```

For example, the address portion of a local naming configuration file (`tnsnames.ora`) might include the following lines:

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

Set up configuration files so that indentation reflects what keyword is the parent or owner of other keyword-value pairs.

Even if you do not choose to indent your files in this way, you must indent a wrapped line by at least one space, or it will be misread as a new parameter. The following layout is acceptable:

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

The following layout is not acceptable:

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

Further Syntax Rules for Configuration Files

The following rules apply to the syntax of configuration files:

- Any keyword in a configuration file that begins a parameter that includes one or more keyword-value pairs must be in the far left column of a line. If it is indented by one or more spaces, it is interpreted as a continuation of the previous line.
- All characters must belong to the network character set

  See Also: "Network Character Set" on page 3-2

- Keywords are not case sensitive. Values may be case sensitive, depending on the operating system and protocol.
- Spaces around the "=" sign are optional in keyword-value pairs.
- There is a hierarchy of keywords in that some keywords are always followed by others. At any level of the hierarchy, keywords can be listed in any order. For example, the following entries are equally valid:

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

- Keywords cannot contain spaces. Values must not contain spaces unless enclosed within double quotes (") or single quotes (').
- The maximum length of a connect descriptor is 4 KB
- Comments can be included using the pound sign # at the beginning of a line. Anything following the sign to the end of the line is considered a comment.
- If the keyword-value pair consists of a single word or a concatenation of words on either side of the equal sign, no parentheses are needed.

Network Character Set

The network character set for keyword values consists of the following characters. Connect descriptors must be made up of single-byte characters.

```
A-Z, a-z

0-9

( ) < > / \

, . : ; ' "=- _

$ + * # & ! % ? @
```

Within this character set, the following symbols are reserved:

```
(  )=\ " ' #
```

Reserved symbols are used as delimiters, not as part of a keyword or a value unless the keyword or value is quoted. Either single or double quotes can be used to enclose a value containing reserved symbols. To include a quote within a value that is

surrounded by quotes, use different quote types. The backslash (\) is used as an escape character.

The following characters may be used within a connect descriptor, but not in a keyword or value:

<Space> <Tab> <Carriage Return> <Newline>

Character Set

The listener name and net service name are limited to the following character set:

```
[a...z] [A...Z] [0...9] _
```

The first character must be an alphabetical character. In general, up to 64 characters is acceptable. A database service name must match the global database name defined by the database administrator, which consists of a database name (originally limited to eight characters), and the database domain. Net service names and global database names are not case sensitive.

4

Protocol Address Configuration

A network object is identified by a **protocol address**. When a connection is made, the client and the receiver of the request (**listener** 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, and grants a connection based on its address information matching the client information.

This chapter contains these topics:

ADDRESSes and ADDRESS_LISTs

Protocol address are comprised of ADDRESS and ADDRESS_LIST elements.

ADDRESS

Purpose

The `ADDRESS` parameter defines a protocol address.

Embed this parameter under an `ADDRESS_LIST` or `DESCRIPTION` parameter. A `DESCRIPTION` is used in a `tnsnames.ora` or a `listener.ora` file.

> **See Also:** "Protocol Parameters" on page 4-2 for each protocol's required parameters

Example

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

ADDRESS_LIST

Purpose

The ADDRESS_LIST parameter defines a list of protocol addresses that share common characteristics.

Example

```
(ADDRESS_LIST=
 (LOAD_BALANCE=on)
 (ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales-server)
  (PORT=1521))
 (ADDRESS=
  (PROTOCOL=tcp)
  (HOST=hr-server)
  (PORT=1521)))
(ADDRESS_LIST=
 (ADDRESS=
  (PROTOCOL=tcp)
  (HOST=finance-server)
  (PORT=1521)))
```

Protocol Parameters

The listener and Oracle Connection Manager are identified by protocol addresses. Table 4–1, " Protocol-Specific Parameters" describes the parameters used by the Oracle protocol support.

Table 4–1 Protocol-Specific Parameters

Protocol	Parameter	Description
IPC	PROTOCOL	Specify ipc as the value.
	KEY	Specify a unique name for the service. Oracle Corporation recommends using the service name or the **Oracle System Identifier (SID)** of the service. **Example:** (PROTOCOL=ipc)(KEY=sales)
Named Pipes	PROTOCOL	Specify nmp as the value.
	SERVER	Specify the name of the Oracle server computer.
	PIPE	Specify the pipe name you used to connect to the database server (the same PIPE keyword you specified on server with Named Pipes). This name can be any arbitrary name. **Example:** (PROTOCOL=nmp)(SERVER=sales)(PIPE=dbpipe0)
SDP	PROTOCOL	Specify sdp as the value.
	HOST	Specify the host name or IP address of the computer.

Table 4–1 (Cont.) Protocol-Specific Parameters

Protocol	Parameter	Description
	PORT	Specify the listening port number. **Example:** (PROTOCOL=sdp)(HOST=sales-server)(PORT=1521) (PROTOCOL=sdp)(HOST=44.25.186.204)(PORT=1521) **See Also:** "Recommended Port Numbers" on page 4-3
TCP/IP	PROTOCOL	Specify tcp as the value.
	HOST	Specify the host name or IP address of the computer.
	PORT	Specify the listening port number. **Example:** (PROTOCOL=tcp)(HOST=sales-server)(PORT=1521) (PROTOCOL=tcp)(HOST=44.25.186.204)(PORT=1521) **See Also:** "Recommended Port Numbers" on page 4-3
TCP/IP with SSL	PROTOCOL	Specify tcps as the value.
	HOST	Specify the host name or IP address of the computer.
	PORT	Specify the listening port number. **Example:** (PROTOCOL=tcps)(HOST=sales-server)(PORT=2484) (PROTOCOL=tcps)(HOST=44.25.186.204)(PORT=2484) **See Also:** "Recommended Port Numbers" on page 4-3

Recommended Port Numbers

Table 4–2, " Recommended Port Numbers" lists the recommends the port numbers.

Table 4–2 Recommended Port Numbers

Port	Description
1521	Default listening port for client connections to the listener. In future releases, this port number may change to the officially registered port number of 2483 for TCP/IP and 2484 for TCP/IP with SSL.
1521	Default and officially registered listening port for client connections to Oracle Connection Manager
1830	Default and officially registered listening port for administrative commands to Oracle Connection Manager

Port Number Limitations

Oracle Corporation allows port numbers from 1 to 65535. Port numbers less than 1024 are reserved for use by privileged processes on many operating systems.

On certain operating systems, only privileged processes can listen for TCP connections on ports less than 1024. If you need to configure listener to listen on a port number less than 1024, follow these general steps. Your operating system may require different procedures.

1. Use Oracle Net Configuration Assistant or Oracle Net Manager to configure the listener with protocol addresses and other configuration parameters.

See Also: *Oracle Database Net Services Administrator's Guide*

2. Log in as super user (`root`) and set file ownership and access permissions for the listener executable (`tnslsnr`) and the dependent shared libraries so that these files can be modified only by the super user.

   Ensure that the permissions of the individual directories found in the path names to these files, starting with the root directory, are also modified in the same way.

3. Start the listener as `root`.

   At the operating system prompt, enter `tnslsnr` with optional command line arguments. The usage is as follows:

   ```
   tnslsnr [listener_name] [-user user] [-group group]
   ```
   where:

Table 4–3 tnslsnr Utility Options

Option	Description
listener_name	Specify the name of the listener. If omitted, the default name `LISTENER` will be used.
`-user` *user*	Specify the user whose privileges the listener will use when super user (`root`) privileges are not needed. After performing the privileged operations, the listener will give up `root` privileges irreversibly.
`-group` *group*	Specify the group whose privileges the listener will use when super user (`root`) group privileges are not needed. After performing the privileged operations, the listener will give up `root` group privileges irreversibly.

   The listener will temporarily switch to the provided user and group immediately after startup. All subsequent operations will be done with the specified user and group privileges, except the system calls necessary to listen on configured endpoints. The listener will revert to super user (`root`) for a short period of time to listen on reserved addresses, such as TCP ports less than 1024. After the listener starts listening on all of its endpoints configured in `listener.ora`, it will switch to the specified user and group irreversibly. Therefore, the listener will give up the `root` privilege that it initially had. In the current release, `-user` and `-group` command line arguments only accept user and group identifiers specified in numeric form.

   For example, to execute a root listener called `mylsnr` and have it use privileges of a user identified as 37555 with a group identifier of 16, enter the following at the operating system command prompt. Note that 37555 could be the identifier for user `oracle` and 16 could be the identifier for the `dba` group.

   ```
   tnslsnr mylsnr -user 37555 -group 16
   ```

4. After the listener has been started, you can administer it with the Listener Control utility.

Important Notes:

- Oracle Corporation recommends that the user under which the listener process runs be `oracle`, as described in the example in Step 3, or whichever user the listener process normally runs as on the operating system.
- Do not leave the listener process running as root because doing so is a security vulnerability.

5

Profile Parameters (sqlnet.ora)

This chapter provides complete listing of the `sqlnet.ora` file configuration parameters.

This chapter contains these topics:

Overview of Profile Configuration File

The `sqlnet.ora` file enables you to:

- Specify the client domain to append to unqualified names
- Prioritize **naming method**s
- 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

By default, `sqlnet.ora` is located in the `$ORACLE_HOME/network/admin` directory on UNIX operating systems and the *`ORACLE_HOME`*`\network\admin` directory on Windows operating systems. `sqlnet.ora` can also be stored in the directory specified by the `TNS_ADMIN` environment variable.

Profile Parameters

This section lists and describes the `sqlnet.ora` file parameters.

BEQUEATH_DETACH

Purpose

Use the parameter `BEQUEATH_DETACH` to turn signal handling on or off for UNIX systems.

Default

`no`

Values

- yes to turn signal handling off
- no to leave signal handling on

Example

```
BEQUEATH_DETACH=yes
```

DEFAULT_SDU_SIZE

Purpose

Use the parameter DEFAULT_SDU_SIZE to specify the **session data unit (SDU)** size, in bytes to connections.

Usage

Oracle Corporation recommends setting this parameter in both the clientside and serverside sqlnet.ora file to ensure the same SDU size is used throughout a connection. When the configured values of client and database server do not match for a session, the lower of the two values is used.

You can override this parameter for a particular client connection by specifying the SDU parameter in the connect descriptor for a client.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for complete SDU usage and configuration information

Default

2048 bytes (2 KB)

Values

512 to 32768 bytes (32 KB)

Example

```
DEFAULT_SDU_SIZE=4096
```

DISABLE_OOB

Purpose

If turned off, the parameter DISABLE_OOB enables Oracle Net to send and receive "break" messages using urgent data provided by the underlying protocol.

If turned 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.

> **See Also:** Operating system-specific documentation to determine if the protocols you are using support urgent data requests. TCP/IP is an example of a protocol that supports this feature.

Default

off

Example

```
DISABLE_OOB=on
```

LOG_DIRECTORY_CLIENT

Purpose

Use the parameter LOG_DIRECTORY_CLIENT to specify the destination directory for the client log file.

Default

Current directory from which the executable is started

Example

```
LOG_DIRECTORY_CLIENT=/oracle/network/log
```

LOG_DIRECTORY_SERVER

Purpose

Use the parameter LOG_DIRECTORY_SERVER to specify the destination directory for the database server log file.

Default

Current directory from which the executable is started

Example

```
LOG_DIRECTORY_SERVER=/oracle/network/log
```

LOG_FILE_CLIENT

Purpose

Use the parameter LOG_FILE_CLIENT to specify the name of the log file for the client.

Default

```
sqlnet.log
```

Example

```
LOG_FILE_CLIENT=client
```

LOG_FILE_SERVER

Purpose

Use the parameter LOG_FILE_SERVER to specify the name of the log file for the database server.

Default

```
sqlnet.log
```

Example

```
LOG_FILE_SERVER=svr.log
```

NAMES.DCE.PREFIX

Purpose

Use the parameter NAMES.DCE.PREFIX to specify the Distributed Computing Environment (DCE) cell name (prefix) to use for name lookups.

Default

/.:/subsys/oracle/names

Example

```
NAMES.DCE.PREFIX=/.:/subsys/oracle/names
```

NAMES.DEFAULT_DOMAIN

Purpose

Use the parameter NAMES.DEFAULT_DOMAIN to set the domain from which the client most often looks up names resolution requests. When this parameter is set, the default domain name is automatically appended to any unqualified net service name or service name.

For example, if the default domain is set to us.acme.com, the connect string CONNECT scott/tiger@sales gets searched as sales.us.acme.com. If the connect string includes the domain extension, such as CONNECT scott/tiger@sales.acme.com, the domain is not appended.

Default

None

Example

```
NAMES.DEFAULT_DOMAIN=acme.com
```

NAMES.DIRECTORY_PATH

Purpose

Use the parameter NAMES.DIRECTORY_PATH to specify the order of the naming methods used for client name resolution lookups.

Default

```
NAMES.DIRECTORY_PATH=(tnsnames, onames, hostname)
```

Values

Table 5–1 NAMES.DIRECTORY_PATH Values

Naming Method Value	Description
tnsnames (**local naming** naming method)	Set to resolve a **net service name** through the tnsnames.ora file on the client. **See Also:** *Oracle Database Net Services Administrator's Guide*

Table 5–1 (Cont.) NAMES.DIRECTORY_PATH Values

Naming Method Value	Description
`ldap` (**directory naming** naming method)	Set to resolve a database service name, net service name, or **net service alias** through a **directory server**. **See Also:** *Oracle Database Net Services Administrator's Guide*
`ezconnect` or `hostname` (**easy connect naming** or **host naming** method)	Select to enable clients to use a TCP/IP connect identifier, consisting of a host name and optional port and service name. **See Also:** *Oracle Database Net Services Administrator's Guide*
`cds` (**CDS** external naming method)	Set to resolve an Oracle database name in a Distributed Computing Environment (DCE) environment. **See Also:** *Oracle Database Advanced Security Administrator's Guide*
`nis` (**Network Information Service (NIS)** external naming method)	Set to resolve service information through an existing NIS. **See Also:** *Oracle Database Net Services Administrator's Guide*

Example

```
NAMES.DIRECTORY_PATH=(tnsnames, onames)
```

NAMES.NIS.META_MAP

Purpose

Use the parameter `NAMES.NIS.META_MAP` to specify the **map** file to be used to map **Network Information Service (NIS)** attributes to an NIS mapname

Default

```
sqlnet.maps
```

Example

```
NAMES.NIS.META_MAP=sqlnet.maps
```

RECV_BUF_SIZE

Purpose

Use the parameter `RECV_BUF_SIZE` to specify the buffer space limit for receive operations of sessions. This parameter is supported by the TCP/IP, TCP/IP with SSL, and SDP protocols.

Note: Additional protocols might support this parameter on certain operating systems. Refer to operating-system specific documentation for information about additional protocols that support this parameter.

See Also: *Oracle Net Services Administrator's Guide* for information about configuring this parameter

Default

The default value for this parameter is operating-system specific. The default for the Solaris 2.6 Operating System is 32768 bytes (32 KB).

Usage

You can override this parameter for a particular client connection by specifying the RECV_BUF_SIZE parameter in the connect descriptor for a client.

Example

```
RECV_BUF_SIZE=11784
```

SEND_BUF_SIZE

Purpose

Use the parameter SEND_BUF_SIZE to specify the buffer space limit for send operations of sessions. This parameter is supported by the TCP/IP, TCP/IP with SSL, and SDP protocols.

Note: Additional protocols might support this parameter on certain operating systems. Refer to operating-system specific documentation for information about additional protocols that support this parameter.

See Also: *Oracle Database Net Services Administrator's Guide* for information about configuring this parameter

Default

The default value for this parameter is operating-system specific. The default for the Solaris 2.6 Operating System is 8192 bytes (8 KB).

Usage

You can override this parameter for a particular client connection by specifying the SEND_BUF_SIZE parameter in the connect descriptor for a client.

Example

```
SEND_BUF_SIZE=11784
```

SQLNET_ALLOWED_LOGON_VERSIONS

Purpose

Use the parameter SQLNET_ALLOWED_LOGON_VERSIONS to specify which authentication protocols are allowed by the client or database. If the client and database server do not have at least one matching version, then authentication fails with an error.

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

Allowed Values

- 10 for Oracle Database 10*g* authentication protocols

- 9 for Oracle9i authentication protocols
- 8 for Oracle8 authentication protocols
- 7.3 for Oracle 7.3 authentication protocols

Any value other than 10 could expose vulnerabilities that may have existed in previous version of the authentication protocols. For complete compatibility, set the list of allowable versions for logon to include all versions of database in the system.

Default

10, 9, 8

Example

If both Oracle 8.1.7 and Oracle9i databases are present, then set the parameter as follows:

```
SQLNET_ALLOWED_LOGON_VERSIONS=(10,9,8)
```

SQLNET.AUTHENTICATION_KERBEROS5_SERVICE

Purpose

Use the parameter SQLNET.AUTHENTICATION_KERBEROS5_SERVICE to define the name of the service used to obtain a Kerberos service ticket.

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

Default

None

Example

```
SQLNET.AUTHENTICATION_KERBEROS5_SERVICE=oracle
```

SQLNET.AUTHENTICATION_SERVICES

Purpose

Use the parameter SQLNET.AUTHENTICATION_SERVICES to enable one or more authentication services. If authentication has been installed, it is recommended that this parameter be set to either none or to one of the authentication methods.

Default

None

Values

Authentication Methods Available with Oracle Net Services:

- none for no authentication methods. A valid username and password can be used to access the database.
- all for all authentication methods
- nts for **Windows NT native authentication**

Authentication Methods Available with Oracle Advanced Security:

- kerberos5 for Kerberos authentication
- radius for RADIUS authentication
- dcegssapi for DCE GSSAPI authentication

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

Example

```
SQLNET.AUTHENTICATION_SERVICES=(kerberos5)
```

SQLNET.CLIENT_REGISTRATION

Purpose

Use the parameter SQLNET.CLIENT_REGISTRATION to set a unique identifier for this client computer. This identifier is passed to the listener with any connection request and is included in the Audit Trail. The identifier can be any alphanumeric string up to 128 characters long.

Default

None

Example

```
SQLNET.CLIENT_REGISTRATION=1432
```

SQLNET.CRYPTO_CHECKSUM_CLIENT

Purpose

Use the parameter SQLNET.CRYPTO_CHECKSUM_CLIENT to specify the checksum behavior for the client.

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

Default

accepted

Values

- accepted to enable the security service if required or requested by the other side
- rejected to disable the security service, even if the required by the other side
- requested to enable the security service if the other side allows it
- required to enable the security service and disallow the connection if the other side is not enabled for the security service

Example

```
SQLNET.CRYPTO_CHECKSUM_CLIENT=accepted
```

SQLNET.CRYPTO_CHECKSUM_SERVER

Purpose

Use the parameter SQLNET.CRYPTO_CHECKSUM_SERVER to specify the checksum behavior for the database server.

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

Default

accepted

Values

- accepted to enable the security service if required or requested by the other side
- rejected to disable the security service, even if the required by the other side
- requested to enable the security service if the other side allows it
- required to enable the security service and disallow the connection if the other side is not enabled for the security service

Example

```
SQLNET.CRYPTO_CHECKSUM_SERVER=accepted
```

SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT

Purpose

Use the parameter SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT to specify a list of crypto-checksum algorithms for the client to use.

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

Default

all available algorithms

Values

- md5 for the RSA Data Security's MD5 algorithm
- sha1 for the Secure Hash algorithm

Example

```
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT=(MD5)
```

SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER

Purpose

Use the parameter SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER to specify a list of crypto-checksum algorithms for the database server to use.

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

Default

all available algorithms

Values

- `md5` for the RSA Data Security's MD5 algorithm
- `sha1` for the Secure Hash algorithm

Example

```
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER=(md5)
```

SQLNET.CRYPTO_SEED

Purpose

Use the parameter `SQLNET.CRYPTO_SEED` to specify the characters used when generating cryptographic keys. The more random the characters are, the stronger the keys are. The string should be 10-70 random characters. This optional parameter is required for when encryption or checksumming are turned on. Encryption is turned on if the SQLNET.ENCRYPTION_CLIENT parameter is specified for the client and the SQLNET.ENCRYPTION_SERVER parameter is specified for the database server; checksumming is turned on if the SQLNET.CRYPTO_CHECKSUM_CLIENT parameter is specified for the client and the SQLNET.CRYPTO_CHECKSUM_SERVER parameter is specified for the database server.

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

Default

```
qwertyuiopasdfghjkl;zxcvbnm,.s1
```

Example

```
SQLNET.CRYPTO_SEED="qwertyuiopasdfghjkl;zxcvbnm,.s1"
```

SQLNET.ENCRYPTION_CLIENT

Purpose

Use the parameter `SQLNET.ENCRYPTION_CLIENT` to turn encryption on for the client.

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

Default

```
accepted
```

Values

- `accepted` to enable the security service if required or requested by the other side
- `rejected` to disable the security service, even if the required by the other side
- `requested` to enable the security service if the other side allows it
- `required` to enable the security service and disallow the connection if the other side is not enabled for the security service

Example

```
SQLNET.ENCRYPTION_CLIENT=accepted
```

SQLNET.ENCRYPTION_SERVER

Purpose

Use the parameter SQLNET.ENCRYPTION_SERVER to turn encryption on for the database server.

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

Default

accepted

Values

- accepted to enable the security service if required or requested by the other side
- rejected to disable the security service, even if the required by the other side
- requested to enable the security service if the other side allows it
- required to enable the security service and disallow the connection if the other side is not enabled for the security service

Example

```
SQLNET.ENCRYPTION_SERVER=accepted
```

SQLNET.ENCRYPTION_TYPES_CLIENT

Purpose

Use the parameter SQLNET.ENCRYPTION_TYPES_CLIENT to specify a list of encryption algorithms for the client to use.

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

Default

All available algorithms.

Values

One or more of the following:

- 3des112 for triple DES with a two-key (112 bit) option
- 3des168 for triple DES with a three-key (168 bit) option
- des for standard 56 bit key size
- des40 for 40 bit key size
- rc4_40 for 40 bit key size
- rc4_56 for 56 bit key size
- rc4_128 for 128 bit key size
- rc4_256 for 256 bit key size

Example

```
SQLNET.ENCRYPTION_TYPES_CLIENT=(rc4_56)
```

SQLNET.ENCRYPTION_TYPES_SERVER

Purpose

Use the parameter SQLNET.ENCRYPTION_TYPES_SERVER to specify a list of encryption algorithms for the database server to use.

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

Default

All available algorithms

Values

One or more of the following:

- 3des112 for triple DES with a two-key (112 bit) option
- 3des168 for triple DES with a three-key (168 bit) option
- des for standard 56 bit key size
- des40 for 40 bit key size
- rc4_40 for 40 bit key size
- rc4_56 for 56 bit key size
- rc4_128 for 128 bit key size
- rc4_256 for 256 bit key size

Example

```
SQLNET.ENCRYPTION_TYPES_SERVER=(rc4_56, des, ...)
```

SQLNET.EXPIRE_TIME

Purpose

Use parameter SQLNET.EXPIRE_TIME to specify a the time interval, in minutes, to send a probe to verify that client/server connections are active. Setting a value greater than 0 ensures that connections are not left open indefinitely, due to an abnormal client termination. If the probe finds a terminated connection, or a connection that is no longer in use, it returns an error, causing the server process to exit. This parameter is primarily intended for the database server, which typically handles multiple connections at any one time.

Limitations on using this terminated connection detection feature are:

- It 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 degraded network performance.

Default

0

Minimum Value

0

Recommended Value

10

Example

```
SQLNET.EXPIRE_TIME=10
```

SQLNET.INBOUND_CONNECT_TIMEOUT

Purpose

Use the SQLNET.INBOUND_CONNECT_TIMEOUT parameter to specify the time, in seconds, for a client to connect with the database server and provide the necessary authentication information.

If the client fails to establish a connection and complete authentication in the time specified, then the database server terminates the connection. In addition, the database server logs the IP address of the client and an ORA-12170: TNS:Connect timeout occurred error message to the sqlnet.log file. The client receives either an ORA-12547: TNS:lost contact or an ORA-12637: Packet receive failed error message.

Without this parameter, a client connection to the database server can stay open indefinitely without authentication. Connections without authentication can introduce possible denial-of-service attacks, whereby malicious clients attempt to flood database servers with connect requests that consume resources.

To protect both the database server and the listener, Oracle Corporation recommends setting this parameter in combination with the INBOUND_CONNECT_TIMEOUT_*listener_name* parameter in the listener.ora file. When specifying values for these parameters, consider the following recommendations:

- Set both parameters to an initial low value.
- Set the value of the INBOUND_CONNECT_TIMEOUT_*listener_name* parameter to a lower value than the SQLNET.INBOUND_CONNECT_TIMEOUT parameter.

For example, you can set INBOUND_CONNECT_TIMEOUT_*listener_name* to 2 seconds and INBOUND_CONNECT_TIMEOUT parameter to 3 seconds. If clients are unable to complete connections within the specified time due to system or network delays that are normal for the particular environment, then increment the time as needed.

> **See Also:** *Oracle Net Services Administrator's Guide* for information about configuring these parameters

Default

60 seconds

Example

```
SQLNET.INBOUND_CONNECT_TIMEOUT=3
```

SQLNET.KERBEROS5_CC_NAME

Purpose

Use the parameter SQLNET.KERBEROS5_CC_NAME to specify the complete path name to the Kerberos credentials cache file.

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

Default

/usr/tmp/krbcache on UNIX operating systems and c:\tmp\krbcache on Windows operating systems

Example

```
SQLNET.KERBEROS5_CC_NAME=/usr/tmp/krbcache
```

SQLNET.KERBEROS5_CLOCKSKEW

Purpose

Use the parameter SQLNET.KERBEROS5_CLOCKSKEW to specify how many seconds can pass before a Kerberos credential is considered out of date.

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

Default

```
300
```

Example

```
SQLNET.KERBEROS5_CLOCKSKEW=1200
```

SQLNET.KERBEROS5_CONF

Purpose

Use the parameter SQLNET.KERBEROS5_CONF to specify the complete path name to the Kerberos configuration file, which contains the realm for the default Key Distribution Center (KDC) and maps realms to KDC hosts. The KDC maintains a list of user principals and is contacted through the kinit program for the user's initial ticket.

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

Default

/krb5/krb.conf on UNIX operating systems and c:\krb5\krb.conf on Windows operating systems

Example

```
SQLNET.KERBEROS5_CONF=/krb5/krb.conf
```

SQLNET.KERBEROS5_KEYTAB

Purpose

Use the parameter SQLNET.KERBEROS5_KEYTAB to specify the complete path name to the Kerberos principal/secret key mapping file, which is used to extract keys and decrypt incoming authentication information.

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

Default

/etc/v5srvtab on UNIX operating systems and c:\krb5\v5srvtab on Windows operating systems

Example

```
SQLNET.KERBEROS5_KEYTAB=/etc/v5srvtab
```

SQLNET.KERBEROS5_REALMS

Purpose

Use the parameter SQLNET.KERBEROS5_REALMS to specify the complete path name to the Kerberos realm translation file, which provides a mapping from a host name or domain name to a realm.

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

Default

/krb5/krb.realms on UNIX operating systems and c:\krb5\krb.realms on Windows operating systems

Example

```
SQLNET.KERBEROS5_REALMS=/krb5/krb.realms
```

SQLNET.RADIUS_ALTERNATE

Purpose

Use the parameter SQLNET.RADIUS_ALTERNATE to specify an alternate RADIUS server to use in case the primary server is unavailable. The value can be either the IP address or host name of the server.

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

Default

None

Example

```
SQLNET.RADIUS_ALTERNATE=radius2
```

SQLNET.RADIUS_ALTERNATE_PORT

Purpose

Use the parameter SQLNET.RADIUS_ALTERNATE_PORT to specify the listening port of the alternate RADIUS server.

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

Default

1645

Example

```
SQLNET.RADIUS_ALTERNATE_PORT=1667
```

SQLNET.RADIUS_ALTERNATE_RETRIES

Purpose

Use the parameter SQLNET.RADIUS_ALTERNATE_RETRIES to specify the number of times the database server should resend messages to the alternate RADIUS server.

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

Default

3

Example

```
SQLNET.RADIUS_ALTERNATE_RETRIES=4
```

SQLNET.RADIUS_AUTHENTICATION

Purpose

Use the parameter SQLNET.RADIUS_AUTHENTICATION to specify the location of the primary RADIUS server, either by its host name or IP address.

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

Default

Local host

Example

```
SQLNET.RADIUS_AUTHENETICATION=officeacct
```

SQLNET.RADIUS_AUTHENTICATION_INTERFACE

Purpose

Use the parameter SQLNET.RADIUS_AUTHENTICATION_INTERFACE to specify the class containing the user interface used to interact with the user.

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

Default

DefaultRadiusInterface

Example

```
SQLNET.RADIUS_AUTHENTICATION_INTERFACE=DefaultRadiusInterface
```

SQLNET.RADIUS_AUTHENTICATION_PORT

Purpose

Use the parameter SQLNET.RADIUS_AUTHENTICATION_PORT to specify the listening port of the primary RADIUS server.

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

Default

1645

Example

```
SQLNET.RADIUS_AUTHENTICATION_PORT= 1667
```

SQLNET.RADIUS_AUTHENTICATION_RETRIES

Purpose

Use the parameter SQLNET.RADIUS_AUTHENTICATION_RETRIES to specify the number of times the database server should resend messages to the primary RADIUS server.

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

Default

3

Example

```
SQLNET.RADIUS_AUTHENTICATION_RETRIES=4
```

SQLNET.RADIUS_AUTHENTICATION_TIMEOUT

Purpose

Use the parameter SQLNET.RADIUS_AUTHENTICATION_TIMEOUT to specify the time, in seconds, that the database server should wait for a response from the primary RADIUS server.

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

Default

5

Example

```
SQLNET.RADIUS_AUTHENTICATION_TIMEOUT=10
```

SQLNET.RADIUS_CHALLENGE_RESPONSE

Purpose

Use the parameter SQLNET.RADIUS_CHALLENGE_RESPONSE to turn challenge response on or off.

Default

off

Values

on | off

Example

```
SQLNET.RADIUS_CHALLENGE_RESPONSE=on
```

SQLNET.RADIUS_SECRET

Purpose:

Use the parameter SQLNET.RADIUS_SECRET to specify the location of the RADIUS secret key.

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

Default

The $ORACLE_HOME/network/security/radius.key file on UNIX operating systems and the *ORACLE_HOME*\network\security\radius.key file on Windows

Example

```
SQLNET.RADIUS_SECRET=oracle/bin/admin/radiuskey
```

SQLNET.RADIUS_SEND_ACCOUNTING

Purpose

Use the parameter SQLNET.RADIUS_SEND_ACCOUNTING to turn accounting on and off. If enabled, packets are sent to the active RADIUS server at listening port plus one. The default port is 1646.

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

Default

off

Values

on | off

Example

```
SQLNET.RADIUS_SEND_ACCOUNTING=on
```

SQLNET.RECV_TIMEOUT

Purpose

Use the parameter SQLNET.RECV_TIMEOUT to specify the time, in seconds, for a database server to wait for client data after connection establishment. A client must send some data within the time interval.

For environments in which clients shut down on occasion or abnormally, setting this parameter is recommended. If a client does not send any data in time specified, then the database server logs an ORA-12535: TNS:operation timed out and ORA-12609: TNS: Receive timeout occurred to the sqlnet.log file. Without this parameter, the database server may continue to wait for data from clients that may be down or are experiencing difficulties.

You can also set this parameter on the clientside to specify the time, in seconds, for a client to wait for response data from the database server after connection establishment. Without this parameter, the client may wait for a long period of time for a response from a database server saturated with requests.

Set the value for this parameter to an initial low value and adjust according to system and network capacity. If necessary, use this parameter in conjunction with the SQLNET.SEND_TIMEOUT parameter.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for information about configuring these parameters

Default

None

Example

```
SQLNET.RECV_TIMEOUT=3
```

SQLNET.SEND_TIMEOUT

Purpose

Use to specify the time, in seconds, for a database server to complete a send operation to clients after connection establishment.

For environments in which clients shut down on occasion or abnormally, setting this parameter is recommended. If the database server is unable to complete a send operation in the time specified, then it logs an ORA-12535: TNS:operation timed out and ORA-12608: TNS: Send timeout occurred to the sqlnet.log file. Without this parameter, the database server may continue to send responses to clients that are unable to receive data due to a downed computer or a busy state.

You can also set this parameter on the clientside to specify the time, in seconds, for a client to complete send operations to the database server after connection establishment. Without this parameter, the client may continue to send requests to a database server already saturated with requests.

Set the value for this parameter to an initial low value and adjust according to system and network capacity. If necessary, use this parameter in conjunction with the SQLNET.RECV_TIMEOUT parameter.

See Also: *Oracle Database Net Services Administrator's Guide* for information about configuring these parameters

Default

None

Example

```
SQLNET.SEND_TIMEOUT=3
```

SSL_CERT_REVOCATION

Purpose

Use the `SSL_CRT_REVOCATION` parameter to configure a revocation check for a certificate.

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

Default

`none`

Values

- `none` to turn off certificate revocation checking
- `requested` to perform certificate revocation in case a Certificate Revocation List (CRL) is available. Reject SSL connection if the certificate is revoked. If no appropriate CRL is found to determine the revocation status of the certificate and the certificate is not revoked, then accept the SSL connection
- `required` to perform certificate revocation when a certificate is available. If a certificate is revoked and no appropriate CRL is found, then reject the SSL connection If no appropriate CRL is found to ascertain the revocation status of the certificate and the certificate is not revoked. then accept the SSL connection.

Example

```
SSL_CERT_REVOCATION=required
```

SSL_CERT_FILE

Purpose

Use the parameter `SSL_CRL_FILE` to specify the name of the file where you can assemble the CRL of CAs for client authentication.

This file contains the PEM-encoded CRL files, in order of preference. You can use this file alternatively or in additional to the SSL_CERT_PATH parameter. This parameter is only valid if SSL_CERT_REVOCATION is set to either `requested` or `required`.

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

Default

None

SSL_CERT_PATH

Purpose

Use the parameter SSL_CRL_PATH to specify the destination directory of the CRL of CA. The files in this directory are hashed symbolic links created by Oracle Wallet Manager. This parameter is only valid if SSL_CERT_REVOCATION is set to either requested or required.

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

Default

None

Default

None

SSL_CIPHER_SUITES

Purpose

Use the parameter SSL_CIPHER_SUITES to control what combination of encryption and data integrity is used by the **Secure Sockets Layer (SSL)**.

Default

None

Values

See Also: *Oracle Advanced Security Administrator's Guide* for further information about cipher suite values

Example

```
SSL_CIPHER_SUITE=(ssl_rsa_with_rc4_138_md5)
```

SSL_CLIENT_AUTHENTICATION

Purpose

Use the parameter SSL_CLIENT_AUTHENTICATION to specify whether or not a client—in addition to the database server—is authenticated using SSL.

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

Default

```
true
```

Values

```
true | false
```

Example

```
SSL_CLIENT_AUTHENTICATION=true
```

SSL_SERVER_DN_MATCH

Purpose

Use the parameter SSL_SERVER_DN_MATCH to enforce that the **distinguished name (DN)** for the database server matches its service name. If you enforce the match verifications, then SSL ensures that the certificate is from the server. If you select to not enforce the match verification, then SSL performs the check but allows the connection, regardless if there is a match. Not enforcing the match allows the server to potentially fake its identify.

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

Default

no

Values

- yes | on | true to specify to enforce a match. If the DN matches the service name, then the connection succeeds. If the DN does not match the service name, then the connection fails.
- no | off | false to specify to not enforce a match. If does not match the service name, then the connection is successful, but an error is logged to the sqlnet.log file.

Usage Notes

In addition to the sqlnet.ora file, configure the tnsnames.ora parameter SSL_SERVER_CERT_DN to enable server DN matching.

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

Example

```
SSL_SERVER_DN_MATCH=yes
```

SSL_VERSION

Purpose

Use the parameter SSL_VERSION to force the version of the SSL connection.

Clients and database servers must use a compatible version.

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

Default

undetermined

Values

undetermined | 2.0 | 3.0

Example

```
SSL_VERSION=2.0
```

TCP.EXCLUDED_NODES

Purpose

Use the parameter TCP.EXCLUDED_NODES to specify which clients are denied access to the database.

Syntax

TCP.EXCLUDED_NODES=(*hostname* | *ip_address*, *hostname* | *ip_address*, ...)

Example

```
TCP.EXCLUDED_NODES=(finance.us.acme.com, mktg.us.acme.com, 144.25.5.25)
```

TCP.INVITED_NODES

Purpose

Use the parameter TCP.INVITED_NODES to specify which clients are allowed access to the database. This list takes precedence over the TCP.EXCLUDED_NODES parameter if both lists are present.

Syntax

TCP.INVITED_NODES=(*hostname* | *ip_address*, *hostname* | *ip_address*, ...)

Example

```
TCP.INVITED_NODES=(sales.us.acme.com, hr.us.acme.com, 144.185.5.73)
```

TCP.VALIDNODE_CHECKING

Purpose

Use the parameter TCP.VALIDNODE_CHECKING to check for the TCP.INVITED_NODES and TCP.EXCLUDED_NODES to determine which clients to allow or deny access.

Default

no

Values

yes | no

Example

```
TCP.VALIDNODE_CHECKING=yes
```

TCP.NODELAY

Purpose

Use the parameter TCP.NODELAY to preempt delays in buffer flushing within the TCP/IP protocol stack.

Default

yes

Values

yes | no

Example

TCP.NODELAY=yes

TNSPING.TRACE_DIRECTORY

Purpose

Use the parameter TNSPING.TRACE_DIRECTORY to specify the destination directory for the TNSPING utility trace file, tnsping.trc.

Default

The $ORACLE_HOME/network/trace directory on UNIX operating systems and the *%ORACLE_HOME%*\network\trace directory on Windows operating systems

Example

TNSPING.TRACE_DIRECTORY=/oracle/traces

TNSPING.TRACE_LEVEL

Purpose

Use the parameter TNSPING.TRACE_LEVEL to turn TNSPING utility tracing on, at a specific level, or off.

Default

off

Values

- off for no trace output
- user for user trace information
- admin for administration trace information
- support for Oracle Support Services trace information

Example

TNSPING.TRACE_LEVEL=admin

TRACE_DIRECTORY_CLIENT

Purpose

Use the parameter TRACE_DIRECTORY_CLIENT to specify the destination directory for the client trace file.

Default

The $ORACLE_HOME/network/trace directory on UNIX operating systems and the *ORACLE_HOME*\network\trace directory on Windows operating systems

Example

```
TRACE_DIRECTORY_CLIENT=/oracle/traces
```

TRACE_DIRECTORY_SERVER

Purpose

Use the parameter TRACE_DIRECTORY_SERVER to specify the destination directory for the database server trace file.

Default

The $ORACLE_HOME/network/trace directory on UNIX operating systems and the *%ORACLE_HOME%*\network\trace directory on Windows

Example

```
TRACE_DIRECTORY_SERVER=/oracle/traces
```

TRACE_FILE_CLIENT

Purpose

Use the parameter TRACE_FILE_CLIENT to specify the name of the client trace file.

Default

```
sqlnet.trc
```

Example

```
TRACE_FILE_CLIENT=clientsqlnet.trc
```

TRACE_FILE_SERVER

Purpose

Use the parameter TRACE_FILE_SERVER to specify the name of the database server trace file

Default

svr_*pid*.trc

Example

```
TRACE_FILE_SERVER=svrsqlnet.trc
```

TRACE_FILELEN_CLIENT

Purpose

Use the parameter TRACE_FILELEN_CLIENT to specify the size of the client trace files in kilobytes (KB). When the size is met, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO_CLIENT parameter.

Example

```
TRACE_FILELEN_CLIENT=100
```

TRACE_FILELEN_SERVER

Purpose

Use the parameter TRACE_FILELEN_SERVER to specify the size of the database server trace files in kilobytes (KB). When the size is met, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO_SERVER parameter.

Example

```
TRACE_FILELEN_SERVER=100
```

TRACE_FILENO_CLIENT

Purpose

Use the parameter TRACE_FILENO_CLIENT to specify the number of trace files for client tracing. When this parameter is set along with the TRACE_FILELEN_CLIENT parameter, trace files are used in a cyclical fashion. The first file is filled first, then the second file, and so on. When the last file has been filled, the first file is re-used, and so on.

The trace file names are distinguished from one another by their sequence number. For example, if the default trace file of `sqlnet.trc` is used, and this parameter is set to 3, the trace files would be named `sqlnet1.trc`, `sqlnet2.trc` and `sqlnet3.trc`.

In addition, trace events in the trace files are preceded by the sequence number of the file.

Default

None

Example

```
TRACE_FILENO_CLIENT=3
```

TRACE_FILENO_SERVER

Purpose

Use the parameter TRACE_FILENO_SERVER to specify the number of trace files for database server tracing. When this parameter is set along with the TRACE_FILELEN_SERVER parameter, trace files are used in a cyclical fashion. The first file is filled first, then the second file, and so on. When the last file has been filled, the first file is re-used, and so on.

The trace file names are distinguished from one another by their sequence number. For example, if the default trace file of `svr_pid.trc` is used, and this parameter is set to 3, the trace files would be named `svr1_pid.trc`, `svr2_pid.trc` and `svr3_pid.trc`.

In addition, trace events in the trace files are preceded by the sequence number of the file.

Default

None

Example

```
TRACE_FILENO_SERVER=3
```

TRACE_LEVEL_CLIENT

Purpose

Use the parameter TRACE_LEVEL_CLIENT to turn client tracing on, at a specific level, or off.

Default

```
off
```

Values

- `off` for no trace output
- `user` for user trace information
- `admin` for administration trace information
- `support` for Oracle Support Services trace information

Example

```
TRACE_LEVEL_CLIENT=user
```

TRACE_LEVEL_SERVER

Purpose

Use the parameter TRACE_LEVEL_SERVER to turn server tracing on, at a specific level, or off.

Default

```
off
```

Values

- `off` for no trace output
- `user` for user trace information
- `admin` for administration trace information
- `support` for Oracle Support Services trace information

Example

```
TRACE_LEVEL_SERVER=admin
```

TRACE_TIMESTAMP_CLIENT

Purpose

Use the parameter TRACE_TIMESTAMP_CLIENT to add a time stamp in the form of *dd-mon-yyyy hh:mi:ss:mil* to every trace event in the client trace file, which has a default name of `sqlnet.trc`.

Default

on

Values

on or true | off or false

Example

```
TRACE_TIMESTAMP_SERVER=true
```

TRACE_TIMESTAMP_SERVER

Purpose

Use the parameter TRACE_TIMESTAMP_SERVER to add a time stamp in form of *dd-mon-yyyy hh:mi:ss:mil* to every trace event in the database server trace file, which has a default name of svr_*pid*.trc.

Default

off

Values

on or true | off or false

Example

```
TRACE_TIMESTAMP_SERVER=true
```

TRACE_UNIQUE_CLIENT

Purpose

Use the parameter TRACE_UNIQUE_CLIENT to specify whether or not a unique trace file is created for each client trace session. When the value is set to on, a process identifier is appended to the name of each trace file, enabling several files to coexist. For example, trace files named sqlnet*pid*.trc are created if default trace file name sqlnet.trc is used. When the value is set to off, data from a new client trace session overwrites the existing file.

Default

on

Values

on or off

Example

```
TRACE_UNIQUE_CLIENT=on
```

USE_CMAN

Purpose

If set to true, the parameter USE_CMAN routes the client to a protocol address for an Oracle Connection Manager.

The following example shows two address lists. While the first address list routes the client to an Oracle Connection Manager, the second address list routes the client directly to a listener.

```
sales=
 (DESCRIPTION=
   (LOAD_BALANCE=on)
   (FAILOVER=on)
   (ADDRESS_LIST=
     (SOURCE_ROUTE=yes)
     (ADDRESS=(PROTOCOL=tcp)(HOST=host1)(PORT=1630))
     (ADDRESS=(PROTOCOL=tcp)(HOST=host2)(PORT=1521)))
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=host3)(PORT=1521)))
   (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)))
```

Without `USE_CMAN=true`, the client picks one of the address lists at random and fails over to the other address list if the chosen `ADDRESS_LIST` fails. With `USE_CMAN=true`, the client always uses the first address list.

If no Oracle Connection Manager addresses are available, connections are routed through any available listener address.

Default

`false`

Values

`true | false`

Example

```
USE_CMAN=true
```

USE_DEDICATED_SERVER

Purpose

If set to `on`, the parameter `USE_DEDICATED_SERVER` automatically appends `(SERVER=dedicated)` to the connect data for a connect descriptor. This way connections from this client use a **dedicated server** process, even if **shared server** is configured.

This parameter adds `(SERVER=dedicated)` to the `CONNECT_DATA` section of the connect descriptor used by the client. It overrides the current value of the SERVER parameter in the `tnsnames.ora` file.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for complete configuration information

Default

`off`

Values

- `on` to append `(SERVER=dedicated)`
- `off` to hand off requests to existing server processes

Example

```
USE_DEDICATED_SERVER=on
```

WALLET_LOCATION

Purpose

Use the parameter WALLET_LOCATION to specify the location of wallets. Wallets are certificates, keys, and trustpoints processed by SSL.

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

Syntax

Oracle wallets on the file system:

```
WALLET_LOCATION=
  (SOURCE=
    (METHOD=file)
    (METHOD_DATA=
       (DIRECTORY=directory)
       [(PKCS11=TRUE/FALSE)]))
```

Microsoft certificate store:

```
WALLET_LOCATION=
  (SOURCE=
     (METHOD=mcs))
```

Oracle wallets in the Windows registry:

```
WALLET_LOCATION=
   (SOURCE=
      (METHOD=reg)
      (METHOD_DATA=
         (KEY=registry_key)))
```

Entrust wallets:

```
WALLET_LOCATION=
   (SOURCE=
      (METHOD=entr)
      (METHOD_DATA=
         (PROFILE=file.epf)
         (INIFILE=file.ini)))
```

Subparameters

WALLET_LOCATION supports the following subparameters:

SOURCE: Specify the type of storage for wallets and storage location.

METHOD: Specify the type of storage.

METHOD_DATA: Specify the storage location.

DIRECTORY: Specify the location of Oracle wallets on file system.

KEY: Specify the wallet type and location in the Windows registry.

PROFILE: Specify the Entrust profile file (.epf).

INIFILE: Specify the Entrust initialization file (.ini).

Default

None

Usage Notes

- The key/value pair for Microsoft's certificate store (MCS) omits the METHOD_DATA parameter because MCS does not use wallets. Instead, Oracle PKI (public key infrastructure) applications obtain certificates, trustpoints and private keys directly from the user's profile.
- If an Oracle wallet is stored in the Windows registry and the wallet's key (KEY) is SALESAPP, the storage location of the encrypted wallet is HKEY_CURRENT_USER\SOFTWARE\ORACLE\WALLETS\SALESAPP\EWALLET.P12. The storage location of the decrypted wallet is HKEY_CURRENT_USER\SOFTWARE\ORACLE\WALLETS\SALESAPP\CWALLET.SSO.

Values

true | false

Examples

Oracle wallets on file system:

```
WALLET_LOCATION=
  (SOURCE=
      (METHOD=file)
      (METHOD_DATA=
         (DIRECTORY=/etc/oracle/wallets/databases)))
```

Microsoft certificate store:

```
WALLET_LOCATION=
   (SOURCE=
     (METHOD=mcs))
```

Oracle Wallets in the Windows registry:

```
WALLET_LOCATION=
   (SOURCE=
     (METHOD=REG)
     (METHOD_DATA=
        (KEY=SALESAPP)))
```

Entrust Wallets:

```
WALLET_LOCATION=
   (SOURCE=
     (METHOD=entr)
     (METHOD_DATA=
       (PROFILE=/etc/oracle/wallets/test.epf)
       (INIFILE=/etc/oracle/wallets/test.ini)))
```

WALLET_OVERRIDE

Purpose

This parameter determines whether the client should override the strong authentication credential with the password credential in the secret store to log into the database.

Syntax

None.

Subparameters

None.

Usage Notes

- Users may have batch jobs that require logging into the database. There may be scripts that access databases and are shared by administrators. This project provides a way for them to use `connect /` instead of specifying `username/password` explicitly. It simplifies the maintenance of the scripts and secures the password management for the applications.
- Middle-tier applications create an Oracle Applications wallet at install time to store the application's specific identity. The password may be randomly generated rather than hardcoded. When an Oracle application accesses the database, it sets appropriate values for `SQLNET.AUTHENTICATION_SERVICES` and `WALLET_LOCATION`. The new wallet-based password authentication code uses the password credential in the Oracle Applications wallet to log on to the database.

Examples

New commands will be implemented for mkstore to manage the entries in the secret store.

To create a wallet:

```
mkstore -wrl <wallet location> -create
```

To create an entry:

```
mkstore -wrl <wallet location> -createCredential <alias> <username> <password>
```

To modify an entry:

```
mkstore -wrl <wallet location> -modifyCredential <alias> <username> <password>
```

To delete an entry:

```
mkstore -wrl <wallet location> -deleteCredential <alias>
```

To list all entries:

```
mkstore -wrl <wallet location> -listCredential
```

6

Local Naming Parameters (tnsnames.ora)

This chapter provides a complete listing of the `tnsnames.ora` file configuration parameters.

This chapter contains these topics:

Overview of Local Naming Parameters

This `tnsnames.ora` file is a configuration file that contains **net service name**s mapped to **connect descriptor**s for the **local naming** method, or net service names mapped to listener **protocol address**es.

A net service name is an alias mapped to a database network address contained in a connect descriptor. A connect descriptor contains the location of the listener through a protocol address and the service name of the database to which to connect. Clients and database servers (that are clients of other database servers) use the net service name when making a connection with an application.

By default, `tnsnames.ora` is located in the `$ORACLE_HOME/network/admin` directory on UNIX operating systems and in the *`ORACLE_HOME`*`\network\admin` directory on Windows operating systems. `tnsnames.ora` can also be stored the following locations:

- The directory specified by the `TNS_ADMIN` environment variable or registry value
- On UNIX operating systems, the global configuration directory. For example, on the Solaris Operating System, this directory is `/var/opt/oracle`.

See Also: Oracle operating system-specific documentation

General Syntax of tnsnames.ora

The basic syntax for a `tnsnames.ora` file is shown in Figure 6–1. `DESCRIPTION` contains the connect descriptor, `ADDRESS` contains the protocol address, and `CONNECT_DATA` contains the database service identification information.

Example 6–1 Basic Format of tnsnames.ora File

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS=(protocol_address_information))
   (CONNECT_DATA=
     (SERVICE_NAME=service_name)))
```

Multiple Descriptions in tnsnames.ora

A `tnsnames.ora` file can contain net service names with one or more connect descriptors. Each connect descriptor can contain one or more protocol addresses. Example 6–2 shows two connect descriptors with multiple addresses. `DESCRIPTION_LIST` defines a list of connect descriptors.

> **Note:** Oracle Net Manager does not support the creation of multiple connect descriptors for a net service name.

Example 6–2 Net Service Name with Multiple Connect Descriptors in tnsnames.ora

```
net_service_name=
 (DESCRIPTION_LIST=
  (DESCRIPTION=
   (ADDRESS=(protocol_address_information))
   (ADDRESS=(protocol_address_information))
   (ADDRESS=(protocol_address_information))
   (CONNECT_DATA=
     (SERVICE_NAME=service_name)))
  (DESCRIPTION=
   (ADDRESS=(protocol_address_information))
   (ADDRESS=(protocol_address_information))
   (ADDRESS=(protocol_address_information))
   (CONNECT_DATA=
     (SERVICE_NAME=service_name))))
```

Multiple Address Lists in tnsnames.ora

The `tnsnames.ora` file also supports connect descriptors with multiple lists of addresses, each with its own characteristics. In Figure 6–3, two address lists are presented. The first address list features **client load balancing** and no **connect-time failover**, affecting only those protocol adresses within the `ADDRESS_LIST`. The second protocol address list features connect-time failover and no client load loading balancing, affecting only those protocol addresses within the `ADDRESS_LIST`. The client first tries either the first or second protocol address at random, then tries protocol addresses three and four sequentially.

> **Note:** **Oracle Net Manager** supports only the creation of one protocol address list for a connect descriptor.

Example 6–3 Multiple Address Lists in tnsnames.ora

```
net_service_name=
 (DESCRIPTION=
  (ADDRESS_LIST=
   (LOAD_BALANCE=on)
   (FAILOVER=off)
```

```
 (ADDRESS=(protocol_address_information))
 (ADDRESS=(protocol_address_information)))
(ADDRESS_LIST=
 (LOAD_BALANCE=off)
 (FAILOVER=on)
 (ADDRESS=(protocol_address_information))
 (ADDRESS=(protocol_address_information)))
(CONNECT_DATA=
 (SERVICE_NAME=service_name)))
```

Note: Protocol address lists do not have to be embedded in an ADDRESS_LIST if there is only one list, as was the case prior to release 8.1.

Connect-Time Failover and Client Load Balancing with Oracle Connection Managers

When a connect descriptor in a tnsnames.ora file contains at least two protocol addresses for **Oracle Connection Manager**, parameters for connect-time failover and load balancing can be included in the file.

Figure 6–4 illustrates failover of multiple Oracle Connection Manager protocol addresses.

Example 6–4 Multiple Oracle Connection Manager Addresses in tnsnames.ora

```
sample1=
 (DESCRIPTION=
   (SOURCE_ROUTE=yes)
   (ADDRESS=(PROTOCOL=tcp)(HOST=host1)(PORT=1630))     # hop 1
   (ADDRESS_LIST=
     (FAILOVER=on)
     (LOAD_BALANCE=off)                                 # hop 2
     (ADDRESS=(PROTOCOL=tcp)(HOST=host2a)(PORT=1630))
     (ADDRESS=(PROTOCOL=tcp)(HOST=host2b)(PORT=1630)))
   (ADDRESS=(PROTOCOL=tcp)(HOST=host3)(PORT=1521))     # hop 3
   (CONNECT_DATA=(SERVICE_NAME=Sales.us.acme.com)))
```

In Figure 6–4:

1. The client is instructed to connect to an protocol address of the first Oracle Connection Manager, as indicated by:

   ```
   (ADDRESS=(PROTOCOL=tcp)(HOST=host1)(PORT=1630))
   ```

2. The first Oracle Connection Manager is then instructed to connect to the first protocol address of another Oracle Connection Manager. If the first protocol address fails, then it tries the second protocol address. This sequence is specified with the following configuration:

   ```
   (ADDRESS_LIST=
     (FAILOVER=ON)
     (LOAD_BALANCE=off)
     (ADDRESS=(PROTOCOL=tcp)(HOST=host2a)(PORT=1630))
     (ADDRESS=(PROTOCOL=tcp)(HOST=host2b)(PORT=1630))
   ```

3. The Oracle Connection Manager then connects to the database service using the following protocol address:

```
    (ADDRESS=(PROTOCOL=tcp)(HOST=host3)(PORT=1521))
```

Figure 6–5 illustrates client load balancing among two Oracle Connection Managers and two protocol addresses:

Example 6–5 Client Load Balancing in tnsnames.ora

```
sample2=
 (DESCRIPTION=
   (LOAD_BALANCE=on)
   (FAILOVER=on)
   (ADDRESS_LIST=
     (SOURCE_ROUTE=yes)
     (ADDRESS=(PROTOCOL=tcp)(HOST=host1)(PORT=1630))
     (ADDRESS=(PROTOCOL=tcp)(HOST=host2)(PORT=1521)))
  (ADDRESS_LIST=
     (SOURCE_ROUTE=yes)
     (ADDRESS=(PROTOCOL=tcp)(HOST=host3)(port=1630))
     (ADDRESS=(PROTOCOL=tcp)(HOST=host4)(port=1521)))
   (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)))
```

In Figure 6–5:

1. The client is instructed to pick an ADDRESS_LIST at random and to failover to the other if the chosen ADDRESS_LIST fails. This is indicated by the LOAD_BALANCE and FAILOVER parameters being set to on.
2. When an ADDRESS_LIST is chosen, the client first connects to the Oracle Connection Manager, using the Oracle Connection Manager protocol address that uses port 1630 indicated for the ADDRESS_LIST.
3. The Oracle Connection Manager then connects to the database service, using the protocol address indicated for the ADDRESS_LIST.

Local Naming Parameters

This section lists and describes the tnsnames.ora file parameters that comprise connect descriptors. Configuration parameters fall into the following categories:

- Connect Descriptor Descriptions
- Protocol Address Section
- Optional Parameters for Lists
- Connect Data Section
- Security Section

Connect Descriptor Descriptions

Each connect descriptor is contained within a DESCRIPTION parameter. Multiple connect descriptors are characterized by the DESCRIPTION_LIST parameter. These parameters are described next.

DESCRIPTION

Purpose

Use the DESCRIPTION parameter as a container for a connect descriptor.

Embed this parameter under the DESCRIPTION_LIST parameter.

Example

```
net_service_name=
(DESCRIPTION=
  (ADDRESS=...)
  (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)))
```

DESCRIPTION_LIST

Purpose

Use he DESCRIPTION_LIST parameter defines a list of connect descriptors for a particular net service name.

Example

```
net_service_name=
(DESCRIPTION_LIST=
 (DESCRIPTION=
  (ADDRESS=...)
  (CONNECT_DATA=(SERVICE_NAME=sales.acme.com)))
 (DESCRIPTION=
  (ADDRESS=...)
  (CONNECT_DATA=(SERVICE_NAME=sales2.us.acme.com))))
```

Protocol Address Section

The protocol address section of the tnsnames.ora file specifies the protocol addresses of the listener.

ADDRESS

Purpose

Use the parameter ADDRESS to define a single listener protocol address.

Embed this parameter under either the ADDRESS_LIST parameter or the DESCRIPTION parameter.

> **See Also:** Chapter 4 for descriptions of the correct parameters to use for each protocol

Example

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

ADDRESS_LIST

Purpose

Use the parameter ADDRESS_LIST to define a list of protocol addresses. If there is only address list, ADDRESS_LIST is not necessary.

Embed this parameter under either the DESCRIPTION parameter or the DESCRIPTION_LIST parameter.

Example

```
net_service_name=
 (DESCRIPTION=
  (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521)))
  (ADDRESS_LIST=
   (FAILOVER=on)
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)))
```

Optional Parameters for Lists

For multiple addresses, the following parameters are available for usage:

- FAILOVER
- LOAD_BALANCE
- RECV_BUF_SIZE
- SDU
- SEND_BUF_SIZE
- SOURCE_ROUTE
- TYPE_OF_SERVICE

FAILOVER

Purpose

Use the parameter FAILOVER to enable or disable connect-time failover for multiple protocol addresses.

When you set the parameter to on, yes, or true, Oracle Net, at connect time, fails over to a different address if the first protocol address fails. When you set the parameter to off, no, or false, Oracle Net tries one protocol address.

Embed this parameter under either the DESCRIPTION_LIST parameter, the DESCRIPTION parameter, or the ADDRESS_LIST parameter.

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 connect-time failover.

Default

on for DESCRIPTION_LISTs, DESCRIPTIONs, and ADDRESS_LISTs

Values

on | off | yes | no | true | false

Example

```
net_service_name=
 (DESCRIPTION=
  (FAILOVER=on)
```

```
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)))
```

LOAD_BALANCE

Purpose

Use the parameter LOAD_BALANCE to enable or disable client load balancing for multiple protocol addresses.

When you set the parameter to on, yes, or true, Oracle Net progresses through the list of addresses in a random sequence, balancing the load on the various listener or Oracle Connection Manager protocol addresses. When you set the parameter to off, no, or false, Oracle Net tries the protocol addresses sequentially until one succeeds.

Embed this parameter under either the DESCRIPTION_LIST parameter, the DESCRIPTION parameter, or the ADDRESS_LIST parameter.

Default

on for DESCRIPTION_LISTs

Values

on | off | yes | no | true | false

Example

```
net_service_name=
 (DESCRIPTION=
  (LOAD_BALANCE=on)
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com))
```

RECV_BUF_SIZE

Purpose

Use the parameter RECV_BUF_SIZE to specify, in bytes, the buffer space for receive operations of sessions. This parameter is supported by the TCP/IP, TCP/IP with SSL, and SDP protocols.

> **Note:** Additional protocols might support this parameter on certain operating systems. Refer to operating-system specific documentation for information about additional protocols that support this parameter.

Embed this parameter under the DESCRIPTION parameter or at the end of the protocol address.

Default

The default value for this parameter is operating-system specific. The default for the Solaris 2.6 Operating System is 32768 bytes.

Usage

Setting this parameter in the connect descriptor for a client overrides the RECV_BUF_SIZE parameter at the clientside `sqlnet.ora` file.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for information about configuring this parameter

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521)
        (RECV_BUF_SIZE=11784))
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)
        (RECV_BUF_SIZE=11784))
   (CONNECT_DATA=
     (SERVICE_NAME=sales.us.acme.com)))
net_service_name=
 (DESCRIPTION=
   (RECV_BUF_SIZE=11784)
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=hr1-server)(PORT=1521)
     (ADDRESS=(PROTOCOL=tcp)(HOST=hr2-server)(PORT=1521))
   (CONNECT_DATA=
     (SERVICE_NAME=hr.us.acme.com)))
```

SDU

Purpose

Use the parameter `SDU` to instruct Oracle Net to optimize the transfer rate of data packets being sent across the network with the **session data unit (SDU)** size you specify.

Embed this parameter under the `DESCRIPTION` parameter.

Default

2048 bytes (2KB)

Values

512 bytes to 32768 (32 KB)

Usage

Setting this parameter in the connect descriptor for a client overrides the DEFAULT_SDU_SIZE parameter at clientside `sqlnet.ora` file.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for complete SDU usage and configuration information

Example

```
net_service_name=
 (DESCRIPTION=
  (SDU=2085)
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521))
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
```

```
  (CONNECT_DATA=
    (SERVER_NAME=sales.us.acme.com))
```

SEND_BUF_SIZE

Purpose

Use the parameter SEND_BUF_SIZE to specify, in bytes, the buffer space for send operations of sessions. This parameter is supported by the TCP/IP, TCP/IP with SSL, and SDP protocols.

> **Note:** Additional protocols might support this parameter on certain operating systems. Refer to operating-system specific documentation for information about additional protocols that support this parameter.

Embed this parameter under the DESCRIPTION parameter or at the end of the protocol address.

Default

The default value for this parameter is operating-system specific. The default for the Solaris 2.6 Operating System is 8192 bytes.

Usage

Setting this parameter in the connect descriptor for a client overrides the SEND_BUF_SIZE parameter at the clientside sqlnet.ora file.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for information about configuring this parameter

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521)
        (SEND_BUF_SIZE=11784))
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)
        (SEND_BUF_SIZE=11784))
   (CONNECT_DATA=
     (SERVICE_NAME=sales.us.acme.com)))
net_service_name=
 (DESCRIPTION=
   (SEND_BUF_SIZE=11784)
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=hr1-server)(PORT=1521)
     (ADDRESS=(PROTOCOL=tcp)(HOST=hr2-server)(PORT=1521))
   (CONNECT_DATA=
     (SERVICE_NAME=hr.us.acme.com)))
```

SOURCE_ROUTE

Purpose

Use the parameter SOURCE_ROUTE to enable routing through multiple protocol addresses.

When you set to on or yes, Oracle Net uses each address in order until the destination is reached.

To use Oracle Connection Manager, an initial connection from the client to Oracle Connection Manager is required, and a second connection from Oracle Connection Manager to the listener is required.

Embed this parameter under either the DESCRIPTION_LIST parameter, the DESCRIPTION parameter, or the ADDRESS_LIST parameter.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for complete configuration information

Default

off

Values

yes | no | on | off

Example

```
net_service_name=
 (DESCRIPTION=
  (SOURCE_ROUTE=on)
  (ADDRESS=(PROTOCOL=tcp)(HOST=cman-pc)(PORT=1630))
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com))
```

TYPE_OF_SERVICE

Purpose

Use the parameter TYPE_OF_SERVICE parameter to specify the type of service to use for an Oracle Rdb database. This parameter should only be used if the application supports both an Oracle Rdb and Oracle database service, and you want the application to load balance between the two.

Embed this parameter under the DESCRIPTION parameter.

Example

```
net_service_name=
 (DESCRIPTION_LIST=
  (DESCRIPTION=
   (ADDRESS=...)
   (CONNECT_DATA=
    (SERVICE_NAME=generic)
    (RDB_DATABASE=[.mf]mf_personal.rdb)
    (GLOBAL_NAME=alpha5))
   (TYPE_OF_SERVICE=rdb_database))
  (DESCRIPTION=
   (ADDRESS=...)
   (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com))
   (TYPE_OF_SERVICE=oracle9_database)))
```

Connect Data Section

The connection data section of the tnsnames.ora file specifies the name of the destination service.

CONNECT_DATA

Purpose

Use the parameter CONNECT_DATA to define the service to which to connect.

Embed this parameter under the DESCRIPTION parameter.

Usage Notes

CONNECT_DATA permits the following subparameters:

- FAILOVER_MODE
- GLOBAL_NAME
- HS
- INSTANCE_NAME
- RDB_DATABASE
- SERVER
- SERVICE_NAME
- SID

Example

```
net_service_name=
 (DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521))
  (CONNECT_DATA=
   (SERVICE_NAME=sales.us.acme.com)))
```

FAILOVER_MODE

Purpose

Use the parameter FAILOVER_MODE to instruct Oracle Net to fail over to a different listener if the first listener fails during runtime. Depending upon the configuration, session or any SELECT statements which were in progress are automatically failed over.

This type of failover is called **Transparent Application Failover (TAF)** and should not be confused with the connect-time failover FAILOVER parameter.

Embed this parameter under the CONNECT_DATA parameter.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for complete configuration information

Subparameters

FAILOVER_MODE supports the following subparameters:

BACKUP: Specify the failover node by its net service name. A separate net service name must be created for the failover node.

TYPE: Specify the type of failover. Three types of Oracle Net failover functionality are available by default to **Oracle Call Interface (OCI)** applications:

session: Fails over the session; that is, 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: Allows 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, in which no failover functionality is used. This can also be explicitly specified to prevent failover from happening.

METHOD: Specify how fast failover is to occur from the primary node to the backup node:

basic: Establishes connections at failover time. This option requires almost no work on the backup database server until failover time.

preconnect: Pre-establishes 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: If a callback function is registered, then RETRIES and DELAY subparameters are ignored.

Example

See Also: *Oracle Database Net Services Administrator's Guide* for implementation examples

GLOBAL_NAME

Purpose

Use the parameter GLOBAL_NAME to identify the Oracle Rdb database.

Embed this parameter under the CONNECT_DATA parameter.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS=...)
   (ADDRESS=...)
   (CONNECT_DATA=
    (SERVICE_NAME=generic)
    (RDB_DATABASE=[.mf]mf_personal.rdb)
    (GLOBAL_NAME=alpha5)))
```

HS

Purpose

Use the parameter HS to instruct Oracle Net to connect to a non-Oracle system through **Heterogeneous Services**.

Embed this parameter under the CONNECT_DATA parameter.

See Also: *Oracle Database Net Services Administrator's Guide* for complete configuration information

Default

None

Values

ok

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS=...)
   (ADDRESS=...)
   (CONNECT_DATA=
    (SID=sales6)
    (HS=ok))
```

INSTANCE_NAME

Purpose

Use the parameter INSTANCE_NAME to identify the database instance to access. Set the value to the value specified by the INSTANCE_NAME parameter in the initialization parameter file.

Embed this parameter under the CONNECT_DATA parameter.

See Also: *Oracle Database Net Services Administrator's Guide* for information about the use of INSTANCE_NAME

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS=...)
   (ADDRESS=...)
   (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)
    (INSTANCE_NAME=sales1))
```

RDB_DATABASE

Purpose

Use the parameter RDB_DATABASE parameter to specify the file name of an Oracle Rdb database.

Embed this parameter under the CONNECT_DATA parameter.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS=...)
   (ADDRESS=...)
   (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)
    (RDB_DATABASE= [.mf]mf_personal.rdb))
```

SERVER

Purpose

Use the parameter SERVER to instruct the listener to connect the client to a specific type of **service handler**.

Embed this parameter under the CONNECT_DATA parameter.

Values

- dedicated to specify that client requests be served by **dedicated server**
- shared to specify that client request be served by **shared server**

> **Note:** Shared server must be configured in the database initialization file in order for the client to connect to the database with a shared server process. See the *Oracle Database Net Services Administrator's Guide* for configuration information.

> **Note:** The USE_DEDICATED_SERVER parameter in the sqlnet.ora file overrides this parameter.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS=...)
   (ADDRESS=...)
   (CONNECT_DATA=
    (SERVER_NAME=sales.us.acme.com)
    (SERVER=dedicated)))
```

SERVICE_NAME

Purpose

Use the parameter SERVICE_NAME to identify the Oracle9i or Oracle8 database service to access. Set the value to a value specified by the SERVICE_NAMES parameter in the initialization parameter file.

Embed this parameter under the CONNECT_DATA parameter.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for information about the use of the SERVICE_NAME parameter

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS=...)
   (ADDRESS=...)
   (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com)))
```

SID

Purpose

Use the parameter SID to identify the Oracle8 database instance by its **Oracle System Identifier (SID)**. If the database is Oracle9i or Oracle8, use the SERVICE_NAME parameter rather than the SID parameter.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for information about the use of SID

Embed this parameter under the CONNECT_DATA parameter.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS=...)
   (ADDRESS=...)
   (CONNECT_DATA=
    (SID=sales)))
```

Security Section

The security section of the tnsnames.ora file specifies security-related parameters for use with Oracle Advanced Security features.

SECURITY

Purpose

Use the parameter SECURITY to enable secure connections.

Embed this parameter under the DESCRIPTION parameter.

Usage Notes

SECURITY permits the SSL_SERVER_CERT_DN subparameter.

Example

```
net_service_name=
 (DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521))
  (CONNECT_DATA=
   (SERVICE_NAME=sales.us.acme.com))
   (SECURITY=
    (SSL_SERVER_CERT_DN="cn=sales,cn=OracleContext,dc=us,dc=acme,dc=com")))
```

SSL_SERVER_CERT_DN

Purpose

Use the parameter SSL_SERVER_CERT_DN to specify the **distinguished name (DN)** of the database server. The client uses this information to obtain the list of DNs it expects for each of the servers, enforcing the database server DN to match its service name.

Usage Notes

Use this parameter in conjunction with the `sqlnet.ora` parameter SSL_SERVER_DN_MATCH to enable server DN matching.

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

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS=...)
   (ADDRESS=...)
   (CONNECT_DATA=
    (SERVICE_NAME=finance.us.acme.com))
   (SECURITY=
    (SSL_SERVER_CERT_DN="cn=finance,cn=OracleContext,dc=us,dc=acme,dc=com")))
```

7

Listener Parameters (listener.ora)

This chapter provides a complete listing of the `listener.ora` file configuration parameters.

This chapter contains these topics:

Overview of Listener Configuration File

Listener configuration, stored in the `listener.ora` file, consists of the following elements:

- Name of the listener
- Protocol addresses that the listener is accepting connection requests on
- Database services

  Dynamic **service registration**, a feature of Oracle9i and Oracle8, eliminates the need for static configuration of supported services. However, static service configuration is required if you plan to use Oracle Enterprise Manager.
- Control parameters

By default, the `listener.ora` file is located in the `$ORACLE_HOME/network/admin` directory on UNIX operating systems and the *`ORACLE_HOME`*`\network\admin` directory on Windows. `listener.ora` can also be stored the following locations:

- The directory specified by the `TNS_ADMIN` environment variable or registry value
- On UNIX operating systems, the global configuration directory. For example, on the Solaris Operating System, this directory is `/var/opt/oracle`.

See Also: Oracle operating system-specific documentation

It is possible to configure multiple listeners, each with unique name, in one `listener.ora` file. Multiple listener configuration is possible because each of the top-level configuration parameters has a suffix of the listener name or is the listener name itself.

Note: It is often useful to configure multiple listeners in one `listener.ora` file. However, Oracle Corporation recommends running only one listener for each node in most customer environments.

Example 7–1 shows a `listener.ora` file for a listener named `LISTENER`, which is the default name of the listener.

Example 7–1 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=/oracle10g)
      (SID_NAME=sales))
    (SID_DESC=
      (SID_NAME=plsextproc)
      (ORACLE_HOME=/oracle10g)
      (PROGRAM=extproc)))
```

Listener Parameters

This section lists and describes the `listener.ora` file parameters. Listener configuration parameters fall into the following categories:

- Protocol Address Section
- Static Service Registration (SID_LIST) Section
- Control Parameters

Protocol Address Section

The **protocol address** section of the `listener.ora` file defines the protocol addresses that the listener is accepting connection requests on. Discussed next are the most common parameters used in protocol addresses. Note that `ADDRESS_LIST` is also supported.

See Also: Chapter 4, "Protocol Address Configuration" for information about the `ADDRESS_LIST` parameter

DESCRIPTION

Purpose

Use the parameter `DESCRIPTION` as a container for listener protocol addresses.

Example

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

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

ADDRESS

Purpose

Use the parameter ADDRESS to specify a single listener protocol address.

Embed this parameter under the DESCRIPTION parameter.

> **See Also:** Chapter 4, "Protocol Address Configuration" for descriptions of the correct parameters to use for each type of support protocol

Example

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

QUEUESIZE

Purpose

Use the parameter QUEUESIZE to specify the number of concurrent connection requests that the listener can accept on a TCP/IP or IPC listening endpoint (protocol address).

Embed this parameter at the end of the protocol address with its value set to the expected number of concurrent connection requests.

Default

The default number of concurrent connection requests is operating system specific. Following are the defaults for the Solaris Operating System and Windows:

- Solaris Operating System: 5
- Windows NT 4.0 Workstation: 5
- Windows NT 4.0 Server: 50

Usage Notes

> **See Also:** *Oracle Net Services Administrator's Guide* for information about configuring this parameter

Example

```
listener_name=
 (DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=hr-server)(PORT=1521)(QUEUESIZE=20)))
```

RECV_BUF_SIZE

Purpose

Use the parameter RECV_BUF_SIZE to specify, in bytes, the buffer space for receive operations of sessions. This parameter is supported by the TCP/IP, TCP/IP with SSL, and SDP protocols.

> **Note:** Additional protocols might support this parameter on certain operating systems. Refer to operating-system specific documentation for information about additional protocols that support thi parameter.

Embed this parameter under the DESCRIPTION parameter or at the end of the protocol address with its value set to the expected number of bytes.

Default

The default value for this parameter is operating-system specific. The default for the Solaris 2.6 Operating System is 32768 bytes (32 KB).

Usage Notes

> **See Also:** *Oracle Net Services Administrator's Guide* for information about configuring this parameter

Example

```
listener_name=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)
       (RECV_BUF_SIZE=11784))
    (ADDRESS=(PROTOCOL=ipc)(KEY=extproc)
       (RECV_BUF_SIZE=11784)))
listener_name=
  (DESCRIPTION=
    (RECV_BUF_SIZE=11784))
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)
    (ADDRESS=(PROTOCOL=ipc)(KEY=extproc)))
```

SEND_BUF_SIZE

Purpose

Use the parameter SEND_BUF_SIZE to specify, in bytes, the buffer space for send operations of sessions. This parameter is supported by the TCP/IP, TCP/IP with SSL, and SDP protocols.

> **Note:** Additional protocols might support this parameter on certain operating systems. Refer to operating-system specific documentation for information about additional protocols that support thi parameter.

Embed this parameter under the DESCRIPTION parameter or at the end of the protocol address.

Default

The default value for this parameter is operating-system specific. The default for the Solaris 2.6 Operating System is 8192 bytes (8 KB).

Usage Notes

See Also: *Oracle Database Net Services Administrator's Guide* for information about configuring this parameter

Example

```
listener_name=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)
       (SEND_BUF_SIZE=11280))
      (ADDRESS=(PROTOCOL=ipc)(KEY=extproc)
       (SEND_BUF_SIZE=11280)))
listener_name=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (SEND_BUF_SIZE=11280))
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)
      (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))))
```

Static Service Registration (SID_LIST) Section

You can use the SID_LIST section of the listener.ora to statically configure service information for the listener.

The SID_LIST section is required for Oracle8 release 8.0 or Oracle7 database services, as well as **external procedure** calls and **Heterogeneous Services**, and some management tools, including Oracle Enterprise Manager.

```
SID_LIST_listener_name=
  (SID_LIST=
   (SID_DESC=
    (GLOBAL_DBNAME=global_database_name)
    (SID_NAME=sid)
    (ORACLE_HOME=oracle_home))
   (SID_DESC=...))
```

For later database releases, the listener uses the dynamic service information about the database and instance it has received through service registration *before using statically configured information* in the listener.ora file. Therefore, the SID_LIST is not required, unless Oracle Enterprise Manager is used to monitor an Oracle9i or Oracle8 database.

SID_LIST

Purpose

Use the parameter SID_LIST to identify a list of SID descriptions.

Example

```
SID_LIST_listener_name=
  (SID_LIST=
   (SID_DESC=...)
   (SID_DESC=...))
```

SID_DESC

Purpose

Use the parameter SID_DESC to specify service information for a specific database instance or a non-database service.

Embed this parameter under the SID_LIST parameter.

Example

```
SID_LIST_listener_name=
(SID_LIST=
   (SID_DESC=...)
   (SID_DESC=...))
```

SID_DESC permits the following parameters:

- ENVS
- GLOBAL_DBNAME
- ORACLE_HOME
- PROGRAM
- SID_NAME
- SDU

ENVS

Purpose

Use the parameter ENVS to specify environment variables for the listener to set prior to executing (as a child process) a dedicated server program or an executable specified with the PROGRAM parameter.

Embed this parameter under the SID_DESC parameter.

> **Note:** This parameter in not supported on Windows. Any process spawned by the listener will simply inherit the listener's environment.

Usage Notes

Enclose an environment variable and its value within double quotes (").

```
(ENVS="variable=value")
```

A list of environment variables and their values can be specified. Enclose the list within double quotes, from end to end, and separate environment variable definitions with commas and no space.

```
(ENVS="variable=value,variable=value")
```

> **Note:** Single quotes (') are supported for backward compatibility.

The use of the following characters within the environment variable or its value definition (ENVS="*variable=value*") are not supported:

- Comma (,)

- Single quotes (')
- Double quotes (")
- Equal sign (=)

Example

```
SID_LIST_listener_name=
 (SID_LIST=
  (SID_DESC=
   (SID_NAME=plsextproc)
   (ORACLE_HOME=/oracle10g)
   (PROGRAM=extproc)
   (ENVS="LD_LIBRARY_PATH=/oracle/10g/lib:/oracle/10g/ctx/lib"))
 (SID_DESC=
  (SID_NAME=test)
  (PROGRAM=/tmp/myexec)
  (ENVS="LD_LIBRARY_
PATH=/private/xpm/lib:/private/mylibs,MYPATH=/usr/ucb:/usr/local/packages,APL_ENV_
FILE=/apl/conf/env.txt")))
```

GLOBAL_DBNAME

Purpose

Use the parameter GLOBAL_DBNAME to identify the database service.

While processing a client connection request, the listener tries to match the value of this parameter with the value of the SERVICE_NAME parameter in the client connect descriptor. If the client connect descriptor uses the SID parameter, then the listener does not attempt to map the values. This parameter is primarily intended for configurations with Oracle8 release 8.0 or Oracle7 databases (where dynamic service registration is not supported for dedicated servers). This parameter may also be required for use with Oracle9i and Oracle8 database services by some configurations and management tools.

The value for this parameter is typically obtained from the combination of the DB_NAME and DB_DOMAIN parameters (DB_NAME.DB_DOMAIN) in the initialization parameter file, but the value can also contain any valid name used by clients to identify the service.

Embed this parameter under the SID_DESC parameter.

Example

```
SID_LIST_listener_name=
  (SID_LIST=
   (SID_DESC=
    (GLOBAL_DBNAME=sales.us.acme.com)
    (SID_NAME=sales)
    (ORACLE_HOME=/usr/oracle)))
```

ORACLE_HOME

Purpose

Use the parameter ORACLE_HOME to identify the Oracle home location of the service.

Embed this parameter under the SID_DESC parameter.

Example

```
SID_LIST_listener_name=
  (SID_LIST=
   (SID_DESC=
     (SID_NAME=extproc)
     (ORACLE_HOME=/usr/oracle)
     (PROGRAM=extproc)))
```

PROGRAM

Purpose

Use the parameter PROGRAM to identify the service executable program name.

Embed this parameter under the SID_DESC parameter.

Example

```
SID_LIST_listener_name=
  (SID_LIST=
   (SID_DESC=
     (SID_NAME=sales)
     (ORACLE_HOME=/usr/oracle)
     (PROGRAM=extproc))
```

SID_NAME

Purpose

Use the parameter SID_NAME to identify the **Oracle System Identifier (SID)** of the instance. You can obtain the SID value from the INSTANCE_NAME parameter in the initialization parameter file.

Embed this parameter under the SID_DESC parameter.

Example

```
SID_LIST_listener_name=
  (SID_LIST=
   (SID_DESC=
    (GLOBAL_DBNAME=sales.us.acme.com)
    (SID_NAME=sales)
    (ORACLE_HOME=/usr/oracle)))
```

SDU

Purpose

Use the parameter SDU to instruct Oracle Net to optimize the transfer rate of data packets being sent across the network with the **session data unit (SDU)** size you specify.

Embed this parameter under the SID_DESC parameter.

Usage

> **See Also:** *Oracle Database Net Services Administrator's Guide* for information about configuring this parameter

Default

2048 bytes (2KB)

Values

512 bytes to 32768 (32 KB)

Example

```
SID_LIST_listener_name=
  (SID_LIST=
   (SID_DESC=
    (SDU=2085)
    (GLOBAL_DBNAME=sales.us.acme.com)
    (SID_NAME=sales)
    (ORACLE_HOME=/usr/oracle)))
```

Control Parameters

This section describes the following parameters that control the behavior of the listener:

- ADMIN_RESTRICTIONS_listener_name
- INBOUND_CONNECT_TIMEOUT_listener_name
- LOG_DIRECTORY_listener_name
- LOG_FILE_listener_name
- LOGGING_listener_name
- PASSWORDS_listener_name
- SAVE_CONFIG_ON_STOP_listener_name
- SSL_CLIENT_AUTHENTICATION
- STARTUP_WAIT_TIME_listener_name
- TRACE_DIRECTORY_listener_name
- TRACE_FILE_listener_name
- TRACE_FILELEN_listener_name
- TRACE_FILENO_listener_name
- TRACE_LEVEL_listener_name
- TRACE_TIMESTAMP_listener_name
- WALLET_LOCATION

ADMIN_RESTRICTIONS_*listener_name*

Purpose

Use the parameter ADMIN_RESTRICTIONS_*listener_name* to restrict runtime administration of the listener. The parameter is useful if the listener is not password-protected.

Setting ADMIN_RESTRICTIONS_*listener_name*=on disables the runtime modification of parameters in `listener.ora`. That is, the listener will refuse to accept SET commands that alter its parameters. To change any of the parameters in `listener.ora`, including ADMIN_RESTRICTIONS_*listener_name* itself, modify

the `listener.ora` file manually and reload its parameters (with the RELOAD command) for the new changes to take effect without explicitly stopping and restarting the listener.

Oracle Corporation recommends establishing a password to secure the listener. To establish an encrypted password, use either the Listener Control utility CHANGE_PASSWORD command or Oracle Net Manager.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for further information about password security of the listener

Default

`off`

Example

```
ADMIN_RESTRICTIONS_listener=on
```

INBOUND_CONNECT_TIMEOUT_*listener_name*

Purpose

Use the `INBOUND_CONNECT_TIMEOUT_`*`listener_name`* parameter to specify the time, in seconds, for the client to complete its connect request to the listener after the network connection had been established.

If the listener does not receive the client request in the time specified, then it terminates the connection. In addition, the listener logs the IP address of the client and an `ORA-12525:TNS: listener has not received client's request in time allowed` error message to the `listener.log` file.

To protect both the listener and the database server, Oracle Corporation recommends setting this parameter in combination with the SQLNET.INBOUND_CONNECT_TIMEOUT parameter in the `sqlnet.ora` file. When specifying values for these parameters, consider the following recommendations:

- Set both parameters to an initial low value.
- Set the value of the `INBOUND_CONNECT_TIMEOUT_`*`listener_name`* parameter to a lower value than the `SQLNET.INBOUND_CONNECT_TIMEOUT` parameter.

For example, you can set `INBOUND_CONNECT_TIMEOUT_`*`listener_name`* to 2 seconds and `INBOUND_CONNECT_TIMEOUT` parameter to 3 seconds. If clients are unable to complete connections within the specified time due to system or network delays that are normal for the particular environment, then increment the time as needed.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for information about configuring these parameters

Default

60 seconds

Example

```
INBOUND_CONNECT_TIMEOUT_listener=2
```

LOG_DIRECTORY_*listener_name*

Purpose

Use the parameter LOG_DIRECTORY_*listener_name* to specify the destination directory of the listener log file.

Default

The $ORACLE_HOME/network/log directory on UNIX operating systems and the *ORACLE_HOME*\network\log directory on Windows operating systems

Example

```
LOG_DIRECTORY_listener=/oracle/network/admin/log
```

LOG_FILE_*listener_name*

Purpose

Use the parameter LOG_FILE_*listener_name* to specify the name of the log file for the listener.

Default

```
listener.log
```

Example

```
LOG_FILE_listener=list.log
```

LOGGING_*listener_name*

Purpose

Use the parameter LOGGING_*listener_name* to turn logging on or off.

Default

```
on
```

Values

```
on | off
```

Example

```
LOGGING_listener=on
```

PASSWORDS_*listener_name*

Purpose

Use the parameter PASSWORDS_*listener_name* to store an encrypted password for a listener, so that certain privileges operations, such as SAVE_CONFIG and STOP, used from the Listener Control utility are secure. An encrypted password can be set using either the Listener Control utility CHANGE_PASSWORD command or Oracle Net Manager.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for further information about password security of the listener

Example

```
PASSWORDS_LISTENER=(2D6C48144CF753AC)
```

SAVE_CONFIG_ON_STOP_*listener_name*

Purpose

Use the parameter SAVE_CONFIG_ON_STOP_*listener_name* to specify whether or not runtime configuration changes are saved into the `listener.ora` file.

When you set the parameter to `true`, any parameters that were modified while the listener was running using the Listener Control utility SET command are saved to the `listener.ora` file when the STOP command is issued. When you set the parameter to `false`, the Listener Control utility does not save the runtime configuration changes to the `listener.ora` file.

Default

```
false
```

Values

```
true | false
```

Example

```
SAVE_CONFIG_ON_STOP_listener=true
```

SSL_CLIENT_AUTHENTICATION

Purpose

Use the parameter SSL_CLIENT_AUTHENTICATION to specify whether or not a client is authenticated using the **Secure Sockets Layer (SSL)**.

Default

```
true
```

Values

```
true | false
```

Usage Notes

The database server authenticates the client. Therefore, this value should be set to `false`. If this parameter is set to `true`, the listener attempts to authenticate the client, which can result in a failure.

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

Example

```
SSL_CLIENT_AUTHENTICATION=true
```

STARTUP_WAIT_TIME_*listener_name*

> **Note:** This parameter is deprecated and will be desupported in a future release. If you require this parameter to run the listener, please notify Oracle Support Services.

Purpose

Use the parameter STARTUP_WAIT_TIME_*listener_name* to set the number of seconds that the listener waits before responding to a Listener Control utility START command.

Default

0

Example

```
STARTUP_WAIT_TIME_listener=5
```

TRACE_DIRECTORY_*listener_name*

Purpose

Use the parameter TRACE_DIRECTORY_*listener_name* to specify the destination directory of the listener trace file.

Default

The $ORACLE_HOME/network/trace directory on UNIX operating systems and the *ORACLE_HOME*\network\trace directory on Windows

Example

```
TRACE_DIRECTORY_listener=/oracle/network/admin/trace
```

TRACE_FILE_*listener_name*

Purpose

Use the parameter TRACE_FILE_*listener_name* to specify the name of the trace file for the listener.

Default

listener.trc

Example

```
TRACE_FILE_listener=list.trc
```

TRACE_FILELEN_*listener_name*

Purpose

Use the parameter TRACE_FILELEN_*listener_name* to specify the size of the listener trace files in kilobytes (KB). When the size is met, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO_ listener_name parameter.

Default

Unlimited

Example

```
TRACE_FILELEN_listener=100
```

TRACE_FILENO_*listener_name*

Purpose

Use the parameter TRACE_FILENO_*listener_name* to specify the number of trace files for listener tracing. When this parameter is set along with the TRACE_FILELEN_listener_name parameter, trace files are used in a cyclical fashion. The first file is filled first, then the second file, and so on. When the last file has been filled, the first file is re-used, and so on.

The trace file names are distinguished from one another by their sequence number. For example, if the default trace file of `listener.trc` is used, and this parameter is set to 3, the trace files would be named `listener1.trc`, `listener2.trc` and `listener3.trc`.

In addition, trace events in the trace files are preceded by the sequence number of the file.

Default

`1`

Example

```
TRACE_FILENO_listener=3
```

TRACE_LEVEL_*listener_name*

Purpose

Use the parameter TRACE_LEVEL_*listener_name* to turn listener tracing on, at a specific level, or off.

Default

`off`

Values

- `off` for no trace output
- `user` for user trace information
- `admin` for administration trace information
- `support` for Oracle Support Services trace information

Example

```
TRACE_LEVEL_listener=admin
```

TRACE_TIMESTAMP_*listener_name*

Purpose

When parameter TRACE_LEVEL_listener_name is set to a specific tracing level, you can use the parameter TRACE_TIMESTAMP_*listener_name* to add a time stamp in the form of *dd-mon-yyyy hh:mi:ss:mil* to every trace event in the trace file for the listener.

Default

`on`

Values

on or true | off or false

Example

```
TRACE_TIMESTAMP_listener=true
```

WALLET_LOCATION

Purpose

Use the parameter WALLET_LOCATION to specify the location of wallets. Wallets are certificates, keys, and trustpoints processed by SSL that allow for secure connections.

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

Syntax

Oracle wallets on file system:

```
WALLET_LOCATION=
  (SOURCE=
    (METHOD=file)
    (METHOD_DATA=
       (DIRECTORY=directory)
       [(PKCS11=TRUE/FALSE)]))
```

Microsoft certificate store:

```
WALLET_LOCATION=
  (SOURCE=
     (METHOD=mcs))
```

Oracle wallets in the Windows registry:

```
WALLET_LOCATION=
   (SOURCE=
      (METHOD=reg)
      (METHOD_DATA=
         (KEY=registry_key)))
```

Entrust wallets:

```
WALLET_LOCATION=
   (SOURCE=
      (METHOD=entr)
      (METHOD_DATA=
         (PROFILE=file.epf)
         (INIFILE=file.ini)))
```

Subparameters

WALLET_LOCATION supports the following subparameters:

SOURCE: Specify the type of storage for wallets and storage location.

METHOD: Specify the type of storage.

METHOD_DATA: Specify the storage location.

DIRECTORY: Specify the location of Oracle wallets on file system.

KEY: Specify the wallet type and location in the Windows registry.

PROFILE: Specify the Entrust profile file (.epf).

INIFILE: Specify the Entrust initialization file (.ini).

Default

None

Usage Notes

- The key/value pair for Microsoft's certificate store (MCS) omits the METHOD_DATA parameter because MCS does not use wallets. Instead, Oracle PKI (public key infrastructure) applications obtain certificates, trustpoints and private keys directly from the user's profile.
- If an Oracle wallet is stored in the Windows registry and the wallet's key (KEY) is SALESAPP, the storage location of the encrypted wallet is HKEY_CURRENT_USER\SOFTWARE\ORACLE\WALLETS\SALESAPP\EWALLET.P12. The storage location of the decrypted wallet is HKEY_CURRENT_USER\SOFTWARE\ORACLE\WALLETS\SALESAPP\CWALLET.SSO.

Examples

Oracle wallets on file system:

```
WALLET_LOCATION=
  (SOURCE=
      (METHOD=file)
      (METHOD_DATA=
         (DIRECTORY=/etc/oracle/wallets/databases)))
```

Microsoft certificate store:

```
WALLET_LOCATION=
   (SOURCE=
     (METHOD=mcs))
```

Oracle Wallets in the Windows registry:

```
WALLET_LOCATION=
   (SOURCE=
     (METHOD=REG)
     (METHOD_DATA=
        (KEY=SALESAPP)))
```

Entrust Wallets:

```
WALLET_LOCATION=
   (SOURCE=
     (METHOD=entr)
     (METHOD_DATA=
       (PROFILE=/etc/oracle/wallets/test.epf)
       (INIFILE=/etc/oracle/wallets/test.ini)))
```

8

Oracle Connection Manager Parameters (cman.ora)

This chapter provides a complete listing of the `cman.ora` file configuration parameters.

This chapter contains these topics:

- Overview of Oracle Connection Manager Configuration File
- Oracle Connection Manager Parameters

Overview of Oracle Connection Manager Configuration File

Oracle Connection Manager configuration information, stored in the `cman.ora` file, consists of the following elements:

- Protocol address of the Oracle Connection Manager listener
- Access control parameters
- Performance parameters

By default, the `cman.ora` file is located in the `$ORACLE_HOME/network/admin` directory on UNIX operating systems and in the `%`*`ORACLE_HOME`*`%\network\admin` directory on Windows. `cman.ora` can also be stored the following locations:

- The directory specified by the `TNS_ADMIN` environment variable or registry value
- On UNIX operating systems, the global configuration directory. For example, on the Solaris Operating System, this directory is `/var/opt/oracle`.

> **See Also:** Oracle operating system-specific documentation

Example 8–1 shows an example `cman.ora` file.

Example 8–1 Example cman.ora File

```
CMAN=
  (CONFIGURATION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=proxysvr)(PORT=1521))
    (RULE_LIST=
      (RULE=(SRC=206.62.226.32/27)(DST=sales-server)(SRV=*)(ACT=accept))
        (ACTION_LIST=(AUT=on)(MCT=120)(MIT=30)))
      (RULE=(SRC=foo)(DST=foobar)(SRV=cmon)(ACT=accept)))
    (PARAMETER_LIST=
      (MAX_GATEWAY_PROCESSES=8)
      (MIN_GATEWAY_PRCESSSES=3)
```

```
(REMOTE_ADMIN=YES)))
```

Oracle Connection Manager Parameters

This section lists and describes the `cman.ora` file parameters. Configuration parameters fall into the following categories:

- Listening Endpoint
- Rule List
- Parameter List

Listening Endpoint

ADDRESS

Purpose

Use the parameter `ADDRESS` to specify the protocol address of Oracle Connection Manager.

Syntax

```
(ADDRESS= ...)
```

Example (Default)

```
(ADDRESS=(PROTOCOL=tcp)(HOST=local_host)(PORT=1521))
```

Rule List

RULE

Purpose

Use the parameter `RULE` to specify an access control rule list to filter incoming connections. A rule list specifies which connections are accepted, rejected, or dropped.

Syntax

```
(RULE_LIST=
  (RULE=
    (SRC=host)
    (DST=host)
    (SRV=service_name)
    (ACT={accept|reject|drop})
    (ACTION_LIST=AUT=on|off
    ((CONN_STATS=yes|no)(MCT=time)(MIT=time)(MOCT=time)))
  (RULE= ...))
```

Subparameters

The `RULE` parameter filters a connection or group of connections using the following subparameters:

`SRC`: Specify the source host name or IP address in dot notation of the client.

DST: Specify the destination server host name or IP address in dot notation of the database server.

SRV: Specify database **service name** of the Oracle Database 10*g*, Oracle9i, or Oracle8 database (obtained from the SERVICE_NAME parameter in the initialization parameter file).

ACT: Specify accept to accept incoming requests or reject to reject incoming requests.

ACTION_LIST: Specify rule-level parameter settings for some parameters. These parameters are as follows:

- AUT–Oracle Advanced Security authentication on client side
- CONN_STATS—log input and output statistics
- MCT—maximum connect time
- MIT—maximum idle timeout
- MOCT—maximum outbound connect time

Rule-level parameters override their global counterparts.

Usage Notes

- If no rules are specified, all connections are rejected.
- The source and destination can be a host name, IP address, or subnet mask.
- You must enter at least one rule for client connections and one rule for CMCTL connections. Omitting one or the other results in the rejection of all connections for the rule type omitted. The last rule in the example that follows is a CMCTL rule.
- If the CMCTL connection is remote, the REMOTE_ADMIN parameter in cman.ora must be set to on, regardless of the rules specified.
- Oracle Connection Manager does not support wildcards for partial IP addresses. If you use a wildcard, use it in place of a full IP address. The IP address of the client may, for example, be (SRC=*).
- Oracle Connection Manager supports only the /nn notation for subnet addresses. In the first rule in the example, /27 represents a subnet mask that comprises 27 left-most bits.

Example

```
(RULE_LIST=
  (RULE=
    (SRC=client1-pc)
    (DST=sales-server)
    (SRV=sales.us.acme.com)
    (ACT=reject))
  (RULE=
    (SRC=144.25.23.45)
    (DST=144.25.187.200)
    (SRV=db1)
    (ACT=accept))
  (RULE=
    (SRC=foo)
    (DST=foobar)
    (SRV=cmon)
    (ACT=accept)))
```

Parameter List

PARAMETER_LIST

Purpose

The PARAMETER_LIST parameter specifies the attributes for an Oracle Connection Manager. To override the default setting for a parameter, enter the parameter and its nondefault value.

Default Parameters

ASO_AUTHENTICATION_FILTER=OFF

CONNECTION_STATISTICS=NO

EVENT_GROUP=none

IDLE_TIMEOUT=0

INBOUND_CONNECT_TIMEOUT=60

LOG_DIRECTORY=The $ORACLE_HOME/network/log directory on UNIX operating systems and the *ORACLE_HOME*\network\log directory on Windows

LOG_LEVEL=SUPPORT

MAX_CMCTL_SESSIONS=4

MAX_CONNECTIONS=256

MAX_GATEWAY_PROCESSES=16

MIN_GATEWAY_PROCESSES=2

OUTBOUND_CONNECT_TIMEOUT=0

PASSWORD_*instance_name*=Value is the encrypted instance password, if one has been set. Default is no value.

REMOTE_ADMIN=NO

SESSION_TIMEOUT=0

TRACE_DIRECTORY=The $ORACLE_HOME/network/trace directory on UNIX operating systems and the *ORACLE_HOME*\network\trace directory on Windows

TRACE_FILELEN=0

TRACE_FILENO=0

TRACE_LEVEL=OFF

TRACE_TIMESTAMP=OFF

Allowed Values of Parameters

ASO_AUTHENTICATION_FILTER=[off | on]

CONNECTION_STATISTICS=[no | yes]

EVENT_GROUP=[init_and_term | memory_ops | conn_hdlg | proc_mgmt | reg_and_load | wake_up | timer | cmd_proc | relay

IDLE_TIMEOUT=0 or greater

INBOUND_CONNECT_TIMEOUT=0 or greater

LOG_DIRECTORY=*log_directory*

LOG_LEVEL=[off | user | admin | support]

MAX_CMCTL_SESSIONS=Any positive number

MAX_CONNECTIONS=[1 to 1024]

MAX_GATEWAY_PROCESSES=Any number greater than the minimum number of gateway processes up to 64

MIN_GATEWAY_PROCESSES=Any positive number less than or equal to 64. Must be less than or equal to the maximum number of gateway processes.

OUTBOUND_CONNECT_TIMEOUT=0 or greater

REMOTE_ADMIN=[no | yes]

SESSION_TIMEOUT=0 or greater

TRACE_DIRECTORY=*trace_directory*

TRACE_FILELEN=Any positive number

TRACE_FILENO=Any positive number

TRACE_LEVEL=[off | user | admin | support]

TRACE_TIMESTAMP=[off | on]

Note: The event group ALERT cannot be turned off.

Example

```
(PARAMETER_LIST=
   (ASO_AUTHENTICATION_FILTER=ON)
   (CONNECTION_STATISTICS=NO)
   (EVENT_GROUP=INIT_AND_TERM,MEMORY_OPS,PROCESS_MGMT)
   (IDLE_TIMEOUT=30)
   (INBOUND_CONNECT_TIMEOUT=30)
   (LOG_DIRECTORY=/home/user/network/admin/log)
   (LOG_LEVEL=SUPPORT)
   (MAX_CMCTL_SESSIONS=6)
   (MAX_CONNECTIONS=512)
   (MAX_GATEWAY_PROCESSES=10)
   (MIN_GATEWAY_PROCESSES=4)
   (OUTBOUND_CONNECT_TIMEOUT=30)
   (REMOTE_ADMIN=YES)
   (SESSION_TIMEOUT=60)
   (TRACE_DIRECTORY=/home/user/network/admin/trace)
   (TRACE_FILELEN=100)
   (TRACE_FILENO=2)
   (TRACE_LEVEL=SUPPORT)
   (TRACE_TIMESTAMP=ON))
```

Note: You cannot add the parameter PASSWORD_*instance_name* directly to cman.ora. The parameter is added when you issue the command SAVE_PASSWD.

ASO_AUTHENTICATION_FILTER

Use the ASO_AUTHENTICATION_LEVEL parameter to specify whether Oracle Advanced Security authentication settings must be used by the client. The global setting can be overridden by a rule-level setting in ACTION_LIST. This parameter accepts the following values:

- on to instruct Oracle Connection Manager to reject connect requests that are not using Secure Network Services (SNS). SNS is part of the Oracle Advanced Security.
- off (default) to instruct Oracle Connection Manager not to check for SNS between the client and server

CONNECTION_STATISTICS

Use the CONNECTION_STATISTICS parameter to specify whether the SHOW_CONNECTIONS command displays connection statistics. The global setting can be overridden by a rule-level setting in ACTION_LIST. This parameter accepts the following values:

- yes to display statistics
- no (default) to not display statistics

EVENT_GROUP

Use the EVENT_GROUP parameter to specify which event groups are logged. Multiple events may be designated using a comma-separated list. This parameter accepts the following values:

- INIT_AND_TERM—initialization and termination
- MEMORY_OPS—memory operations
- CONN_HDLG—connection handling
- PROC_MGMT—process management
- REG_AND_LOAD—Registration and load update
- WAKE_UP—events related to CMADMIN wakeup queue
- TIMER—gateway timeouts
- CMD_PROC—command processing
- RELAY—events associated with connection control blocks

IDLE_TIMEOUT

Use the IDLE_TIMEOUT parameter to specify the amount of time that an established connection can remain active without transmitting data. The global setting can be overridden by a rule-level setting in ACTION_LIST. This parameter accepts the following values:

0 (default) to disable the timeout

n>0 to enable the timeout, where *n* equals the timeout period in seconds

INBOUND_CONNECT_TIMEOUT

Use the INBOUND_CONNECT_TIMEOUT parameter to specify how long the Oracle Connection Manager listener waits for a valid connection from a client or another instance of Oracle Connection Manager. This parameter accepts the following values:

60 (default) to disable the timeout

n>0 to enable the timeout, where *n* equals the timeout period in seconds

LOG_DIRECTORY

Use the LOG_DIRECTORY parameter to specify the location of Oracle Connection Manager log files.

LOG_LEVEL

Use the LOG_LEVEL parameter to specify the level of logging performed by Oracle Connection Manager. This parameter accepts four log levels:

- off (default) for no logging
- user for user log information
- admin for administrative log information
- support for Oracle Support Services information

There are three kinds of log files: *instance-name_pid*.log for the listener, *instance-name*_cmadmin_*pid*.log for CMADMIN, and *instance-name*_cmgw_*pid*.log for the gateway processes. The log files are located in the $ORACLE_HOME/network/log directory on UNIX operating systems and the *%ORACLE_HOME%*\network\log directory on Windows.

MAX_CMCTL_SESSIONS

Use the MAX_CMCTL_SESSIONS parameter to specify the maximum number of concurrent local or remote sessions of the Oracle Connection Manager control utility allowable for a given instance. One of these sessions must be a local session. Any number of sessions can be designated.

MAX_CONNECTIONS

Use the MAX_CONNECTIONS parameter to specify the maximum number of connections that a gateway process can handle.

This parameter accepts a range of:

1 to 1024

MAX_GATEWAY_PROCESSES

Use the MAX_GATEWAY_PROCESSES parameter to specify the maximum number of gateway processes that an instance of Oracle Connection Manager supports. The maximum is 64. The number designated must be greater than the minimum number of gateway processes.

MIN_GATEWAY_PROCESSES

Use the MIN_GATEWAY_PROCESSES parameter to specify the minimum number of gateway processes that an instance of Oracle Connection Manager must support. Any number of sessions can be designated up to 64.

OUTBOUND_CONNECT_TIMEOUT

Use the OUTBOUND_CONNECT_TIMEOUT parameter to specify the length of time that the Oracle Connection Manager instance waits for a valid connection to be established

with the database server or with another Oracle Connection Manager instance. This parameter accepts the following values:

60 (default) to disable the timeout

n>0 to enable the timeout, where *n* equals the timeout period in seconds

PASSWORD_instance_name

Use the PASSWORD_*instance_name* parameter to specify the encrypted instance password, if one has been set.

REMOTE_ADMIN

Use the parameter REMOTE_ADMIN to specify whether or not remote access to an Oracle Connection Manager is allowed. This parameter accepts the following values:

- yes to allow access from a remote Oracle Connection Manager Control utility session to Oracle Connection Manager
- no to allow only access to the local Oracle Connection Manager. This value prevents a user running a remote Oracle Connection Manager Control utility from accessing Oracle Connection Manager.

> **See Also:** "Distributed Operations" on page 2-2 for configuration details

SESSION_TIMEOUT

Use the SESSION_TIMEOUT parameter to specify the maximum time allowed for a user session. The global setting can be overridden by a rule-level setting in ACTION_LIST. This parameter accepts the following values:

0 (default) to disable the timeout

n>0 to enable the timeout, where *n* equals the timeout period in seconds

TRACE_DIRECTORY

Use the parameter TRACE_DIRECTORY to specify the location of the Oracle Connection Manager trace files.

TRACE_FILELEN

Use the parameter TRACE_FILELEN to specify the size, in kilobytes, of the trace file. When the size is met, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO parameter. Any size can be designated.

TRACE_FILENO

Use the parameter TRACE_FILENO to specify the number of trace files for Oracle Connection Manager tracing. When this parameter is set along with the TRACE_FILELEN parameter, trace files are used in a cyclical fashion. The first file is filled first, then the second file, and so on. When the last file has been filled, the first file is reused, and so on. Any number of files can be designated.

The trace file names are distinguished from one another by their sequence number. For example, if this parameter is set to 3, the gateway trace files would be named *instance-name*_cmgw1_*pid*.trc, *instance_name*_cmgw2_*pid*.trc and *instance_name*_cmgw3_*pid*.trc.

In addition, trace events in the trace files are preceded by the sequence number of the file.

TRACE_LEVEL

Use the parameter TRACE_LEVEL to specify the trace level for the Oracle Connection Manager instance. This parameter accepts four trace levels:

- off (default) for no logging
- user for user log information
- admin for administrative log information
- support for Oracle Support Services information

There are three kinds of trace files: *instance-name_pid*.trc for the listener, *instance-name*_cmadmin_*pid*.trc for CMADMIN, and *instance-name*_cmgw_*pid*.trc for the gateway processes. The log files are located in the $ORACLE_HOME/network/trace directory on UNIX operating systems and the *%ORACLE_HOME%*\network\trace directory on Windows.

TRACE_TIMESTAMP

When the parameter TRACE_LEVEL is enabled, you can use the TRACE_TIMESTAMP parameter to add a time stamp in the form of *dd-mon-yyyy hh:mi:ss:mil* to every trace event in the trc files.

9

Directory Usage Parameters (ldap.ora)

This chapter provides a complete listing of the `ldap.ora` file configuration parameters.

This chapter contains these topics:

- Overview of Directory Server Usage File
- Directory Usage Parameters

Overview of Directory Server Usage File

The `ldap.ora` file contains directory usage configuration parameters created by **Oracle Internet Directory** Configuration Assistant or **Oracle Net Configuration Assistant**. Do not modify these parameters or their settings.

When created with Oracle Internet Directory Configuration Assistant, `ldap.ora` is located in the `$ORACLE_HOME/ldap/admin` directory on UNIX operating systems and the *ORACLE_HOME*`\ldap\admin` directory on Windows operating systems. When created with Oracle Net Configuration Assistant, `ldap.ora` is located in the `$ORACLE_HOME/network/admin` directory on UNIX operating systems and the *ORACLE_HOME*`\network\admin` directory on Windows operating systems. `ldap.ora` can also be stored in the directory specified by the `LDAP_ADMIN` or `TNS_ADMIN` environment variable.

Directory Usage Parameters

This section lists and describes the `ldap.ora` file configuration parameters.

DIRECTORY_SERVERS

Purpose

Use the parameter `DIRECTORY_SERVERS` to list the host names and port number of the primary and alternate LDAP **directory server**s.

Values

host`:`*port*`[:`*sslport*`]`

Example

```
DIRECTORY_SERVERS=(ldap-server:389, raffles:400:636)
```

DIRECTORY_SERVER_TYPE

Purpose

Use the parameter `DIRECTORY_SERVER_TYPE` to specify the type of directory server that is being used.

Values

- `oid` for Oracle Internet Directory
- `ad` for Microsoft Active Directory

Example

```
DIRECTORY_SERVER_TYPE=oid
```

DEFAULT_ADMIN_CONTEXT

Purpose

Use the parameter `DEFAULT_ADMIN_CONTEXT` to specify the default directory entry that contains an **Oracle Context** from which connect identifiers can be created, modified, or looked up.

Values

Valid **distinguished name (DN)**

Example

```
DEFAULT_ADMIN_CONTEXT="o=OracleSoftware,c=US"
```

Part III

Appendixes

This part contains the following appendixes:

- Appendix A, "Features Not Supported in this Release"
- Appendix B, "Upgrade Considerations for Oracle Net Services"
- Appendix C, "LDAP Schema for Oracle Net Services"

A

Features Not Supported in this Release

This appendix describes features no longer supported by Oracle Net Services.

This appendix contains these topics:

Overview of 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:

Oracle Names

Oracle Names is no longer supported as a **naming method** in Oracle Database 10*g* . You must migrate to **directory naming**.

> **See Also:** *Oracle Net Services Administrator's Guide* for information about migrating to directory naming

Identix and SecurID Authentication Methods

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

- 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.

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.

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

See Also: Chapter 5 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.

Prespawned Dedicated Servers

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

Protocols

Protocol addresses using the SPX or LU6.2 protocol must be replaced. Oracle Net provides support for the following network protocols:

- TCP/IP
- TCP/IP with SSL
- Named Pipes
- SDP

See Also: "Protocol Parameters" on page 4-2 for protocol parameter configuration

Unsupported Parameters

Table A–1 describes the networking parameters no longer supported.

Table A–1 Unsupported Networking Parameters

File	Parameter	Description	Last Supported Release
tnsnames.ora	COMMUNITY	The parameter was a required part of all network service addresses. Thus, it appears anywhere you might find an address (for example, local naming and listener configuration files).	8.0
sqlnet.ora	AUTOMATIC_IPC	This parameter was used to force sessions through IPC addresses. Due to performance issues, this parameter has been removed. Configure an IPC address instead.	8.0
sqlnet.ora	NAMES.DEFAULT_ZONE	This parameter used to be included in profiles as slight variants of the NAMES.DEFAULT_DOMAIN parameter.	8.0
sqlnet.ora	NAMES.NDS.NAME.CONTEXT	This parameter was used to configure naming contexts for NDS external naming.	8.1
sqlnet.ora	OSS.SOURCE_MY_WALLET	This parameter's name has changed to WALLET_LOCATION.	8.1
sqlnet.ora	SQLNET.CRYPTO_SEED	This parameter was used to seed a random number generator for Oracle Advanced Security. In 10*i*, Oracle Advanced Security uses a random number generator that does not to require a user-supplied seed value.	9.2
sqlnet.ora	SQLNET.IDENTIX_FINGERPRINT_DATABASE SQLNET.IDENTIX_FINGERPRINT_DATABASE_USER SQLNET.IDENTIX_FINGERPRINT_DATABASE_PASSWORD SQLNET.IDENTIX_FINGERPRINT_METHOD	These parameters supported the Identix authentication method.	8.1
listener.ora	CONNECT_TIMEOUT	This parameter specified the amount of time that the listener waited for a client's request after the transport connection had been established. Use the INBOUND_CONNECT_TIMEOUT_listener_name parameter.	8.1
listener.ora	PRESPAWN_DESC PRESPAWN_LIST PRESPAWN_MAX	These parameters were used for prespawned dedicated server configuration. Prespawned dedicated servers are no longer supported. Use shared server instead.	8.1
listener.ora	USE_PLUG_AND_PLAY_*listener_name*	This parameter instructed the listener to register database information with an Oracle Names server during startup.	8.1
names.ora	All parameters	Oracle Names is no longer supported.	9.2

Unsupported Control Utility Commands

Table A–2 describes the control utility commands not supported in release 9.0.

Table A–2 Unsupported Network Control Utility Commands

Control Utility	Commands	Description	Last Supported Release
Oracle Names Control Utility	All commands	Oracle Names is no longer supported.	9.2

Listener Control Utility	`DBSNMP_START`
	`DBSNMP_STATUS`
	`DBSNAMP_STOP`
		These commands controlled the Oracle Intelligent Agent for use with Oracle Enterprise Manager. You can now control the Oracle Intelligent Agent through the Oracle Enterprise Manager Console.	8.1

Listener Control Utility	`SET CONNECT_TIMEOUT` `SHOW CONNECT_TIMEOUT`	These commands specified the amount of time that the listener waited for a client's request after the transport connection had been established.	8.1
Listener Control Utility	`SET USE_PLUGANDPLAY` `SHOW USE_PLUGANDPLAY`	These commands instructed the listener to register database information with an Oracle Names server.	8.1

B

Upgrade Considerations for Oracle Net Services

This appendix describes coexistence and upgrade issues for Oracle Net Services. This appendix covers the following topics:

- Overview of Unsupported Oracle Net Services Features
- Unsupported Parameters and Control Utility Commands
- Client and Database Coexistence Issues
- Using the Oracle Net Manager to Handle Compatibility Issues
- Upgrading to Oracle Net Services

Overview of Unsupported Oracle Net Services 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
- protocol.ora File
- Prespawned Dedicated Servers
- Protocols

Identix and SecurID Authentication Methods

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

- RADIUS
- Kerberos
- SSL

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

NDS External Naming and NDS Authentication

Support for Novell Directory Services (NDS) as an authentication method and as an external naming method are no longer supported. If you are using NDS as an external naming method, Oracle Corporation recommends using directory naming instead.

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.

protocol.ora File

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 Database Net Services Reference* for a description of these parameters

If you have a `protocol.ora` file in `$ORACLE_HOME/network/admin` on UNIX and *`ORACLE_HOME`*`\network\admin` on Windows, Oracle Net Manager, when first started, will automatically merge its 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.

Prespawned Dedicated Servers

Prespawned dedicated server processes are no longer supported. Instead, configure shared server (formerly named multi-threaded server) to improve scalability and system resource usage.

Protocols

Protocol addresses using the SPX or LU6.2 protocol must be replaced. Oracle Net provides support for the following network protocols:

- TCP/IP
- TCP/IP with SSL
- Named Pipes

> **See Also:** *Oracle Database Net Services Reference* for protocol parameter information

Unsupported Parameters and Control Utility Commands

> **See Also:** *Oracle Database Net Services Reference* for further information about unsupported configuration parameters and control utility commands

Client and Database Coexistence Issues

Clients and database servers require compatible releases of Oracle Net Services or Net8. For example, an Oracle9i client requires an installation of Oracle Net Services, and an Oracle9i database requires an installation of Oracle Net Services with the Oracle Net Listener.

Consider the following client-to-database connection issues before you decide if upgrading is appropriate for your environment:

- Oracle9i Database Connections
- Oracle8i or Oracle7 Database Connections
- Oracle Names

Oracle9i Database Connections

Connect descriptors, created for connections to an Oracle9i or an Oracle8 database, identify a database by its service name with the `SERVICE_NAME` parameter.

A connect descriptor to an Oracle9i or Oracle8 database uses the parameter `SERVICE_NAME`, as shown in the following example:

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

Connect descriptors that are currently configured with the `SID` parameter can remain. However, to take advantage of new features, such as client load balancing and connect-time failover, Oracle Corporation recommends replacing `SID` with `SERVICE_NAME`.

To modify a connect descriptor to use `SERVICE_NAME`, use the Oracle Net Manager's compatibility mode, as described in "Using the Oracle Net Manager to Handle Compatibility Issues" on page B-5.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for information about database identification by `SERVICE_NAME` rather than `SID`

Consider the following questions for an environment with release 8.0 clients connecting to an Oracle9i database:

- *Will my third-party applications be able use features of Oracle Net Services?*

  No. You must rebuild or upgrade applications to work with Oracle Net libraries.

- *Do my clients require Oracle Net to connect to a remote Oracle9i database?*

  No. If a client needs to connect to a *remote* Oracle9i database, only Net8 Client release 8.0 needs to be configured on the client. However, new features of Oracle Net Services are not available to these clients.

- *Do my clients require Oracle Net to connect to a local Oracle9i database?*

  No. The client requires an installation of Net8 Client release 8.0 in its Oracle home and the Oracle9i requires an installation of Oracle Net and Oracle Net Listener in its Oracle home.

Oracle8*i* or Oracle7 Database Connections

A connect descriptor to an Oracle release 8.0 or Oracle7 database uses `SID`, as shown in the following example:

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

In addition, the `listener.ora` file on the database server must be configured with the description of the `SID` for the release 8.0 database. In the following example, the listener is configured to listener for a database service called `sales.us.acme.com` that has a SID of `sales`:

```
SID_LIST_listener=
(SID_LIST=
 (SID_DESC=
  (GLOBAL_DBNAME=sales.us.acme.com)
  (SID_NAME=sales)))
```

> **See Also:** *Oracle Database Net Services Administrator's Guide* for information about database identification by `SID`

Consider the following questions for an environment with Oracle9i clients connecting to a release 8.0 database.

- *Do my clients require Net8 Client release 8.0 to connect to a remote Oracle release 8.0 database?*

  No. If a client needs to connect to a *remote* release 8.0 database, only Net8 Client of a compatible release needs to be configured on the client. The only limitation is that the new features available with Oracle Net Services are unavailable with this connection type.

- *Do my clients require Net8 Client release 8.0 to connect to a local release 8.0 database?*

  Yes. The client requires an installation of Oracle Net in its Oracle home and the release 8.0 database requires an installation of Net8 Server in its Oracle home.

Oracle Names

If you upgrade all or part of your network to Oracle9*i*, you should upgrade all the Oracle Names Servers in the region to version 9.

- *Can my release 8.0 clients use Oracle Names version 9 to resolve service names?*

  Yes.

- *Can my release 8.0 clients then use the connect descriptor returned from Oracle Names version 9 to connect to an Oracle version 8 database?*

  Yes, if the connect descriptor was specified correctly when it was entered into Oracle Names.

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 upgrading an existing Oracle Names configuration to directory naming, as described in the *Oracle Database Net Services Administrator's Guide*.

Using the Oracle Net Manager to Handle Compatibility Issues

Because some parameters are enabled only for Oracle9*i* and release 8.1, Oracle Net Manager offers two options that permit you to set the proper parameters in the `tnsnames.ora` file for clients connecting to a particular release of the database. These options are described in Table B–1.

Table B–1 Oracle Net Manager Options

Oracle Net Manager Option	Description
Use Options Compatible with Net8 8.0 Clients	Enables you to configure multiple addresses parameters for a client. If selected, enables the `SOURCE_ROUTE` parameter for pre-release 8.1 clients requiring Oracle Connection Manager connections. If turned off, enables you to use the `SOURCE_ROUTE`, `LOAD_BALANCE`, and `FAILOVER` parameters for Oracle9*i* and release 8.1 clients. **See Also:** *Oracle Database Net Services Administrator's Guide* for information about configuring address list parameters
Use Oracle8 Release 8.0 Compatible Identification	Enables you to configure parameters specific to a database release in the `CONNECT_DATA` section of a connect descriptor. If turned on, allows you to enter the SID of the release 8.0 or Oracle7 database. If turned off, enables you to enter the Oracle9i or Oracle8 database service name (`SERVICE_NAME`). **Note:** The *Advanced Service Options* dialog box, which is visible when the Advanced button in the Service Identification group is chosen, is also affected by whether this option is turned on or off. Some settings are only available for connections to an Oracle9i or Oracle8 database service. **See Also:** *Oracle Database Net Services Administrator's Guide* for information about configuring advanced connect data parameters

Upgrading to Oracle Net Services

To upgrade from SQL*Net release 2.*x* to Oracle Net Services or upgrade from Net8 release 8.0 or 8.1, complete these tasks:

Step 1: Verify Service Name and Instance Name

Step 2: Perform Software Upgrade on the Database Server

Step 3: Perform Software Upgrade on the Client

Step 4: Perform Functional Upgrade

Step 1: Verify Service Name and Instance Name

If you want to identify a service and its instance in the `tnsnames.ora` file, ensure that the `SERVICE_NAMES` and `INSTANCE_NAMES` initialization parameters are set in the initialization parameter file.

Table B–2 SERVICE_NAMES and INSTANCE NAMES Parameters

Parameter	Description
`SERVICE_NAMES`	Specifies one or more names for the database service to which this instance connects. You can specify multiple services names in order to distinguish among different uses of the same database. For example: `SERVICE_NAMES = sales.us.acme.com, widgetsales.us.acme.com` If you do not qualify the names in this parameter with a domain, Oracle qualifies them with the value of the `DB_DOMAIN` parameter. If `DB_DOMAIN` is not specified, Oracle uses the domain of your local database as it currently exists in the data dictionary. **Note:** You can change the value of `SERVICE_NAMES` parameter dynamically with the SQL `ALTER SYSTEM` when the database is running. See the *Oracle Database Reference* for further information about this parameter
`INSTANCE_NAME`	Specifies the unique name of this instance. Set the instance name to the value of the Oracle System Identifier (SID).

Step 2: Perform Software Upgrade on the Database Server

To perform a software upgrade on the database server, install the latest release of Oracle Net and Oracle Net Listener from the Oracle Universal Installer to receive the latest executables.

You are prompted to upgrade a database with the Database Upgrade Assistant if the Oracle Universal Installer detects a pre-release 9.2 database on your system. If you do not want to upgrade during the installation process, then you can choose to install this assistant and use it later.

The Oracle Universal Installer automatically performs these tasks:

- Stops older listener
- Starts release 9.2 listener

Step 3: Perform Software Upgrade on the Client

To perform a software upgrade on the client, install the latest release of Oracle Net Services from the Oracle Universal Installer to receive the latest executables.

Step 4: Perform Functional Upgrade

After the software is upgraded, it is not required to upgrade the configuration files unless you want to use the Oracle9i features. To take advantage of new features, review the following configuration files:

- `sqlnet.ora`
- `tnsnames.ora`
- `listener.ora`
- `protocol.ora`

Replace obsolete or renamed parameters.

See Also: *Oracle Database Net Services Reference* for further information about unsupported configuration parameters

tnsnames.ora

Replace the `SID` parameter with the `SERVICE_NAME` parameter to connect to a release 8.1 or higher service, as in the following example.

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

If you have multiple addresses, you can configure client load balancing and connect-time failover features, as in the following example.

```
sales=
(DESCRIPTION=
 (ADDRESS_LIST=
  (FAILOVER=on)
  (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)))
```

See Also:

- "Using the Oracle Net Manager to Handle Compatibility Issues" on page B-5 for information about configuring the service name and multiple address features
- *Oracle Database Net Services Administrator's Guide* for information about multiple addresses

listener.ora

Because instance information is registered with the listener in Oracle9*i*, it is no longer necessary to include the instance information with the `SID_LIST_`*`listener_name`* section of the `listener.ora` file.

However, Oracle Enterprise Manager still requires static information in the `listener.ora` file. If you are using Oracle Enterprise Manager to manage database objects, the `listener.ora` file must be configured with information about the database in the following manner:

```
SID_LIST_listener_name=
  (SID_LIST=
     (SID_DESC=
        (GLOBAL_DBNAME=global_database_name)
        (ORACLE_HOME=oracle_home)
        (SID_NAME=sid)))
```

Table B–3 Parameter Descriptions

Parameter	Description
`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.

Table B–3 (Cont.) Parameter Descriptions

Parameter	Description
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	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, such as in a Real Application Clusters environment, Oracle Corporation recommends not setting the GLOBAL_DBNAME parameter.

See Also: *Oracle Database Net Services Administrator's Guide* for information about configuring service information and connect-time failover and Transparent Application Failover (TAF)

C

LDAP Schema for Oracle Net Services

This appendix describes the **Oracle schema object class**es and **attribute**s defined in the **directory server** for Oracle Net Services objects. It does not describe object classes and attributes reserved for future functionality or used by other Oracle products.

This appendix contains these topics:

Structural Object Classes

The Oracle schema supports the following structural object classes for Oracle Net **directory naming** lookups:

- orclDBServer
- orclNetService
- orclNetServiceAlias
- orclNetDescription
- orclNetDescriptionList
- orclNetAddress
- orclNetAddressList

orclDBServer

Description

Defines the attributes for database service entries

Attributes

`orclNetDescName`

`orclVersion`

orclNetService

Description

Defines the attributes for **net service name** entries

Attributes

orclNetDescName

orclVersion

orclNetServiceAlias

Description

Defines the attributes for **net service alias** entries

Attributes

orclNetDescName

orclVersion

orclNetDescription

Description

Specifies a **connect descriptor** containing the **protocol address** of the **listener** and the connect information to the service

Attributes

- orclNetAddrList
- orclNetInstanceName
- orclNetConnParamList
- orclNetFailover
- orclNetLoadBalance
- orclNetSdu
- orclNetServiceName
- orclNetSourceRoute
- orclSid
- orclVersion

orclNetDescriptionList

Description

Specifies a list of connect descriptors

Attributes

- orclNetDescList
- orclVersion

orclNetAddress

Description

Specifies a listener protocol address

Attributes

- orclNetAddressString
- orclNetProtocol
- orclVersion

orclNetAddressList

Description

Specifies a list of protocol addresses

Attributes

- orclNetAddrList
- orclNetFailover
- orclNetLoadBalance
- orclNetSourceRoute
- orclVersion

orclNetDescriptionAux1

Attributes

orclNetSendBufSize

orclNetReceiveBufSize

orclNetFailoverModeString

orclNetInstanceRole

orclNetAddressAux1

Attributes

orclNetHostname

Attributes

Table C–1 lists the attributes used for the object classes. This list is subject to change.

Table C–1 LDAP Schema Attributes for Oracle Net Services

Attribute	Description
orclCommonContextMap	Allows the mapping of more than one default oracleContext in the directory server.
orclNetAddrList	Identifies one or more listener protocol addresses
orclNetAddressString	Defines a listener protocol address
orclNetConnParamList	Placeholder for future connect data parameters
orclNetDescList	Identifies one or more connect descriptors
orclNetDescName	Identifies a connect descriptor or a list of connect descriptors
orclNetFailover	Turns connect-time failover on for a protocol address list
orclNetFailoverModeString	
orclNetHostname	
orclNetInstanceName	Specifies the instance name to access
orclNetInstanceRole	Specifies a connection to the primary or secondary instance of aRAC configuration
orclNetLoadBalance	Turns client load balancing on for a protocol address list
orclNetProtocol	Identifies the protocol used in the orclAddressString attribute
orclNetReceiveBufSize	Specifies the buffer space limit for receive operations of sessions.
orclNetSdu	Specifies the **session data unit (SDU)** size
orclNetSendBufSize	Specifies the buffer space limit for send operations of sessions.
orclNetServiceName	Specifies the database service name in the CONNECT_DATA portion
orclNetSourceRoute	Instructs Oracle Net to use each address in order until the destination is reached
orclSid	Specifies the **Oracle System Identifier (SID)** in the CONNECT_DATA portion of a connection descriptor
orclVersion	Specifies the version of software used to create the entry

Glossary

access control list (ACL)

The group of access directives that you define. The directives grant levels of access to specific data for specific clients 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

See **protocol address**.

alias

An alternative name for a network object in an Oracle Names server. An alias stores the name of the object is referencing. When a client requests a lookup of an alias, Oracle completes the lookup as if it is the referenced object.

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:

- RADIUS
- Kerberos
- **SSL**
- **Windows NT native authentication**

cache

Memory that stores recently-accessed data to 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.

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 method**s, client and server **logging** and **tracing**, the domain from which to request names, and other client options for **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.

CMADMIN (Connection Manager Administration)

An **Oracle Connection Manager** process that monitors the health of the listener and Oracle Connection Manager gateway processes, shutting down and starting processes as needed. CMADMIN registers information about gateway processes with the

listener and processes commands executed with the Oracle Connection Manager Control utility.

CMGW (Connection Manager gateway)

An **Oracle Connection Manager** process that receives client connections screened and forwarded by the listener located at the Oracle Connection Manager instance. The gateway process forwards the requests to the database server. In addition, it can multiplex or funnel multiple client connections through a single protocol connection.

connect data

A portion of the **connect descriptor** that defines the destination database **service name** or **Oracle System Identifier (SID)**. In the following example, `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 Oracle9*i* or Oracle8*i* 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 **connect descriptor** or a name that maps to a connect descriptor. A connect identifier can be a **net service name**, database **service name**, or **net service alias**. Users initiate a connect request by passing a username and password along with a connect identifier in a connect string for the service to which they wish to connect:

```
CONNECT username/password@connect_identifier
```

connect string

Information the user passes to a service to connect, such as username, password, and **connect identifier**:

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

connect-time failover

A client connect request is forwarded to a another listener if a 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.

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 Configuration Assistant

A tool that enables you to create, delete, and modify a database.

database link

A pointer that defines a one-way communication path from an Oracle database server to another database server. The link pointer is actually defined as an entry in a data dictionary table. To access the link, you must be connected to the local database that contains the data dictionary entry.

A database link connection is one-way in the sense that a client connected to local database A can use a link stored in database A to access information in remote database B, but users connected to database B cannot use the same link to access data in database A. If local users on database B want to access data on database A, then they must define a link that is stored in the data dictionary of database B.

The following database links types are supported:

- A **private database link** in a specific schema of a database. Only the owner of a private database link can use it.
- A **public database link** for a database. All users in the database can use it.

dedicated server

A server process that is dedicated to one client connection. Contrast with **shared server process**.

default domain

The **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.

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, **net service name**, or **net service alias** to a **connect descriptor** stored in a central directory server. A **directory server** provides central administration of directory naming objects, reducing the work effort associated with adding or relocating services.

directory server

A directory server that is accessed with the **Lightweight Directory Access Protocol (LDAP)**. Support of LDAP-compliant directory servers provides a centralized vehicle for managing and configuring a distributed Oracle network. The directory server can replace clientside and serverside localized `tnsnames.ora` 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 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 server in that domain. This enables an Oracle 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 **domain*s***. 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.

DNS

Domain Name System. See **Domain Name System (DNS)**.

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 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 third-party naming service, such as **NIS** or **CDS**.

external procedure

Function or procedure written in a third-generation language (3GL) that can be called from PL/SQL code. Only C is supported for external procedures.

failover

See **connect-time failover**.

firewall support

See **access control**.

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 caching performance.

FTP protocol

File Transfer Protocol. A client/server protocol which allows a user on one computer to transfer files to and from another computer over a TCP/IP network.

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.

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.

identity management realm

A collection of identities, all of which are governed by the same administrative policies. In an enterprise, all employees having access to the intranet may belong to one realm, while all external users who access the public applications of the enterprise may belong to another realm. An identity management realm is represented in the directory by a specific entry with a special object class associated with it.

instance

System Global Area (SGA) 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**.

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.

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**.

Java Database Connectivity (JDBC) Driver

A driver that provides Java applications and applets access to an Oracle database.

JDBC OCI Driver

A Type II driver for use with client/server Java applications. This driver requires an Oracle client installation.

JDBC Thin Driver

A Type IV driver for Oracle JDBC applets and applications. Because it is written entirely in Java, this driver is platform-independent. It does not require any additional Oracle software on the client side. The Thin driver communicates with the server using **Two-Task Common (TTC)**, a protocol developed by Oracle to access the database server.

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.

latency

Networking round-trip time.

Lightweight Directory Access Protocol (LDAP)

A standard, extensible directory access protocol. It is a common language that LDAP clients and servers use to communicate. The framework of design conventions supporting industry-standard **directory server**s.

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 Internet Directory Configuration Assistant or Oracle Net Configuration Assistant that contains the following directory server access information:

- Type of directory server
- Location of the directory server
- Default Oracle Context that the client or server will use to look up or configure connect identifiers for connections to database services

When created with Oracle Internet Directory Configuration Assistant, `ldap.ora` is located in the `$ORACLE_HOME/ldap/admin` directory on UNIX operating systems and the *`ORACLE_HOME`*`\ldap\admin` directory on Windows operating systems. When created with Oracle Net Configuration Assistant, `ldap.ora` is located in the `$ORACLE_HOME/network/admin` directory on UNIX operating systems and the *`ORACLE_HOME`*`\network\admin` directory on Windows operating systems.

link qualifier

A qualifier appended to a global database link to provide alternate settings for the database username 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.

When a client requests a network session with a database server, a listener receives the actual request. If the client information matches the listener information, then the listener grants a connection to the database 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.

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.

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.

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 four naming methods:

- **local naming**
- **directory naming**
- **easy connect naming**
- **external naming**

net service alias

An alternative name for a **directory naming** object in a directory server. A directory server stores net service aliases for any defined **net service name** or database service.

A net service alias entry does not have connect descriptor information. Instead, it only references the location of the object for which it is an alias. When a client requests a directory lookup of a net service alias, the directory determines that the entry is a net service alias and completes the lookup as if it was actually the entry it is referencing.

net service name

A simple name for a service that resolves to a **connect descriptor**. Users initiate a connect request by passing a username 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
- 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 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.

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)**.

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**.

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 following it and provides a service to the layer preceding.

Oracle Advanced Security

A product that provides a comprehensive suite of security features to protect enterprise networks and securely extends 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 Internet Directory** automatically creates an Oracle Context at the root of the DIT structure. This root Oracle Context has a DN of `dn:cn=OracleContext`.

Oracle Enterprise Manager

A separate Oracle product that combines a graphical console, agents, common services, and tools to provide an integrated and comprehensive systems management platform for managing Oracle products.

Oracle Identity Management

An infrastructure enabling deployments to manage centrally and securely all enterprise identities and their access to various applications in the enterprise.

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 Net

Communication software 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 identifier**s
- 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 listener

See **listener**.

Oracle Net Manager

A 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 identifier**s and map them to **connect descriptor**s to identify the network location and identification of a service. Oracle Net Manager supports configuration of connect descriptors in a local `tnsnames.ora` file or directory 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 is comprised of **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 XML DB

A high-performance XML storage and retrieval technology provided with Oracle database server. It is based on the W3C XML data model.

Oracle Real Application Clusters

An architecture that allows multiple instances to access a shared database of datafiles. Real Application Clusters is also a software component that provides the necessary cluster database scripts, initialization files, and datafiles needed for the Oracle Enterprise Edition and Real Application Clusters.

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)**.

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.

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.

presentation layer

A networking communication layer that manages the representation of information that application layer entities either communicate or reference in their communication. **Two-Task Common (TTC)** is an example of presentation layer.

private database link

A database link created by one user for his or her exclusive use.

See also **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** 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 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** and **private database link**.

realm Oracle Context

An Oracle Context contained in each **identity management realm**. It stores the following information:

- User naming policy of the identity management realm—that is, how users are named and located
- Mandatory authentication attributes
- Location of groups in the identity management realm
- Privilege assignments for the identity management realm—for example: who has privileges to add more users to the realm.
- Application specific data for that Realm including authorizations

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` is the RDN.

root Oracle Context

In the **Oracle Identity Management** infrastructure, the The root Oracle Context is an entry in Product_Name containing a pointer to the default **identity management realm** in the infrastructure. It also contains information on how to locate an identity management realm given a simple name of the realm.

RPC

Remote Procedure Call

SDP protocol

Sockets Direct Protocol (SDP).

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 comprising 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 (**dispatcher** or **dedicated server**) 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.

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

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 information for Oracle8*i* or later instances is automatically registered with the listener. Static configuration is also required for other services, such as **external procedure** calls and **Heterogeneous Services**.

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
- 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**.

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 maps **net service name***s* to **connect descriptor**s. This file is used for the **local naming** 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.

UPI

User Program Interface

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.

WebDAV protocol

World Wide Web Distributed Authoring and Versioning. A protocol with a set of extensions to the **HTTP protocol** which allows users to manage files on remote Web servers.

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

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W