Oracle® Database

Net Services Reference

18c

E83751-08

July 2021

Oracle Database Net Services Reference, 18c

E83751-08

Primary Author: Binika Kumar

Contributing Authors: Bharathi Jayathirtha, Robert Achacoso, Alan Williams, Abhishek Dadhich, Sarma Namuduri, Kevin Neel, Santanu Datta, Steve Ding, Feroz Khan, Thanigai Nallathambi, Yi Ouyang, Russ Lowenthal, Krishna Itikarlapalli, Peter Knaggs, Bhaskar Mathur, Scot McKinley, Sweta Mogra, Srinivas Pamu, Kant Patel, Hector Pujol, Murali Purayathu, Saravanakumar Ramasubramanian, Sudeep Reguna, Ching Tai, Norman Woo

Contents

Preface

Changes in This Release for Oracle Database Net Services Reference

Part I Control Utilities

1 Listener Control Utility

Part II Configuration Parameters

3 Syntax Rules for Configuration Files

4 Protocol Address Configuration

5 Parameters for the sqlnet.ora File

8 Oracle Connection Manager Parameters (cman.ora)

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 18c that apply to the Microsoft Windows and UNIX operating systems.

This preface contains the following 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 should be familiar with the networking concepts and configuration tasks described in *Oracle Database Net Services Administrator's Guide*.

Documentation Accessibility

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

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

For additional information, see the following Oracle resources:

- *Oracle Database Net Services Administrator's Guide*
- Online Help for Oracle Net Services tools and utilities
- Oracle Database 18c documentation set
- *Oracle Database Global Data Services Concepts and Administration Guide*

A glossary of Oracle Net Services terms is available in *Oracle Database 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 additional information about how these schemas were created and how you can use them yourself.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Changes in This Release for Oracle Database Net Services Reference

The following are the changes in *Oracle Database Net Services Reference* for Oracle Database 18c:

- New Features
- Deprecated Features

New Features

The following are the new features in Oracle Net Services:

- Read-only Oracle Home Support

  An Oracle home can be configured in a read-only mode, which prevents creation or modification of files inside the Oracle home (`ORACLE_HOME`) directory. A read-only Oracle home can be used as a software image that can be shared across multiple independent servers. This simplifies patching and mass rollout as only one Oracle home image needs to be updated to distribute a patch to multiple servers. In the read-only Oracle home mode, `ORACLE_BASE_`*HOME* is a home-specific directory located at `ORACLE_BASE/homes/`*HOME_NAME*.

  > **See Also:**
  >
  > – *Oracle Database Installation Guide for Linux*
  >
  > – Overview of Oracle Net Listener Configuration File

- New sqlnet.ora Parameters
  - `ACCEPT_MD5_CERTS` parameter replaces the `ORACLE_SSL_ALLOW_MD5_CERT_SIGNATURES` environment variable
  - `ACCEPT_SHA1_CERTS` parameter
  - `ADD_SSLV3_TO_DEFAULT` parameter

  > **See Also:**
  >
  > sqlnet.ora Profile Parameters

- Ability to Create a Keystore for Each Pluggable Database

  Starting with this release, each pluggable database (PDB) can have its own keystore, instead of there being only one keystore for the entire container database (CDB). The advantage of this feature is that it enables independent key management operations to be performed by each tenant (PDB) in a multitenant environment rather than having to share a keystore at the CDB root level. This feature benefits both multitenant and non-multitenant environments because it provides parameters to facilitate the configuration of the keystore location and the keystore type, eliminating the need for editing the `sqlnet.ora` file.

  This feature provides the following new functionality:

  - For multitenant environments, the following two modes:
    - United mode, in which the keystores and master encryption keys are primarily managed from the CDB root, and can be accessed from the united mode PDB. Within the PDB, the keystore can be opened and closed just for that PDB. You also can create a PDB-specific master encryption key for this keystore.
    - Isolated mode, in which the keystore and encryption keys are managed in an individual PDB. This way, each PDB can configure its own keystore type independently, and create and manage this keystore after configuring it.

    To accommodate these modes, the `ADMINISTER KEY MANAGEMENT` SQL statement has been enhanced to behave differently in the two modes.
  - For both non-multitenant and multitenant environments, the following are the new features:
    - Addition of the `WALLET_ROOT` static instance initialization parameter, to specify the keystore path. In this guide, `WALLET_ROOT` refers to the configuration of software keystores, hardware keystores, and Oracle Key Vault keystores, but this parameter can be used to designate the wallet location for other products as well: Enterprise User Security, Secure Sockets Layer, Oracle XML DB, and Secure External Password Store.
    - Addition of the `TDE_CONFIGURATION` dynamic instance initialization parameter, to specify the type of keystore to use. You can set this parameter for TDE software keystores, hardware security module keystores (HSMs), and Oracle Key Vault.
    - Modification to the behavior of the `SQLNET.ENCRYPTION_WALLET_LOCATION` parameter, to enable its use only if the `WALLET_ROOT` parameter has not been set

- Integration of Active Directory Services with Oracle Database

  With centrally managed users (CMU) Oracle database users and roles can map directly to Active Directory users and groups without using Oracle Enterprise User Security (EUS) or another intermediate directory service. EUS is not being replaced or deprecated; this new feature is another simpler option if you only want to authenticate and authorize users with Active Directory.

  The direct integration with directory services supports better security through faster and easier configuration with the enterprise identity management architecture. In the past, users may have avoided integrating the database with directory services due to the difficulty and complexity. Centrally managed users allows the Oracle database to directly connect with Active Directory

- Support for Oracle Connection Manager in Traffic Director Mode

  This feature provides improved high availability and performance for both planned and unplanned outages with the help of new `cman.ora` parameters. Some of the existing parameters that support Oracle Connection Manager in Traffic Director Mode are inbound_connect_timeout, min_gateway_processes, max_gateway_processes, and max_connections.

> **See Also:**
>
> – Oracle Connection Manager in Traffic Director Mode Parameters
> – *Oracle Database Net Services Administrator's Guide*

Deprecated Features

The following feature is deprecated in this release:

Deprecation of Weak Native Network Encryption and Integrity Algorithms

The DES, DES40, 3DES112, 3DES168, RC4_40, RC4_56, RC4_128, RC4_256, and MD5 algorithms are deprecated in this release.

As a result of this deprecation, Oracle recommends that you review your network encryption and integrity configuration to check if you have specified any of the deprecated weak algorithms.

To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

Part I

Control Utilities

Oracle Net Services provides control utilities to administer listeners, and Oracle Connection Manager. 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:

- Listener Control Utility
- Oracle Connection Manager Control Utility

1 Listener Control Utility

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

> **Note:**
>
> The terms "SQL*Net" and "Net Services" are used interchangeably throughout Oracle documentation and both these terms refer to the same functionality.

This chapter contains the following topics:

- Listener Control Utility Overview
- SET and SHOW Commands of the Listener Control utility
- Distributed Operations
- Oracle Net Listener Security
- Listener Control Utility Commands

1.1 Listener Control Utility Overview

The Listener Control utility enables you to administer listeners.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
```

In the preceding command, *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
```

You can combine commands in a standard text file, and then run them as a sequence of commands. To run 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 processing.

For most 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 must 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
- Naming method, for example, a `tnsnames.ora` file

If none of the preceding mechanisms resolve the listener name, then the Listener Control utility uses the default listener name `LISTENER`, resolves the host name IP address, and uses port 1521.

The Listener Control utility supports the following types of commands:

- Operational commands, such as START, and STOP.
- Modifier commands, such as SET TRC_LEVEL.
- Informational commands, such as STATUS and SHOW LOG_FILE.

1.2 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 to administer using the `SET CURRENT_LISTENER` command. Parameter values remain in effect until the listener is shut down. If you want these settings to persist, then 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.

1.3 Distributed Operations

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

The following procedure describes how to set up a computer to remotely administer a listener:

1. Ensure that the Listener Control utility (`lsnrctl`) executable is installed in the `ORACLE_HOME/bin` directory.
2. Ensure that the name of the listener to administer can be resolved through a `listener.ora` file or a naming method, as described in "Listener Control Utility Overview".

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.

1.4 Oracle Net Listener Security

Local listener administration is secure through local operating system authentication, which restricts listener administration to the user who started the listener or to the super user. By default, remote listener administration is disabled.

Oracle recommends that you perform listener administration in the default mode, and access the system remotely using a remote login. When you administer the listener remotely, use Oracle Enterprise Manager Cloud Control or Secure Shell (SSH) to access the remote host.

1.5 Listener Control Utility Commands

This section describes the following Listener Control utility commands:

- EXIT
- HELP
- QUIT
- RELOAD
- SAVE_CONFIG
- SERVICES
- SET
- SET CURRENT_LISTENER
- SET DISPLAYMODE
- SET INBOUND_CONNECT_TIMEOUT
- SET LOG_DIRECTORY
- SET LOG_FILE
- SET LOG_STATUS
- SET SAVE_CONFIG_ON_STOP
- SET TRC_DIRECTORY
- SET TRC_FILE
- SET TRC_LEVEL
- SHOW
- SPAWN
- START
- STATUS
- STOP
- TRACE
- VERSION

1.5.1 EXIT

Purpose

To exit from the Listener Control utility, and return to the operating system prompt.

Prerequisites

None

Syntax

From the Listener Control utility:

```
LSNRCTL> EXIT
```

Arguments

None

Usage Notes

This command is identical to the QUIT command.

Example

```
LSNRCTL> EXIT
```

1.5.2 HELP

Purpose

To provide a list of all the Listener Control utility commands or provide syntax help for a particular Listener Control utility command.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl HELP command
```

From the Listener Control utility:

```
LSNRCTL> HELP command
```

Arguments

command: The Listener Control utility 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:
exit
quit
reload
services
set*
show*
spawn
start
status
stop
trace
version
```

1.5.3 QUIT

Purpose

To exit from the Listener Control utility and return to the operating system prompt.

Prerequisites

None

Syntax

From the Listener Control utility:

```
LSNRCTL> QUIT
```

Arguments

None

Usage Notes

This command is identical to the EXIT command.

Example

```
LSNRCTL> QUIT
```

1.5.4 RELOAD

Purpose

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 are unregistered and subsequently registered again.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl RELOAD listener_name
```

From the Listener Control utility:

```
LSNRCTL> RELOAD listener_name
```

Arguments

listener_name: 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
```

1.5.5 SAVE_CONFIG

Purpose

To save 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. Before modification of the `listener.ora` file, a backup of the file, called `listener.bak`, is created.

Syntax

From the operating system:

```
lsnrctl SAVE_CONFIG listener_name
```

From the Listener Control utility:

```
LSNRCTL> SAVE_CONFIG listener_name
```

Arguments

listener_name: 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.

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
```

1.5.6 SERVICES

Purpose

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

Syntax

From the operating system:

```
lsnrctl SERVICES listener_name
```

From the Listener Control utility:

```
LSNRCTL> SERVICES listener_name
```

Arguments

listener_name: The listener name, if the default name of `LISTENER` is not used.

Usage Notes

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

> **See Also:**
>
> *Oracle Database Net Services Administrator's Guide* for a complete description of `SERVICES` 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.example.com` and `sales2.us.example.com`, with a total of three service handlers.
- Service `sales1.us.example.com` is handled by one dispatcher only.
- Service `sales2.us.example.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.example.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.example.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
```

1.5.7 SET

Purpose

To alter the parameter values for the listener. Parameter value 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

Syntax

From the operating system:

```
lsnrctl SET parameter
```

From the Listener Control utility:

```
LSNRCTL> SET parameter
```

Arguments

parameter: 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, then use the SET CURRENT_LISTENER command to set the name of the listener to administer.

Example

```
LSNRCTL> SET
The following operations are available with set.
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
trc_file
trc_directory
trc_level
```

1.5.8 SET CURRENT_LISTENER

Purpose

To set the name of the listener to administer. Subsequent commands that would normally require `listener_name` can be issued without it.

Syntax

From the Listener Control utility:

```
LSNRCTL> SET CURRENT_LISTENER listener_name
```

Arguments

listener_name: 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 that was set. You do not have to specify the name of the listener.

Example

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

1.5.9 SET DISPLAYMODE

Purpose

To change the format and level of detail for the SERVICES and STATUS commands.

Syntax

From the Listener Control utility:

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

Arguments

Specify one of the following modes:

`compat`: Output that is compatible with earlier releases of the listener.

`normal`: Output that is formatted and descriptive. Oracle recommends this mode.

`verbose`: All data received from the listener in a formatted and descriptive output.

`raw`: All data received from the listener without any formatting. This argument should be used only if recommended by Oracle Support Services.

Example

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

1.5.10 SET INBOUND_CONNECT_TIMEOUT

Purpose

To specify the time, in seconds, for the client to complete its connect request to the listener after establishing the network connection.

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 additional information about specifying the time out for client connections

Syntax

From the Listener Control utility:

```
LSNRCTL> SET INBOUND_CONNECT_TIMEOUT time
```

Arguments

time: 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.
```

1.5.11 SET LOG_DIRECTORY

> **Note:**
>
> This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled, and use the log directory `ORACLE_HOME/log/diag/`*`product_type`*.

Purpose

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.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl SET LOG_DIRECTORY directory
```

From the Listener Control utility:

```
LSNRCTL> SET LOG_DIRECTORY directory
```

Arguments

directory: 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
```

1.5.12 SET LOG_FILE

> **Note:**
>
> This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled, and use the log directory `ORACLE_HOME/log/diag/`*`product_type`*.

Purpose

To set the name for the listener log file. By default, the log file name is `listener.log`.

Prerequisites

None

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: The 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
```

1.5.13 SET LOG_STATUS

Purpose

To turn listener logging on or off.

Prerequisites

None

Syntax

From the operating system:

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

From the Listener Control utility:

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

Arguments

`on`: To turn logging on.

`off`: 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
```

1.5.14 SET SAVE_CONFIG_ON_STOP

Purpose

To specify whether 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. Before modification of the `listener.ora` file, a backup of the file, called `listener.bak`, is created.

To have all parameters saved immediately, use the SAVE_CONFIG 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: To save configuration to `listener.ora`.

`off`: 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
```

1.5.15 SET TRC_DIRECTORY

> **Note:**
>
> This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled, and use the log directory `ORACLE_HOME/log/diag/`*`product_type`*.

Purpose

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.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl SET TRC_DIRECTORY directory
```

From the Listener Control utility:

```
LSNRCTL> SET TRC_DIRECTORY directory
```

Arguments

directory: 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
```

1.5.16 SET TRC_FILE

> **Note:**
>
> This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled, and use the log directory `ORACLE_HOME/log/diag/`*`product_type`*.

Purpose

To set the name of the listener trace file. By default, the trace file name is `listener.trc`.

Prerequisites

None

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: 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
```

1.5.17 SET TRC_LEVEL

Purpose

To set a specific level of tracing for the listener.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl SET TRC_LEVEL level
```

From the Listener Control utility:

```
LSNRCTL> SET TRC_LEVEL level
```

Arguments

level: 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
```

1.5.18 SHOW

Purpose

To view the current parameter values for the listener.

All the SET parameters have equivalent `SHOW` parameters.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl SHOW parameter
```

From the Listener Control utility:

```
LSNRCTL> SHOW parameter
```

Arguments

parameter: A `SHOW` parameter to view its configuration settings. Parameters are shown in the example output.

When you enter `SHOW` 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
trc_file
trc_directory
trc_level
```

1.5.19 SPAWN

Purpose

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

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: The listener name, if the default name of `LISTENER` is not used.

alias: The alias of the program to be spawned off is specified by a `listener.ora` file entry, similar to the following:

```
alias = (PROGRAM=(NAME=)(ARGS=)(ENVS=))
```

For example:

```
nstest = (PROGRAM=(NAME=nstest)(ARGS=test1)(ENVS='ORACLE_HOME=/usr/oracle'))
```

Example

The `nstest` program, shown in the preceding section, can then be spawned off using the following command:

```
lsnrctl SPAWN listener_name nstest
```

1.5.20 START

Purpose

To start the named listener.

Prerequisites

Listener must not be running.

Syntax

From the operating system:

```
lsnrctl START listener_name
```

From the Listener Control utility:

```
LSNRCTL> START listener_name
```

> **Note:**
>
> The utility may prompt for a password on Microsoft Windows if the database was installed with the Oracle Home User. The password is the operating system password for the Oracle Home User. The prompt is displayed only if the listener service does not exist and needs to be created as part of starting the listener.

Arguments

listener_name: 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/sales_group/sales/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 18.0.0.0.0
System parameter file is $ORACLE_HOME/network/admin/listener.ora
Log messages written to $ORACLE_BASE/diag/tnslsnr/node_name/listener/alert/log.xml
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 Linux: Version 18.0.0.0.0
Start Date                21-JAN-2018 21:50:49
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
```

```
Security                  ON: Local OS Authetication
SNMP                      OFF
Listener Parameter File  $ORACLE_HOME/network/admin/listener.ora
Listener Log File        $ORACLE_BASE/diag/tnslsnr/node_name/listener/alert/
log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)))
The listener supports no services
The command completed successfully
```

> **See Also:**
>
> *Oracle Database Platform Guide for Microsoft Windows* for information about the Oracle Home User

1.5.21 STATUS

Purpose

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.

> **Note:**
>
> You can also obtain the status of the listener through the Oracle Enterprise Manager Cloud Control console.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl STATUS listener_name
```

From the Listener Control utility:

```
LSNRCTL> STATUS listener_name
```

Arguments

listener_name: The listener name, if the default name of `LISTENER` is not used.

Usage Notes

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

See Also:

Oracle Database Net Services Administrator's Guide for a complete description of `STATUS` 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 Linux: Version 18.0.0.0.0 -
Production
Start Date                12-JAN-2018 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.example.com" has 1 instance(s).
  Instance "sales", status READY, has 1 handler(s) for this service...
Service "sales2.us.example.com" has 1 instance(s).
  Instance "sales", status READY, has 2 handler(s) for this service...
The command completed successfully
```

1.5.22 STOP

Purpose

To stop the named listener.

Prerequisites

The listener must be running.

Syntax

From the operating system:

```
lsnrctl STOP listener_name
```

From the Listener Control utility:

```
LSNRCTL> STOP listener_name
```

Arguments

listener_name: 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
```

1.5.23 TRACE

Purpose

To set tracing for the listener.

Syntax

From the operating system:

```
lsnrctl trace level listener_name
```

From the Listener Control utility:

```
LSNRCTL> trace level listener_name
```

Arguments

level: 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 the `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
```

1.5.24 VERSION

Purpose

To display the current version of Listener Control utility.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl VERSION listener_name
```

From the Listener Control utility:

```
LSNRCTL> VERSION listener_name
```

Arguments

listener_name: The listener name, if the default name of `LISTENER` is not used.

Example

```
LSNRCTL> version listener
Connecting to ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
TNSLSNR for Linux: Version 18.0.0.0.0
        TNS for Linux: Version 18.0.0.0.0
        Oracle Bequeath NT Protocol Adapter for Linux: Version 18.0.0.0.0
        Unix Domain Socket IPC NT Protocol Adaptor for Linux: Version 18.0.0.0.0
        TCP/IP NT Protocol Adapter for Linux: Version 18.0.0.0.0
The command completed successfully
```

2

Oracle Connection Manager Control Utility

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

This chapter contains the following topics:

2.1 Oracle Connection Manager Control Utility Overview

The Oracle Connection Manager Control utility enables you to administer Oracle Connection Managers. 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.

2.2 Command Modes and Syntax

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

```
cmctl command [argument]
```

The Oracle Connection Manager Control utility supports the following 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:**
>
> You can use SET commands to dynamically alter configuration parameters. The changes only remain in effect until 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 using the command SAVE_PASSWD .

You can use the Oracle Connection Manager Control utility in command mode, or batch mode.

- Using command mode:
  - From the Oracle Connection Manager Control utility:

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

```
cmctl
CMCTL> command
```

- From the operating system:

  Enter the entire command from the operating system command prompt:

  ```
  cmctl [command] [argument1 . . . argumentN] [-c instance_name]
  ```

  Each command issued this way can have an Oracle Connection Manager instance name appended as an argument. If an Oracle Connection Manager instance name is not provided, then the default instance name is assumed. The default name is `cman_`*`hostname`*. You may be prompted for a password if one was set in a previous CMCTL session. Issuing commands from an Oracle Connection Manager Control utility session of Oracle Connection Manager requires that a password be entered once, at the beginning of the session, if one has been set.

> **Caution:**
>
> There is an option to specify the password on the command line. However, doing so exposes the password on the screen, and is a potential security risk. Oracle recommends not using the password option (`-p`) on the command line.

- Using batch mode:

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

  ```
  cmctl @input_file
  ```

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

2.3 Oracle Connection Manager Control Utility Commands

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

- ADMINISTER
- CLOSE CONNECTIONS
- EXIT
- HELP
- QUIT
- RELOAD

- RESUME GATEWAYS
- SAVE_PASSWD
- SET
- SET ASO_AUTHENTICATION_FILTER
- SET CONNECTION_STATISTICS
- SET EVENT
- SET IDLE_TIMEOUT
- SET INBOUND_CONNECT_TIMEOUT
- SET LOG_DIRECTORY
- SET LOG_LEVEL
- SET OUTBOUND_CONNECT_TIMEOUT
- SET PASSWORD
- SET SESSION_TIMEOUT
- SET TRACE_DIRECTORY
- SET TRACE_LEVEL
- SHOW
- SHOW ALL
- SHOW CONNECTIONS
- SHOW DEFAULTS
- SHOW EVENTS
- SHOW GATEWAYS
- SHOW PARAMETERS
- SHOW RULES
- SHOW SERVICES
- SHOW STATUS
- SHOW VERSION
- SHUTDOWN
- STARTUP
- SUSPEND GATEWAY

2.3.1 ADMINISTER

Purpose

To select an Oracle Connection Manager instance.

Prerequisites

None

Syntax

From the Oracle Connection Manager Control utility:

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

Arguments

instance_name: The instance name of Oracle Connection Manager that you would like to administer. Instances are defined in the `cman.ora` file.

Usage Notes

You can issue the `ADMINISTER` command only within the utility. You cannot issue the command from the operating system.

`ADMINISTER` enables you to choose which Oracle Connection Manager instance to administer. To start the Oracle Connection Manager instance, you must issue the STARTUP command.

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

Use the `-c` option when to administer an instance that is not the local instance.

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

Example

```
CMCTL> ADMINISTER cman_indl040ad
Enter CMAN password: password
Current instance cman_indl040ad is already started
Connections refer to (address=(protocol=TCP)(host=indl040ad)(port=1560)).
The command completed successfully
```

2.3.2 CLOSE CONNECTIONS

Purpose

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]
```

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

state: 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, then `CLOSE CONNECTIONS` defaults to all possible states. If the time qualifier is included under these conditions, then the time specified is the amount of time that has elapsed since a client initiated a connection.

time: The time format. Use the following format to specify connections greater than the time indicated:

```
gt[hh:mm:]ss
```

source: The source address. Use one of the following formats to specify the source address:

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

destination: The destination address. Use one of the following formats to specify the destination address:

- `to` *`IP`*
- `to` *`hostname`*
- `to` *`subnet`*

service: The service name. Use the `service_name` parameter to specify the service, such as `sales.us.example.com`.

gateway_process_id: The gateway process identifier is a number. Use this number to specify connections that are proxied by the gateway process indicated. To determine the gateway process identifier, use the Oracle Connection Manager control utility `show gateways` command.

connect_identifier_list: The connection identifiers. Use a space between multiple connection identifiers in a list.

Usage Notes

Because the `CLOSE CONNECTIONS` command terminates 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 example 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, and the destination is the specified host name.

```
CMCTL> CLOSE CONNECTIONS gt 1:30:00 from 192.0.2.32/24 to host1
```

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

```
CMCTL> CLOSE idle CONNECTIONS gt 30:00 using 0
```

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

```
CMCTL> CLOSE established CONNECTIONS for sales.us.example.com
```

2.3.3 EXIT

Purpose

To exit from the Oracle Connection Manager Control utility.

Prerequisites

None

Syntax

From the operating system:

```
cmctl EXIT [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> EXIT
```

Usage Notes

This command is identical to the QUIT command.

Example

```
CMCTL> EXIT
```

2.3.4 HELP

Purpose

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]
```

From the Oracle Connection Manager Control utility:

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

Arguments

command: Specify a HELP command. Commands are shown in the following sample 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> HELP
The following operations are available
An asterisk (*) denotes a modifier or extended command:

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

2.3.5 QUIT

Purpose

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 EXIT command.

Example

```
CMCTL> QUIT
```

2.3.6 RELOAD

Purpose

To dynamically reread parameters and rules.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl RELOAD [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> RELOAD
```

Usage Notes

Configuration information modified using the `RELOAD` command applies only to new connections. Existing connections are unaffected. The `SET RELOAD` command restores configurations set in `cman.ora`, and override the `SET` command.

`RELOAD` reregisters gateways with the Oracle Connection Manager listener during which some new connections might be refused until the registration process is complete.

Example

```
CMCTL> RELOAD
The command completed successfully
```

2.3.7 RESUME GATEWAYS

Purpose

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]
```

From the Oracle Connection Manager Control utility:

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

Arguments

gateway_process_id: One or more gateway processes to reopen. Separate multiple gateway processes using a space between the process identifiers.

Usage Notes

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

Example

```
CMCTL> RESUME GATEWAYS 1
The command completed successfully
```

2.3.8 SAVE_PASSWD

Purpose

To save the current password to the `cman.ora` file, the configuration file for Oracle Connection Manager.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SAVE_PASSWD [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SAVE_PASSWD
```

Usage Notes

If you run this command, then the next session of Oracle Connection Manager uses the password. The password is stored in an encrypted format in the `cman.ora` file.

Example

```
CMCTL> SAVE_PASSWD
```

2.3.9 SET

Purpose

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> 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
```

2.3.10 SET ASO_AUTHENTICATION_FILTER

Purpose

To indicate whether the client must use Oracle Database 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]
```

From the Oracle Connection Manager Control utility:

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

Arguments

`on`: To reject connections that are not using Secure Network Service (SNS) to perform client authentication. SNS is part of Oracle Database security.

`off`: To specify whether no authentication is required for client connections. This is the default.

Example

```
CMCTL> set aso_authentication_filter ON
CMAN_user.us.example.com parameter aso_authentication_filter set to ON
The command completed successfully
```

2.3.11 SET CONNECTION_STATISTICS

Purpose

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]
```

From the Oracle Connection Manager Control utility:

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

Arguments

`yes`: To have gateway processes collect connection statistics.

`no`: To not have gateway processes collect connection statistics. This is the default.

Usage Notes

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

Example

```
CMCTL> set connection_statistics ON
CMAN_user.us.example.com parameter connection_statistics set to ON
The command completed successfully
```

2.3.12 SET EVENT

Purpose

To log information for a particular event.

Syntax

From the operating system:

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

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 event group.
- `memory_ops`: Memory operations event group.
- `conn_hdlg`: Connection handling event group.
- `proc_mgmt`: Process management event group.
- `reg_and_load`: Registration and load update event group.
- `wake_up`: Events related to Connection Manager Administration (CMADMIN) wakeup queue event group.
- `timer`: Gateway timeouts event group.

- `cmd_proc`: Command processing event group.
- `relay`: Events associated with connection control blocks event group.

`on`: To turn an event group on.

`off`: To turn an event group off.

Usage Notes

The `SET EVENT` command accepts only one argument. To log multiple events, you must issue the command for each event separately.

Example

```
CMCTL> set event memory_ops off
cman11 event memory_ops set to OFF.
The command completed successfully
```

2.3.13 SET IDLE_TIMEOUT

Purpose

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]
```

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 (zero), which disables this feature.

Example

```
CMCTL> SET IDLE_TIMEOUT 30
CMAN_user.us.example.com parameter idle_timeout set to 30
The command completed successfully
```

2.3.14 SET INBOUND_CONNECT_TIMEOUT

Purpose

To specify the maximum amount of time the Oracle Connection Manager listener waits 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]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET INBOUND_CONNECT_TIMEOUT [time]
```

Arguments

time: The inbound connect timeout in seconds. The default is 0 (zero), which disables this feature.

Example

```
CMCTL> SET INBOUND_CONNECT_TIMEOUT 30
CMAN_user.us.example.com parameter inbound_connect_timeout set to 30
The command completed successfully
```

2.3.15 SET LOG_DIRECTORY

> **Note:**
>
> This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled, and use the log directory `ORACLE_HOME/log`.

Purpose

To designate where the log files for 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]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET LOG_DIRECTORY [directory_path]
```

Arguments

directory_path: The location of the log directory. The default path is as follows:

- Linux and UNIX:

`ORACLE_HOME/network/log directory`

- Microsoft Windows:

  `ORACLE_HOME\network\log directory`

Usage Notes

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

Example

```
CMCTL>
SET LOG_DIRECTORY /disk1/user_cman_test/oracle/network/admin

CMAN_user.us.example.com parameter log_directory set to
/disk1/user_cman_test/oracle/network/admin

The command completed successfully
```

2.3.16 SET LOG_LEVEL

Purpose

To set the log level for Oracle Connection Manager.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET LOG_LEVEL [level] [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET LOG_LEVEL [level]
```

Arguments

level: Specify one of the following log levels:

- `off`: No logging.
- `user`: User log information.
- `admin`: Administrative log information.
- `support`: Oracle Support Services log information. This is the default.

Usage Notes

Specify `off` to capture the minimum amount of log information. Specify `support` to capture the maximum amount.

Example

```
CMCTL> SET LOG_LEVEL SUPPORT
CMAN_user.us.example.com parameter log_level set to SUPPORT
The command completed successfully
```

2.3.17 SET OUTBOUND_CONNECT_TIMEOUT

Purpose

To specify the maximum amount of time the Oracle Connection Manager instance waits 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]
```

From the From the Oracle Connection Manager Control utility:

```
CMCTL> SET OUTBOUND_CONNECT_TIMEOUT [time]
```

Arguments

time: The outbound connect timeout in seconds. The default is `0`.

Example

```
CMCTL> SET OUTBOUND_CONNECT_TIMEOUT 30
CMAN_user.us.example.com parameter outbound_connect_timeout set to 30
The command completed successfully
```

2.3.18 SET PASSWORD

Purpose

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 after you have set it, run the SAVE_PASSWD command.

Example

```
CMCTL> SET PASSWORD

Enter Old password: old_password
Enter New password: new_password
Reenter New password: new_password

The command completed successfully
```

2.3.19 SET SESSION_TIMEOUT

Purpose

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]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET SESSION_TIMEOUT [time]
```

Arguments

time: The session timeout in seconds. The default is 0 (zero), which disables this feature.

Example

```
CMCTL> SET SESSION_TIMEOUT 60
CMAN_user.us.example.com parameter session_timeout set to 60
The command completed successfully
```

2.3.20 SET TRACE_DIRECTORY

> **Note:**
>
> This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled.

Purpose

To designate where the trace files for an Oracle Connection Manager instance are written.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET TRACE_DIRECTORY [directory_path] [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET TRACE_DIRECTORY [directory_path]
```

Arguments

directory_path: The location of the trace directory. The default path is `ORACLE_HOME/network/trace`.

Usage Notes

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

Example

```
CMCTL> SET TRACE_DIRECTORY /disk1/mpurayat_newtest/oracle/network/trace
cman1 parameter trace_directory set to /disk1/mpurayat_newtest/oracle/network
/trace
The command completed successfully
```

2.3.21 SET TRACE_LEVEL

Purpose

To set the trace level for an Oracle Connection Manager instance.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET TRACE_LEVEL [level] [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET TRACE_LEVEL [level]
```

Arguments

level: Specify one of the following log levels:

- `off`: No tracing. This is the default.
- `user`: User trace information.
- `admin`: Administrative trace information.
- `support`: Oracle Support Services trace information.

Usage Notes

Specify `off` to capture the minimum amount of trace information. Specify `support` to capture the maximum amount.

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

Example

```
CMCTL> SET TRACE_LEVEL USER
CMAN_user.us.example.com parameter trace_level set to USER
The command completed successfully
```

2.3.22 SHOW

Purpose

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]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW
```

Example

```
CMCTL> 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
```

2.3.23 SHOW ALL

Purpose

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]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW ALL
```

Example

```
CMCTL> SHOW ALL
listener_address          |
(address=(protocol=tcp)(host=users.us.example.com)(port=1630))
aso_authentication_filter |   OFF
connection_statistics     |   OFF
event_group               |   OFF
log_directory             | /disk1/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
trace_directory           | /disk1/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
```

2.3.24 SHOW CONNECTIONS

Purpose

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]
```

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: Specify one of the following values to display information about connections. Information categories include connection identifier, source, destination, service, current state, total idle time, and total elapsed time.

- `count`: The total number of connections that meet the criteria specified by the other qualifiers. This is the default.
- `detail`: All information about connections specified by the other qualifiers.

state: Specify 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, then `SHOW CONNECTIONS` defaults to all possible states. If the time qualifier is included under these conditions, then the time specified is the amount of time that has elapsed since a client initiated a connection.

time: Use the following format to specify connections greater than the time indicated:

`gt[hh:mm:]ss`

source: Specify one of the following formats to specify the source address:

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

destination: Specify one of the following formats to specify the destination address:

- `to` *`IP`*
- `to` *`hostname`*
- `to` *`subnet`*

service: Use the *service_name* format to request a service:

gateway_process_id: Use the following format to specify connections that are proxied by the gateway process indicated:

```
using gateway_process_id
```

connect_identifier_list: Separate multiple connection identifiers using a space.

Usage Notes

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

Issuing `SHOW CONNECTIONS` without an argument displays all connections.

Examples

The following command 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 192.0.2.32/24 to host1
```

The following command displays the number of connections proxied by Oracle Connection Manager using the gateway process identifier 0 that have been in the idle state more than 30 minutes:

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

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

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

2.3.25 SHOW DEFAULTS

Purpose

To display default parameter settings.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW DEFAULTS [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW DEFAULTS
```

Example

```
CMCTL> SHOW DEFAULTS
listener_address          |
(address=(protocol=tcp)(host=users.us.example.com)(port=1521))
aso_authentication_filter |   OFF
connection_statistics     |   OFF
event_group               |   OFF
log_directory             | /disk1/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
trace_directory           | /disk1/user_cman_test/oracle/network/trace/
trace_level               |   OFF
trace_timestamp           |   OFF
trace_filelen             |     0
trace_fileno              |     0
The command completed successfully
```

2.3.26 SHOW EVENTS

Purpose

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]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW EVENTS
```

Example

```
CMCTL> SHOW EVENTS
Event Groups:
memory_ops
The command completed successfully
```

2.3.27 SHOW GATEWAYS

Purpose

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]
```

From the Oracle Connection Manager Control utility:

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

Arguments

gateway: The identifier of the gateway or gateways whose status to display.

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

Usage Notes

To display multiple gateways, use a space to separate the identifiers when entering the command.

Example

```
CMCTL> 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
```

2.3.28 SHOW PARAMETERS

Purpose

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]
```

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> SHOW PARAMETERS
listener_address          |
(address=(protocol=tcp)(host=users.us.example.com)(port=1630))
aso_authentication_filter |    ON
connection_statistics     |    ON
event_group               | (memory_ops)
log_directory             | /disk1/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
trace_directory           | /disk1/user_cman_test/oracle/network/trace/
trace_level               | SUPPORT
trace_timestamp           |   OFF
trace_filelen             |     0
trace_fileno              |     0
The command completed successfully
```

2.3.29 SHOW RULES

Purpose

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]
```

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> SHOW RULES
Number of filtering rules currently in effect: 5
(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=users.us.example.com)
    (dst=users.us.example.com)
    (srv=cmon)
    (act=accept)
    (action_list=(mit=100)(mct=1130)(moct=200)(aut=on))
  )
)
```

2.3.30 SHOW SERVICES

Purpose

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, both refused and current.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW SERVICES [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW SERVICES
```

Example

```
CMCTL> 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=users)(PORT=33168))
The command completed successfully
```

2.3.31 SHOW STATUS

Purpose

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> SHOW STATUS
Status of the Instance
----------------------
Instance name             CMAN_user.us.example.com
Version                   CMAN for Linux: Version 18.0.0.0.0
Start date                12-JAN-2018 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      /disk1/user_cman_test/oracle/network/admin/cman.ora
Instance Log directory    /disk1/user_cman_test/oracle/network/log/
Instance Trace directory  /disk1/user_cman_test/oracle/network/trace/
The command completed successfully
```

2.3.32 SHOW VERSION

Purpose

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]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW VERSION
```

Examples

```
CMCTL> SHOW VERSION
CMAN for Linux: Version 18.0.0.0.0
The command completed successfully
```

2.3.33 SHUTDOWN

Purpose

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]
```

From the Oracle Connection Manager Control utility:

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

Arguments

`gateways`: To shut down a specific gateway.

normal: To reject new connections and terminate after existing connections close. This is the default.

abort: To shut down Oracle Connection Manager immediately, and close all open connections.

To specify more than one gateway, separate gateways using a space.

Usage Notes

Issuing SHUTDOWN without an argument shuts down all gateways.

Example

```
CMCTL> SHUTDOWN GATEWAYS 0
The command completed successfully
```

2.3.34 STARTUP

Purpose

To start Oracle Connection Manager.

Prerequisites

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

Syntax

From the operating system:

```
cmctl STARTUP [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> STARTUP
```

Usage Notes

Before issuing this command, you must use the ADMINISTER command to select an instance to start.

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

The utility may prompt for a password if Oracle Connection Manager was installed with secure installation option.

Example

```
CMCTL> STARTUP
Starting Oracle Connection Manager instance cman_1. Please wait...
CMAN for Linux: Version 18.0.0.0.0
Status of the Instance
----------------------
Instance name             cman_1
Version                   CMAN for Linux: Version 18.0.0.0.0
Start date                22-JAN-2018 01:16:55
```

```
Uptime                    0 days 0 hr. 0 min. 9 sec
Num of gateways started   8
Average Load level        0
Log Level                 SUPPORT
Trace Level               OFF
Instance Config file      $ORACLE_HOME/network/admin/cman.ora
Instance Log directory    $ORACLE_BASE/diag/netcman/node_name/cman_1/alert
Instance Trace directory  $ORACLE_BASE/diag/netcman/node_name/cman_1/trace
The command completed successfully
```

2.3.35 SUSPEND GATEWAY

Purpose

To specify which gateway processes 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]
```

From the Oracle Connection Manager Control utility:

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

Arguments

gateway_process_id: The gateway process that will no longer accept new connections. Specify multiple gateway processes by putting a space 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> 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:

- Syntax Rules for Configuration Files
- Protocol Address Configuration
- Parameters for the sqlnet.ora File
- Local Naming Parameters in the tnsnames.ora File
- Oracle Net Listener Parameters in the listener.ora File
- Oracle Connection Manager Parameters (cman.ora)
- Directory Usage Parameters in the ldap.ora File

3

Syntax Rules for Configuration Files

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

This chapter contains the following topics:

3.1 Overview of Configuration File Syntax

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=
    (keyword1=value1)
    (keyword2=value2))
```

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

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

The following syntax is not acceptable:

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

3.2 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, then it is interpreted as a continuation of the previous line.
- All characters must belong to the network character set.
- Keywords are not case sensitive. However, values may be case sensitive, depending on the operating system and protocol.
- Spaces around the equal sign (=) are optional in keyword-value pairs.
- There is a hierarchy of keywords such 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 quotation marks (") or single quotation marks (').
- If the keyword-value pair consists of a single word or a concatenation of words on either side of the equal sign, then no parentheses are needed.
- The maximum length of a connect descriptor is 4 KB.
- Comments can be included using the number sign (#) at the beginning of a line. Anything following the number sign to the end of the line is considered a comment.

3.3 Network Character Set for Keywords

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 has quotation marks. Either single or double quotation marks can be used to enclose a value containing reserved symbols. To include a quotation marks within a value that is surrounded by quotation marks, use different quotation marks. 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

3.4 Character Set for Listener and Net Service Names

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 alphanumeric 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, 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 the following topics:

4.1 Protocol Addresses

The protocol address is comprised of ADDRESS and ADDRESS_LIST elements.

4.1.1 ADDRESS

Purpose

To define a protocol address.

Usage Notes

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

Example

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

> **See Also:**
>
> - "Protocol Parameters" for each protocol's required parameters
> - *Oracle Database Global Data Services Concepts and Administration Guide* for information about management of global services

4.1.2 ADDRESS_LIST

Purpose

To define a list of protocol addresses that share common characteristics.

Usage Notes

This parameter is not mandatory when specifying multiple addresses.

Example

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

4.2 Protocol Parameters

The listener and Oracle Connection Manager are identified by protocol addresses.

The following table lists the parameters used by the Oracle protocol support:

Table 4-1   Protocol-Specific Parameters

Protocol	Parameter	Description
IPC	PROTOCOL	Specify ipc as the value.
IPC	KEYPATH	On UNIX variants, IPC protocol uses the UNIX domain socket and this socket creates an internal file for client/server communication. The parameter keypath specifies the location where this file is created. If keypath is used, then use the same value of version greater than 18 on the client and listener sides.
IPC	KEY	Specify a unique name for the service. Oracle 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.
Named Pipes	SERVER	Specify the name of the Oracle server.
Named Pipes	PIPE	Specify the pipe name used to connect to the database server. This is the same PIPE keyword specified on server with Named Pipes. This name can be any name. Example: `(PROTOCOL=nmp)(SERVER=sales)(PIPE=dbpipe0)`
SDP	PROTOCOL	Specify sdp as the value.

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

Protocol	Parameter	Description
SDP	`HOST`	Specify the host name or IP address of the computer.

SDP	`PORT`	Specify the listening port number.
		Example:
		`(PROTOCOL=sdp)(HOST=sales-server)(PORT=1521)`
		`(PROTOCOL=sdp)(HOST=192.0.2.204)(PORT=1521)`

TCP/IP	`PROTOCOL`	Specify `tcp` as the value.
TCP/IP	`HOST`	Specify the host name or IP address of the computer.

TCP/IP	`PORT`	Specify the listening port number.
		Example:
		`(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)`
		`(PROTOCOL=tcp)(HOST=192.0.2.204)(PORT=1521)`

TCP/IP with SSL	`PROTOCOL`	Specify `tcps` as the value.
TCP/IP with SSL	`HOST`	Specify the host name or IP address of the computer.

TCP/IP with SSL	`PORT`	Specify the listening port number.
		Example:
		`(PROTOCOL=tcps)(HOST=sales-server)(PORT=2484)`
		`(PROTOCOL=tcps)(HOST=192.0.2.204)(PORT=2484)`

Exadirect	`PROTOCOL`	Specify `exadirect` as the value.
Exadirect	`HOST`	Specify the IP address of the InfiBand interface.

Exadirect	`PORT`	Specify the listening port number.
		Example:
		`(PROTOCOL=exadirect)(HOST=sales-server)(PORT=2484)`
		`(PROTOCOL=tcps)(HOST=192.0.2.204)(PORT=1522)`

4.3 Recommended Port Numbers

Table 4-2 lists the recommends the port numbers.

Table 4-2 Recommended Port Numbers

Port	Description
1521	Default listening port for client connections to the listener. 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.

4.4 Port Number Limitations

Oracle allows port numbers from 1 to 65535. However, many operating systems reserve port numbers less than 1024. For example, on certain operating systems, only privileged processes can listen for TCP connections on ports less than 1024.

If you need to configure a listener to listen on a port number less than 1024, then do the following procedure:

> **Note:**
>
> Your operating system may require a different procedure.

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 the `root` user on the machine that has the listener.
3. 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 `root` user.
4. Ensure that the permissions of the individual directories found in the path names to these files, starting with the `root` directory have the same ownership and access permissions.
5. Start the listener as the `root` user.
6. Enter the following command at the system prompt:

   ```
   tnslsnr listener_name -user user -group group
   ```

   In the preceding command, the following options are used:

   Table 4-3 tnslsnr Utility Options

Options	Description
listener_name	Specify the name of the listener. If omitted, then the default name `LISTENER` is used.
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.

Table 4-3 (Cont.) tnslsnr Utility Options

Options	Description
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.

During this step, the listener switches to the specified user and group. All operations are done with the specified user and group privileges, except the system calls necessary to listen on configured endpoints. The listener reverts to the `root` user 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 switches to the specified user and group irreversibly. Therefore, the listener will give up the `root` privilege that it initially had. The `-user` and `-group` command line arguments only accept user and group identifiers specified in numeric form.

For example, to run a listener with root privileges 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:

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

In the preceding example, 37555 could be the identifier for the `oracle` user, and 16 could be the identifier for the `dba` group.

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

> **Important Notes:**
>
> - Oracle recommends that the user which the listener process runs be the `oracle` user, or a user that the listener process normally runs on the operating system.
> - Do not leave the listener process running as the `root` user because doing so is a security vulnerability.

5

Parameters for the sqlnet.ora File

This chapter provides a complete listing of the `sqlnet.ora` file configuration parameters. This chapter includes the following topics:

5.1 Overview of Profile Configuration File

The `sqlnet.ora` file is the profile configuration file. It resides on the client machines and the database server. Profiles are stored and implemented using this file. The database server can be configured with access control parameters in the `sqlnet.ora` file. These parameters specify whether clients are allowed or denied access based on the protocol.

The `sqlnet.ora` file enables you to do the following:

- Specify the client domain to append to unqualified names
- Prioritize naming methods
- Enable logging and tracing features
- Route connections through specific processes
- Configure parameters for external naming
- Configure Oracle Advanced Security
- Use protocol-specific parameters to restrict access to the database

By default, the `sqlnet.ora` file is located in the `ORACLE_HOME/network/admin` directory. The `sqlnet.ora` file can also be stored in the directory specified by the `TNS_ADMIN` environment variable.

Note:

- The settings in the `sqlnet.ora` file apply to all pluggable databases (PDBs) in a multitenant container database environment.
- Oracle Net Services supports the IFILE parameter in the `sqlnet.ora` file, with up to three levels of nesting. The parameter is added manually to the file. The following is an example of the syntax:

  ```
  IFILE=/tmp/listener_em.ora
  IFILE=/tmp/listener_cust1.ora
  IFILE=/tmp/listener_cust2.ora
  ```

  Refer to *Oracle Database Reference* for additional information.
- In the read-only Oracle home mode,, the `sqlnet.ora` file default location is `*ORACLE_BASE_HOME*/network/admin`.
- In the read-only Oracle home mode, the parameters that default to `ORACLE_HOME` location change to default to `ORACLE_BASE_HOME` location.

5.2 sqlnet.ora Profile Parameters

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

- ACCEPT_MD5_CERTS
- ACCEPT_SHA1_CERTS
- ADD_SSLV3_TO_DEFAULT
- BEQUEATH_DETACH
- EXADIRECT_FLOW_CONTROL
- EXADIRECT_RECVPOLL
- DEFAULT_SDU_SIZE
- DISABLE_OOB
- HTTPS_SSL_VERSION
- IPC.KEYPATH
- NAMES.DEFAULT_DOMAIN
- NAMES.DIRECTORY_PATH
- NAMES.LDAP_AUTHENTICATE_BIND
- NAMES.LDAP_CONN_TIMEOUT
- NAMES.LDAP_PERSISTENT_SESSION
- NAMES.NIS.META_MAP
- RECV_BUF_SIZE
- SDP.PF_INET_SDP
- SEC_USER_AUDIT_ACTION_BANNER

- SEC_USER_UNAUTHORIZED_ACCESS_BANNER
- SEND_BUF_SIZE
- SQLNET.ALLOW_WEAK_CRYPTO
- SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS
- SQLNET.ALLOWED_LOGON_VERSION_CLIENT
- SQLNET.ALLOWED_LOGON_VERSION_SERVER
- SQLNET.AUTHENTICATION_SERVICES
- SQLNET.CLIENT_REGISTRATION
- SQLNET.COMPRESSION
- SQLNET.COMPRESSION_LEVELS
- SQLNET.COMPRESSION_THRESHOLD
- SQLNET.CRYPTO_CHECKSUM_CLIENT
- SQLNET.CRYPTO_CHECKSUM_SERVER
- SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT
- SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER
- SQLNET.DBFW_PUBLIC_KEY
- SQLNET.DOWN_HOSTS_TIMEOUT
- SQLNET.ENCRYPTION_SERVER
- SQLNET.ENCRYPTION_TYPES_CLIENT
- SQLNET.ENCRYPTION_TYPES_SERVER
- SQLNET.EXPIRE_TIME
- SQLNET.INBOUND_CONNECT_TIMEOUT
- SQLNET.KERBEROS5_CC_NAME
- SQLNET.KERBEROS5_CLOCKSKEW
- SQLNET.KERBEROS5_CONF
- SQLNET.KERBEROS5_CONF_LOCATION
- SQLNET.KERBEROS5_KEYTAB
- SQLNET.KERBEROS5_REALMS
- SQLNET.KERBEROS5_REPLAY_CACHE
- SQLNET.OUTBOUND_CONNECT_TIMEOUT
- SQLNET.RADIUS_ALTERNATE
- SQLNET.RADIUS_ALTERNATE_PORT
- SQLNET.RADIUS_ALTERNATE_RETRIES
- SQLNET.RADIUS_AUTHENTICATION
- SQLNET.RADIUS_AUTHENTICATION_INTERFACE
- SQLNET.RADIUS_AUTHENTICATION_PORT
- SQLNET.RADIUS_AUTHENTICATION_RETRIES

- SQLNET.RADIUS_AUTHENTICATION_TIMEOUT
- SQLNET.RADIUS_CHALLENGE_RESPONSE
- SQLNET.RADIUS_SECRET
- SQLNET.RADIUS_SEND_ACCOUNTING
- SQLNET.RECV_TIMEOUT
- SQLNET.SEND_TIMEOUT
- SQLNET.WALLET_OVERRIDE
- SSL_CERT_REVOCATION
- SSL_CERT_FILE
- SSL_CERT_PATH
- SSL_CIPHER_SUITES
- SSL_EXTENDED_KEY_USAGE
- SSL_SERVER_DN_MATCH
- SSL_VERSION
- TCP.CONNECT_TIMEOUT
- TCP.EXCLUDED_NODES
- TCP.INVITED_NODES
- TCP.NODELAY
- TCP.QUEUESIZE
- TCP.VALIDNODE_CHECKING
- TNSPING.TRACE_DIRECTORY
- TNSPING.TRACE_LEVEL
- USE_CMAN
- USE_DEDICATED_SERVER
- WALLET_LOCATION

5.2.1 ACCEPT_MD5_CERTS

Purpose

To accept MD5 signed certificates, in addition to `sqlnet.ora`, this parameter must also be set in `listener.ora`.

Default

`FALSE`

Values

- `TRUE` to accept MD5 signed certificates
- `FALSE` to not accept MD5 signed certficates

5.2.2 ACCEPT_SHA1_CERTS

Purpose

To not accept SHA1 signed certificates, in addition to `sqlnet.ora`, this parameter must also be set in `listener.ora`.

Default

TRUE

Values

- `TRUE` to accept SHA1 signed certificates
- `FALSE` to not accept SHA1 signed certificates

5.2.3 ADD_SSLV3_TO_DEFAULT

Purpose

If the server wants to accept `SSL_VERSION=3.0` in its default list of `SSL_VERSION`s, then in addition to `sqlnet.ora`, this parameter must also be set in `listener.ora`.

Default

`FALSE`

Values

- If set to `TRUE` and `SSL_VERSION` is not specified or is set to "undetermined", then `SSL_VERSION` includes versions `1.2`, `1.1`, `1.0`, and `3.0`.
- If set to `FALSE` and `SSL_VERSION` is not specified or is set to "undetermined", then `SSL_VERSION` includes versions `1.2`, `1.1`, and `1.0`

5.2.4 BEQUEATH_DETACH

Purpose

To turn signal handling on or off for Linux and UNIX systems.

Default

no

Values

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

Example

```
BEQUEATH_DETACH=yes
```

5.2.5 EXADIRECT_FLOW_CONTROL

Purpose

To enable or disable Exadirect flow control.

Usage Notes

If turned on, the parameter enables Oracle Net to broadcast available receive window to the sender. The sender limits the sends based on the receiver broadcast window.

Default

```
off
```

Example

```
EXADIRECT_FLOW_CONTROL=on
```

5.2.6 EXADIRECT_RECVPOLL

Purpose

To specify the time that a receiver polls for incoming data.

Usage Notes

The parameter can be set to a fixed value or `AUTO` for auto tuning of the polling value.

Default

```
0
```

Example

```
EXADIRECT_RECVPOLL = 10
```

```
EXADIRECT_RECVPOLL = AUTO
```

5.2.7 DEFAULT_SDU_SIZE

Purpose

To specify the session data unit (SDU) size, in bytes to connections.

Usage Notes

Oracle recommends setting this parameter in both the client-side and server-side `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

8192 bytes (8 KB)

Values

512 to 2097152 bytes

Example

```
DEFAULT_SDU_SIZE=4096
```

5.2.8 DISABLE_OOB

Purpose

To enable or disable Oracle Net to send or receive out-of-band break messages using urgent data provided by the underlying protocol.

Usage Notes

If turned `off`, then the parameter enables Oracle Net to send and receive break messages. If turned `on`, then the parameter disables the ability to send and receive break messages. Once enabled, this feature applies to all protocols used by this client.

Default

off

Example

```
DISABLE_OOB=on
```

> **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.

5.2.9 HTTPS_SSL_VERSION

Purpose

To control the Secure Sockets Layer (SSL) version used by XDB HTTPS connections separately.

Usage Notes

In particular, the `SSL_VERSION` parameter no longer controls the SSL version used by HTTPS. You can set this parameter to any valid `SSL_VERSION` values.

Default

`1.1` or `1.2`, meaning `TLSv1.1` or `TLSv1.2`.

Values

Any valid `SSL_VERSION` value

5.2.10 IPC.KEYPATH

Purpose

To specify the destination directory where the internal file is created for UNIX domain sockets.

Usage Notes

This parameter applies only to Oracle Net's usage of UNIX domain socket and does not apply to other usages of UNIX domain sockets in the database, such as clusterware. If `keypath` is used, then the same value should be used on both the client and the listener sides with version greater than 18.

Default

The directory path is either `/var/tmp/.oracle` for Oracle Linux, Oracle Solaris or `/tmp/.oracle` for other UNIX variants.

Example

```
ipc.keypath=/home/oracleuser.
```

5.2.11 NAMES.DEFAULT_DOMAIN

Purpose

To set the domain from which the client most often looks up names resolution requests.

Usage Notes

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.example.com`, then the connect string `CONNECT scott@sales` gets searched as `sales.us.example.com`. If the connect string includes the domain extension, such as `CONNECT scott@sales.us.example.com`, then the domain is not appended to the string.

Default

None

Example

```
NAMES.DEFAULT_DOMAIN=example.com
```

5.2.12 NAMES.DIRECTORY_PATH

Purpose

To specify the order of the naming methods used for client name resolution lookups.

Default

NAMES.DIRECTORY_PATH=(tnsnames, ldap, ezconnect)

Values

The following table shows the NAMES.DIRECTORY_PATH values for the naming methods.

Naming Method Value	Description
`tnsnames` (local naming method)	Set to resolve a network service name through the `tnsnames.ora` file on the client.
`ldap (`directory naming method)	Set to resolve a database service name, net service name, or network service alias through a directory server.
`ezconnect` or `hostname` (Easy Connect naming method)	Select to enable clients to use a TCP/IP connect identifier, consisting of a host name and optional port and service name.
`nis` (external naming method)	Set to resolve service information through an existing Network Information Service (NIS).

Example

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

5.2.13 NAMES.LDAP_AUTHENTICATE_BIND

Purpose

To specify whether the LDAP naming adapter should attempt to authenticate using a specified wallet when it connects to the LDAP directory to resolve the name in the connect string.

Usage Notes

The parameter value is Boolean.

If the parameter is set to `TRUE`, then the LDAP connection is authenticated using a wallet whose location must be specified in the WALLET_LOCATION parameter.

If the parameter is set to `FALSE`, then the LDAP connection is established using an anonymous bind.

Default

false

Example

```
NAMES.LDAP_AUTHENTICATE_BIND=true
```

5.2.14 NAMES.LDAP_CONN_TIMEOUT

Purpose

To specify number of seconds for a non-blocking connect timeout to the LDAP server.

Usage Notes

The parameter value -1 is for infinite timeout.

Default

15 seconds

Values

Values are in seconds. The range is `-1` to the number of seconds acceptable for your environment. There is no upper limit.

Example

```
names.ldap_conn_timeout = -1
```

5.2.15 NAMES.LDAP_PERSISTENT_SESSION

Purpose

To specify whether the LDAP naming adapter should leave the session with the LDAP server open after name lookup is complete.

Usage Notes

The parameter value is Boolean.

If the parameter is set to `TRUE`, then the connection to the LDAP server is left open after the name lookup is complete. The connection will effectively stay open for the duration of the process. If the connection is lost, then it is re-established as needed.

If the parameter is set to `FALSE`, then the LDAP connection is terminated as soon as the name lookup completes. Every subsequent lookup opens the connection, performs the lookup, and closes the connection. This option prevents the LDAP server from having a large number of clients connected to it at any one time.

Default

false

Example

```
NAMES.LDAP_PERSISTENT_SESSION=true
```

5.2.16 NAMES.NIS.META_MAP

Purpose

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
```

5.2.17 RECV_BUF_SIZE

Purpose

To specify the buffer space limit for receive operations of sessions.

Usage Notes

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

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 the operating system-specific documentation for additional information about additional protocols that support this parameter.

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

Default

The default value for this parameter is operating system specific. The default for Linux 2.6 operating system is 87380 bytes.

Example

```
RECV_BUF_SIZE=11784
```

5.2.18 SDP.PF_INET_SDP

Purpose

To specify the protocol family or address family constant for the SDP protocol on your system.

Default

27

Values

Any positive integer

Example

```
SDP.PF_INET_SDP=30
```

5.2.19 SEC_USER_AUDIT_ACTION_BANNER

Purpose

To specify a text file containing the banner contents that warn the user about possible user action auditing.

Usage Notes

The complete path of the text file must be specified in the `sqlnet.ora` file on the server. Oracle Call Interface (OCI) applications can make use of OCI features to retrieve this banner and display it to the user.

Default

None

Values

Name of the file for which the database owner has read permissions.

Example

```
SEC_USER_AUDIT_ACTION_BANNER=/opt/oracle/admin/data/auditwarning.txt
```

5.2.20 SEC_USER_UNAUTHORIZED_ACCESS_BANNER

Purpose

To specify a text file containing the banner contents that warn the user about unauthorized access to the database.

Usage Notes

The complete path of the text file must be specified in the `sqlnet.ora` file on the server. OCI applications can make use of OCI features to retrieve this banner and display it to the user.

Default

None

Values

Name of the file for which the database owner has read permissions.

Example

```
SEC_USER_UNAUTHORIZED_ACCESS_BANNER=/opt/oracle/admin/data/unauthwarning.txt
```

5.2.21 SEND_BUF_SIZE

Purpose

To specify the buffer space limit for send operations of sessions.

Usage Notes

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

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 the operating system-specific documentation for additional information about additional protocols that support this parameter.

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

Default

The default value for this parameter is operating system specific. The default for Linux 2.6 operating system is 16 KB.

Example

```
SEND_BUF_SIZE=11784
```

5.2.22 SQLNET.ALLOW_WEAK_CRYPTO

Use the `sqlnet.ora` compatibility parameter `SQLNET.ALLOW_WEAK_CRYPTO` to configure your client-side network connection by reviewing the specified encryption and crypto-checksum algorithms.

Purpose

To configure your client-side network connection by reviewing the encryption and crypto-checksum algorithms enabled on the client and server. This ensures that the connection does not encounter compatibility issues and your configuration uses supported strong algorithms.

Usage Notes

- The `DES`, `DES40`, `3DES112`, `3DES168`, `RC4_40`, `RC4_56`, `RC4_128`, `RC4_256`, and `MD5` algorithms are deprecated in this release.

  As a result of this deprecation, Oracle recommends that you review your network encryption and integrity configuration to check if you have specified any of the deprecated weak algorithms.

  To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.
- If you set this parameter to `TRUE`, then you can specify deprecated algorithms for backward compatibility. This configuration allows patched clients to connect to unpatched servers, and thus such a connection is less secure.
- If you set this parameter to `FALSE`, then you can specify only supported algorithms so that clients and servers can communicate in a fully patched environment. The server enforces key fold-in for all Kerberos and JDBC thin clients. This configuration strengthens the connection between clients and servers by using strong native network encryption and integrity capabilities.

  Using this setting, if native network encryption or checksumming is enabled and a patched server or client attempts to communicate with an unpatched old client or server, then the connection fails with an error message.

Values

- `TRUE`
- `FALSE`

Default Value

`TRUE`

Recommended Value

`FALSE`

> **Note:**
>
> Before setting this parameter to FALSE, you must remove all deprecated algorithms listed in the server and client `sqlnet.ora` files.

Example

```
SQLNET.ALLOW_WEAK_CRYPTO = FALSE
```

5.2.23 SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS

Use the `sqlnet.ora` compatibility parameter `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS` to configure your server-side network connection by reviewing the specified encryption and crypto-checksum algorithms.

Purpose

To configure your server-side network connection by reviewing the encryption and crypto-checksum algorithms enabled on the client and server. This ensures that the connection does not encounter compatibility issues and your configuration uses supported strong algorithms.

Usage Notes

- The `DES`, `DES40`, `3DES112`, `3DES168`, `RC4_40`, `RC4_56`, `RC4_128`, `RC4_256`, and `MD5` algorithms are deprecated in this release.

  As a result of this deprecation, Oracle recommends that you review your network encryption and integrity configuration to check if you have specified any of the deprecated weak algorithms.

  To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.
- If you set this parameter to `TRUE`, then you can specify deprecated algorithms for backward compatibility. This configuration allows patched servers to connect to unpatched clients, and thus such a connection is less secure.
- If you set this parameter to `FALSE`, then you can specify only supported algorithms so that clients and servers can communicate in a fully patched environment. The server enforces key fold-in for all Kerberos and JDBC thin clients. This configuration strengthens the connection between clients and servers by using strong native network encryption and integrity capabilities.

  Using this setting, if native network encryption or checksumming is enabled and a patched server or client attempts to communicate with an unpatched old client or server, then the connection fails with an error message.

Values

- `TRUE`
- `FALSE`

Default Value

`TRUE`

Recommended Value

FALSE

> **Note:**
>
> Before setting this parameter to FALSE, you must remove all deprecated algorithms listed in the server and client sqlnet.ora files.

Example

```
SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS = FALSE
```

5.2.24 SQLNET.ALLOWED_LOGON_VERSION_CLIENT

Purpose

To set the minimum authentication protocol allowed for clients, and when a server is acting as a client, such as connecting over a database link, when connecting to Oracle Database instances.

Usage Notes

The term VERSION in the parameter name refers to the version of the authentication protocol, not the Oracle Database release.

If the version does not meet or exceed the value defined by this parameter, then authentication fails with an ORA-28040: No matching authentication protocol error.

> **See Also:**
>
> *Oracle Database Security Guide*

Values

- 12a for Oracle Database 12*c* Release 1 (12.1.0.2) or later (strongest protection)

  > **Note:**
  >
  > Using this setting, the clients can only authenticate using a de-optimized password version. For example, the 12C password version.

- 12 for the critical patch updates CPUOct2012 and later Oracle Database 11*g* authentication protocols (stronger protection)

> **Note:**
>
> Using this setting, the clients can only authenticate using a password hash value that uses salt. For example, the `11G` or `12C` password versions.

- `11` for Oracle Database 11*g* authentication protocols (default)
- `10` for Oracle Database 10*g* authentication protocols
- `8` for Oracle8*i* authentication protocol

Default

11

Example

If an Oracle Database 12*c* database hosts a database link to an Oracle Database 10*g* database, then the `SQLNET.ALLOWED_LOGON_VERSION_CLIENT` parameter should be set as follows in order for the database link connection to proceed:

```
SQLNET.ALLOWED_LOGON_VERSION_CLIENT=10
```

> **See Also:**
>
> *Oracle Database Reference*

5.2.25 SQLNET.ALLOWED_LOGON_VERSION_SERVER

Purpose

To set the minimum authentication protocol allowed when connecting to Oracle Database instances.

Usage Notes

The term `VERSION` in the parameter name refers to the version of the authentication protocol, not the Oracle Database release.

The authentication fails with an `ORA-28040: No matching authentication protocol` error or an `ORA-03134: Connections to this server version are no longer supported` error if the client does not have the ability listed in the "Ability Required of the Client" column corresponding to the row matching the value of the SQLNET.ALLOWED_LOGON_VERSION_SERVER parameter in Table 1.

> **See Also:**
>
> *Oracle Database Security Guide*

A setting of `8` permits all password versions, and allows any combination of the `DBA_USERS.PASSWORD_VERSIONS` values `10G`, `11G`, and `12C`.

A setting of 12a permits only the 12C password version.

A greater value means the server is less compatible in terms of the protocol that clients must understand in order to authenticate. The server is also more restrictive in terms of the password version that must exist to authenticate any specific account. Whether a client can authenticate to a specific account depends on both the server's setting of its SQLNET.ALLOWED_LOGON_VERSION_SERVER parameter, as well as on the password versions which exist for the specified account. The list of password versions can be seen in DBA_USERS.PASSWORD_VERSIONS.

Note the following implications of setting the value to 12 or 12a:

- A value of FALSE for the SEC_CASE_SENSITIVE_LOGON Oracle instance initialization parameter must not be used because password case insensitivity requires the use of the 10G password version. If the SEC_CASE_SENSITIVE_LOGON Oracle instance initialization parameter is set to FALSE, then user accounts and secure roles become unusable because Exclusive Mode excludes the use of the 10G password version. The SEC_CASE_SENSITIVE_LOGON Oracle instance initialization parameter enables or disables password case sensitivity. However, since Exclusive mode is enabled by default in this release, disabling the password case sensitivity is not supported.

  > **Note:**
  >
  > – The use of the Oracle instance initialization parameter SEC_CASE_SENSITIVE_LOGON is deprecated in favor of setting the SQLNET.ALLOWED_LOGON_VERSION_SERVER parameter to 12 to ensure that passwords are treated in a case-sensitive fashion.
  >
  > – Disabling password case sensitivity is not supported in Exclusive mode (when SQLNET.ALLOWED_LOGON_VERSION_SERVER is set to 12 or 12a.)

- Releases of OCI clients earlier than Oracle Database 10*g* cannot authenticate to the Oracle database using password-based authentication.
- If the client uses Oracle Database 10*g*, then the client will receive an ORA-03134: Connections to this server version are no longer supported error message. To allow the connection, set the SQLNET.ALLOWED_LOGON_VERSION_SERVER value to 8. Ensure the DBA_USERS.PASSWORD_VERSIONS value for the account contains the value 10G. It may be necessary to reset the password for that account.

Note the following implication of setting the value to 12a:

- To take advantage of the new 12C password version introduced in Oracle Database release 12.2, user passwords should be expired to encourage users to change their passwords and cause the new 12C password version to be generated for their account. By default in this release, new passwords are treated in a case-sensitive fashion. When an account password is changed, the earlier 10G case-insensitive password version is automatically removed, and the new 12C password version is generated.
- When an account password is changed, the earlier 10G case-insensitive password version and the 11G password version are both automatically removed.

- JDBC Thin Client Support:

  In Oracle Database release 12.1.0.2 and later, if you set the `sqlnet.ora` parameter `SQLNET.ALLOWED_LOGON_VERSION_SERVER` to `12a` and you create a new account or change the password of an existing account, then only the new `12C` password version is generated. The `12C` password version is based on a `SHA-2 (Secure Hash Algorithm) SHA-512` salted cryptographic hash deoptimized using the `PBKDF2` (Password-Based Key Derivation Function 2) algorithm. When the database server is running with `ALLOWED_LOGON_VERSION_SERVER` set to `12a`, it is running in Exclusive Mode. In this mode, to log in using a JDBC client, the JRE version must be at least version 8. The JDBC client enables its `O7L_MR` capability flag only when it is running with at least version 8 of the JRE.

  > **Note:**
  >
  > Check the `PASSWORD_VERSIONS` column of the `DBA_USERS` catalog view to see the list of password versions for any given account.

  If you set the `sqlnet.ora` parameter `SQLNET.ALLOWED_LOGON_VERSION_SERVER` to `12`, the server runs in Exclusive Mode and only the `11G` and `12C` password versions (the `SHA-1` and `PBKDF2 SHA-2` based hashes of the password, respectively) are generated and allowed to be used. In such cases, fully-patched JDBC clients having the CPUOct2012 patch can connect because these JDBC clients provide the `O5L_NP` client ability.

  Older JDBC clients which do not have the CPUOct2012 containing the fix for the stealth password cracking vulnerability CVE-2012-3132, do not provide the `O5L_NP` client ability. Therefore, ensure that all the JDBC clients are patched properly.

The client must support certain abilities of the authentication protocol before the server will authenticate. If the client does not support a specified authentication ability, then the server rejects the connection with an `ORA-28040: No matching authentication protocol` error message.

The following is the list of all client abilities. Some clients do not have all abilities. Clients that are more recent have all the capabilities of the older clients, but older clients tend to have less abilities than more recent clients.

- `O7L_MR`: The ability to perform the Oracle Database 10*g* authentication protocol using the `12C` password version. For JDBC clients, only those running on at least JRE version 8 offer the O7L_MR capability.
- `O5L_NP`: The ability to perform the Oracle Database 10*g* authentication protocol using the `11G` password version, and generating a session key encrypted for critical patch update CPUOct2012.
- `O5L`: The ability to perform the Oracle Database 10*g* authentication protocol using the `10G` password version.
- `O4L`: The ability to perform the Oracle9*i* database authentication protocol using the `10G` password version.
- `O3L`: The ability to perform the Oracle8*i* database authentication protocol using the `10G` password version.

An ability which appears higher in the above list is more recent and secure than an ability which appears lower in the list. Clients that are more recent have all the capabilities of the older clients.

The following table describes:

- the allowed settings of the SQLNET.ALLOWED_LOGON_VERSION_SERVER parameter
- its effect on the generated password versions when an account is created or a password is changed
- the ability flag required of the client to authenticate while the server has this setting
- and whether the setting is considered to be an Exclusive Mode.

Table 5-1 SQLNET.ALLOWED_LOGON_VERSION_SERVER Settings

Value of the ALLOWED_LOGON_VERSION_SERVER Parameter	Generated Password Version	Ability Required of the Client	Meaning for Clients	Server Runs in Exclusive Mode
`12a`	`12C`	`O7L_MR`	Only Oracle Database 12*c* release 1 (12.1.0.2 or later) clients can connect to the server.	Yes because it excludes the use of both `10G` and `11G` password versions.
`12`	`11G`, `12C`	`O5L_NP`	Oracle Database 11*g* release 2 (11.2.0.3 or later) clients can connect to the server. Older clients need the critical patch update CPUOct2012 or later, to gain the O5L_NP ability. Only older clients which have applied critical patch update CPUOct2012 or later can connect to the server.	Yes because it excludes the use of the `10G` password version.

Table 5-1 (Cont.) SQLNET.ALLOWED_LOGON_VERSION_SERVER Settings

Value of the ALLOWED_LOGON_VERSION_SERVER Parameter	Generated Password Version	Ability Required of the Client	Meaning for Clients	Server Runs in Exclusive Mode
`11`	`10G`, `11G`, `12C`	`O5L`	Clients using Oracle Database 10*g* and later can connect to the server. Clients using releases earlier than Oracle Database release 11.2.0.3 that have not applied critical patch update CPUOct2012 or later patches must use the `10G` password version.	No
`10`	`10G`, `11G`, `12C`	`O5L`	It has the same meaning as the earlier row.	No
`9`	`10G`, `11G`, `12C`	`O4L`	It has the same meaning as the earlier row.	No
`8`	`10G`, `11G`, `12C`	`O3L`	It has the same meaning as the earlier row.	No

Values

- `12a` for Oracle Database 12*c* release 12.1.0.2 or later authentication protocols (strongest protection)
- `12` for Oracle Database 12*c* release 12.1 authentication protocols (default and recommended value)
- `11` for Oracle Database 11*g* authentication protocols
- `10` for Oracle Database 10*g* authentication protocols
- `9` for Oracle9*i* Database authentication protocol
- `8` for Oracle8*i* Database authentication protocol

> **Note:**
>
> - Starting with Oracle Database 12*c* Release 2 (12.2), the default value is 12.
> - For earlier releases, the value 12 can be used after the critical patch updates CPUOct2012 and later are applied.

Default

12

Example

```
SQLNET.ALLOWED_LOGON_VERSION_SERVER=12
```

5.2.26 SQLNET.AUTHENTICATION_SERVICES

Purpose

To enable one or more authentication services. If authentication has been installed, then it is recommended that this parameter be set to either `none` or to one of the listed authentication methods.

Usage Notes

When using the `SQLNET.AUTHENTICATION_SERVICES` value `all`, the server attempts to authenticate using each of the following methods. The server falls back to the ones lower on the list if the ones higher on the list were unsuccessful.

- Authentication based on a service external to the database, such as a service on the network layer, Kerberos, or RADIUS.
- Authentication based on the operating system user's membership in an administrative operating system group. Group names are platform-specific. This authentication is applicable to administrative connections only.
- Authentication performed by the database.
- Authentication based on credentials stored in a directory server.

Operating system authentication allows access to the database using any user name and any password when an administrative connection is attempted, such as using the `AS SYSDBA` clause when connecting using SQL*Plus. An example of a connection is as follows.

```
sqlplus ignored_username/ignored_password AS SYSDBA
```

When the operating-system user who issued the preceding command is already a member of the appropriate administrative operating system group, then the connection is successful. This is because the user name and password are ignored by the server due to checking the group membership first.

> **See Also:**
>
> *Oracle Database Security Guide* for additional information about authentication methods

Default

`all`

> **Note:**
>
> When installing the database with Database Configuration Assistant (DBCA), this parameter may be set to `nts` in the `sqlnet.ora` file.

Values

Authentication methods available with Oracle Net Services:

- `none` for no authentication methods, including Microsoft Windows native operating system authentication. When `SQLNET.AUTHENTICATION_SERVICES` is set to `none`, a valid user name and password can be used to access the database.
- `all` for all authentication methods.
- `beq` for native operating system authentication for operating systems other than Microsoft Windows
- `kerberos5` for Kerberos authentication
- `nts` for Microsoft Windows native operating system authentication
- `radius` for Remote Authentication Dial-In User Service (RADIUS) authentication
- `tcps` for SSL authentication

Example

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

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.27 SQLNET.CLIENT_REGISTRATION

Purpose

To set a unique identifier for the client computer.

Usage Notes

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
```

5.2.28 SQLNET.COMPRESSION

Purpose

To enable or disable data compression. If both the server and client have this parameter set to `ON`, then compression is used for the connection.

> **Note:**
>
> The `SQLNET.COMPRESSION` parameter applies to all database connections, except for Oracle Data Guard streaming redo and SecureFiles LOBs (Large Objects).

Default

off

Values

- `on` to enable data compression.
- `off` to disable data compression.

Example

```
SQLNET.COMPRESSION=on
```

5.2.29 SQLNET.COMPRESSION_ACCELERATION

Purpose

To specify the use of hardware accelerated version of compression using this parameter if it is available for that platform.

Usage Notes

This parameter can be specified under Oracle Connection Manager alias description.

Default

on

Values

- `on`
- `off`
- `0`
- `1`

Example 5-1 Example

```
compression_acceleration = on
```

5.2.30 SQLNET.COMPRESSION_LEVELS

Purpose

To specify the compression level.

Usage Notes

The compression levels are used at time of negotiation to verify which levels are used at both ends, and to select one level.

For Database Resident Connection Pooling (DRCP), only the compression level `low` is supported.

Default

low

Values

- `low` to use low CPU usage and low compression ratio.
- `high` to use high CPU usage and high compression ratio.

Example

```
SQLNET.COMPRESSION_LEVELS=(high)
```

5.2.31 SQLNET.COMPRESSION_THRESHOLD

Purpose

To specify the minimum data size, in bytes, for which compression is needed.

Usage Notes

Compression is not be done if the size of the data to be sent is less than this value.

Default

1024 bytes

Example

```
SQLNET.COMPRESSION_THRESHOLD=1024
```

5.2.32 SQLNET.CRYPTO_CHECKSUM_CLIENT

Purpose

To specify the checksum behavior for the client.

> **See Also:**
>
> *Oracle Database Security 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 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
```

5.2.33 SQLNET.CRYPTO_CHECKSUM_SERVER

Purpose

To specify the checksum behavior for the database server.

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 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
```

See Also:

Oracle Database Security Guide

5.2.34 SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT

Purpose

To specify a list of crypto-checksum algorithms for the client to use.

Default

All available algorithms

Values

- `MD5` for the RSA Data Security MD5 algorithm.

  The `MD5` algorithm is deprecated in this release. To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.
- `SHA1` for the Secure Hash Algorithm.
- `SHA256` for SHA-2 uses 256 bits with the hashing algorithm.
- `SHA384` for SHA-2 uses 384 bits with the hashing algorithm.
- `SHA512` for SHA-2 uses 512 bits with the hashing algorithm.

Example

```
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT=(SHA256, MD5)
```

See Also:

Oracle Database Security Guide

5.2.35 SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER

Purpose

To specify a list of crypto-checksum algorithms for the database server to use.

Default

All available algorithms

Values

- `MD5` for the RSA Data Security's MD5 algorithm

The MD5 algorithm is deprecated in this release. To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

- SHA1 for the Secure Hash algorithm.
- SHA256 for SHA-2 uses 256 bits with the hashing algorithm.
- SHA384 for SHA-2 uses 384 bits with the hashing algorithm.
- SHA512 for SHA-2 uses 512 bits with the hashing algorithm.

Example

```
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER=(SHA256, MD5)
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.36 SQLNET.DBFW_PUBLIC_KEY

Purpose

To provide Oracle Database Firewall public keys to Advanced Security Option (ASO) by specifying the file that stores the Oracle Database Firewall public keys.

Default

None

Values

Full path name of the operating system file that has the public keys.

Example

```
SQLNET.DBFW_PUBLIC_KEY="/path_to_file/dbfw_public_key_file.txt"
```

> **See Also:**
>
> "SQLNET.ENCRYPTION_TYPES_SERVER"

5.2.37 SQLNET.DOWN_HOSTS_TIMEOUT

Purpose

To specify the amount of time in seconds that information about the down state of server hosts is kept in client process cache.

Usage Notes

Clients discover the `down` state of server hosts when attempting connections. When a connection attempt fails, the information about the `down` state of the server host is added to the client process cache. Subsequent connection attempts by the same client process move the `down` hosts to the end of the address list, thereby reducing the priority of such hosts. When the time specified by the `SQLNET.DOWN_HOSTS_TIMEOUT` parameter has passed, the host is purged from the process cache, and its priority in the address list is restored.

Default

600 seconds (10 minutes)

Values

Any positive integer

Example

```
SQLNET.DOWN_HOSTS_TIMEOUT=60
```

5.2.38 SQLNET.ENCRYPTION_CLIENT

Purpose

To turn encryption on for the client.

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 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
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.39 SQLNET.ENCRYPTION_SERVER

Purpose

To turn encryption on for the database server.

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 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
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.40 SQLNET.ENCRYPTION_TYPES_CLIENT

Purpose

To specify a list of encryption algorithms for the client to use.

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
- `aes128` for AES (128-bit key size)
- `aes192` for AES (192-bit key size)
- `aes256` for AES (256-bit key size)
- `des` for standard DES (56-bit key size)
- `des40` for DES (40-bit key size)

- rc4_40 for RSA RC4 (40-bit key size)
- rc4_56 for RSA RC4 (56-bit key size)
- rc4_128 for RSA RC4 (128-bit key size)
- rc4_256 for RSA RC4 (256-bit key size)

The DES, DES40, 3DES112, 3DES168, RC4_40, RC4_56, RC4_128, and RC4_256 algorithms are deprecated in this release.

To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

Example

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

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.41 SQLNET.ENCRYPTION_TYPES_SERVER

Purpose

To specify a list of encryption algorithms for the database server to use.

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
- aes128 for AES (128-bit key size)
- aes192 for AES (192-bit key size)
- aes256 for AES (256-bit key size)
- des for standard DES (56-bit key size)
- des40 for DES40 (40-bit key size)
- rc4_40 for RSA RC4 (40-bit key size)
- rc4_56 for RSA RC4 (56-bit key size)
- rc4_128 for RSA RC4 (128-bit key size)
- rc4_256 for RSA RC4 (256-bit key size)

The DES, DES40, 3DES112, 3DES168, RC4_40, RC4_56, RC4_128, and RC4_256 algorithms are deprecated in this release.

To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

Example

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

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.42 SQLNET.EXPIRE_TIME

Purpose

To specify a time interval, in minutes, to send a check to verify that client/server connections are active.

Usage Notes

Setting a value greater than 0 ensures that connections are not left open indefinitely, due to an abnormal client termination. If the system supports TCP keepalive tuning, then Oracle Net Services automatically uses the enhanced detection model, and tunes the TCP keepalive parameters

If the probe finds a terminated connection, or a connection that is no longer in use, then 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
```

5.2.43 SQLNET.INBOUND_CONNECT_TIMEOUT

Purpose

To specify the time, in `ms`, `sec`, or `min`, for a client to connect with the database server and provide the necessary authentication information.

Usage Notes

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.

The default value of this parameter is appropriate for typical usage scenarios. However, if you need to explicitly set a different value, then Oracle recommends setting this parameter in combination with the INBOUND_CONNECT_TIMEOUT_listener_name parameter in the `listener.ora` file. When specifying the values for these parameters, note 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.

It accepts different timeouts with or without space between the value and the unit. In case, no unit is mentioned, the default unit is `sec`. For example, you can set `INBOUND_CONNECT_TIMEOUT_`*`listener_name`* to 2 seconds and `SQLNET.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.

Default

60 seconds

Example

```
SQLNET.INBOUND_CONNECT_TIMEOUT=3ms
```

> **See Also:**
>
> - "Control Parameters" for additional information about `INBOUND_CONNECT_TIMEOUT_`*`listener_name`*
> - *Oracle Database Net Services Administrator's Guide* for additional information about configuring these parameters

5.2.44 SQLNET.FALLBACK_AUTHENTICATION

Purpose

To specify whether password-based authentication is going to be attempted if Kerberos authentication fails. This is relevant for direct connections as well as database link connections.

Default

`FALSE`

Example

```
SQLNET.FALLBACK_AUTHENTICATION=TRUE
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.45 SQLNET.KERBEROS5_CC_NAME

Purpose

To specify the complete path name to the Kerberos credentials cache file.

Usage Notes

The `MSLSA` option specifies the file is on Microsoft Windows, and is running Microsoft KDC.

The `OS_MEMORY` option specifies that an operating system-managed memory credential is used for the credential cache file. This option is supported for all operating systems with such a feature.

Default

`/usr/tmp/krbcache` on Linux and UNIX operating systems

`c:\tmp\krbcache` on Microsoft Windows operating systems

Examples

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

```
SQLNET.KERBEROS5_CC_NAME=MSLSA
```

```
SQLNET.KERBEROS5_CC_NAME=OS_MEMORY
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.46 SQLNET.KERBEROS5_CLOCKSKEW

Purpose

To specify how many seconds can pass before a Kerberos credential is considered out of date.

Default

300

Example

```
SQLNET.KERBEROS5_CLOCKSKEW=1200
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.47 SQLNET.KERBEROS5_CONF

Purpose

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.

Usage Notes

The KDC maintains a list of user principals and is contacted through the `kinit` program for the user's initial ticket.

The `AUTO_DISCOVER` option allows the automatic discovery of KDC and realms. It is the default configuration for Kerberos clients. If there are multiple realms to be specified, then Oracle recommends creating configuration files instead of using the `AUTO_DISCOVER` option. This option is supported for all operating systems with such a feature.

Default

`/krb5/krb.conf` on Linux and UNIX operating systems

`c:\krb5\krb.conf` on Microsoft Windows operating systems

Values

- Directory path to `krb.conf` file
- `AUTO_DISCOVER`

Example

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

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.48 SQLNET.KERBEROS5_CONF_LOCATION

Purpose

To specify the directory for the Kerberos configuration file. The parameter also specifies the file is created by the system, and not by the client.

Usage Notes

The configuration file uses DNS lookup to obtain the realm for the default KDC, and maps realms to the KDC hosts. This option is supported for all operating systems with such a feature.

Default

`/krb5` on Linux and UNIX operating systems

`c:\krb5` on Microsoft Windows operating systems

Example

```
SQLNET.KERBEROS5_CONF_LOCATION=/krb5
```

5.2.49 SQLNET.KERBEROS5_KEYTAB

Purpose

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.

Default

`/etc/v5srvtab` on Linux and UNIX operating systems

`c:\krb5\v5srvtab` on Microsoft Windows operating systems

Example

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

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.50 SQLNET.KERBEROS5_REALMS

Purpose

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.

Default

`/krb5/krb.realms` on Linux and UNIX operating systems

`c:\krb5\krb.realms` on Microsoft Windows operating systems

Example

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

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.51 SQLNET.KERBEROS5_REPLAY_CACHE

Purpose

To specify replay cache is stored in operating system-managed memory on the server, and that file-based replay cache is not used.

Usage Notes

The `OS_MEMORY` option specifies the replay cache is stored in operating system-managed memory on the server, and file-based replay cache is not used.

Example

```
SQLNET_KERBEROS5_REPLAY_CACHE=OS_MEMORY
```

5.2.52 SQLNET.OUTBOUND_CONNECT_TIMEOUT

Purpose

To specify the time, in `ms`, `sec`, or `min`, for a client to establish an Oracle Net connection to the database instance.

Usage Notes

If an Oracle Net connection is not established in the time specified, then the connect attempt is terminated. The client receives an `ORA-12170: TNS:Connect timeout occurred` error.

The outbound connect timeout interval is a superset of the TCP connect timeout interval, which specifies a limit on the time taken to establish a TCP connection. Additionally, the outbound connect timeout interval includes the time taken to be connected to an Oracle instance providing the requested service. It accepts different timeouts with or without space between the value and the unit.

Without this parameter, a client connection request to the database server may block for the default TCP connect timeout duration (60 `seconds`) when the database server host system is unreachable. In case, no unit is mentioned, the default unit is `sec`.

The outbound connect timeout interval is only applicable for TCP, TCP with SSL, and IPC transport connections.

This parameter is overridden by the CONNECT_TIMEOUT parameter in the address description.

Default

None

Example

```
SQLNET.OUTBOUND_CONNECT_TIMEOUT=10 ms
```

5.2.53 SQLNET.RADIUS_ALTERNATE

Purpose

To specify an alternate RADIUS server to use in case the primary server is unavailable.

Usage Notes

The value can be either the IP address or host name of the server.

Default

None

Example

```
SQLNET.RADIUS_ALTERNATE=radius2
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.54 SQLNET.RADIUS_ALTERNATE_PORT

Purpose

To specify the listening port of the alternate RADIUS server.

Default

1645

Example

```
SQLNET.RADIUS_ALTERNATE_PORT=1667
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.55 SQLNET.RADIUS_ALTERNATE_RETRIES

Purpose

To specify the number of times the database server should resend messages to the alternate RADIUS server.

Default

3

Example

```
SQLNET.RADIUS_ALTERNATE_RETRIES=4
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.56 SQLNET.RADIUS_AUTHENTICATION

Purpose

To specify the location of the primary RADIUS server, either by its host name or IP address.

Default

Local host

Example

```
SQLNET.RADIUS_AUTHENETICATION=officeacct
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.57 SQLNET.RADIUS_AUTHENTICATION_INTERFACE

Purpose

To specify the class containing the user interface used to interact with the user.

Default

DefaultRadiusInterface

Example

```
SQLNET.RADIUS_AUTHENTICATION_INTERFACE=DefaultRadiusInterface
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.58 SQLNET.RADIUS_AUTHENTICATION_PORT

Purpose

To specify the listening port of the primary RADIUS server.

Default

1645

Example

```
SQLNET.RADIUS_AUTHENTICATION_PORT=1667
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.59 SQLNET.RADIUS_AUTHENTICATION_RETRIES

Purpose

To specify the number of times the database server should resend messages to the primary RADIUS server.

Default

3

Example

```
SQLNET.RADIUS_AUTHENTICATION_RETRIES=4
```

See Also:

Oracle Database Security Guide

5.2.60 SQLNET.RADIUS_AUTHENTICATION_TIMEOUT

Purpose

To specify the time, in seconds, that the database server should wait for a response from the primary RADIUS server.

Default

5

Example

```
SQLNET.RADIUS_AUTHENTICATION_TIMEOUT=10
```

See Also:

Oracle Database Security Guide

5.2.61 SQLNET.RADIUS_CHALLENGE_RESPONSE

Purpose

To turn challenge response on or off.

Default

off

Values

`on | off`

Example

```
SQLNET.RADIUS_CHALLENGE_RESPONSE=on
```

See Also:

Oracle Database Security Guide

5.2.62 SQLNET.RADIUS_SECRET

Purpose:

To specify the location of the RADIUS secret key.

Default

The `ORACLE_HOME/network/security/radius.key` file.

Example

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

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.63 SQLNET.RADIUS_SEND_ACCOUNTING

Purpose

To turn accounting `on` and `off`. If enabled, then packets are sent to the active RADIUS server at listening port plus one.

Usage Notes

The default port is 1646.

Default

off

Values

`on | off`

Example

```
SQLNET.RADIUS_SEND_ACCOUNTING=on
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.64 SQLNET.RECV_TIMEOUT

Purpose

To specify the time, in `ms`, `sec`, or `min`, for a database server to wait for client data after establishing a connection. It accepts different timeouts with or without space between the value and the unit A client must send some data within the time interval.

Usage Notes

Setting this parameter is recommended for environments in which clients shut down on occasion or abnormally. If a client does not send any data in time specified, then the database server logs `ORA-12535: TNS:operation timed out` and `ORA-12609: TNS: Receive timeout occurred` messages 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 client-side to specify the time, in `ms`, `sec`, or `min`, for a client to wait for response data from the database server after connection establishment. Without this parameter, the client waits a long period of time for a response from a database server saturated with requests. If you choose to set the value, then set the value to an initial low value and adjust according to system and network capacity. If necessary, use this parameter with the SQLNET.SEND_TIMEOUT parameter. In case, no unit is mentioned, the default unit is `sec`.

Default

None

Example

```
SQLNET.RECV_TIMEOUT=10ms
```

or

```
SQLNET.RECV_TIMEOUT=10 ms
```

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

5.2.65 SQLNET.SEND_TIMEOUT

Purpose

To specify the time, in `ms`, `sec`, or `min` , for a database server to complete a send operation to clients after establishing a connection.

Usage Notes

Setting this parameter is recommended for environments in which clients shut down occasionally or abnormally.

If the database server cannot complete a send operation in the time specified, then it logs `ORA-12535: TNS:operation timed out` and `ORA-12608: TNS: Send timeout occurred` messages 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 client-side to specify the time, in `ms`, `sec`, or `min` , for a client to complete send operations to the database server after connection establishment. It accepts different timeouts with or without space between the value and the unit. In case, no unit is mentioned, the default unit is `sec`.Without this parameter, the client may continue to send requests to a database server already saturated with requests. If you choose to set the value, then set the value to an initial low value and adjust according to system and network capacity. If necessary, use this parameter with the SQLNET.RECV_TIMEOUT parameter.

Default

None

Example

```
SQLNET.SEND_TIMEOUT=3 ms
```

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

5.2.66 SQLNET.USE_HTTPS_PROXY

Purpose

To enable forward HTTP proxy tunneling client connections.

Usage Notes

If turned `on`, the clients can tunnel secure connections over forward HTTP proxy using HTTP CONNECT method. This helps in accessing the public cloud database service as it eliminates the requirement to open an outbound port on a client side firewall.

This parameter is applicable with Oracle Connection Manager on the server side.

Default

`off`

Example

```
SQLNET.USE_HTTPS_PROXY=on
```

5.2.67 SQLNET.WALLET_OVERRIDE

OracleMetaLink note 340559.1.

Purpose

To determine whether the client should override the strong authentication credential with the password credential in the stored wallet to log in to the database.

Usage Notes

When wallets are used for authentication, the database credentials for user name and password are securely stored in an Oracle wallet. The auto-login feature of the wallet is turned on so the database does not need a password to open the wallet. From the wallet, the database gets the credentials to access the database for the user.

Wallet usage can simplify large-scale deployments that rely on password credentials for connecting to databases. When this feature is configured, application code, batch jobs, and scripts do not need embedded user names and passwords. Risk is reduced because such passwords are no longer exposed in the clear, and password management policies are more easily enforced without changing application code whenever user names or passwords change.

Users connect using the `connect /@`*`database_name`* command instead of specifying a user name and password explicitly. This simplifies the maintenance of the scripts and secures the password management for the applications.

Middle-tier applications create an Oracle Applications wallet at installation 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.

Values

```
true | false
```

Examples

```
SQLNET.WALLET_OVERRIDE=true
```

> **See Also:**
>
> In order to use wallets, a wallet must be configured on the client. Refer to *Oracle Database Security Guide* for additional information about configuring the clients.

5.2.68 SSL_CERT_REVOCATION

Purpose

To configure a revocation check for a certificate.

> **See Also:**
>
> *Oracle Database Security Guide*

Default

`none`

Values

- `none` disables certificate revocation status checking. This is the default value.

  > **Note:**
  >
  > Oracle recommends that you do not set the `SSL_CERT_REVOCATION` parameter to `none` because this removes a critical component in certificate-based authentication. Without certificate revocation status checking, you cannot protect against stolen certificates that are used for authentication. Set the `none` value only in cases where mitigating controls safeguard the use of certificates for authentication, such as network access control lists or Oracle Database Vault policies that limit the database connection to trusted clients.

- `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
```

5.2.69 SSL_CRL_FILE

Purpose

To specify the name of the file where you can assemble the CRL for client authentication.

Usage Notes

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

Default

None

Example

```
SSL_CRL_FILE=
```

5.2.70 SSL_CRL_PATH

Purpose

To specify the destination directory of the CRL of CA.

Usage Notes

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`.

Default

None

Example

```
SSL_CRL_PATH=
```

5.2.71 SSL_CIPHER_SUITES

Purpose

To control which combination of encryption and data integrity is used by the Secure Sockets Layer (SSL). Cipher suites that use Advanced Encryption Standard (AES) only work with Transport Layer Security (TLS 1.0).

Default

None

Values

Ciphers updated based on bug 19139199.

- `SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`
- `SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA`
- `SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256`
- `SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA`
- `SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384`
- `SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384`
- `SSL_RSA_WITH_AES_128_CBC_SHA256`
- `SSL_RSA_WITH_AES_128_GCM_SHA256`
- `SSL_RSA_WITH_AES_128_CBC_SHA`
- `SSL_RSA_WITH_AES_256_CBC_SHA`

- `SSL_RSA_WITH_AES_256_CBC_SHA256`
- `SSL_RSA_WITH_AES_256_GCM_SHA384`
- `SSL_RSA_WITH_RC4_128_MD5`
- `SSL_RSA_WITH_RC4_128_SHA`
- `SSL_RSA_WITH_3DES_EDE_CBC_SHA`
- `SSL_DH_anon_WITH_3DES_EDE_CBC_SHA`
- `SSL_DH_anon_WITH_RC4_128_MD5`

> **Note:**
>
> `SSL_DH_anon_WITH_3DES_EDE_CBC_SHA` and `SSL_DH_anon_WITH_RC4_128_MD5` do not the provide authentication of the communicating parties, and can be vulnerable to man-in-the-middle attacks. Oracle recommends that you do not use these cipher suites to protect sensitive data. However, they are useful if the communicating parties want to remain anonymous or simply do not want the overhead caused by mutual authentication.

Example

```
SSL_CIPHER_SUITES=(ssl_rsa_with_aes_128_cbc_sha256)
```

> **See Also:**
>
> *Oracle Database Security Guide* for additional information about cipher suite values

5.2.72 SSL_EXTENDED_KEY_USAGE

Purpose

To specify the purpose of the key in the certificate.

Usage Notes

When this parameter is specified, the certificate with the matching extended key is used.

Values

`client authentication`

Example

```
SSL_EXTENDED_KEY_USAGE="client authentication"
```

5.2.73 SSL_SERVER_DN_MATCH

Purpose

To enforce that the distinguished name (DN) for the database server matches its service name.

Usage Notes

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.

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

Default

no

Values

- `yes` | `on` | `true` 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 not enforce a match. If the DN does not match the service name, then the connection is successful, but an error is logged to the `sqlnet.log` file.

Example

```
SSL_SERVER_DN_MATCH=yes
```

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.74 SSL_VERSION

Purpose

To limit allowable SSL or TLS versions used for connections.

Usage Notes

Clients and database servers must use a compatible version. This parameter should only be used when absolutely necessary for backward compatibility. The current default uses TLS version 1.2 which is the version required for multiple security compliance requirements.

If you set `SSL_VERSION` to `undetermined`, then by default it uses `3.0`.

Default

```
1.2
```

Values

> **Note:**
>
> The `sqlnet.ora parameter ADD_SSLV3_TO_DEFAULT` has no impact on this parameter.

```
undetermined | 3.0 | 1.0| 1.1 | 1.2
```

If you want to specify one version or another version, then use “or”. The following values are permitted:

```
1.0 or 3.0 | 1.2 or 3.0 | 1.1 or 1.0 | 1.2 or 1.0 | 1.2 or 1.1 | 1.1 or 1.0 or 
3.0 |
1.2 or 1.0 or 3.0 | 1.2 or 1.1 or 1.0 | 1.2 or 1.1 or 3.0 |1.2 or 1.1 or 1.0 or 
3.0
```

Example

```
SSL_VERSION=1.2
```

The remaining version numbers correspond to the TLS versions, such as, TLSv1.0, TLSv1.1, and TLSv1.2.

> **See Also:**
>
> *Oracle Database Security Guide*

5.2.75 TCP.CONNECT_TIMEOUT

Purpose

To specify the time, in `ms`, `sec`, or `min`, for a client to establish a TCP connection (`PROTOCOL=tcp` in the TNS connect address) to the database server.

Usage Notes

If a TCP connection to the database host is not established in the specified time, then the connection attempt is terminated. The client receives an `ORA-12170: TNS:Connect timeout occurred` error.

The timeout applies to each IP address that resolves to a host name. It accepts different timeouts with or without space between the value and the unit. For example, if a host name resolves to an IPv6 and an IPv4 address, and if the host is not reachable through the network, then the connection request times out twice because there are two IP addresses. In this example, the default timeout setting of 60 causes a timeout in 120 `seconds`. In case, no unit is mentioned, the default unit is `sec`.

Default

60

Example

```
TCP.CONNECT_TIMEOUT=10 ms
```

5.2.76 TCP.EXCLUDED_NODES

Purpose

To specify which clients are denied access to the database.

Usage Notes

This parameter is only valid when the TCP.VALIDNODE_CHECKING parameter is set to `yes`.

This parameter can use wildcards for IPv4 addresses and CIDR notation for IPv4 and IPv6 addresses.

Syntax

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

Example

```
TCP.EXCLUDED_NODES=(finance.us.example.com, mktg.us.example.com, 192.0.2.25,
 172.30.*, 2001:DB8:200C:417A/32)
```

5.2.77 TCP.INVITED_NODES

Purpose

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, ...)
```

Usage Notes

- This parameter is only valid when the TCP.VALIDNODE_CHECKING parameter is set to `yes`.
- This parameter can use wildcards for IPv4 addresses and CIDR notation for IPv4 and IPv6 addresses.

Example

```
TCP.INVITED_NODES=(sales.us.example.com, hr.us.example.com, 192.0.*,
 2001:DB8:200C:433B/32)
```

5.2.78 TCP.NODELAY

Purpose

To preempt delays in buffer flushing within the TCP/IP protocol stack.

Default

yes

Values

```
yes | no
```

Example

```
TCP.NODELAY=yes
```

5.2.79 TCP.QUEUESIZE

Purpose

To configure the maximum length of the queue for pending connections on a TCP listening socket.

Default

System-defined maximum value. The defined maximum value for Linux is 128.

Values

Any integer value up to the system-defined maximum.

Examples

```
TCP.QUEUESIZE=100
```

5.2.80 TCP.VALIDNODE_CHECKING

Purpose

To enable and disable valid node checking for incoming connections.

Usage Notes

If this parameter is set to `yes`, then incoming connections are allowed only if they originate from a node that conforms to list specified by TCP.INVITED_NODES or TCP.EXCLUDED_NODES parameters.

The TCP.INVITED_NODES and TCP.EXCLUDED_NODES parameters are valid only when the TCP.VALIDNODE_CHECKING parameter is set to `yes`.

This parameter and the depending parameters, TCP.INVITED_NODES and TCP.EXCLUDED_NODES must be set in the `sqlnet.ora` file of the listener. This is important in an Oracle RAC environment where the listener runs out of the Oracle Grid Infrastructure home. Setting the parameter in the database home does not have any effect in Oracle RAC environments. In such environments, the address of all Single Client Access Name (SCANs), Virtual IPs (VIPs), local IP must be included in the TCP.INVITED_NODES list.

In VLAN environments, the `sqlnet.ora` file present in the Oracle Grid Infrastructure home should include all the addresses of all the VLANs. The VLANs perform the network segregation, whereas the INVITED_NODES allows or restricts access to databases within the VLANs.

If multiple databases within the same VLAN require different INVITED_NODE lists, then separate listeners are required.

Default

no

Values

`yes | no`

Example

```
TCP.VALIDNODE_CHECKING=yes
```

5.2.81 TNSPING.TRACE_DIRECTORY

Purpose

To specify the destination directory for the TNSPING utility trace file, `tnsping.trc`.

Default

The `ORACLE_HOME/network/trace` directory.

Example

```
TNSPING.TRACE_DIRECTORY=/oracle/traces
```

5.2.82 TNSPING.TRACE_LEVEL

Purpose

To turn TNSPING utility tracing on at a specified level or to turn it 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
```

5.2.83 USE_CMAN

Purpose

To specify client routing to Oracle Connection Manager.

Usage Notes

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

If set to `false`, then 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, then connections are routed through any available listener address.

Default

false

Values

`true | false`

Example

```
USE_CMAN=true
```

5.2.84 USE_DEDICATED_SERVER

Purpose

To append `(SERVER=dedicated)` to the `CONNECT_DATA` section of the connect descriptor used by the client.

Usage Notes

It overrides the current value of the SERVER parameter in the `tnsnames.ora` file.

If set to `on`, then 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.

Default

off

Values

- `on` to append `(SERVER=dedicated)`

- `off` to send requests to existing server processes

Example

```
USE_DEDICATED_SERVER=on
```

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

5.2.85 WALLET_LOCATION

Purpose

To specify the location of wallets. Wallets are certificates, keys, and trustpoints processed by SSL.

Usage Notes

The key/value pair for Microsoft 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 Microsoft Windows registry and the wallet's key (`KEY`) is `SALESAPP`, then 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`.

Syntax

The syntax depends on the wallet, as follows:

- 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 Microsoft 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)))
```

Additional Parameters

`WALLET_LOCATION` supports the following parameters:

- `SOURCE`: The type of storage for wallets, and storage location.
- `METHOD`: The type of storage.
- `METHOD_DATA`: The storage location.
- `DIRECTORY`: The location of Oracle wallets on file system.
- `KEY`: The wallet type and location in the Microsoft Windows registry.
- `PROFILE`: The Entrust profile file (`.epf`).
- `INIFILE`: The Entrust initialization file (`.ini`).

Default

None

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 Microsoft 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)))
```

> **See Also:**
>
> *Oracle Database Enterprise User Security Administrator's Guide*

5.3 ADR Diagnostic Parameters in sqlnet.ora

Since Oracle Database 11*g*, Oracle Database includes an advanced fault diagnosability infrastructure for preventing, detecting, diagnosing, and resolving problems. The problems are critical errors such as those caused by database code bugs, metadata corruption, and customer data corruption.

When a critical error occurs, it is assigned an incident number, and diagnostic data for the error, such as traces and dumps, is immediately captured and tagged with the incident number. The data is then stored in the Automatic Diagnostic Repository (ADR), a file-based repository outside the database.

This section describes the parameters used when ADR is enabled. Non-ADR parameters listed in the `sqlnet.ora` file are ignored when ADR is enabled. "Non-ADR Diagnostic Parameters in sqlnet.ora" Non-ADR Diagnostic Parameters in sqlnet.ora describes the parameters used when ADR is disabled. ADR is enabled by default.

The following `sqlnet.ora` parameters are used when ADR is enabled (when `DIAG_ADR_ENABLED` is set to `on`):

- ADR_BASE
- DIAG_ADR_ENABLED
- TRACE_LEVEL_CLIENT
- TRACE_LEVEL_SERVER
- TRACE_TIMESTAMP_CLIENT
- TRACE_TIMESTAMP_SERVER

5.3.1 ADR_BASE

It is a diagnostic parameter in the `sqlnet.ora` file and it specifies the base location of the ADR files.

Purpose

To specify the base directory into which tracing and logging incidents are stored when ADR is enabled.

Usage Notes

This parameter is applicable only to clients. On the server side, the ADR base location is defined by the `DIAGNOSTIC_DEST` initialization parameter in the `init.ora` file. See DIAGNOSTIC_DEST in *Oracle Database Reference*.

Default

ORACLE_BASE or ORACLE_HOME/log (if ORACLE_BASE is not defined)

Values

Any valid directory path to a directory with write permission.

Example

```
ADR_BASE=/oracle/network/trace
```

5.3.2 DIAG_ADR_ENABLED

Purpose

To specify whether ADR tracing is enabled.

Usage Notes

If the DIAG_ADR_ENABLED parameter is set to OFF, then non-ADR file tracing is used.

Default

on

Values

on | off

Example

```
DIAG_ADR_ENABLED=on
```

5.3.3 TRACE_LEVEL_CLIENT

Purpose

To turn client tracing on at a specified level or to turn it off.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

off or 0

Values

- off or 0 for no trace output
- user or 4 for user trace information
- admin or 10 for administration trace information
- support or 16 for Oracle Support Services trace information

Example

```
TRACE_LEVEL_CLIENT=user
```

5.3.4 TRACE_LEVEL_SERVER

Purpose

To turn server tracing on at a specified level or to turn it off.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

off or 0

Values

- `off` or `0` for no trace output
- `user` or `4` for user trace information
- `admin` or `10` for administration trace information
- `support` or `16` for Oracle Support Services trace information

Example

```
TRACE_LEVEL_SERVER=admin
```

5.3.5 TRACE_TIMESTAMP_CLIENT

Purpose

To add a time stamp in the form of `dd-mmm-yyyy hh:mm:ss:mil` to every trace event in the client trace file, which has a default name of `sqlnet.trc`.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

on

Values

`on` or `true` | `off` or `false`

Example

```
TRACE_TIMESTAMP_CLIENT=true
```

5.3.6 TRACE_TIMESTAMP_SERVER

Purpose

To add a time stamp in the form of `dd-mmm-yyyy hh:mm:ss:mil` to every trace event in the database server trace file, which has a default name of `svr_*pid*.trc`.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

on

Values

`on` or `true` | `off` or `false`

Example

```
TRACE_TIMESTAMP_SERVER=true
```

5.4 Non-ADR Diagnostic Parameters in sqlnet.ora

This section lists the `sqlnet.ora` parameters used when ADR is disabled.

> **Note:**
>
> The default value of DIAG_ADR_ENABLED is `on`. Therefore, the `DIAG_ADR_ENABLED` parameter must explicitly be set to `off` in order for non-ADR tracing to be used.

- LOG_DIRECTORY_CLIENT
- LOG_DIRECTORY_SERVER
- LOG_FILE_CLIENT
- LOG_FILE_SERVER
- TRACE_DIRECTORY_CLIENT
- TRACE_DIRECTORY_SERVER
- TRACE_FILE_CLIENT
- TRACE_FILE_SERVER
- TRACE_FILELEN_CLIENT
- TRACE_FILELEN_SERVER
- TRACE_FILENO_CLIENT
- TRACE_FILENO_SERVER
- TRACE_UNIQUE_CLIENT

5.4.1 LOG_DIRECTORY_CLIENT

Purpose

To specify the destination directory for the client log file.

Usage Notes

Use this parameter when ADR is not enabled.

Default

```
ORACLE_HOME/network/log
```

Values

Any valid directory path.

Example

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

5.4.2 LOG_DIRECTORY_SERVER

Purpose

To specify the destination directory for the database server log file.

Usage Notes

Use this parameter when ADR is not enabled.

Default

```
ORACLE_HOME/network/trace
```

Values

Any valid directory path to a directory with write permission.

Example

```
LOG_DIRECTORY_SERVER=/oracle/network/trace
```

5.4.3 LOG_FILE_CLIENT

Purpose

To specify the name of the log file for the client.

Usage Notes

Use this parameter when ADR is not enabled.

Default

```
ORACLE_HOME/network/log/sqlnet.log
```

Values

The default value cannot be changed.

5.4.4 LOG_FILE_SERVER

Purpose

To specify the name of the log file for the database server.

Usage Notes

Use this parameter when ADR is not enabled.

Default

`sqlnet.log`

Values

Any valid directory path to a directory with write permission.

Example

```
LOG_FILE_SERVER=svr.log
```

5.4.5 TRACE_DIRECTORY_CLIENT

Purpose

To specify the destination directory for the client trace file.

Usage Notes

Use this parameter when ADR is not enabled.

Default

`ORACLE_HOME/network/trace`

Values

Any valid directory path to a directory with write permission.

Example

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

5.4.6 TRACE_DIRECTORY_SERVER

Purpose

To specify the destination directory for the database server trace file. Use this parameter when ADR is not enabled.

Default

```
ORACLE_HOME/network/trace
```

Values

Any valid directory path to a directory with write permission.

Example

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

5.4.7 TRACE_FILE_CLIENT

Purpose

To specify the name of the client trace file.

Usage Notes

Use this parameter when ADR is not enabled.

Default

```
ORACLE_HOME/network/trace/cli.trc
```

Values

Any valid file name.

Example

```
TRACE_FILE_CLIENT=clientsqlnet.trc
```

5.4.8 TRACE_FILE_SERVER

Purpose

To specify the destination directory for the database server trace output.

Usage Notes

Use this parameter when ADR is not enabled.

Default

```
ORACLE_HOME/network/trace/svr_pid.trc
```

Values

Any valid file name. The process identifier (pid) is appended to the name automatically.

Example

```
TRACE_FILE_SERVER=svrsqlnet.trc
```

5.4.9 TRACE_FILEAGE_CLIENT

Purpose

To specify the maximum age of client trace files in minutes.

Usage Notes

When the age limit is reached, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO_CLIENT parameter. Use this parameter when ADR is not enabled.

Default

Unlimited

This is the same as setting the parameter to `0`.

Example 5-2   Example

```
TRACE_FILEAGE_CLIENT=60
```

5.4.10 TRACE_FILEAGE_SERVER

Purpose

To specify the maximum age of database server trace files in minutes.

Usage Notes

When the age limit is reached, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO_SERVER parameter. Use this parameter when ADR is not enabled.

Default

Unlimited

This is the same as setting the parameter to`0`.

Example 5-3   Example

```
TRACE_FILEAGE_SERVER=60
```

5.4.11 TRACE_FILELEN_CLIENT

Purpose

To specify the size of the client trace files in kilobytes (KB).

Usage Notes

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. Use this parameter when ADR is not enabled.

Example

```
TRACE_FILELEN_CLIENT=100
```

5.4.12 TRACE_FILELEN_SERVER

Purpose

To specify the size of the database server trace files in kilobytes (KB).

Usage Notes

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. Use this parameter when ADR is not enabled.

Example

```
TRACE_FILELEN_SERVER=100
```

5.4.13 TRACE_FILENO_CLIENT

Purpose

To specify the number of trace files for client tracing.

Usage Notes

When this parameter is set 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, then the first file is re-used, and so on.

When this parameter is set with theTRACE_FILEAGE_CLIENT parameter, trace files are cycled based on their age. The first file is used until the age limit is reached, then the second file is used, and so on. When the last file's age limit is reached, the first file is re-used, and so on.

When this parameter is set with both the TRACE_FILELEN_CLIENT and TRACE_FILEAGE_CLIENT parameters, trace files are cycled when either the size limit or the age limit is reached.

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, then 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. Use this parameter when ADR is not enabled.

Default

None

Example

```
TRACE_FILENO_CLIENT=3
```

5.4.14 TRACE_FILENO_SERVER

Purpose

To specify the number of trace files for database server tracing.

Usage Notes

When this parameter is set 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, then the first file is re-used, and so on.

When this parameter is set with theTRACE_FILEAGE_SERVER parameter, trace files are cycled based on the age of the trace file. The first file is used until the age limit is reached, then the second file is used, and so on. When the last file's age limit is reached, the first file is re-used, and so on.

When this parameter is set with both the TRACE_FILELEN_SERVER and TRACE_FILEAGE_SERVER parameters, trace files are cycled when either the size limit or the age limit is reached.

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, then 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. Use this parameter when ADR is not enabled.

Default

None

Example

```
TRACE_FILENO_SERVER=3
```

5.4.15 TRACE_UNIQUE_CLIENT

Purpose

To specify whether a unique trace file is created for each client trace session.

Usage Notes

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. Use this parameter when ADR is not enabled.

Default

on

Values

`on` or `off`

Example

```
TRACE_UNIQUE_CLIENT=on
```

6

Local Naming Parameters in the tnsnames.ora File

This chapter provides a complete listing of the `tnsnames.ora` file configuration parameters. This chapter contains the following topics:

6.1 Overview of Local Naming Parameters

The `tnsnames.ora` file is a configuration file that contains network service names mapped to connect descriptors for the local naming method, or net service names mapped to listener protocol addresses.

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, the `tnsnames.ora` file is located in the `ORACLE_HOME/network/admin` directory. Oracle Net will check the other directories for the configuration file. For example, the order checking the `tnsnames.ora` file is as follows:

1. The directory specified by the `TNS_ADMIN` environment variable. If the file is not found in the directory specified, then it is assumed that the file does not exist.
2. If the `TNS_ADMIN` environment variable is not set, then Oracle Net checks the `ORACLE_HOME/network/admin` directory.

> **Note:**
>
> On Microsoft Windows, the TNS_ADMIN environment variable is used if it is set in the environment of the process. If the TNS_ADMIN environment variable is not defined in the environment, or the process is a service which does not have an environment, then Microsoft Windows scans the registry for a TNS_ADMIN parameter.

> **See Also:**
>
> - *Oracle Database Global Data Services Concepts and Administration Guide* for information about management of global services
> - Oracle operating system-specific documentation

6.2 General Syntax of tnsnames.ora

The basic syntax for a tnsnames.ora file is shown in Example 6-1.

Example 6-1 Basic Format of tnsnames.ora File

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

In the preceding example, DESCRIPTION contains the connect descriptor, ADDRESS contains the protocol address, and CONNECT_DATA contains the database service identification information.

6.3 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.

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

```
net_service_name=
 (DESCRIPTION_LIST=
  (DESCRIPTION=

   (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521))
   (CONNECT_DATA=
     (SERVICE_NAME=sales.us.example.com)))
  (DESCRIPTION=
   (ADDRESS=(PROTOCOL=tcp)(HOST=hr1-svr)(PORT=1521))
   (ADDRESS=(PROTOCOL=tcp)(HOST=hr2-svr)(PORT=1521))
   (CONNECT_DATA=
     (SERVICE_NAME=hr.us.example.com))))
```

Note:

Oracle Net Manager does not support the creation of multiple connect descriptors for a net service name when using Oracle Connection Manager.

6.4 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 Example 6-3, two address lists are presented. The first address list features client load balancing and no connect-time failover, affecting only those protocol addresses within its `ADDRESS_LIST`. The second protocol address list features no client load loading balancing, but does have connect-time failover, affecting only those protocol addresses within its `ADDRESS_LIST`. The client first tries the first or second protocol address at random, then tries protocol addresses three and four sequentially.

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:

- Oracle Net Manager supports only the creation of one protocol address list for a connect descriptor.
- Oracle Net Services supports the IFILE parameter in the `tnsnames.ora` file, with up to three levels of nesting. The parameter is added manually to the file. The following is an example of the syntax:

  ```
  IFILE=/tmp/listener_em.ora
  IFILE=/tmp/listener_cust1.ora
  IFILE=/tmp/listener_cust2.ora
  ```

  Refer to *Oracle Database Reference* for additional information.

6.5 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.

Example 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_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=host1)(PORT=1630))    # 1
     (ADDRESS_LIST=
       (FAILOVER=on)
       (LOAD_BALANCE=off)                               #  2
       (ADDRESS=(PROTOCOL=tcp)(HOST=host2a)(PORT=1630))
       (ADDRESS=(PROTOCOL=tcp)(HOST=host2b)(PORT=1630)))
     (ADDRESS=(PROTOCOL=tcp)(HOST=host3)(PORT=1521)))    #  3
   (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

In Example 6-4, the syntax does the following:

1. The client is instructed to connect to the 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 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. Oracle Connection Manager connects to the database service using the following protocol address:

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

Example 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)                                    # 1
   (FAILOVER=on)
   (ADDRESS_LIST=
     (SOURCE_ROUTE=yes)
     (ADDRESS=(PROTOCOL=tcp)(HOST=host1)(PORT=1630))    # 2
```

```
     (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.example.com)))  # 3
```

In Example 6-5, the syntax does the following:

1. The client is instructed to pick an ADDRESS_LIST at random and to fail over 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 Oracle Connection Manager, using the Oracle Connection Manager protocol address that uses port 1630 indicated for the ADDRESS_LIST.
3. Oracle Connection Manager then connects to the database service, using the protocol address indicated for the ADDRESS_LIST.

6.6 Connect Descriptor Descriptions

Each connect descriptor is contained within the DESCRIPTION parameter. Multiple connect descriptors are characterized by the DESCRIPTION_LIST parameter. These parameters are described in this section.

6.6.1 DESCRIPTION

Purpose

To specify a container for a connect descriptor.

Usage Notes

When using more than one DESCRIPTION parameter, put the parameters under the DESCRIPTION_LIST parameter.

Example

```
net_service_name=
(DESCRIPTION=
  (ADDRESS=...)
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

6.6.2 DESCRIPTION_LIST

Purpose

To define a list of connect descriptors for a particular net service name.

Example

```
net_service_name=
(DESCRIPTION_LIST=
 (DESCRIPTION=
  (ADDRESS=...)
  (CONNECT_DATA=(SERVICE_NAME=sales.example.com)))
 (DESCRIPTION=
  (ADDRESS=...)
  (CONNECT_DATA=(SERVICE_NAME=sales2.us.example.com))))
```

6.7 Protocol Address Section

The protocol address section of the `tnsnames.ora` file specifies the protocol addresses of the listener. If there is only one listener protocol address, then use the ADDRESS parameter. If there is more than one address, then use the ADDRESS_LIST parameter.

6.7.1 ADDRESS

Purpose

To define a single listener protocol address.

Usage Notes

Put this parameter under either the `ADDRESS_LIST` parameter or the `DESCRIPTION` parameter.

Example

```
net_service_name=
(DESCRIPTION=
 (ADDRESS=(PROTOCOL=tcp)(HOST=sales-svr)(PORT=1521))
 (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

> **See Also:**
>
> Protocol Address Configuration for descriptions of the correct parameters to use for each protocol

6.7.1.1 HTTPS_PROXY

Purpose

To specify HTTP proxy host name for tunneling SSL client connections.

Usage Notes

The clients can tunnel secure connections over forward HTTP proxy using HTTP CONNECT method. This helps in accessing the public cloud database service as it eliminates the requirement to open an outbound port on a client side firewall. This parameter is applicable only to the connect descriptors where `PROTOCOL=TCPS`. This is similar to the web browser setting for intranet users who want to connect to internet hosts. Increase the forward web proxy read timeout for requests to a higher value depending on client queries. Otherwise, the forward web proxy closes the connection assuming that no requests are made from the client.

Successful connection depends on specific proxy configurations. The performance of data transfers depends on proxy capacity. Oracle recommends not to use this feature in production environments where performance is critical.

Configuring `tnsnames.ora` for the HTTP proxy may not be enough depending your organization's network configuration and security policies. For example, some networks require a username and password for the HTTP proxy.

Oracle Client versions earlier than 18c does not support connections through HTTP proxy.

Contact your network administrator to open outbound connections to hosts in the `oraclecloud.com` domain using the relevant port, without going through an HTTP proxy. For example, port 1522.

Default

None

Values

HTTP proxy host name that can make an outbound connection to the internet hosts.

Example

```
HTTPS_PROXY=www-proxy.mycompany.com
```

6.7.1.2 HTTPS_PROXY_PORT

Purpose

To specify forward HTTP proxy host port for tunneling SSL client connections.

Usage Notes

It forwards the HTTP proxy host port that receives HTTP CONNECT method. This parameter should be used along with `HTTPS_PROXY_PORT`. This value takes effect only when `SQLNET.USE_HTTPS_PROXY=1` is set in `sqlnet.ora`.

Default

none

Values

port number

Example

```
HTTPS_PROXY_PORT=80
```

6.7.2 ADDRESS_LIST

Purpose

To define a list of protocol addresses.

Usage Notes

If there is only one listener protocol address, then ADDRESS_LIST is not necessary.

Put 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)))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

6.8 Optional Parameters for Address Lists

For multiple addresses, the following parameters are available:

- ENABLE
- FAILOVER
- LOAD_BALANCE
- RECV_BUF_SIZE
- SDU
- SEND_BUF_SIZE
- SOURCE_ROUTE
- TYPE_OF_SERVICE

6.8.1 ENABLE

Purpose

To allow the caller to detect a terminated remote server, typically it takes 2 hours or more to notice.

Usage Notes

The keepalive feature on the supported TCP transports can be enabled for a net service client by putting (ENABLE=broken) under the DESCRIPTION parameter in the connect string. On the client side, the default for tcp_keepalive is off. Operating system TCP configurables, which vary by platform, define the actual keepalive timing details.

Values

broken

Example

```
net_service_name=
 (DESCRIPTION=
  (ENABLE=broken)
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

Although the preceding example has multiple addresses, the `ADDRESS_LIST` parameter was not used. This is because the `ADDRESS_LIST` parameter is not mandatory.

6.8.2 FAILOVER

Purpose

To enable or disable connect-time failover for multiple protocol addresses.

Usage Notes

When you set the parameter to `on`, `yes`, or `true`, Oracle Net fails over at connect time 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.

Put this parameter under the `DESCRIPTION_LIST` parameter, the `DESCRIPTION` parameter, or the `ADDRESS_LIST` parameter.

> **Note:**
>
> 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 the `DESCRIPTION_LIST`, `DESCRIPTION`, and `ADDRESS_LIST` parameters

Values

- `yes` | `on` | `true`
- `no` | `off` | `false`

Example

```
net_service_name=
 (DESCRIPTION=
  (FAILOVER=on)
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

6.8.3 LOAD_BALANCE

Purpose

To enable or disable client load balancing for multiple protocol addresses.

Usage Notes

When you set the parameter to `on`, `yes`, or `true`, Oracle Net goes 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.

Put this parameter under the `DESCRIPTION_LIST` parameter, the `DESCRIPTION` parameter, or the `ADDRESS_LIST` parameter.

Default

`on` for `DESCRIPTION_LIST`

Values

- `yes` | `on` | `true`
- `no` | `off` | `false`

Example

```
net_service_name=
 (DESCRIPTION=
  (LOAD_BALANCE=on)
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

6.8.4 RECV_BUF_SIZE

Purpose

To specify, in bytes, the buffer space for receive operations of sessions.

Usage Notes

This parameter is supported by the TCP/IP, TCP/IP with SSL, and SDP protocols.

Put this parameter under the `DESCRIPTION` parameter or at the end of the protocol address.

Setting this parameter in the connect descriptor for a client overrides the RECV_BUF_SIZE parameter at the client-side `sqlnet.ora` file.

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

Default

The default value for this parameter is specific to the operating system. The default for the Linux 2.6 operating system is 87380 bytes.

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.example.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.example.com)))
```

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

6.8.5 SDU

Purpose

To instruct Oracle Net to optimize the transfer rate of data packets being sent across the network with a specified session data unit (SDU) size.

Usage Notes

Put this parameter under the `DESCRIPTION` parameter.

Setting this parameter in the connect descriptor for a client overrides the `DEFAULT_SDU_SIZE` parameter at client-side `sqlnet.ora` file.

Default

8192 bytes (8 KB)

Values

512 to 2097152 bytes.

Example

```
net_service_name=
 (DESCRIPTION=
  (SDU=8192)
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521))
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com))
```

6.8.6 SEND_BUF_SIZE

Purpose

To specify, in bytes, the buffer space for send operations of sessions.

Usage Notes

This parameter is supported by the TCP/IP, TCP/IP with SSL, and SDP protocols.

Put this parameter under the `DESCRIPTION` parameter or at the end of the protocol address.

Setting this parameter in the connect descriptor for a client overrides the SEND_BUF_SIZE parameter at the client-side `sqlnet.ora` file.

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

Default

The default value for this parameter is operating system specific. The default for the Linux 2.6 operating system is 16 KB.

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.example.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.example.com)))
```

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

6.8.7 SOURCE_ROUTE

Purpose

To enable routing through multiple protocol addresses.

Usage Notes

When you set this parameter 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.

Put this parameter under either the `DESCRIPTION_LIST` parameter, the `DESCRIPTION` parameter, or the `ADDRESS_LIST` parameter.

Default

off

Values

- `yes` | `on`
- `no` | `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.example.com))
```

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

6.8.8 TYPE_OF_SERVICE

Purpose

To specify the type of service to use for an Oracle Rdb database.

Usage Notes

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.

Put 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.example.com))
   (TYPE_OF_SERVICE=oracle11_database)))
```

6.9 Connection Data Section

The connection data section of the `tnsnames.ora` file specifies the name of the destination service. The following parameters are available:

- CONNECT_DATA
- FAILOVER_MODE
- GLOBAL_NAME
- HS
- INSTANCE_NAME
- RDB_DATABASE
- SERVER
- SERVICE_NAME

6.9.1 CONNECT_DATA

Purpose

To define the service to which to connect, such as `SERVICE_NAME`.

Usage Notes

Put this parameter under the DESCRIPTION parameter.

CONNECT_DATA permits the following additional parameters:

- FAILOVER_MODE
- GLOBAL_NAME
- HS
- INSTANCE_NAME
- RDB_DATABASE
- SHARDING_KEY
- SUPER_SHARDING_KEY
- SERVER
- SERVICE_NAME

Example

```
net_service_name=
 (DESCRIPTION=
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com)))
```

6.9.1.1 SHARDING_KEY

Purpose

To route the database request to a particular shard.

Usage Notes

Put this parameter under the CONNECT_DATA parameter.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=...)
     (ADDRESS=...))
   (CONNECT_DATA=
     (SERVICE_NAME=sales.us.example.com)
     ((SHARDING_KEY=40598230))))
```

> **See Also:**
>
> *Oracle Database Net Services Administrator's Guide* for additional information about the use of the SHARDING_KEY parameter

6.9.1.2 SUPER_SHARDING_KEY

Purpose

To route the database request to a collection of shards.

Usage Notes

Put this parameter under the CONNECT_DATA parameter.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=...)
     (ADDRESS=...))
   (CONNECT_DATA=
     (SERVICE_NAME=sales.us.example.com)
     ((SHARDING_KEY=40598230)(SUPER_SHARDING_KEY=gold)))
```

> **See Also:**
>
> *Oracle Database Net Services Administrator's Guide* for additional information about the use of the SUPER_SHARDING_KEY parameter

6.9.2 FAILOVER_MODE

Purpose

To instruct Oracle Net to fail over to a different listener if the first listener fails during run time.

Usage Notes

Depending upon the configuration, the 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.

Put this parameter under the CONNECT_DATA parameter.

Additional Parameters

FAILOVER_MODE supports the following parameters:

- BACKUP: Specifies the failover node by its net service name. A separate net service name must be created for the failover node.
- TYPE: Specifies 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. For example, if a user's connection is lost, then 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 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: Specifies 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.
- TRANSACTION: Allows the database to complete the current database transaction following a recoverable error. This parameter is used with the COMMIT_OUTCOME=TRUE parameter.
- RETRIES: Specifies the number of times to attempt to connect after a failover. If DELAY is specified, then RETRIES defaults to five retry attempts.
- DELAY: Specifies the amount of time in seconds to wait between connect attempts. If RETRIES is specified, then DELAY defaults to one second.

Note:

If a callback function is registered, then RETRIES and DELAY parameters are ignored.

See Also:

Oracle Database Net Services Administrator's Guide for additional configuration information

6.9.3 GLOBAL_NAME

Purpose

To identify the Oracle Rdb database.

Usage Notes

Put this parameter under the CONNECT_DATA parameter.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS_LIST=
    (ADDRESS=...)
    (ADDRESS=...))
   (CONNECT_DATA=
    (SERVICE_NAME=generic)
    (RDB_DATABASE=[.mf]mf_personal.rdb)
    (GLOBAL_NAME=alpha5)))
```

6.9.4 HS

Purpose

To direct Oracle Net to connect to a non-Oracle system through Heterogeneous Services.

Usage Notes

Put this parameter under the `CONNECT_DATA` parameter.

Default

None

Values

`ok`

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=...)
     (ADDRESS=...))
   (CONNECT_DATA=
    (SID=sales6)
    )
(HS=ok))
```

See Also:

Oracle Database Net Services Administrator's Guide for complete configuration information

6.9.5 INSTANCE_NAME

Purpose

To identify the database instance to access.

Usage Notes

Set the value to the value specified by the INSTANCE_NAME parameter in the initialization parameter file.

Put this parameter under the CONNECT_DATA parameter.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=...)
     (ADDRESS=...))
   (CONNECT_DATA=
     (SERVICE_NAME=sales.us.example.com)
     (INSTANCE_NAME=sales1)))
```

> **See Also:**
>
> *Oracle Database Net Services Administrator's Guide* for additional information about the use of INSTANCE_NAME

6.9.6 RDB_DATABASE

Purpose

To specify the file name of an Oracle Rdb database.

Usage Notes

Put this parameter under the CONNECT_DATA parameter.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=...)
     (ADDRESS=...))
   (CONNECT_DATA=
     (SERVICE_NAME=sales.us.example.com)
     (RDB_DATABASE= [.mf]mf_personal.rdb)))
```

6.9.7 SERVER

Purpose

To direct the listener to connect the client to a specific type of service handler.

Usage Notes

Put this parameter under the CONNECT_DATA parameter.

Values

- `dedicated` to specify whether client requests be served by dedicated server.
- `shared` to specify whether client requests be served by a dispatcher or shared server.
- `pooled` to get a connection from the connection pool if database resident connection pooling is enabled on the 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.
> - The USE_DEDICATED_SERVER parameter in the `sqlnet.ora` file overrides this parameter.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=...)
     (ADDRESS=...))
   (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com)
    (SERVER=dedicated)))
```

6.9.8 SERVICE_NAME

Purpose

To identify the Oracle Database database service to access.

Usage Notes

Set the value to a value specified by the `SERVICE_NAMES` parameter in the initialization parameter file.

Put this parameter under the `CONNECT_DATA` parameter.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=...)
     (ADDRESS=...))
   (CONNECT_DATA=
     (SERVICE_NAME=sales.us.example.com)))
```

See Also:

Oracle Database Net Services Administrator's Guide for additional information about the use of the SERVICE_NAME parameter

6.10 Security Section

The security section of the tnsnames.ora file specifies the following security-related parameters for use with Oracle security features:

- SECURITY
- SSL_SERVER_CERT_DN

6.10.1 SECURITY

Purpose

To change the security properties of the connection. Put this parameter under the DESCRIPTION parameter.

Usage Notes

The parameters permitted under SECURITY are SSL_SERVER_CERT_DN and AUTHENTICATION_SERVICE.

Example

```
net_service_name=
 (DESCRIPTION=
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com))
    (SECURITY=
      (SSL_SERVER_CERT_DN="cn=sales,cn=OracleContext,dc=us,dc=acme,dc=com")))
```

6.10.2 SSL_SERVER_CERT_DN

Purpose

To specify the distinguished name (DN) of the database server.

Usage Notes

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.

Use this parameter with the sqlnet.ora parameter SSL_SERVER_DN_MATCH to enable server DN matching.

Example

```
net_service_name=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=...)
     (ADDRESS=...))
   (CONNECT_DATA=
     (SERVICE_NAME=finance.us.example.com))
   (SECURITY=
     (SSL_SERVER_CERT_DN="cn=finance,cn=OracleContext,dc=us,dc=acme,dc=com")))
```

> **See Also:**
>
> *Oracle Database Security Guide*

6.11 Timeout Parameters

/? 25800-1 IPv6?/

The timeout section of the `tnsnames.ora` file provides the ability to specify timeout and retry configuration through the TNS connect string. The following parameters can be set at the `DESCRIPTION` level of a connect string:

- CONNECT_TIMEOUT
- RETRY_COUNT
- RETRY_DELAY
- TRANSPORT_CONNECT_TIMEOUT

6.11.1 CONNECT_TIMEOUT

Purpose

To specify the timeout duration in `ms`, `sec`, or `min` for a client to establish an Oracle Net connection to an Oracle database.

Usage Notes

Put this parameter under the `DESCRIPTION` parameter.

The timeout interval specified by `CONNECT_TIMEOUT` is a superset of the TCP connect timeout interval. It includes the time to be connected to the database instance providing the requested service, not just the duration of the TCP connection. It accepts different timeouts with or without space between the value and the unit. In case, no unit is mentioned, the default unit is `sec`.

The timeout interval is applicable for each `ADDRESS` in an `ADDRESS_LIST`, and each IP address to which a host name is mapped.

The `CONNECT_TIMEOUT` parameter is equivalent to the `sqlnet.ora` parameter SQLNET.OUTBOUND_CONNECT_TIMEOUT, and overrides it.

Example

```
net_service_name=
 (DESCRIPTION=
  (CONNECT_TIMEOUT=10 ms)(RETRY_COUNT=3)
  (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521)))
  (CONNECT_DATA=
   (SERVICE_NAME=sales.us.example.com)))
```

6.11.2 RETRY_COUNT

Purpose

To specify the number of times an ADDRESS list is traversed before the connection attempt is terminated.

Usage Notes

Put this parameter under the DESCRIPTION parameter.

When a DESCRIPTION_LIST is specified, each DESCRIPTION is traversed multiple times based on the specified number of retries.

Example

```
net_service_name=
(DESCRIPTION_LIST=
 (DESCRIPTION=
  (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
  (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales1a-svr)(PORT=1521))
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales1b-svr)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales1.example.com)))
 (DESCRIPTION=
  (CONNECT_TIMEOUT=60)(RETRY_COUNT=1)
  (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales2a-svr)(PORT=1521))
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales2b-svr)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales2.us.example.com))))
```

6.11.3 RETRY_DELAY

Purpose

To specify the delay in seconds between subsequent retries for a connection. This parameter works in conjunction with RETRY_COUNT parameter.

Usage Notes

Put this parameter under the DESCRIPTION parameter.

When a DESCRIPTION_LIST is specified, each DESCRIPTION is traversed multiple times based on the specified number of retries, and the specific delay for the description.

Example

```
net_service_name=
(DESCRIPTION_LIST=
 (DESCRIPTION=
  (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)(RETRY_DELAY=2)
  (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales1a-svr)(PORT=1521))
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales1b-svr)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales1.example.com)))
 (DESCRIPTION=
  (CONNECT_TIMEOUT=60)(RETRY_COUNT=2)(RETRY_DELAY=1)
  (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales2a-svr)(PORT=1521))
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales2b-svr)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales2.us.example.com))))
```

6.11.4 TRANSPORT_CONNECT_TIMEOUT

Purpose

To specify the transport connect timeout duration in ms, sec, or min for a client to establish an Oracle Net connection to an Oracle database.

Usage Notes

This parameter is put under the DESCRIPTION parameter.

The TRANSPORT_CONNECT_TIMEOUT parameter specifies the time, in ms, sec, or min, for a client to establish a TCP connection to the database server. It accepts different timeouts with or without space between the value and the unit. The default value is 60 seconds. In case, no unit is mentioned, the default unit is sec.

The timeout interval is applicable for each ADDRESS in an ADDRESS_LIST description, and each IP address that a host name is mapped. The TRANSPORT_CONNECT_TIMEOUT parameter is equivalent to the sqlnet.ora parameter TCP.CONNECT_TIMEOUT, and overrides it.

Example

```
net_service_name =
  (DESCRIPTION=
    (TRANSPORT_CONNECT_TIMEOUT=10 ms)
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521)))
    (CONNECT_DATA=
      (SERVICE_NAME=sales.us.example.com)))
```

6.12 Compression Parameters

The compression section of the tnsnames.ora file provides the ability to enable compression and specify compression levels. The following parameters can be set at the DESCRIPTION level of a connect string:

- COMPRESSION

- COMPRESSION_LEVELS

6.12.1 COMPRESSION

Purpose

To enable or disable data compression.

Usage Notes

Put this parameter under the DESCRIPTION parameter.

Setting this parameter in the connect descriptor for a client overrides the SQLNET.COMPRESSION parameter in the client-side sqlnet.ora file.

Default

off

Values

- on to enable data compression.
- off to disable data compression.

Example

```
net_service_name=
 (DESCRIPTION=
   (COMPRESSION=on)
      (ADDRESS_LIST=
         (ADDRESS= (PROTOCOL=tcp) (HOST=sales1-server) (PORT=1521))
         (ADDRESS= (PROTOCOL=tcp) (HOST=sales2-server) (PORT=1521)))
   (CONNECT_DATA=
       (SERVICE_NAME=sales.us.example.com)))
```

6.12.2 COMPRESSION_LEVELS

Purpose

To specify the compression level.

Usage Notes

The compression levels are used at the time of negotiation to verify which levels are used at both ends, and select one level. Put this parameter under the DESCRIPTION parameter.

This parameter is used with the COMPRESSION parameter. Setting this parameter in the connect descriptor for a client overrides the SQLNET.COMPRESSION_LEVELS parameter in the client-side sqlnet.ora file.

Default

low

Values

- low for low CPU usage and a low compression ratio.

- high for high CPU usage and a high compression ratio.

Example

```
net_service_name=
 (DESCRIPTION=
  (COMPRESSION=on)
  (COMPRESSION_LEVELS=(LEVEL=low)(LEVEL=high))
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521))
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com)))
```

7

Oracle Net Listener Parameters in the listener.ora File

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

This chapter contains the following topics:

7.1 Overview of Oracle Net Listener Configuration File

Oracle Net 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
- Valid nodes that the listener allows to register with the database
- Database services
- Control parameters

Dynamic service registration, eliminates the need for static configuration of supported services. However, static service configuration is required if you plan to use Oracle Enterprise Manager Cloud Control.

By default, the `listener.ora` file is located in the `ORACLE_HOME/network/admin` directory. The `listener.ora` file can also be stored the following locations:

- The directory specified by the `TNS_ADMIN` environment variable or registry value.
- On Linux and UNIX operating systems, it is the global configuration directory. For example, on the Oracle Solaris operating system, the directory is `/var/opt/oracle`.

> **See Also:**
>
> – *Oracle Database Global Data Services Concepts and Administration Guide* for information about management of global services
>
> – Oracle operating system-specific documentation

- In the read-only Oracle home mode, the `listener.ora` file default location is *`ORACLE_BASE_HOME`*`/network/admin`.
- In the read-only Oracle home mode, the parameters that default to `ORACLE_HOME` location change to default to `ORACLE_BASE_HOME` location.

It is possible to configure multiple listeners, each with a unique name, in one `listener.ora` file. Multiple listener configurations are 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 recommends running only one listener for each node in most customer environments.
> - Oracle Net Services supports the IFILE parameter in the `listener.ora` file, with up to three levels of nesting. The parameter is added manually to the file. The following is an example of the syntax:
>
>   ```
>   IFILE=/tmp/listener_em.ora
>   IFILE=/tmp/listener_cust1.ora
>   IFILE=/tmp/listener_cust2.ora
>   ```
>
>   Refer to *Oracle Database Reference* for additional information.

Example 7-1 shows a `listener.ora` file for a listener named `LISTENER`, which is the default name of the listener.

Example 7-1 listener.ora File

```
LISTENER=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sale-server)(PORT=1521))
      (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))))
```

7.2 Protocol Address Parameters

The **protocol address** section of the `listener.ora` file defines the protocol addresses on which the listener is accepting connection requests. This section describes the most common parameters used in protocol addresses. The `ADDRESS_LIST` parameter is also supported.

See Also:

Protocol Address Configuration for additional information about the ADDRESS_LIST parameter

This section lists and describes the following parameters:

- ADDRESS
- DESCRIPTION
- Firewall
- IP
- QUEUESIZE
- RECV_BUF_SIZE
- SEND_BUF_SIZE

7.2.1 ADDRESS

Purpose

To specify a single listener protocol address.

Usage Notes

Put this parameter under the DESCRIPTION parameter.

Example

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

See Also:

Protocol Address Configuration for descriptions of the correct parameters to use for each type of support protocol

7.2.2 DESCRIPTION

Purpose

To contain listener protocol addresses.

Example

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

7.2.3 Firewall

Purpose

It can be set in endpoint to enable firewall functionality.

7.2.4 IP

Purpose

To determine which IP address the listener listens on when a host name is specified.

Usage Notes

This parameter is only applicable when the `HOST` parameter specifies a host name.

Values

- `first`

  Listen on the first IP address returned by the DNS resolution of the host name. If the user wants the listener to listen on the first IP to which the specified host name resolves, then the address must be qualified with `(IP=first)`.

- `v4_only`

  Listen only on IPv4 addresses.

- `v6_only`

  Listen only on IPv6 addresses.

Default

This feature is disabled by default.

Example

```
listener_name=
 (DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=rancode1-vip)(PORT=1522)(IP=v6_only))
```

7.2.5 QUEUESIZE

Purpose

To specify the number of concurrent connection requests that the listener can accept on a TCP/IP or IPC listening endpoint (protocol address).

Usage Notes

The number of concurrent connection requests is dependent on the platform and listener usage scenarios. If the listener is heavily-loaded, then set the parameter to a higher number.

Put 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.

Example

```
listener_name=
 (DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=hr-server)(PORT=1521)(QUEUESIZE=20)))
```

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

7.2.6 RECV_BUF_SIZE

Purpose

To specify, in bytes, the buffer space for receive operations of sessions.

Usage Notes

Put this parameter under the `DESCRIPTION` parameter or at the end of the protocol address with its value set to the expected number of bytes.

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 the operating system-specific documentation for information about additional protocols that support this parameter.

Default

The default value for this parameter is operating system specific. The default for the Linux operating system is 87380 bytes.

Example

```
listener_name=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)
        (RECV_BUF_SIZE=11784))
      (ADDRESS=(PROTOCOL=ipc)(KEY=extproc)
        (RECV_BUF_SIZE=11784))))
listener_name=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (RECV_BUF_SIZE=11784))
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)
      (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))))
```

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

7.2.7 SEND_BUF_SIZE

Purpose

To specify, in bytes, the buffer space for send operations of sessions.

Usage Notes

Put this parameter under the DESCRIPTION parameter or at the end of the protocol address.

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 additional information about additional protocols that support this parameter.

Default

The default value for this parameter is operating system specific. The default for the Linux operating system is 16 KB.

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=
    (SEND_BUF_SIZE=11280)
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)
      (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))))
```

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

7.3 Connection Rate Limiter Parameters

The connection rate limiter feature in Oracle Net Listener enables a database administrator to limit the number of new connections handled by the listener. When this feature is enabled, Oracle Net Listener imposes a user-specified maximum limit on the number of new connections handled by the listener every second.

Depending on the configuration, the rate can be applied to a collection of endpoints, or to a specific endpoint.

This feature is controlled through the following `listener.ora` configuration parameters:

- CONNECTION_RATE_listener name
- RATE_LIMIT

7.3.1 CONNECTION_RATE_*listener_name*

Purpose

To specify a global rate that is enforced across all listening endpoints that are rate-limited.

Usage Notes

When this parameter is specified, it overrides any endpoint-level numeric rate values that might be specified.

Syntax

```
CONNECTION_RATE_listener_name=number_of_connections_per_second
```

7.3.2 RATE_LIMIT

Purpose

To indicate that a particular listening endpoint is rate limited.

Usage Notes

The parameter is specified in the `ADDRESS` section of the listener endpoint configuration.

Syntax

```
LISTENER=
   (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521)(RATE_LIMIT=yes))
```

- bug 17734748

  When the `RATE_LIMIT` parameter is set to `yes` for an endpoint, that endpoint is included in the enforcement of the global rate configured by the `CONNECTION_RATE_`*`listener_name`* parameter. The global rate limit is enforced individually at each endpoint that has `RATE_LIMIT` set to `yes`.

- Dynamic endpoints for listeners managed by Oracle Clusterware have the `RATE_LIMIT` parameter set to `yes`.
- When the `RATE_LIMIT` parameter is set to a value greater than `0`, then the rate limit is enforced at that endpoint level.

Examples

The following examples use the `CONNECTION_RATE_`*`listener name`* and `RATE_LIMIT` parameters.

Example 1

```
CONNECTION_RATE_LISTENER=10

LISTENER=
  (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521)(RATE_LIMIT=yes))
   (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1522)(RATE_LIMIT=yes))
   (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1523)))
```

In the preceding example, the global rate of new connections is enforced separately for each endpoint. Connections through port 1521 are limited at 10 every second, and the connections through port 1522 are also separately limited at 10 every second. Connections through port 1523 are not limited.

Example 2

```
LISTENER= (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521)(RATE_LIMIT=5))
   (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1522)(RATE_LIMIT=10))
   (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1523))
   )
```

In the preceding example, the connection rates are enforced at the endpoint level. A maximum of 5 connections are processed through port 1521 every second. The limit for connections through port 1522 is 10 every second. Connections through port 1523 are not limited.

> **Note:**
>
> The global `CONNECTON_RATE_`*`listener_name`* parameter is not specified in the preceding configuration. If it is specified, then the limits on ports 1521 and 1522 are ignored, and the global value is used instead.

7.4 Control Parameters

This section describes the following parameters that control the behavior of the listener:

- ADMIN_RESTRICTIONS_listener_name
- ALLOW_MULTIPLE_REDIRECTS_listener_name

- ENABLE_EXADIRECT_*listener_name*
- CRS_NOTIFICATION_listener_name
- DEDICATED_THROUGH_BROKER_listener_name
- DEFAULT_SERVICE_listener_name
- INBOUND_CONNECT_TIMEOUT_listener_name
- LOCAL_REGISTRATION_ADDRESS_listener_name
- MAX_ALL_CONNECTIONS_listener_name
- MAX_REG_CONNECTIONS_listener_name
- REGISTRATION_EXCLUDED_NODES_listener_name
- REGISTRATION_INVITED_NODES_listener_name
- REMOTE_REGISTRATION_ADDRESS_listener_name
- SAVE_CONFIG_ON_STOP_listener_name
- SSL_CLIENT_AUTHENTICATION
- SSL_VERSION
- SUBSCRIBE_FOR_NODE_DOWN_EVENT_listener_name
- USE_SID_AS_SERVICE_listener_name
- VALID_NODE_CHECKING_REGISTRATION_listener_name
- WALLET_LOCATION

7.4.1 ADMIN_RESTRICTIONS_*listener_name*

Purpose

To restrict runtime administration of the listener.

Usage Notes

Setting `ADMIN_RESTRICTIONS_`*`listener_name`*`=on` disables the runtime modification of parameters in `listener.ora`. That is, the listener refuses 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 using the RELOAD command for the new changes to take effect without explicitly stopping and restarting the listener.

Default

off

Example

```
ADMIN_RESTRICTIONS_listener=on
```

7.4.2 ALLOW_MULTIPLE_REDIRECTS_*listener_name*

Purpose

To support multiple redirects of the client.

Usage Notes

This parameter should only be set on the SCAN listener on the Oracle Public Cloud. When set to `on`, multiple redirects of the client are allowed.

Do not set this parameter for a node listener if that is used as a SCAN listener.

Default

off

Values

`on | off`

Example

```
ALLOW_MULTIPLE_REDIRECTS_listener=on
```

7.4.3 ENABLE_EXADIRECT_*listener_name*

Purpose

To enable Exadirect protocol.

Usage Notes

The parameter enables Exadirect support.

Default

`Off`

Values

`on | off`

Example 7-2 Example

```
ENABLE_EXADIRECT_listener=on
```

7.4.4 CRS_NOTIFICATION_*listener_name*

Purpose

To set notification.

Usage Notes

By default, the Oracle Net listener notifies Cluster Ready Services (CRS) when it is started or stopped. These notifications allow CRS to manage the listener in an Oracle Real Application Clusters environment. This behavior can be prevented by setting the `CRS_NOTIFICATION_`*`listener_name`* parameter to `off`.

Default

on

Values

`on | off`

7.4.5 DEDICATED_THROUGH_BROKER_*listener_name*

Bug 14644490

Purpose

To enable the server to spawn a thread or process when a connection to the database is requested through the listener.

Default

off

Values

`on | off`

Example

```
DEDICATED_THROUGH_BROKER_listener=on
```

7.4.6 DEFAULT_SERVICE_*listener_name*

Purpose

To enable users to connect to the database without having to specify a service name from the client side.

Usage Notes

When a client tries to connect to the database, the connection request passes through the listener. The listener may be servicing several different databases. If a service name is configured in this parameter, then users may not necessarily need to specify a service name in the connect syntax. If a user specifies a service name, then the listener connects the user to that specific database, otherwise the listener connects to the service name specified by the `DEFAULT_SERVICE_`*`listener_name`* parameter. For container databases, the client must explicitly specify the service name.

> **See Also:**
>
> *Oracle Database Net Services Administrator's Guide* for additional information about the Easy Connect naming method

Default

There is no default value for the `DEFAULT_SERVICE_`*`listener_name`* parameter. If this parameter is not configured and a user does not specify a fully-qualified service name in the connect syntax, then the connection attempt fails. This parameter only accepts one value.

Example

```
DEFAULT_SERVICE_listener=sales.us.example.com
```

7.4.7 INBOUND_CONNECT_TIMEOUT_*listener_name*

Purpose

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

Usage Notes

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 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 the `INBOUND_CONNECT_TIMEOUT_`*`listener_name`* parameter to 2 seconds and the `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.

Default

60 seconds

Example

```
INBOUND_CONNECT_TIMEOUT_listener=2
```

7.4.8 LOCAL_REGISTRATION_ADDRESS_listener_name

Purpose

To secure registration requests through dedicated secure registration endpoints for local listeners. Service ACLs are accepted by listener only if `LOCAL_REGISTRATION_ADDRESS_`*lsnr alias* is configured. The parameter specifies the group that is allowed to send ACLs.

Usage Notes

The local registration endpoint accepts local registration connections from the specified group. All local registration requests coming on normal listening endpoints are redirected to the local registration endpoint. If the registrar is not a part of the group, then it cannot connect to the endpoint.

Default

`OFF`

Values

`ON, OFF, or IPC endpoint address with group`

When set to ON, listener defaults the group to `oinstall` on UNIX and `ORA_INSTALL` on Windows.

Example 7-3 Example

```
LOCAL_REGISTRATION_ADDRESS_lsnr_alias = (address=(protocol=ipc)(group=xyz))
LOCAL_REGISTRATION_ADDRESS_lsnr_alias =ON
```

7.4.9 MAX_ALL_CONNECTIONS_*listener_name*

Purpose

To specify the maximum number of concurrent registration and client connection sessions that can be supported by Oracle Net Listener.

Usage Notes

This number includes registration connections from databases, and ongoing client connection establishment requests. After a connection is established, the clients do not maintain a connection to the listener. This limit only applies to client connections that are in the initial connection establishment phase from a listener perspective.

Default

Operating system-specific

Example

```
MAX_ALL_CONNECTIONS_listener=4096
```

7.4.10 MAX_REG_CONNECTIONS_*listener_name*

Purpose

To specify the maximum number of concurrent registration connection sessions that can be supported by Oracle Net Listener.

Default

512

Example

```
MAX_REG_CONNECTIONS_listener=2048
```

7.4.11 REGISTRATION_EXCLUDED_NODES_*listener_name*

Purpose

To specify the list of nodes that cannot register with the listener.

Usage Notes

The list can include host names or CIDR notation for IPv4 and IPv6 addresses. The wildcard format (*) is supported for IPv4 addresses. The presence of a host name in the list results in the inclusion of all IP addresses mapped to the host name. The host name should be consistent with the public network interface.

If the `REGISTRATION_INVITED_NODES_`*`listener_name`* parameter and the `REGISTRATION_EXCLUDED_NODES_`*`listener_name`* parameter are set, then the `REGISTRATION_EXCLUDED_NODES_`*`listener_name`* parameter is ignored.

Values

Valid nodes and subnet IP addresses or names.

Example

```
REGISTRATION_EXCLUDED_NODES_listener = (10.1.26.*, 10.16.40.0/24, \
                                        2001:DB8:3eff:fe38, node2)
```

7.4.12 REGISTRATION_INVITED_NODES_*listener_name*

Purpose

To specify the list of node that can register with the listener.

Usage Notes

- The list can include host names or CIDR notation for IPv4 and IPv6 addresses. The wildcard format (*) is supported for IPv4 addresses. The presence of a host name in the list results in the inclusion of all IP addresses mapped to the host name. The host name should be consistent with the public network interface.
- If the `REGISTRATION_INVITED_NODES_`*`listener_name`* parameter and the `REGISTRATION_EXCLUDED_NODES_`*`listener_name`* parameter are set, then the `REGISTRATION_EXCLUDED_NODES_`*`listener_name`* parameter is ignored.
- Starting with Oracle Grid Infrastructure 12c, for a SCAN listener, if the `VALID_NODE_CHECKING_REGISTRATION_`*`listener_name`* and `REGISTRATION_INVITED_NODES_`*`listener_name`* parameters are set in the `listener.ora` file, then the listener agent overwrites these parameters.

Values

Valid nodes and subnet IP addresses or names.

Example

```
REGISTRATION_INVITED_NODES_listener = (10.1.35.*, 10.1.34.0/24, \
                                       2001:DB8:fe38:7303, node1)
```

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for information about valid node checking for registration

7.4.13 REMOTE_REGISTRATION_ADDRESS_*listener_name*

Purpose

To secure registration requests through dedicated secure registration endpoints for SCAN listeners.

Usage Notes

The registration endpoint is on a private network within the cluster. All remote registration requests coming in on normal listening endpoints are redirected to the registration endpoint. Any system which is not a part of the cluster cannot connect to the endpoint. This feature is not supported when `ADMIN_RESTRICTIONS_`*`listener_name`* is set to `ON` as the Cluster Ready Services agent configures the `remote_registration_address` dynamically at run time.

Default

This parameter is configured internally in listeners managed by Oracle Clusterware to restrict registrations to the private network. The value of this parameter should not be modified or specified explicitly. The only supported explicit setting is for turning this feature off by setting the value to `OFF`.

Values

`off`

Example

```
REMOTE_REGISTRATION_ADDRESS_listener=off
```

7.4.14 SAVE_CONFIG_ON_STOP_*listener_name*

Purpose

To specify whether runtime configuration changes are saved to the `listener.ora` file.

Usage Notes

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
```

7.4.15 SSL_CLIENT_AUTHENTICATION

Purpose

To specify whether a client is authenticated using the **Secure Sockets Layer (SSL)**.

Usage Notes

The database server authenticates the client. Therefore, this value should be set to `false.` If this parameter is set to `true`, then the listener attempts to authenticate the client, which can result in a failure.

Default

true

Values

`true | false`

Example

```
SSL_CLIENT_AUTHENTICATION=false
```

> **See Also:**
>
> *Oracle Database Security Guide*

7.4.16 SSL_VERSION

Purpose

To limit allowable SSL or TLS versions used for connections.

Usage Notes

Clients and database servers must use a compatible version. This parameter should only be used when absolutely necessary for backward compatibility. The current default uses TLS version 1.2 which is the version required for multiple security compliance requirements.

If you set `SSL_VERSION` to `undetermined`, then by default it uses `3.0`.

Default

`1.2`

Values

> **Note:**
>
> The `sqlnet.ora parameter ADD_SSLV3_TO_DEFAULT` has no impact on this parameter.

```
undetermined | 3.0 | 1.0| 1.1 | 1.2
```

If you want to specify one version or another version, then use “or”. The following values are permitted:

```
1.0 or 3.0 | 1.2 or 3.0 | 1.1 or 1.0 | 1.2 or 1.0 | 1.2 or 1.1 | 1.1 or 1.0 or 3.0 |
1.2 or 1.0 or 3.0 | 1.2 or 1.1 or 1.0 | 1.2 or 1.1 or 3.0 |1.2 or 1.1 or 1.0 or 3.0
```

Example

```
SSL_VERSION=1.2
```

The remaining version numbers correspond to the TLS versions, such as, TLSv1.0, TLSv1.1, and TLSv1.2.

> **See Also:**
>
> *Oracle Database Security Guide*

7.4.17 SUBSCRIBE_FOR_NODE_DOWN_EVENT_*listener_name*

Purpose

To subscribe to Oracle Notification Service (ONS) notifications for downed events.

Usage Notes

By default, the listener subscribes to the ONS node down event on startup, if ONS is available. This subscription enables the listener to remove the affected service when it receives node down event notification from ONS. The listener uses asynchronous subscription for the event notification. Alter this behavior by setting `SUBSCRIBE_FOR_NODE_DOWN_EVENT_`*`listener_name`*`=off` in `listener.ora`.

Default

on

Values

```
on | off
```

7.4.18 USE_SID_AS_SERVICE_*listener_name*

Purpose

To enable the system identifier (SID) in the connect descriptor to be interpreted as a service name when a user attempts a database connection.

Usage Notes

Database clients with earlier releases of Oracle Database that have hard-coded connect descriptors can use this parameter to connect to a container or pluggable database.

For a container database, the client must specify a service name in order to connect to it. Setting this parameter to `on` instructs the listener to use the SID in the connect descriptor as a service name and connect the client to the specified database.

Default

off

Example

```
USE_SID_AS_SERVICE_listener=on
```

7.4.19 VALID_NODE_CHECKING_REGISTRATION_*listener_name*

Purpose

To determine whether valid node checking registration is performed, or the subnet is allowed.

Usage Notes

- When set to `on`, valid node checking registration is performed at the listener for any incoming registration request, and only local IP addresses are allowed.
- Starting with Oracle Grid Infrastructure 12c, for a SCAN listener, if the `VALID_NODE_CHECKING_REGISTRATION_`*`listener_name`* and `REGISTRATION_INVITED_NODES_`*`listener_name`* parameters are set in the `listener.ora` file, then the listener agent overwrites these parameters.

Default

on

Values

- `off | 0` to specify valid node checking registration is off, and no checking is performed.
- `on | 1 | local` to specify valid node checking registration is on, and all local IP addresses can register. If a list of invited nodes is set, then all IP addresses, host names, or subnets in the list as well as local IP addresses are allowed.
- `subnet | 2` to specify valid node checking registration is on, and all machines in the local subnets are allowed to register. If a list of invited nodes is set, then all

nodes in the local subnets as well as all IP addresses, host names and subnets in the list are allowed.

Example

```
VALID_NODE_CHECKING_REGISTRATION_listener=on
```

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for information about valid node checking for registration

7.4.20 WALLET_LOCATION

Purpose

To specify the location of wallets.

Usage Notes

Wallets are certificates, keys, and trustpoints processed by SSL that allow for secure connections.

The key/value pair for Microsoft 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 Microsoft Windows registry and the wallet's `key` (`KEY`) is `SALESAPP`, then 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`.

Syntax

Table 7-1 shows the syntax for the WALLET_LOCATION parameter based on wallet storage location.

Table 7-1 Syntax for WALLET_LOCATION

Wallet Location	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))`

Table 7-1 (Cont.) Syntax for WALLET_LOCATION

Wallet Location	Syntax

Oracle wallets in the Microsoft 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)))`

Additional Parameters

The following additional parameters are available for `WALLET_LOCATION`:

- `SOURCE`: Type of storage for wallets and storage location.
- `METHOD`: Type of storage.
- `METHOD_DATA`: Storage location.
- `DIRECTORY`: Location of Oracle wallets on file system.
- `KEY`: Wallet type and location in the Microsoft Windows registry.
- `PROFILE`: Entrust profile file (`.epf`).
- `INIFILE`: Entrust initialization file (`.ini`).

Default

None

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 Microsoft 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)))
```

> **See Also:**
>
> *Oracle Database Enterprise User Security Administrator's Guide*

7.5 ADR Diagnostic Parameters for Oracle Net Listener

Since Oracle Database 11*g*, Oracle Database includes an advanced fault diagnosability infrastructure for preventing, detecting, diagnosing, and resolving problems. The problems are critical errors such as those caused by database code bugs, metadata corruption, and customer data corruption.

When a critical error occurs, it is assigned an incident number, and diagnostic data for the error, such as traces and dumps, are immediately captured and tagged with the incident number. The data is then stored in the Automatic Diagnostic Repository (ADR), a file-based repository outside the database.

This section includes the parameters used when ADR is enabled. Non-ADR parameters listed in the listener.ora file are ignored when ADR is enabled. "Non-ADR Diagnostic Parameters for Oracle Net Listener" includes those used when ADR is disabled. ADR is enabled by default.

The following listener.ora parameters are used when ADR is enabled (when DIAG_ADR_ENABLED is set to on):

- ADR_BASE_listener_name
- DIAG_ADR_ENABLED_listener_name
- LOGGING_listener_name
- TRACE_LEVEL_listener_name
- TRACE_TIMESTAMP_listener_name

7.5.1 ADR_BASE_*listener_name*

Purpose

To specify the base directory that stores tracing and logging incidents when ADR is enabled.

Default

The default is `ORACLE_BASE`, or `ORACLE_HOME/log` if `ORACLE_BASE` is not defined.

Values

Any valid directory path to a directory with write permission.

Example

```
ADR_BASE_listener=/oracle/network/trace
```

7.5.2 DIAG_ADR_ENABLED_*listener_name*

Purpose

To indicate whether ADR tracing is enabled.

Usage Notes

When the `DIAG_ADR_ENABLED_`*`listener_name`* parameter is set to `on`, then ADR file tracing is used. When the `DIAG_ADR_ENABLED_`*`listener_name`* parameter is set to `off`, then non-ADR file tracing is used.

Default

on

Values

`on` | `off`

Example

```
DIAG_ADR_ENABLED_listener=on
```

7.5.3 LOGGING_*listener_name*

Purpose

To turn logging on or off.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

on

Values

`on` | `off`

Example

```
LOGGING_listener=on
```

7.5.4 TRACE_LEVEL_*listener_name*

Purpose

To turn listener tracing on, at a specific level, or off.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

off | 0

Values

- `off` or `0` for no trace output
- `user` or `4` for user trace information
- `admin` or `10` for administration trace information
- `support` or `16` for Oracle Support Services trace information

Example

```
TRACE_LEVEL_listener=admin
```

7.5.5 TRACE_TIMESTAMP_*listener_name*

Purpose

To add a time stamp in the form of `dd-mmm-yyyy hh:mi:ss:mil` to every trace event in the trace file for the listener.

Usage Notes

This parameter is used with the TRACE_LEVEL_listener_name parameter. This parameter is also applicable when non-ADR tracing is used.

Default

on

Values

- `on` | `true`
- `off` | `false`

Example

```
TRACE_TIMESTAMP_listener=true
```

7.6 Non-ADR Diagnostic Parameters for Oracle Net Listener

This section lists the parameters used when ADR is disabled. "ADR Diagnostic Parameters for Oracle Net Listener" ADR Diagnostic Parameters for Oracle Net Listener includes the parameters when ADR is enabled.

> **Note:**
>
> The default value of DIAG_ADR_ENABLED_listener_name is `on`. Therefore, the `DIAG_ADR_ENABLED_`*`listener_name`* parameter *must* explicitly be set to `off` to use non-ADR tracing.

- LOG_DIRECTORY_listener_name
- LOG_FILE_listener_name
- TRACE_DIRECTORY_listener_name
- TRACE_FILE_listener_name
- TRACE_FILELEN_listener_name
- TRACE_FILENO_listener_name

7.6.1 LOG_DIRECTORY_*listener_name*

Purpose

To specify the destination directory of the listener log file.

Usage Notes

Use this parameter when ADR is not enabled.

Default

`ORACLE_HOME/network/log`

Example

```
LOG_DIRECTORY_listener=/oracle/network/admin/log
```

7.6.2 LOG_FILE_*listener_name*

Purpose

To specify the name of the log file for the listener.

Usage Notes

Use this parameter when ADR is not enabled.

Default

```
listener.log
```

Example

```
LOG_FILE_listener=list.log
```

7.6.3 TRACE_DIRECTORY_*listener_name*

Purpose

To specify the destination directory of the listener trace file.

Usage Notes

Use this parameter when ADR is not enabled.

Default

```
ORACLE_HOME/network/trace
```

Example

```
TRACE_DIRECTORY_listener=/oracle/network/admin/trace
```

7.6.4 TRACE_FILE_*listener_name*

Purpose

To specify the name of the trace file for the listener.

Usage Notes

Use this parameter when ADR is not enabled.

Default

```
listener.trc
```

Example

```
TRACE_FILE_listener=list.trc
```

7.6.5 TRACE_FILEAGE_*listener_name*

Purpose

To specify the maximum age of listener trace files in minutes.

Usage Notes

When the age limit is reached, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO_listener_name parameter. Use this parameter when ADR is not enabled.

Default

Unlimited

This is the same as setting the parameter to `0`.

Example 7-4 Example

```
TRACE_FILEAGE_listener=60
```

7.6.6 TRACE_FILELEN_*listener_name*

Purpose

To specify the size of the listener trace files in kilobytes (KB).

Usage Notes

When the size is met, the trace information is written to the next file. The number of files is specified using the TRACE_FILENO_listener_name parameter. Use this parameter when ADR is not enabled.

Default

Unlimited

Example

```
TRACE_FILELEN_listener=100
```

7.6.7 TRACE_FILENO_*listener_name*

Purpose

To specify the number of trace files for listener tracing.

Usage Notes

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, then 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. Use this parameter when ADR is not enabled.

Default

1

Example

```
TRACE_FILENO_listener=3
```

7.7 Class of Secure Transports Parameters

The class of secure transports (COST) parameters specify a list of transports that are considered secure for administration and registration of a particular listener. The COST parameters identify which transports are considered secure for that installation and whether the administration of a listener requires secure transports. Configuring these parameters is optional.

The following are the COST parameters:

- DYNAMIC_REGISTRATION_listener_name
- SECURE_CONTROL_listener_name
- SECURE_REGISTER_listener_name
- SECURE_PROTOCOL_listener_name

> **See Also:**
>
> *Oracle Database Net Services Administrator's Guide* for additional information about COST parameters and listener security

7.7.1 DYNAMIC_REGISTRATION_*listener_name*

Purpose

To enable or disable dynamic registration.

Usage Notes

Static registrations are not affected by this parameter.

Default

The default value is `on`. Unless this parameter is explicitly set to `off`, all registration connections are accepted.

Values

- `on`: The listener accepts dynamic registration.
- `off`: The listener refuses dynamic registration.

Example

```
DYNAMIC_REGISTRATION_listener_name=on
```

7.7.2 SECURE_CONTROL_*listener_name*

Purpose

To specify the transports on which control commands are to be serviced.

Usage Notes

If the SECURE_CONTROL_*listener_name* parameter is configured with a list of transport names, then the control commands are serviced only if the connection is one of the listed transports. Connections arriving by other transport protocols are refused. The following is an example:

```
SECURE_CONTROL_listener1 = (TCPS,IPC)
```

In the preceding example, administration requests are accepted only on TCPS and IPC transports.

If no values are entered for this parameter, then the listener accepts any connection on any endpoint.

Syntax

```
SECURE_CONTROL_listener_name =
[(]transport1[,transport2, ....,transportn)]
```

In the preceding syntax, transport1, transport2, and transport*n* are valid, installed transport protocol names.

Example

```
LISTENER1=
 (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
      (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))
      (ADDRESS=(PROTOCOL=tcps)(HOST=sales-server)(PORT=1522))))
  SECURE_CONTROL_LISTENER1=tcps
```

7.7.3 SECURE_REGISTER_*listener_name*

Purpose

To specify the transports on which registration requests are to be accepted.

Usage Notes

If the SECURE_REGISTER_*listener_name* parameter is configured with a list of transport names, then only the connections arriving on the specified transports are able to register the service with the listener. Connections arriving by other transport protocols are refused. The following is an example:

```
SECURE_REGISTER_listener1 = (TCPS,IPC)
```

In the preceding example, registration requests are accepted only on TCPS and IPC transports.

If no values are entered for this parameter, then the listener accepts registration requests from any transport.

Syntax

```
SECURE_REGISTER_listener_name =
[(]transport1[,transport2, ....,transportn)]
```

In the preceding example, `transport1`, `transport2`, and `transportn` are valid, installed transport protocol names.

If this parameter and SECURE_CONTROL_listener_name are configured, then they override the SECURE_PROTOCOL_listener_name parameter.

Example

```
LISTENER1=
 (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
      (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))
      (ADDRESS=(PROTOCOL=tcps)(HOST=sales-server)(PORT=1522))))
  SECURE_REGISTER_listener1=tcps
```

7.7.4 SECURE_PROTOCOL_*listener_name*

Purpose

To specify the transports on which administration and registration requests are accepted.

Usage Notes

If this parameter is configured with a list of transport names, then the control commands and service registration can happen only if the connection belongs to the list of transports.

If this parameter is not present and neither SECURE_CONTROL_listener_name or SECURE_REGISTER_listener_name are configured, then all supported transports accept control and registration requests.

If the SECURE_CONTROL_listener_name and SECURE_REGISTER_listener_name parameters are configured, then they override the `SECURE_PROTOCOL_listener_name` parameter.

Syntax

```
SECURE_PROTOCOL_listener_name =
[(]transport1[,transport2, ....,transportn)]
```

In the preceding syntax, `transport1`, `transport2`, and `transportn` are valid, installed transport protocol names.

Example

```
LISTENER1=
 (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
      (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))
      (ADDRESS=(PROTOCOL=tcps)(HOST=sales-server)(PORT=1522))))
  SECURE_PROTOCOL_listener1=tcps
```

7.7.5 Using COST Parameters in Combination

COST parameters can also be used in combination to further control which transports accept service registration and control commands.

In Example 7-5, control commands are accepted only on the IPC channel and the TCPS transport, and service registrations are accepted only on an IPC channel.

Example 7-5 Combining COST Parameters

```
LISTENER1=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
     (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))
     (ADDRESS=(PROTOCOL=tcps)(HOST=sales-server)(PORT=1522))))
  SECURE_CONTROL_listener1=(tcps,ipc)
  SECURE_REGISTER_listener1=ipc
```

In Example 7-6, control commands are accepted only on the TCPS transport, and service registrations are accepted only on the IPC channel.

Example 7-6 Combining COST Parameters

```
LISTENER1=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
     (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))
     (ADDRESS=(PROTOCOL=tcps)(HOST=sales-server)(PORT=1522))))
  SECURE_CONTROL_listener1=tcps
  SECURE_PROTOCOL_listener1=ipc
```

8

Oracle Connection Manager Parameters (cman.ora)

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

This chapter contains the following topics:

- Overview of Oracle Connection Manager Configuration File
- Oracle Connection Manager Parameters
- ADR Diagnostic Parameters for Oracle Connection Manager
- Non-ADR Diagnostic Parameters for Oracle Connection Manager

8.1 Overview of Oracle Connection Manager Configuration File

Oracle Connection Manager configuration information is stored in the cman.ora file, and 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. The cman.ora file can also be stored in the following locations:

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

> **See Also:**
>
> – *Oracle Database Global Data Services Concepts and Administration Guide* for information about management of global services
>
> – Oracle operating system-specific documentation

Example 8-1 shows an sample of a cman.ora file.

Example 8-1 Sample cman.ora File

```
CMAN=
  (CONFIGURATION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=proxysvr)(PORT=1521))
    (RULE_LIST=
      (RULE=(SRC=192.0.2.32/27)(DST=sales-server)(SRV=*)(ACT=accept))
        (ACTION_LIST=(AUT=on)(MCT=120)(MIT=30)))
      (RULE=(SRC=foo)(DST=hr-server)(SRV=cmon)(ACT=accept)))
```

```
(PARAMETER_LIST=
  (MAX_GATEWAY_PROCESSES=8)
  (MIN_GATEWAY_PRCESSSES=3)
  (DIAG_ADR_ENABLED=ON)
  (ADR_BASE=/oracle/log)))
```

The `cman.ora` configuration file consists of the following sections:

- Listening address: Preceded by `ADDRESS=`, this section contains information pertinent to the listener. The `ADDRESS` parameter is required.
- Rule list: Preceded by `RULE_LIST=`, this section contains rule information. The RULE parameter is listed in the rule list section of the file. The `RULE` parameter is required.
- Parameter list: Preceded by `PARAMETER_LIST=`, this section contains all other parameters including those listed in "ADR Diagnostic Parameters for Oracle Connection Manager", and "Non-ADR Diagnostic Parameters for Oracle Connection Manager".

  The following parameters are allowed in the parameter list section of the `cman.ora` file. The default values are bold. To override the default setting for a parameter, enter the parameter and a nondefault value.

  `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. The default value is 60.

  `LOG_DIRECTORY=`*`log_directory`*. The default value is `ORACLE_HOME/network/log`.

  `LOG_LEVEL={off | user | admin |`**`support`**`}`

  `MAX_CMCTL_SESSIONS=` Any positive number. The default value is 4.

  `MAX_CONNECTIONS=` A value between 1 and 1024. The default value is 256.

  `MAX_GATEWAY_PROCESSES=` Any number greater than the minimum number of gateway processes up to 64. The default value is 16.

  `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. The default value is 2.

  `OUTBOUND_CONNECT_TIMEOUT=`**`0`** or greater

  `PASSWORD_`*`instance_name`*`=` Value is the encrypted instance password, if one has been set. The default value is no value.

  `SESSION_TIMEOUT=`**`0`** or greater

  `TRACE_DIRECTORY=`*`trace_directory`*. The default value is `ORACLE_HOME/network/trace`.

  `TRACE_FILELEN=` Any positive number. The default value is 0 (zero).

  `TRACE_FILENO=` Any positive number. The default value is 0 (zero).

  `TRACE_LEVEL={`**`off`**` | user | admin | support}`

TRACE_TIMESTAMP={**off** | on}

> **Note:**
>
> You cannot add the parameter PASSWORD_*instance_name* directly to the cman.ora file. The parameter is added using the SAVE_PASSWD command.

Example 8-2 shows the parameter list section of a cman.ora file.

Example 8-2 Parameter List Section of a cman.ora File

```
   (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)
      (SESSION_TIMEOUT=60)
      (TRACE_DIRECTORY=/home/user/network/admin/trace)
      (TRACE_FILELEN=100)
      (TRACE_FILENO=2)
      (TRACE_LEVEL=SUPPORT)
      (TRACE_TIMESTAMP=ON)
      (VALID_NODE_CHECKING_REGISTRATION=ON)
      (REGISTRATION_EXCLUDED_NODES = 10.1.26.*)
      (REGISTRATION_INVITED_NODES = 10.1.35.*)
)
```

8.2 Oracle Connection Manager Parameters

This section lists and describes the following cman.ora file parameters:

- ADDRESS
- ASO_AUTHENTICATION_FILTER
- COMPRESSION
- COMPRESSION_LEVELS
- COMPRESSION_THRESHOLD
- CONNECTION_STATISTICS
- EVENT_GROUP
- IDLE_TIMEOUT
- INBOUND_CONNECT_TIMEOUT
- LOG_DIRECTORY

- LOG_LEVEL
- MAX_ALL_CONNECTIONS
- MAX_CMCTL_SESSIONS
- MAX_CONNECTIONS
- MAX_GATEWAY_PROCESSES
- MIN_GATEWAY_PROCESSES
- MAX_REG_CONNECTIONS
- OUTBOUND_CONNECT_TIMEOUT
- PASSWORD_instance_name
- REGISTRATION_EXCLUDED_NODES
- REGISTRATION_INVITED_NODES
- RULE
- SDU
- SESSION_TIMEOUT
- TRACE_FILE
- TRACE_FILELEN
- TRACE_FILENO
- TRACE_LEVEL
- TRACE_TIMESTAMP
- VALID_NODE_CHECKING_REGISTRATION
- WALLET_LOCATION

8.2.1 ADDRESS

Purpose

To specify the protocol address of Oracle Connection Manager.

Syntax

```
(ADDRESS=(PROTOCOL=protocol)(HOST=host_name)(PORT=port_number)
```

Example

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

8.2.2 ASO_AUTHENTICATION_FILTER

Purpose

To specify whether Oracle Database security authentication settings must be used by the client.

Usage Notes

The global setting can be overridden by a rule-level setting in `ACTION_LIST`.

Values

- `on` to instruct Oracle Connection Manager to reject connection requests that are not using Secure Network Services (SNS). SNS is part of Oracle Database security.
- `off` to instruct Oracle Connection Manager not to check for SNS between the client and server. This is the default.

8.2.3 COMPRESSION

Purpose

To enable or disable data compression. If both the Oracle Connection Manager and the other end (server or client or Oracle Connection Manager) have this parameter set to `ON`, then compression is used for the connection.

Default

`off`

Values

- `on` to enable data compression.
- `off` to disable data compression.

Example

```
COMPRESSION=on
```

8.2.4 COMPRESSION_LEVELS

Purpose

To specify the compression level.

Usage Notes

The compression levels are used at the time of negotiation to verify which levels are used at both ends, and select one level.

Default

`low`

Values

- `low` for low CPU usage and a low compression ratio.
- `high` for high CPU usage and a high compression ratio.

Example

```
COMPRESSION_LEVELS=high,low
```

8.2.5 COMPRESSION_THRESHOLD

Purpose

To specify the minimum data size, in bytes, for which compression is required.

Usage Notes

Compression is not be done if the size of the data to be sent is less than this value.

Default

`1024 bytes`

Example

```
COMPRESSION_THRESHOLD=1024
```

8.2.6 CONNECTION_STATISTICS

Purpose

To specify whether the `SHOW_CONNECTIONS` command displays connection statistics.

Usage Notes

The global setting can be overridden by a rule-level setting in `ACTION_LIST`.

Values

- `yes` to display statistics.
- `no` to not display statistics. This is the default.

8.2.7 EVENT_GROUP

Purpose

To specify which event groups are logged.

Usage Notes

Multiple events may be designated using a comma-delimited list.

Values

- `alert` for alert notifications.
- `cmd_proc` for command processing.
- `conn_hdlg` for connection handling.
- `init_and_term` for initialization and termination.
- `memory_ops` for memory operations.
- `proc_mgmt` for process management.

- `reg_and_load` for registration and load update.
- `relay` for events associated with connection control blocks.
- `timer` for gateway timeouts.
- `wake_up` for events related to Connection Manager Administration (CMADMIN) wake-up queue.

> **Note:**
>
> The event group `ALERT` cannot be turned off.

8.2.8 IDLE_TIMEOUT

Purpose

To specify the amount of time that an established connection can remain active without transmitting data.

Usage Notes

The global setting can be overridden by a rule-level setting in `ACTION_LIST`.

Values

- `0` to disable the timeout. This is the default.
- Any number greater than 0 to enable the timeout. The number equals the timeout period in seconds.

8.2.9 INBOUND_CONNECT_TIMEOUT

Purpose

To specify how long in seconds the Oracle Connection Manager listener waits for a valid connection from a client or another instance of Oracle Connection Manager.

Values

- `60 sec` is the default. Use value `0` to disable timeout.
- Any number greater than 0 to enable the timeout. The number equals the timeout period in seconds.

8.2.10 LOG_DIRECTORY

Purpose

To specify the directory for the Oracle Connection Manager log files.

Default

`ORACLE_BASE_`*`HOME`*`/network/log`

8.2.11 LOG_LEVEL

Purpose

To specify the level for log messages.

Values

- `off` for no logging. This is the default.
- `user` for user-induced errors log information.
- `admin` for administration log information, such as installation-specific.
- `support` for Oracle Support Services information.

8.2.12 MAX_ALL_CONNECTIONS

Purpose

To specify the maximum number of concurrent registration and client connection sessions that can be supported by Oracle Connection Manager.

Usage Notes

This number includes registration connections from databases, and ongoing client connection establishment requests. After a connection is established, the clients do not maintain a connection to the listener. This limit only applies to client connections that are in the initial connection establishment phase from a listener perspective.

Default

Operating system-specific

Example

```
MAX_ALL_CONNECTIONS=40
```

8.2.13 MAX_CMCTL_SESSIONS

Purpose

To specify the maximum number of concurrent local or remote sessions of the Oracle Connection Manager control utility allowable for a given instance.

Usage Notes

One of the sessions must be a local session.

Values

Any number of sessions can be designated.

8.2.14 MAX_CONNECTIONS

Bug 2447824

Purpose

To specify the maximum number of connection slots that a gateway process can handle.

Values

Any number in the range of 1 to 1024.

8.2.15 MAX_GATEWAY_PROCESSES

Purpose

To specify the maximum number of gateway processes that an instance of Oracle Connection Manager supports.

Values

The number designated must be greater than the minimum number of gateway processes. The maximum is 64.

8.2.16 MAX_REG_CONNECTIONS

Purpose

To specify the maximum number of concurrent registration connection sessions that can be supported by Oracle Connection Manager.

Default

512

Example

```
MAX_REG_CONNECTIONS=20
```

8.2.17 MIN_GATEWAY_PROCESSES

Purpose

To specify the minimum number of gateway processes that an instance of Oracle Connection Manager supports.

Values

Any number of sessions can be designated up to 64.

8.2.18 OUTBOUND_CONNECT_TIMEOUT

Purpose

To specify the length of time in seconds 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.

Values

- `60` to disable the timeout. This is the default.
- Any number greater than 0 to enable the timeout. The number equals the timeout period in seconds.

8.2.19 PASSWORD_*instance_name*

Purpose

To specify the encrypted instance password, if one has been set.

8.2.20 REGISTRATION_EXCLUDED_NODES

Purpose

To specify the list of nodes that cannot register with the listener.

Usage Notes

The list can include host names or CIDR notation for IPv4 and IPv6 addresses. The wildcard format (`*`) is supported for IPv4 addresses. The presence of a host name in the list results in the inclusion of all IP addresses mapped to the host name. The host name should be consistent with the public network interface.

If the `REGISTRATION_INVITED_NODES` parameter and the `REGISTRATION_EXCLUDED_NODES` parameter are set, then the `REGISTRATION_EXCLUDED_NODES` parameter is ignored.

Values

Valid nodes and subnet IP addresses or names.

Example

```
REGISTRATION_EXCLUDED_NODES = 10.1.26.*, 10.16.40.0/24, \
                                   2001:DB8:3eff:fe38, node2
```

8.2.21 REGISTRATION_INVITED_NODES

Purpose

To specify the list of node that can register with the listener.

Usage Notes

The list can include host names or CIDR notation for IPv4 and IPv6 addresses. The wildcard format (`*`) is supported for IPv4 addresses. The presence of a host name in the list results in the inclusion of all IP addresses mapped to the host name. The host name should be consistent with the public network interface.

If the `REGISTRATION_INVITED_NODES` parameter and the `REGISTRATION_EXCLUDED_NODES` parameter are set, then the `REGISTRATION_EXCLUDED_NODES` parameter is ignored.

Values

Valid nodes and subnet IP addresses or names.

Example

```
REGISTRATION_INVITED_NODES = 10.1.35.*, 10.1.34.0/24, \
                                      2001:DB8:fe38:7303, node1
```

8.2.22 RULE

Purpose

To specify an access control rule list to filter incoming connections.

Usage Notes

A rule list specifies which connections are accepted, rejected, or dropped.

If no rules are specified, then all connections are rejected.

The source and destination can be a host name, IP address, or subnet mask.

There must be 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.

Oracle Connection Manager does not support wildcards for partial IP addresses. If you use a wildcard, then 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 Example "**Sample cman.ora File**", /27 represents a subnet mask that comprises 27 left-most bits.

Values

This parameter is listed in the rule list section of the `cman.ora` file preceded by `RULE_LIST=`.

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= ...))
```

Additional Parameters

The `RULE` parameter filters a connection or group of connections using the following parameters:

`SRC`: The source host name or IP address of the client.

`DST`: The destination server host name or IP address of the database server.

SRV: The database service name of Oracle Database obtained from the SERVICE_NAME parameter in the initialization parameter file.

ACT: The action for the connection request. Use accept to accept incoming requests, reject to reject incoming requests, or drop to reject incoming requests without sending an error message.

ACTION_LIST: The rule-level parameter settings for some parameters. These parameters are as follows:

- AUT: Oracle Database 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.

Example

```
(RULE_LIST=
  (RULE=
    (SRC=client1-pc)
    (DST=sales-server)
    (SRV=sales.us.example.com)
    (ACT=reject))
  (RULE=
    (SRC=192.0.2.45)
    (DST=192.0.2.200)
    (SRV=db1)
    (ACT=accept))
  (RULE=
    (SRC=sale-rep)
    (DST=sales1-server)
    (SRV=cmon)
    (ACT=accept)))
```

8.2.23 SDU

Purpose

To specify the session data unit (SDU) size, in bytes, to connections

Usage Notes

Oracle Connection Manager can negotiate large SDU with client and server when configured. When the configured values of client, database server, and Oracle Connection Manager do not match for a session, the least value of all the three values is used.

Default

8192 bytes (8 KB)

Values

512 to 2097152 bytes

Example

```
SDU=32768
```

8.2.24 SESSION_TIMEOUT

Purpose

To specify the maximum time in seconds allowed for a user session.

Usage Notes

The global setting can be overridden by a rule-level setting in `ACTION_LIST`.

Values

- `0` to disable the timeout. This is the default.
- Any number greater than 0 to enable the timeout. The number equals the timeout period in seconds.

8.2.25 TRACE_FILE

Purpose

To specify the directory for Oracle Connection Manager trace files.

8.2.26 TRACE_FILELEN

Purpose

To specify the size of the trace file in KB.

Usage Notes

When the size is reached, the trace information is written to the next file. The number of files is specified with the `TRACE_FILENO` parameter.

8.2.27 TRACE_FILENO

Purpose

To specify the number of trace files.

Usage Notes

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.

8.2.28 TRACE_LEVEL

Purpose

To specify the level for trace messages.

Values

- `off` for no tracing. This is the default.
- `user` for user-induced errors trace information.
- `admin` for administration trace information, such as installation-specific.
- `support` for Oracle Support Services information.

8.2.29 TRACE_TIMESTAMP

Purpose

To specify the use of a timestamp for the tracing logs.

Usage Notes

If the `TRACING` parameter is enabled, then a time stamp in the form of `dd-mmm-yyyy hh:mi:ss:mil` for every trace event in the trace file.

Values

- `off` for no timestamp to be included in the file.
- `on` for timestamp to be included in the file.

8.2.30 VALID_NODE_CHECKING_REGISTRATION

Purpose

To determine whether valid node checking registration is performed, and if the subnet is allowed.

Usage Notes

When set to `on`, valid node checking registration is performed at the listener for any incoming registration request, and only local IP addresses are allowed.

Default

`on`

Values

- `off | 0` to specify valid node checking registration is off, and no checking is performed.
- `on | 1 | local` to specify valid node checking registration is on, and all local IP addresses can register. If a list of invited nodes is set, then all IP addresses, host names, or subnets in the list as well as local IP addresses are allowed.

- subnet | 2 to specify valid node checking registration is on, and all machines in the local subnets are allowed to register. If a list of invited nodes is set, then all nodes in the local subnets as well as all IP addresses, host names and subnets in the list are allowed.

Example

```
VALID_NODE_CHECKING_REGISTRATION = on
```

8.2.31 WALLET_LOCATION

Purpose

To specify the location of wallets. Wallets are certificates, keys, and trustpoints processed by SSL.

Usage Notes

The key/value pair for Microsoft 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 Microsoft Windows registry and the wallet's key (KEY) is SALESAPP, then 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.

> **Note:**
>
> This parameter must be specified outside Oracle Connection Manager alias

Syntax

The syntax depends on the wallet, as follows:

- 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 Microsoft 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)))
  ```

Additional Parameters

WALLET_LOCATION supports the following parameters:

- SOURCE: The type of storage for wallets, and storage location.
- METHOD: The type of storage.
- METHOD_DATA: The storage location.
- DIRECTORY: The location of Oracle wallets on file system.
- KEY: The wallet type and location in the Microsoft Windows registry.
- PROFILE: The Entrust profile file (.epf).
- INIFILE: The Entrust initialization file (.ini).

Default

None

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 Microsoft 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)))
```

8.3 Oracle Connection Manager in Traffic Director Mode Parameters

This section lists and describes the following `cman.ora` file parameters:

- TDM
- TDM_BIND_THREAD
- TDM_DATATYPE_CHECK
- TDM_PRCP_MAX_CALL_WAIT_TIME
- TDM_PRCP_MAX_TXN_CALL_WAIT_TIME
- TDM_SHARED_THREADS_MAX
- TDM_SHARED_THREADS_MIN
- TDM_THREADING_MODE

8.3.1 TDM

Purpose

To configure Oracle Connection Manager to act as Oracle Connection Manager in Traffic Director Mode.

Default

`FALSE`

Values

- `TRUE`
- `FALSE`

Example

```
tdm = TRUE
```

8.3.2 TDM_BIND_THREAD

Purpose

To make the application connection hold on to the TDM thread and has different implications with and without PRCP. This parameter only applies when `TDM_THREADING_MODE` is set to `SHARED`.

Usage Notes

Without PRCP, setting this parameter to yes makes the application connection hold on the TDM worker thread as long as there is a transaction in progress.

With PRCP, setting this parameter to yes makes the application connection hold on to the TDM thread from the time OCISessionGet is done by the application till it does an OCISessionRelease.

Default

no

Values

- yes
- no

Example

```
TDM_BIND_THREAD = yes
```

8.3.3 TDM_DATATYPE_CHECK

Purpose

To validate all the inbound data to the database, of the data type NUMBER, DATE, TIMESTAMP, TIMESTAMP WITH LOCAL TIMEZONE, TIMESTAMP WITH TIMEZONE, BLOB, CLOB, BFILE, UROWID and REF. The following error is received by the application if there is any problem with the data sent to the Oracle Connection Manager in Traffic Director Mode.

```
ORA-03137: malformed TTC packet from client rejected: [3101]
```

Usage Notes

Turning ON/OFF this parameter enables or disables the data validation.

Default

OFF

Values

- ON
- OFF

Example

```
tdm_datatype_check={ON | OFF}
```

8.3.4 TDM_PRCP_MAX_CALL_WAIT_TIME

Purpose

To record the maximum time of inactivity, in seconds, for a client after obtaining a session from the PRCP pool. This parameter is applicable when the Oracle Connection Manager in Traffic Director Mode is configured to have Proxy Resident Connection Pool.

Usage Notes

After obtaining a session from the PRCP pool, if the client application does not issue a database call for the time specified by `TDM_PRCP_MAX_CALL_WAIT_TIME` parameter, then the PRCP session is freed and the client connection is terminated. As a result, if the client application attempts a round trip call on such a connection, then it receives an `ORA-3113` or `ORA-3115` error.

Default

30 seconds

Values

Any non negative value. However, Oracle recommends not to use a value of `0` as that implies that a connection can acquire a PRCP session for an indefinite amount of time

8.3.5 TDM_PRCP_MAX_TXN_CALL_WAIT_TIME

Purpose

To record the maximum time of inactivity, in seconds, for a client after it obtains a session from the Proxy Resident Connection Pool and starts a transaction. This parameter is applicable when the Oracle Connection Manager in Traffic Director Mode is configured to have PRCP.

Usage Notes

If the client application does not issue a database call for the time specified by `TDM_PRCP_MAX_TXN_CALL_WAIT_TIME` parameter while in a transaction, the PRCP session is freed, the transaction is rolled back, and the client connection is terminated. As a result, if the client application attempts a round trip call on such a connection, then it receives an `ORA-3113` or `ORA-3115` error.

Default

`0`

Values

Any nonnegative value. However, it is recommended not to use a value of`0` as it implies that a connection can acquire a PRCP session for an indefinite amount of time.

8.3.6 TDM_SHARED_THREADS_MAX

Purpose

To configure the maximum number of threads that an Oracle Connection Manager process in Traffic Director Mode should have, when `tdm_threading_mode` is set to `SHARED`.

Values

Any number can be designated for the maximum number of threads. For `DEDICATED` mode, the maximum number of threads is same as the maximum number of connections. In `SHARED` mode, though there is no fixed upper bound, it should ideally be proportional to the load.

8.3.7 TDM_SHARED_THREADS_MIN

Purpose

To configure the minimum number of threads that an Oracle Connection Manager process in Traffic Director Mode should have, when `tdm_threading_mode` is set to `SHARED`.

Values

Any number can be designated for the minimum number of threads. For `SHARED` mode, there is no limit enforced. However, the number of threads should be proportional to the load.

8.3.8 TDM_THREADING_MODE

Purpose

To configure the usage of threads by the Oracle Connection Manager in Traffic Director Mode.

Usage Notes

If this parameter is set to `DEDICATED`, then a worker thread is spawned for each inbound connection and the maximum number of threads is determined by the `max_connections` parameter

If this parameter is set to `SHARED`, then a shared pool of worker threads handle all inbound connections. The minimum number of worker threads is specified by the `tdm_shared_threads_min` setting and the maximum number of worker threads is specified by the `tdm_shared_threads_max` setting. The thread pool is internally managed within these bounds.

Default

`DEDICATED`

Values

- `DEDICATED`
- `SHARED`

Example

```
tdm_threading_mode={DEDICATED | SHARED}
```

```
tdm_shared_threads_min = 4
```

```
tdm_shared_threads_max = 5
```

8.4 ADR Diagnostic Parameters for Oracle Connection Manager

Since Oracle Database 11*g*, Oracle Database includes an advanced fault diagnosability infrastructure for preventing, detecting, diagnosing, and resolving problems. The problems are critical errors such as those caused by database code bugs, metadata corruption, and customer data corruption.

When a critical error occurs, it is assigned an incident number, and diagnostic data for the error, such as traces and dumps, are immediately captured and tagged with the incident number. The data is then stored in the Automatic Diagnostic Repository (ADR), a file-based repository outside the database.

This section describes the parameters used when ADR is enabled. Non-ADR parameters listed in the `cman.ora` file are ignored when ADR is enabled. "Non-ADR Diagnostic Parameters for Oracle Connection Manager" Non-ADR Diagnostic Parameters for Oracle Connection Manager describes the parameters used when ADR is disabled. ADR is enabled by default.

- ADR_BASE
- DIAG_ADR_ENABLED
- LOG_LEVEL
- TRACE_LEVEL
- TRACE_TIMESTAMP

8.4.1 ADR_BASE

Purpose

To specify the base directory to store tracing and logging incidents when ADR is enabled.

Default

The default is `ORACLE_BASE`, or `ORACLE_HOME/log` if `ORACLE_BASE` is not defined.

Values

Any valid directory path to a directory with write permission.

Example

```
ADR_BASE=/oracle/network/trace
```

8.4.2 DIAG_ADR_ENABLED

Purpose

To indicate whether ADR tracing is enabled.

Usage Notes

When the `DIAG_ADR_ENABLED` parameter is set to `OFF`, then non-ADR file tracing is used.

Values

`on` | `off`

Example

```
DIAG_ADR_ENABLED=on
```

8.4.3 LOG_LEVEL

Purpose

To specify the level of logging performed by Oracle Connection Manager.

Usage Notes

This parameter is also applicable when non-ADR logging is used.

The following log files are used with Oracle Connection Manager:

- *`instance-name_pid`*`.log` for the listener.
- *`instance-name`*`_cmadmin_`*`pid`*`.log` for CMADMIN.
- *`instance-name`*`_cmgw_`*`pid`*`.log` for the gateway processes.

The log files are located in the `ORACLE_HOME/network/log` directory.

Default

off or 0

Values

- `off` or `0` for no log output.
- `user` or `4` for user log information.
- `admin` or `10` for administration log information.
- `support` or `16` for Oracle Support Services log information.

Example

```
LOG_LEVEL=admin
```

8.4.4 TRACE_LEVEL

Purpose

To specify the trace level for the Oracle Connection Manager instance.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

The following trace files are used with Oracle Connection Manager:

- `*instance-name_pid*.trc` for the listener.
- `*instance-name*_cmadmin_*pid*.trc` for CMADMIN.
- `*instance-name*_cmgw_*pid*.trc` for the gateway processes.

The log files are located in the `ORACLE_HOME/network/log` directory.

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=admin
```

8.4.5 TRACE_TIMESTAMP

Purpose

To add a time stamp in the form of `dd-mmm-yyyy hh:mi:ss:mil` to every trace event in the trace file for the listener.

Usage Notes

This parameter is used with the TRACE_LEVEL parameter. This parameter is also applicable when non-ADR tracing is used.

Default

on

Values

- `on` or `true`
- `off` or `false`

Example

```
TRACE_TIMESTAMP=true
```

8.5 Non-ADR Diagnostic Parameters for Oracle Connection Manager

This section lists the parameters used when ADR is disabled:

> **Note:**
>
> The default value of DIAG_ADR_ENABLED is `on`. Therefore, the `DIAG_ADR_ENABLED` parameter *must* explicitly be set to `off` in order for non-ADR tracing to be used.

- LOG_DIRECTORY
- TRACE_DIRECTORY
- TRACE_FILELEN
- TRACE_FILENO

8.5.1 LOG_DIRECTORY

Purpose

To specify the location of Oracle Connection Manager log files.

Usage Notes

Use this parameter when ADR is not enabled.

Default

`ORACLE_BASE_`*`HOME`*`/network/log`

Values

Any valid directory path to a directory with write permission.

Example

```
LOG_DIRECTORY=/oracle/network/log
```

8.5.2 TRACE_DIRECTORY

Purpose

To specify the location of the Oracle Connection Manager trace files.

Usage Notes

Use this parameter when ADR is not enabled.

Default

`ORACLE_BASE_HOME/network/trace`

Values

Any valid directory path to a directory with write permission.

Example

```
TRACE_DIRECTORY=/oracle/network/admin/trace
```

8.5.3 TRACE_FILELEN

Purpose

To specify the size, in KB, of the trace file.

Usage Notes

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. Use this parameter when ADR is not enabled.

Default

Unlimited

Example

```
TRACE_FILELEN=100
```

8.5.4 TRACE_FILENO

Purpose

To specify the number of trace files for Oracle Connection Manager tracing.

Usage Notes

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`, then 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. Use this parameter when ADR is not enabled.

Default

1

Example

```
TRACE_FILENO=3
```

9

Directory Usage Parameters in the ldap.ora File

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

This chapter contains the following topics:

- Overview of Directory Server Usage File
- Directory Usage Parameters

9.1 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. When created with Oracle Net Configuration Assistant, the `ldap.ora` file is located in the `ORACLE_HOME/network/admin` directory. The `ldap.ora` file can also be stored in the directory specified by the `LDAP_ADMIN` or `TNS_ADMIN` environment variable.

9.2 Directory Usage Parameters

This section lists and describes the following `ldap.ora` file configuration parameters:

- DIRECTORY_SERVERS
- DIRECTORY_SERVER_TYPE
- DEFAULT_ADMIN_CONTEXT

9.2.1 DIRECTORY_SERVERS

Purpose

To list the host names and port number of the primary and alternate LDAP directory servers.

Values

host:*port*[:*sslport*]

Example

```
DIRECTORY_SERVERS=(ldap-server:389, raffles:400:636)
```

9.2.2 DIRECTORY_SERVER_TYPE

Purpose

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
```

9.2.3 DEFAULT_ADMIN_CONTEXT

Purpose

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

Part III contains the following appendixes:

- Features Not Supported in this Release
- Upgrade Considerations for Oracle Net Services
- 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 the following topics:

- Overview of Unsupported Features
- Unsupported Parameters
- Unsupported Control Utility Commands
- Unsupported or Deprecated Protocols

A.1 Overview of Unsupported Features

The following section describe the features and the configuration file that are no longer being supported in Oracle Database. This is based on an effort to streamline configuration and use of Oracle Database.

A.1.1 Oracle Net Connection Pooling

In Oracle Database 12*c* Release 2 (12.2), Oracle Net connection pooling is no longer supported. It was deprecated in Oracle Database 11*g*.

> **See Also:**
>
> My Oracle Support note 1469466.1

A.1.2 Oracle Names

Oracle Names has not been supported as a naming method since Oracle Database 11*g*. You must migrate to directory naming.

> **See Also:**
>
> *Oracle Database Net Services Administrator's Guide* for additional information about migrating to directory naming

A.1.3 Oracle Net Listener Password

In Oracle Database 12*c* Release 2 (12.2), the Oracle Net Listener password feature is no longer supported. This does not cause a loss of security because authentication is enforced through local operating system authentication.

See Also:

"Oracle Net Listener Security"

A.2 Unsupported Parameters

Table A-1 describes the networking parameters no longer supported by this release.

Table A-1 Unsupported Networking Parameters

File	Parameter	Description	Last Supported Release
`sqlnet.ora`	`SQLNET.KERBEROS5_CONF_MIT`	This parameter was used to specify that MIT Kerberos configuration format was used. Starting with Oracle Database 12*c* Release 2 (12.2), only the current MIT Kerberos configuration is supported.	11.2
`sqlnet.ora`	`SQLNET.ALLOWED_LOGON_VERSION`	This parameter has been divided into `SQLNET.ALLOWED_LOGON_VERSION_CLIENT` and `SQLNET.ALLOWED_LOGON_VERSION_SERVER`. **See Also:** "SQLNET.ALLOWED_LOGON_VERSION_CLIENT" and "SQLNET.ALLOWED_LOGON_VERSION_SERVER"	11.2

A.3 Unsupported Control Utility Commands

Table A-2 describes the control utility commands not supported by this release.

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

A.4 Unsupported or Deprecated Protocols

Table A-3 describes the protocols not supported or deprecated since Oracle Database 12*c*.

Table A-3 Unsupported Protocols

Protocol	Description	Last Supported Release
NT LAN Manager (NTLM) protocol for domain authentication	NTLM domain authentication has been deprecated from the Oracle Windows adapter. Only Kerberos authentication is used for the NTS adapter. NTLM is still used for local user authentication, as well as in the case in which the database service runs as a local user.	11.2

B

Upgrade Considerations for Oracle Net Services

This appendix describes coexistence and upgrade issues for Oracle Net Services. This appendix covers the following topic:

- Anonymous Access to Oracle Internet Directory

B.1 Anonymous Access to Oracle Internet Directory

Typical users of directory naming (LDAP) require anonymous access to the Oracle Internet Directory for name lookup. If you upgrade your Oracle Internet Directory software release 11*g* or later, then the default setting for Oracle Internet Directory changes to disallow anonymous access to the directory. The directory administrator must configure the directory to enable anonymous binds after upgrading the directory to release 11*g*. In addition, the way anonymous binds are configured in Oracle Internet Directory changed between Oracle Database 10*g* and Oracle Database 11*g*.

See Also:

Oracle Internet Directory Administrator's Guide for additional information about anonymous binds

C

LDAP Schema for Oracle Net Services

This appendix describes the Oracle schema object classes and attributes 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 the following topics:

- Structural Object Classes
- Attributes

C.1 Structural Object Classes

The Oracle schema supports the structural object classes for Oracle Net directory naming lookups. Table C-1 lists the structural object classes for Oracle Connection Manager. The attributes are described in Table C-2 in the following section.

Table C-1 Oracle Net Structural Object Classes

Object Class	Attributes	Description
orclDBServer	• orclNetDescName • orclVersion	Defines the attributes for database service entries.
orclNetAddress	• orclNetAddressString • orclNetProtocol • orclVersion	Specifies a listener protocol address.
orclNetAddressAux1	• orclNetHostname	Specifies an auxiliary object class to add attributes to an orclNetAddress entry.
orclNetAddressList	• orclNetAddrList • orclNetFailover • orclNetLoadBalance • orclNetSourceRoute • orclVersion	Specifies a list of protocol addresses.
orclNetDescription	• orclNetAddrList • orclNetInstanceName • orclNetConnParamList • orclNetFailover • orclNetLoadBalance • orclNetSdu • orclNetServiceName • orclNetSourceRoute • orclSid • orclVersion	Specifies a connect descriptor containing the protocol address of the listener and the connect information to the service.

Table C-1 (Cont.) Oracle Net Structural Object Classes

Object Class	Attributes	Description
orclNetDescriptionAux1	• orclNetSendBufSize • orclNetReceiveBufSize • orclNetFailoverModeString • orclNetInstanceRole	Specifies auxiliary object class to add attributes to an orclNetDescription entry.
orclNetDescriptionList	• orclNetDescList • orclVersion	Specifies a list of connect descriptors.
orclNetService	• orclNetDescName • orclVersion	Defines the attributes for network service name entries.
orclNetServiceAlias	• orclNetDescName • orclVersion	Defines the attributes for network service alias entries.

C.2 Attributes

Table C-2 lists the attributes used for the object classes. This list is subject to change.

Table C-2 LDAP Schema Attributes for Oracle Net Services

Attribute	Description
`orclCommonContextMap`	Allows the mapping of more than one default Oracle Context in the directory server.
`orclNetAddrList`	Identifies one or more listener protocol addresses.
`orclNetAddressString`	Defines a listener protocol address.
`orclNetConnParamList`	Placeholder for 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`	Instructs Oracle Net to fail over to a different listener if the first listener fails during runtime. Depending on the configuration, session or any `SELECT` statements that were in progress are automatically failed over.
`orclNetHostname`	Specifies the host name.
`orclNetInstanceName`	Specifies the instance name to access.
`orclNetInstanceRole`	Specifies a connection to the primary or secondary instance of an Oracle Real Application Clusters (Oracle RAC) 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.

Table C-2 (Cont.) LDAP Schema Attributes for Oracle Net Services

Attribute	Description
`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.

ADR

See Automatic Diagnostic Repository (ADR).

alias

An alternative name for a network object in a server. An alias stores the name of the object it 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 an 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:

- RADIUS (Remote Authentication Dial-In User Service)
- Kerberos
- SSL
- Microsoft Windows NT native authentication

Automatic Diagnostic Repository (ADR)

Automatic Diagnostic Repository is a systemwide central repository for tracing and logging files. The repository is a file-based hierarchical data store for depositing diagnostic information.

cache

Memory that stores recently-accessed data to so that subsequent requests to access the same data can be processed quickly.

CIDR

Classless Inter-Domain Routing. In CIDR notation, an IPv6 subnet is denoted by the subnet prefix and the size in bits of the prefix (in decimal), separated by the slash (`/`) character. For example, `2001:0db8:0000:0000::/64` denotes a subnet with addresses `2001:0db8:000:0000:0000:0000:0000:0000` through `2001:0db8:000:0000:FFFF:FFFF:FFFF:FFFF`. The CIDR notation includes support for IPv4 addresses. For example, `192.0.2.1/24` denotes the subnet with addresses `192.0.2.1` through `192.0.2.255`.

Classless Inter-Domain Routing (CIDR)

See CIDR.

client

A user, software application, or computer that requests the services, data, or processing from another application or computer. The client is the user process. In a network environment, the client is the local user process and the server may be local or remote.

client load balancing

Load balancing, whereby if more than one listener services a single database, a client can randomly choose between the listeners for its connect requests. This randomization enables all listeners to share the burden of servicing incoming connect requests.

client profile

The properties of a client, which may include the preferred order of naming methods, client and server logging and tracing, the domain from which to request names, and other client options.

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 service for the requests.

cman.ora file

An Oracle Connection Manager 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 run 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 process 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.example.com`:

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)
  (CONNECT_DATA=
     (SERVICE_NAME=sales.us.example.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. 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 network service name, database service name, or network service alias. Users initiate a connect request by passing a user name and password along with a connect identifier in a connect string for the service to which they want to connect:

```
CONNECT username@connect_identifier
```

connect string

Information the user passes to a service to connect, such as user name, and connect identifier:

```
CONNECT username@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 attempt 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

The method for balancing the number of active connections for the same service across the instances and dispatchers. Connection load balancing enables listeners to make routing decisions based on how many connections for each dispatcher and the load on the nodes.

connection pooling

A resource utilization and user scalability feature used 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 link

A pointer that defines a one-way communication path from an Oracle database server to another database server. Public and private database links are a defined entries in a data dictionary table. Global database links are stored in an LDAP directory and can be accessed by all users on the network. To access public and private links, the user must be connected to the local database that contains the data dictionary entry.

A client connected to local database A can use a public or private link stored in database A to access information in remote database B. However, 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 a link must be defined and stored in the data dictionary of database B. Global links may be used between any clients and database on the network.

The following database links 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.
- A global database link is a database link stored in the LDAP directory.

dedicated connection

A dedicated server with a database session.

dedicated server

A server process that is dedicated to one client connection. Contrast with shared server.

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 period (.) character within it.

directory information tree (DIT)

A hierarchical tree-like structure in a directory server of the distinguished names (DNs) of the entries. This structure is specific to x500 and LDAP.

directory naming

A naming method that resolves a database service, network service name, or network 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 method for managing and configuring a distributed Oracle network. The directory server can replace client-side and server-side 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.

distinguished name (DN)

Name of entry in a directory server. The DN specifies where the entry resides in the LDAP directory hierarchy, similar to 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 Net Services supports 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 Name System (DNS)

A system for naming computers and network services that is organized into a hierarchy of domains. DNS is used in TCP/IP networks to locate computers through user-friendly names. DNS resolves a friendly name into an IP address, which is understood by computers.

For Oracle Net Services, DNS translates the host name in a TCP/IP address into an IP address.

DNS

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 by the database administrator of a particular database. An enterprise role can be granted to or revoked from 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 Network Information Service (NIS).

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.

FTP

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 link

A database link definition stored in an LDAP directory which can be accessed by all users on the network. This definition is the same as the one used for client connections to the database (name/connect-descriptor).

Global database links cannot include user or password clauses. They only work when the database initiating the link uses the identity of the existing client to establish the link.

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.example.com`.

The database name portion, `sales`, is a simple name to call a database. The database domain portion, `us.example.com`, specifies the database domain which the database is located, making the global database name unique. When possible, Oracle 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.

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

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

The combination of the System Global Area (SGA) and the Oracle background processes. When a database is started on a database server (regardless of the type of computer), Oracle allocates a memory area called the SGA, and starts one or more Oracle processes. The memory and processes of an instance efficiently manage the associated database data and serve the database users. You can connect to any instance to access information within a cluster database.

instance name

A name of an Oracle database instance. The instance name is identified by the INSTANCE_NAME parameter in the database initialization parameter file. INSTANCE_NAME corresponds to the Oracle system identifier (SID) of the instance. Clients can connect to a specific instance by specifying the INSTANCE_NAME parameter in the connect descriptor.

The instance name is included in the connect data part of the connect descriptor.

IP address

Used to identify a node on a network. Each computer on the network is assigned a unique Internet Protocol (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 `192.0.2.22`.

IPC

Interprocess Communication is 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.

IPv4

Internet Protocol Version 4. IPv4 is the current standard for the IP protocol. IPv4 uses 32-bit (four-byte) addresses, which are typically represented in dotted-decimal notation. The decimal value of each octet is separated by a period, as in `192.0.2.22`.

IPv6

Internet Protocol Version 6. The protocol designed to replace IPv4. In IPv6, an IP address is typically represented in eight fields of hexadecimal values separated by colons, as in `2001:0db8:0000:0000:0000:0000:1428:57AB`. In some cases, fields with `0` values can be compressed, as in `2001:DB8::1428:57AB`.

IP Version 4 (IPv4)

See IPv4.

IP Version 6 (IPv6)

See IPv6.

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

The amount of time it takes to send a request and receive an answer.

LDAP Data Interchange Format (LDIF)

See LDIF

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 use to look up or configure connect identifiers for connections to database services

When created with Oracle Internet Directory Configuration Assistant, the `ldap.ora` file is located in the `ORACLE_HOME/ldap/admin` directory. When created with Oracle Net Configuration Assistant, the `ldap.ora` file is located in the `ORACLE_HOME/network/admin` directory.

LDIF

LDAP Data Interchange Format (LDIF) is the set of standards for formatting an input file for any of the LDAP command line utilities.

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 servers.

link qualifier

An extension to the database link name which specifies the connect name used to connect to the database. It provides alternate settings for the database user name and password credentials. For example, a link qualifier of `fieldrep` can be appended to a global database link of `sales.us.example.com`.

```
SQL> SELECT * FROM emp@sales.us.example.com@fieldrep
```

listener

See Oracle Net Listener.

Listener Control utility

A utility included with Oracle Net Services to control listener functions, such as starting, stopping, and getting the status of the listener.

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 the `ORACLE_HOME/network/admin` directory.

Oracle does not require identification of the database service because of service registration. However, static service configuration is required if you plan to use Oracle Enterprise Manager Cloud Control.

Listener Registration (LREG)

As a part of service registration, LREG registers instance information with the listener. LREG is an instance background process of each database instance that is configured in the database initialization parameter file.

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 Net 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 Microsoft 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.

Microsoft Active Directory can be configured as a directory naming method to store service information that clients can access.

Microsoft Windows NT native authentication

An authentication method that enables a client single login access to a Microsoft Windows NT server and a database running on the 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 protocol.

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).

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:

- Domain Name System (DNS)
- directory naming
- Easy Connect naming
- external naming

network service alias

An alternative name for a directory naming object in a directory server. A directory server stores network service aliases for any defined network service name or database service. A network 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 network service alias, the directory determines that the entry is a network service alias and completes the lookup as if it was actually the entry it is referencing.

network service name

A simple name for a service that resolves to a connect descriptor. Users initiate a connect request by passing a user name and password along with a network service name in a connect string for the service to which they want to connect:

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

Depending on your needs, network 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

network

A group of two or more computers linked 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 uppercase, and lowercase, and some special characters.

Network Information Service (NIS)

The client/server protocol for distributing system configuration data such as user and host names between computers on a network. This service was formerly known as "Sun Microsystems Yellow Pages (yp)."

Network Interface (NI)

A network layer that provides a generic interface for Oracle clients, servers, or external processes to access Oracle Net functions. The network interface 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, such as a listener.

network protocol

See Oracle protocol support.

Network Program Interface

An interface for server-to-server interactions that performs all of the functions that the Oracle Call Interface (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.

NIS

See Network Information Service (NIS).

node

A computer or terminal that is part of a network

object class

In a directory server, a named group of attributes. To assign attributes to an entry, do so by assigning the object classes that hold those attributes to that entry.

All objects associated with the same object class share the attributes of that object class.

OCI

See Oracle Call Interface (OCI).

OPI

See Oracle Program Interface (OPI).

Open Systems Interconnection (OSI)

Open Systems Interconnection is a network architecture model developed by ISO as a framework for international standards in heterogeneous computer network architecture.

The OSI architecture has 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

Oracle Advanced Security

An Oracle product that provides Transparent Data Encryption (TDE) and data redaction.

Oracle Call Interface (OCI)

An application programming interface (API) that enables creation of 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 data types, 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 Oracle Connection Manager can then take advantage of the session multiplexing, access control, or protocol conversion features configured for that Oracle Connection Manager.

Oracle Connection Manager Control utility

A utility included with Oracle Net Services to control various functions, such as starting, stopping, and getting the status of Oracle Connection Manager.

Oracle Context

An entry in an LDAP-compliant Internet directory called `cn=OracleContext`, under which all Oracle software relevant information is kept, including entries for Oracle Net Services directory naming and checksumming security. There may be one or more than one Oracle Context in a directory. An Oracle Context entry can be associated with a directory naming context.

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 Cloud Control

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. After 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 can perform these jobs because it is located on each computer in the network.

Oracle Net Configuration Assistant

A postinstallation tool that configures basic network components after installation, including:

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

Oracle Net Firewall Proxy

Product offered by some firewall vendors that supplies Oracle Connection Manager functionality.

Oracle Net foundation layer

A networking communication layer that is responsible for establishing and maintaining the connection between the client application and server, as well as exchanging messages between them.

Oracle Net 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.

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 identifiers and map them to connect descriptors to identify the network location and identification of a service. Oracle Net Manager supports configuration of connect descriptors in a local `tnsnames.ora` file or directory server.

- Naming Methods

  Configure the 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)

Oracle Program Interface is the 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 after 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 Real Application Clusters (Oracle RAC)

An architecture that allows multiple instances to access a shared database of data files. Oracle RAC is also a software component that provides the necessary cluster database scripts, initialization files, and data files needed for Oracle Enterprise Edition and Oracle RAC.

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 an Oracle database. For any database, there is at least one instance referencing the database.

For Oracle databases earlier than release 8.1, a 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_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, such as connect, accept, redirect, data, and so on. Packet information can be useful in troubleshooting.

PMON process

A process monitor (PMON) database process that performs process recovery when a user process fails. PMON is responsible for cleaning 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.

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 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 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 particularpresentation layer andsession layer combination.

proxy server

A server that substitutes for a 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. It is a fully-qualified X.500 name. For example, `cn=sales,dc=us,dc=example,dc=com`, `cn=sales` is a RDN.

root Oracle Context

In the Oracle Identity Management infrastructure, the root Oracle Context is an entry in Oracle Net Services containing a pointer to the default identity management realm in the infrastructure. It also contains information about how to locate an identity management realm given the simple name of the realm.

RPC

Remote procedure call.

SDP

Sockets Direct Protocol.

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 parameter file

A binary file containing initialization parameter settings that is maintained on the Oracle Database host. You cannot manually edit this file with a text editor. A server parameter file is initially built from a text initialization parameter file by means of the `CREATE SPFILE` statement or created directly.

server process

Database processes that handle a client request on behalf of a database.

service

A program that responds to requests from various clients or performs some operation. 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. The service name is a string that is the global database name, that is, a name comprised of the database name and domain name, entered during installation or database creation. If you are not sure what the global database name is, then 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 Listener Registration (LREG) 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, then 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 dialog 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 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 Oracle databases release 8.0, 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 the `ORACLE_HOME/network/admin` directory.

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

Transmission Control Protocol/Internet Protocol. The 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 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.

TNS

See Transparent Network Substrate (TNS).

tnsnames.ora file

A configuration file that maps network service names to connect descriptors. This file is used for the local naming method. The `tnsnames.ora` file typically resides in the `ORACLE_HOME/network/admin` directory.

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 run. 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 Oracle 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 HTTP which allows users to manage files on remote Web servers.

Index