

## Table of Contents

1. Configuration Management Concepts.....	3
1.1. Configuration Models .....	3
1.2. Localized management .....	3
1.3. Centralized management.....	4
2. Architecture of Oracle Net Services.....	5
2.1. Oracle Net Stack Communication Architecture .....	5
2.1.1. Stack Communication for Client/Server Application Connections .....	5
2.1.2. Stack Communication for Java Application Connections.....	5
2.1.3. Stack Communication for Web Client Connections .....	5
2.2. Listener Architecture .....	6
2.3. Database Server Process Architecture.....	7
2.3.1. Shared Server Processes .....	7
2.3.2. Dedicated Server Processes .....	7
3. Configuration and Administration Tools Overview.....	8
3.1. User Interface Tools.....	8
3.1.1. Oracle EM Database Express.....	8
3.1.2. Oracle Net Manager .....	9
3.2. Oracle Net Control Utility.....	10
3.2.1. Listener Control Utility .....	10
3.2.2. Connection Manager Control Utility.....	10
4. Configuring and Administering Oracle Net Listener .....	11
4.1. Configuration Overview .....	11
4.2. Configuring Service Registration.....	12
4.3. Listener Administration.....	13
4.3.1. Starting and Stopping a Listener .....	13
4.3.2. Determining the Current Status of a Listener .....	13
4.3.3. Monitoring Services of a Listener .....	14
5. Establish A Network and Testing the Connection.....	15
5.1. Confirming Network Availability .....	15

5.2.	Connecting to a Database .....	15
	Task 1: Start the Listener .....	15
	Task 2: Start the Database .....	15
6.	Client Connections .....	16
6.1.	From the command line.....	16
6.2.	From a login screen.....	16
6.3.	From a 3GL application .....	16
6.4.	From within SQL*Plus .....	16

# 1. Configuration Management Concepts

## 1.1. Configuration Models

Configuration information can be stored in a localized configuration file or a centralized repository.

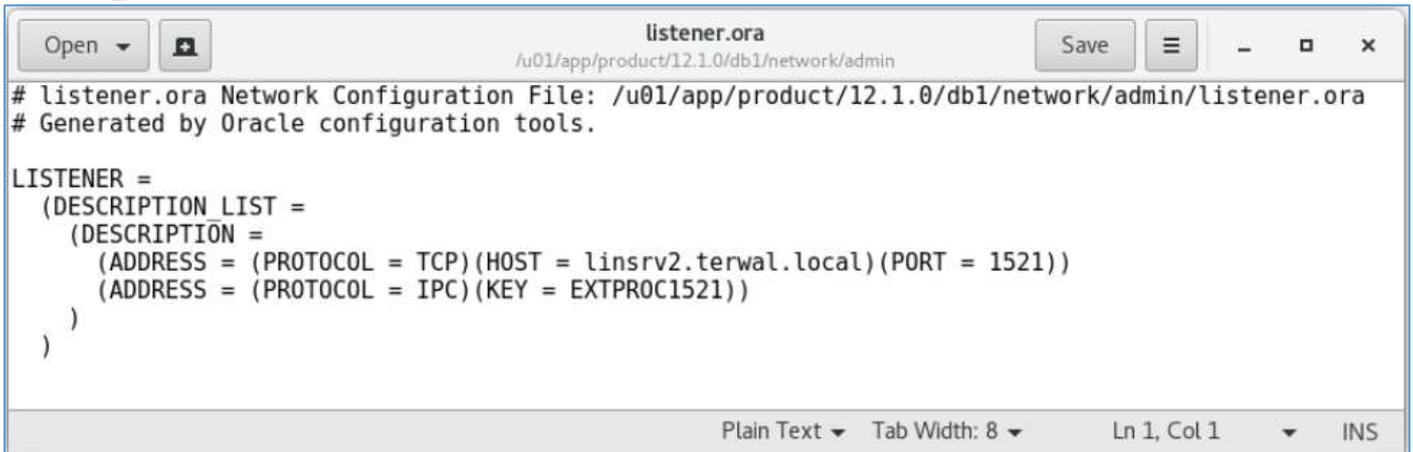
## 1.2. Localized management

Network address information stored in `tnsnames.ora` files on each computer in the network

`cman.ora` => Located on the computer where Oracle Connection Manager runs.

`listener.ora` => Located on the database server

The search order is as follows: `$TNS_ADMIN` directory > Global Configuration Directory `/var/opt/oracle > $ORACLE_HOME/network/admin`



```
# listener.ora Network Configuration File: /u01/app/product/12.1.0/db1/network/admin/listener.ora
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = linsrv2.terwal.local)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )
```

`sqlnet.ora` => Located on client and database server computer

The search order is as follows: `$TNS_ADMIN` directory > `$ORACLE_HOME/network/admin`



```
# sqlnet.ora Network Configuration File: /u01/app/product/12.1.0/db1/network/admin/sqlnet.ora
# Generated by Oracle configuration tools.

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
```

`tnsnames.ora` => Located primarily on the clients, this file contains `net service name` mapped to `connect descriptors`

The search order is as follows: `$TNS_ADMIN` directory > Global Configuration Directory `/var/opt/oracle > $ORACLE_HOME/network/admin`



```
# tnsnames.ora Network Configuration File: /u01/app/product/12.1.0/db1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = linsrv2.terwal.local)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl.terwal.local)
    )
  )
```

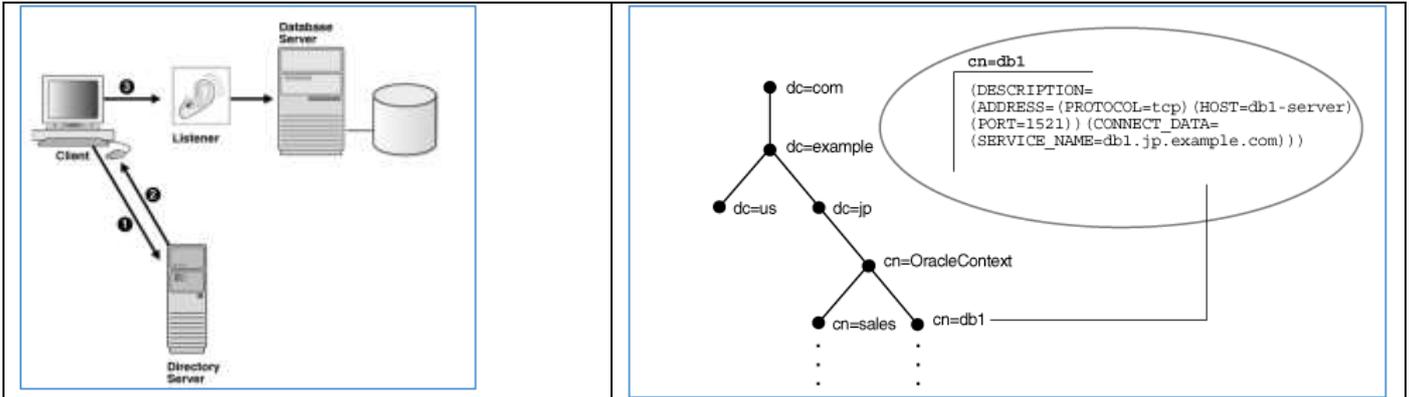
```

oracle@linsrv2:/u01/app/product/12.1.0/db1/network/admin> ls -la
total 16
drwxr-xr-x  3 oracle oinstall  97 Dec 22 22:28 .
drwxr-xr-x 10 oracle oinstall 106 Dec 22 22:06 ..
-rw-r--r--  1 oracle oinstall 336 Dec 22 22:23 listener.ora
drwxr-xr-x  2 oracle oinstall  64 Dec 22 22:05 samples
-rw-r--r--  1 oracle oinstall 373 Oct 31  2013 shrept.lst
-rw-r--r--  1 oracle oinstall 183 Dec 22 22:23 sqlnet.ora
-rw-r----- 1 oracle oinstall 340 Dec 22 22:28 tnsnames.ora

```

### 1.3. Centralized management

Network address information is stored in centralized directory services, including a LDAP-compliant **directory server**.

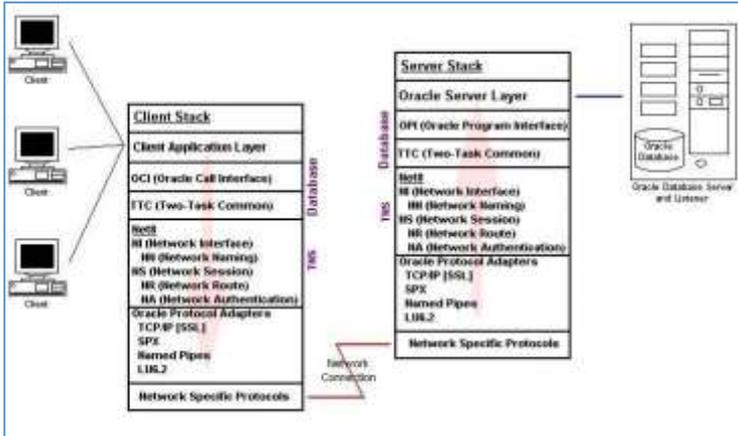


Today, network information is stored in multiple systems and in multiple directory formats. With new requirements for Internet computing and new e-business technologies, a common repository infrastructure is needed as a foundation for management and configuration of all data and resources. This kind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

Support of **Oracle Internet Directory** provides a centralized vehicle for managing and configuring a distributed Oracle network. The directory server can replace client-side and server-side localized **tnsnames.ora** files.

## 2. Architecture of Oracle Net Services

### 2.1. Oracle Net Stack Communication Architecture



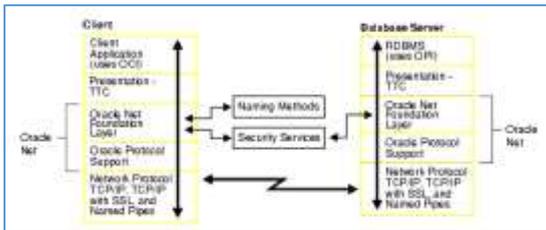
A client is any application that connects to Oracle Database to send or retrieve data. An Oracle Database client application can reside on any computer that has Oracle Database **client software** installed.

**Oracle Net** is a software layer that resides on the client computer and on the Oracle Database host computer. It establishes and maintains the connection between the client application and the database over a network, and exchanges messages between them using industry standard protocols.

For a client application and a database to communicate, the client application must be able to identify the database it wants to connect to, and the database must provide an identification. You can use a service name to connect to a database. A **Service Name** is a logical representation of a database, which is the way a database is presented to clients. A single database can be presented as multiple services.

Service names can provide location transparency so that the client application does not have to know the server's location. If the database is moved to another location, then you must reconfigure only Oracle Net. No changes are necessary to client applications.

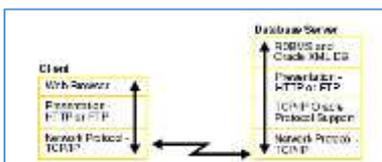
#### 2.1.1. Stack Communication for Client/Server Application Connections



#### 2.1.2. Stack Communication for Java Application Connections

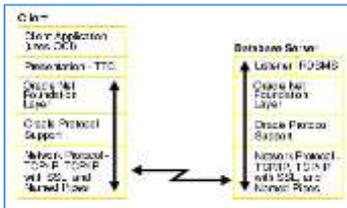


#### 2.1.3. Stack Communication for Web Client Connections



## 2.2. Listener Architecture

The database server receives an initial connection from a client application through the listener. The listener is an application positioned on top of the Oracle Net foundation layer.



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

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

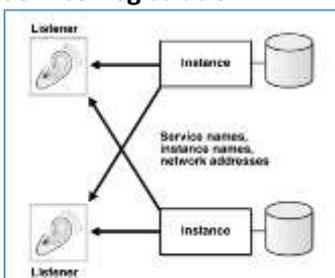
Once a client request has reached the listener, the listener selects an appropriate **service handler** to service the client's request and forwards the client's request to it. The listener determines if a database service and its service handlers are available through **service registration**. During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

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

This information enables the listener to direct a client's request appropriately.

The next figure shows instances registering information with listeners. Note that it does not represent all the information that can be registered.

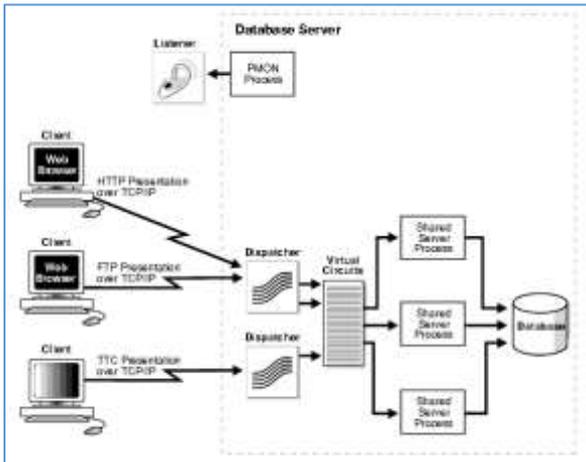
### Service Registration



## 2.3. Database Server Process Architecture

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

### 2.3.1. Shared Server Processes

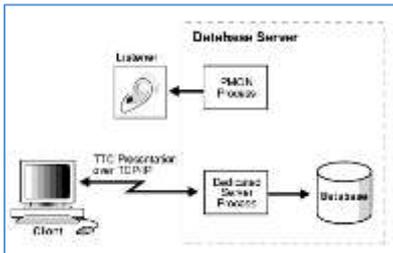


Shared server processes are used in the shared server architecture. With shared server architectures, client processes ultimately connect to a dispatcher. The **PMON** process registers the location and load of the dispatchers with the listener, enabling the listener to forward requests to the least loaded dispatcher.

A dispatcher can support multiple client connections concurrently. Each client connection is bound to a virtual circuit. A virtual circuit is a piece of **shared memory** used by the dispatcher for client database connection requests and replies. The dispatcher places a virtual circuit on a common queue when a request arrives. An idle shared server picks up the virtual circuit from the common queue, services the request, and relinquishes

the virtual circuit before attempting to retrieve another virtual circuit from the common queue. This approach enables a small pool of server processes to serve a large number of clients.

### 2.3.2. Dedicated Server Processes



With a dedicated server architecture, each client process connects to a dedicated server process. The server process is not shared by any other client. PMON registers information about dedicated server processes with the listener. This enables the listener to start up a dedicated server process when a client request arrives and forward the request to it. Dedicated server architectures do not support HTTP, FTP or WebDAV clients. **Only database clients are supported.**

## 3. Configuration and Administration Tools Overview

### 3.1. User Interface Tools

#### 3.1.1. Oracle EM Database Express

Oracle Enterprise Manager Database Express, also referred to as EM Express, is a web-based tool for managing Oracle Database 12c. Built inside the database server, it offers support for basic administrative tasks such as storage and user management, and provides comprehensive solutions for performance diagnostics and tuning.

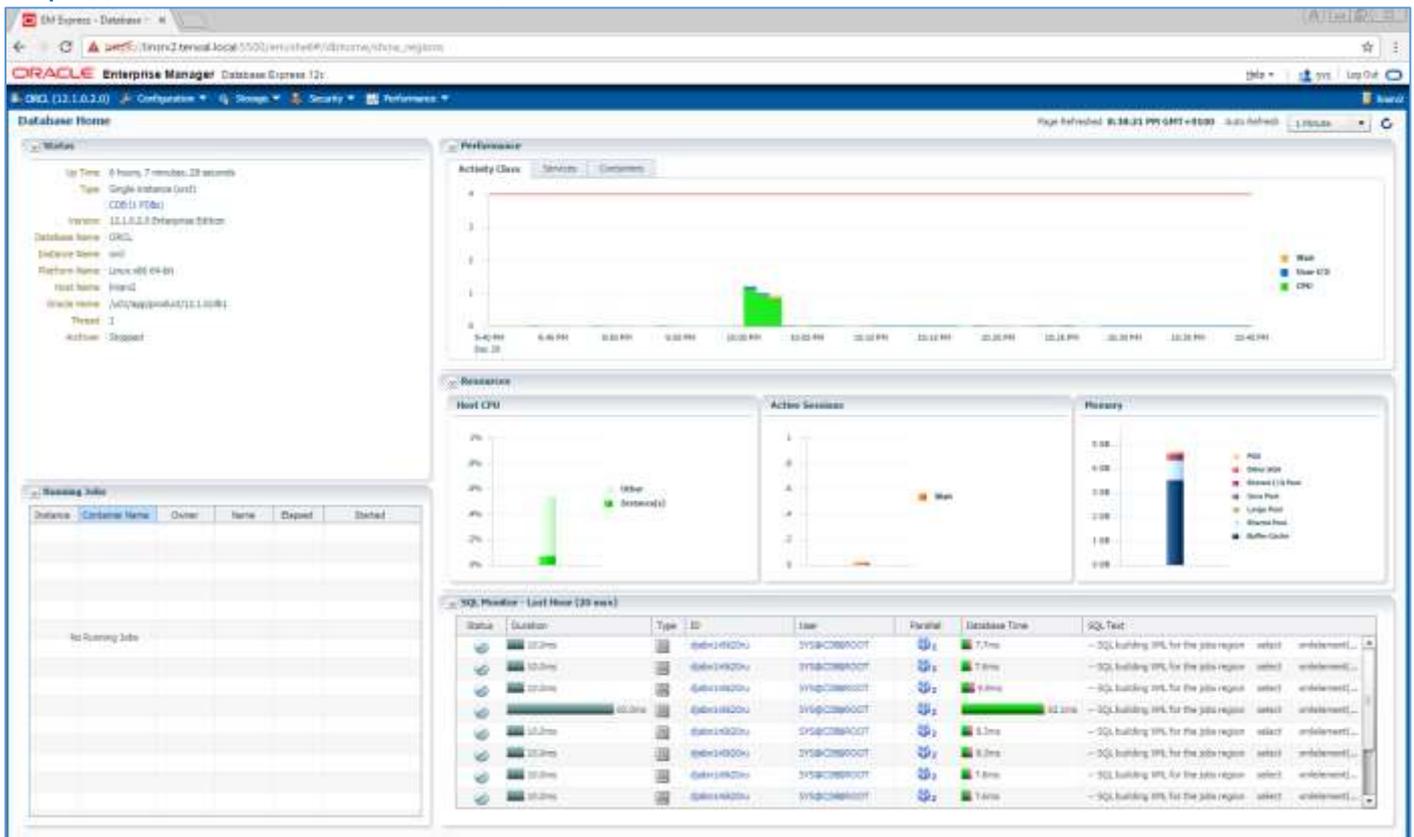
#### EM Express Architecture

EM Express is designed to be lightweight and to incur minimal overhead on the database server. In order to achieve this goal, EM Express is built inside the Oracle Database and only uses internal infrastructure components such as XDB and SQL\*Net. *It does not require any separate middle-tier components.*

Since EM Express is built inside the database, the database **must be open** to use EM Express, and EM Express cannot perform actions outside the database.

EM Express does not have background tasks or processes that periodically collect information. Instead, it utilizes data that is already collected by the database. Data is requested only when the user interacts with the UI and all UI processing is done in the browser, thus minimizing load on the database server.

<https://linsrv2.terwal.local:5500/em>



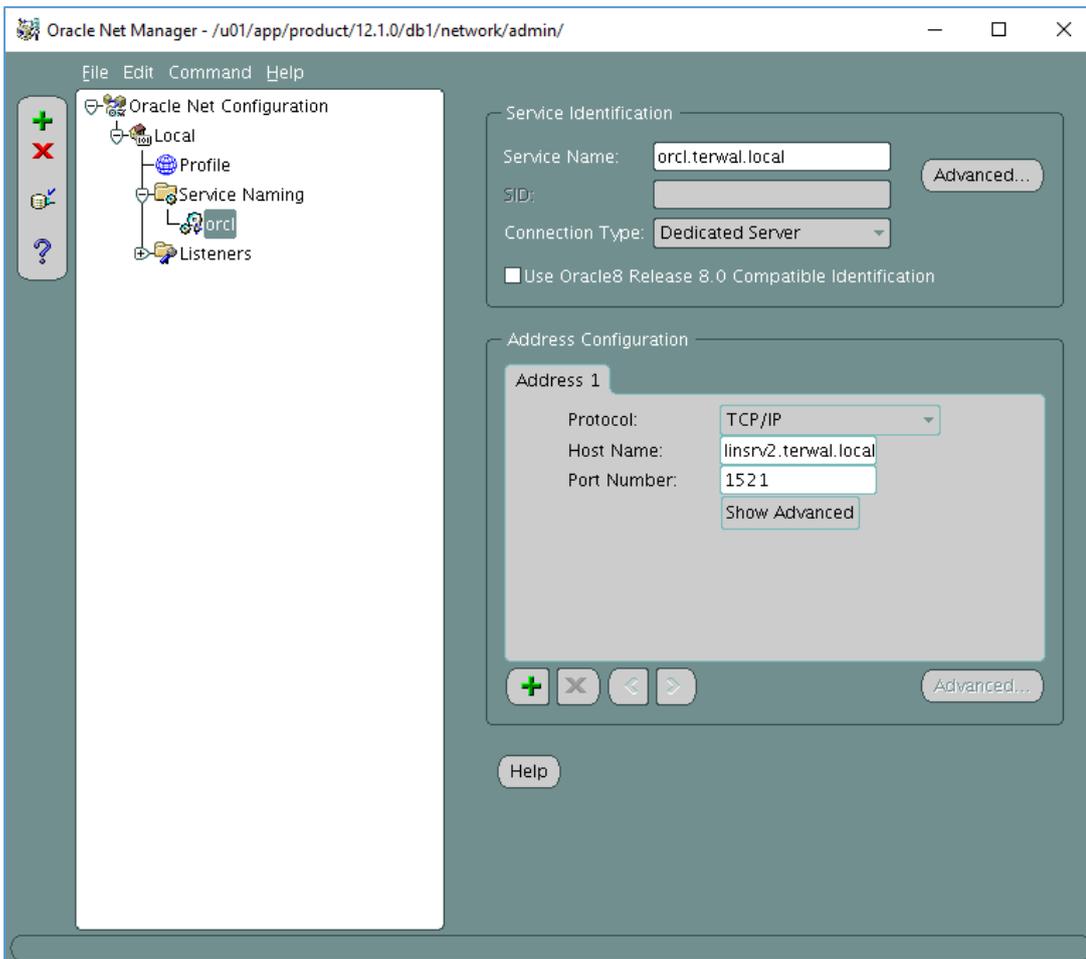
```
linsrv2:~ # su - oracle
oracle@linsrv2:~> lsnrctl status | grep HTTP
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=linsrv2.terwal.local)(PORT=5500))(Security=(my_wallet_directory=/u01/app/admin/orcl/xdb_wallet))(Presentation=HTTP)(Session=RAW))
oracle@linsrv2:~> █
```

On Windows change C:\Windows\System32\drivers\etc => hosts

```
hosts - Notepad
File Edit Format View Help
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com          # source server
#       38.25.63.10      x.acme.com              # x client host
#
# localhost name resolution is handled within DNS itself.
#
#       127.0.0.1         localhost
#       ::1              localhost
192.168.0.41    linsrv2.terwal.local
```

### 3.1.2. Oracle Net Manager

```
linsrv2:~ # su - oracle
oracle@linsrv2:~> netmgr
****DISPLAY environment variable not set!
Oracle Net Manager is a GUI tool which
requires that DISPLAY specify a location
where GUI tools can display.
Set and export DISPLAY, then re-run.
oracle@linsrv2:~> export DISPLAY=192.168.0.20:0.0
oracle@linsrv2:~> netmgr
```



## 3.2. Oracle Net Control Utility

### 3.2.1. Listener Control Utility

```
linsrv2:~ # su - oracle
oracle@linsrv2:~> lsnrctl

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 27-DEC-2016 01:09:58

Copyright (c) 1991, 2014, Oracle. All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> █
```

```
LSNRCTL> help
The following operations are available
An asterisk (*) denotes a modifier or extended command:

start          stop          status        services
version        reload        save_config   trace
spawn          quit          exit          set*
show*
```

### 3.2.2. Connection Manager Control Utility

The Oracle Connection Manager Control utility enables you to administer an Oracle Connection Manager. When you issue commands from the operating system, the basic syntax for this utility is as follows:

```
Cmctl STARTUP -c cman1 -p
```

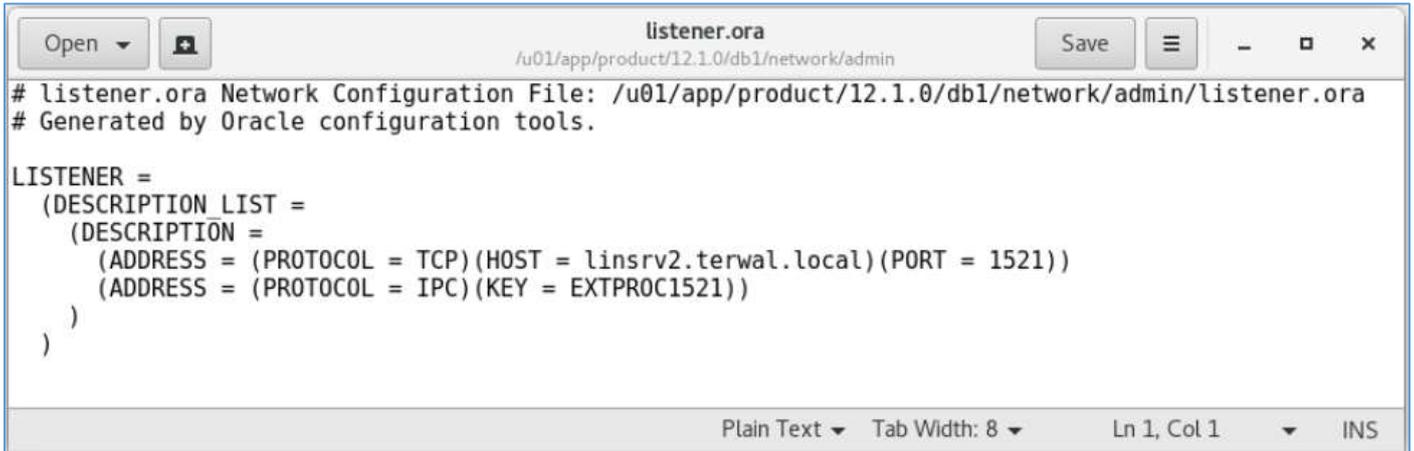
## 4. Configuring and Administering Oracle Net Listener

### 4.1. Configuration Overview

On the database host, the Oracle Net listener (the listener), is a process that listens for client connection requests. It receives incoming client connection requests and manages the traffic of these requests to the database server.

The default listener configuration file is called `listener.ora`, and it is located in the `network/admin` subdirectory of the Oracle home directory. For example, if your Oracle home directory is `/u01/app/oracle/product/11.2.0/dbhome_1`, then the `listener.ora` file is created by default in the `/u01/app/oracle/product/11.2.0/dbhome_1/network/admin` directory.

The file contains a protocol address that identifies the database. This address defines the protocol the listener is listening on and any other protocol-specific information. For example, the listener could be configured to listen at the following protocol address:



The screenshot shows a text editor window titled "listener.ora" with the path "/u01/app/product/12.1.0/db1/network/admin". The window contains the following configuration text:

```
# listener.ora Network Configuration File: /u01/app/product/12.1.0/db1/network/admin/listener.ora
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = linsrv2.terwal.local)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )
```

The status bar at the bottom of the window indicates "Plain Text", "Tab Width: 8", "Ln 1, Col 1", and "INS".

## 4.2. Configuring Service Registration

To ensure service registration works properly, the **initialization parameter file** should contain the following parameters:

- SERVICE\_NAMES for the database service name
- INSTANCE\_NAME for the instance name

For example:

```
SERVICE_NAMES=orcl.terwal.local
INSTANCE_NAME=orcl
```

The value for the **SERVICE\_NAMES** parameter defaults to the global database name, a name comprising the **DB\_NAME** and **DB\_DOMAIN** parameters in the initialization parameter file, entered during installation or database creation. The value for the **INSTANCE\_NAME** parameter defaults to the SID entered during installation or database creation.

Confirm that **database service registration** with the listener has completed using the Listener Control Utility

```
LSNRCTL> services
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=linsrv2.terwal.local) (PORT=1521)))
Services Summary...
Service "orcl.terwal.local" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
  Handler(s):
    "DEDICATED" established:0 refused:0 state:ready
    LOCAL SERVER
Service "orclXDB.terwal.local" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
  Handler(s):
    "D000" established:0 refused:0 current:0 max:1022 state:ready
    DISPATCHER <machine: linsrv2, pid: 10683>
    (ADDRESS=(PROTOCOL=tcp) (HOST=linsrv2.terwal.local) (PORT=41499))
Service "pdborcl.terwal.local" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
  Handler(s):
    "DEDICATED" established:0 refused:0 state:ready
    LOCAL SERVER
The command completed successfully
```

The SERVICES command lists the services supported by the database, along with at least one available service handler. If the database service registration is not listed, then enter the following SQL command:

```
SQL> ALTER SYSTEM REGISTER
```

## 4.3. Listener Administration

### 4.3.1. Starting and Stopping a Listener

```
LSNRCTL> start
Starting /u01/app/product/12.1.0/db1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is /u01/app/product/12.1.0/db1/network/admin/listener.ora
Log messages written to /u01/app/diag/tnslsnr/linsrv2/listener/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=linsrv2.terwal.local) (PORT=1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=linsrv2.terwal.local) (PORT=1521)))
STATUS of the LISTENER
-----
Alias                LISTENER
Version              TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date           28-DEC-2016 23:51:04
Uptime               0 days 0 hr. 0 min. 0 sec
Trace Level          off
Security              ON: Local OS Authentication
SNMP                 OFF
Listener Parameter File /u01/app/product/12.1.0/db1/network/admin/listener.ora
Listener Log File    /u01/app/diag/tnslsnr/linsrv2/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=linsrv2.terwal.local) (PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))
The listener supports no services
The command completed successfully
LSNRCTL> █
```

### 4.3.2. Determining the Current Status of a Listener

```
linsrv2:~ # su - oracle
oracle@linsrv2:~> lsnrctl

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 28-DEC-2016 23:26:59

Copyright (c) 1991, 2014, Oracle. All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> status
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=linsrv2.terwal.local) (PORT=1521)))
TNS-12541: TNS:no listener
TNS-12560: TNS:protocol adapter error
TNS-00511: No listener
Linux Error: 111: Connection refused
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC1521)))
TNS-12541: TNS:no listener
TNS-12560: TNS:protocol adapter error
TNS-00511: No listener
Linux Error: 111: Connection refused
LSNRCTL> █
```

### 4.3.3. Monitoring Services of a Listener

```
LSNRCTL> service
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=linsrv2.terwal.local) (PORT=1521)))
The listener supports no services
The command completed successfully
LSNRCTL> █
```

After starting the Database...

```
LSNRCTL> service
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=linsrv2.terwal.local) (PORT=1521)))
Services Summary...
Service "orcl.terwal.local" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:3 refused:0 state:ready
        LOCAL SERVER
Service "orclXDB.terwal.local" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022 state:ready
        DISPATCHER <machine: linsrv2, pid: 4066>
        (ADDRESS=(PROTOCOL=tcp) (HOST=linsrv2.terwal.local) (PORT=37305))
Service "pdborcl.terwal.local" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:3 refused:0 state:ready
        LOCAL SERVER
The command completed successfully
LSNRCTL> █
```

## 5. Establish A Network and Testing the Connection

### 5.1. Confirming Network Availability

Confirm that the database server computer can communicate itself with a loopback test. A loopback test is a connection from the database server back to itself.

```
linsrv2:~ # tail -3 /etc/hosts
ff02::2      ipv6-allrouters
ff02::3      ipv6-allhosts
192.168.0.41 linsrv2.terwal.local linsrv2
```

Confirm Hardware connectivity

```
linsrv2:~ # ping 192.168.0.41
PING 192.168.0.41 (192.168.0.41) 56(84) bytes of data.
64 bytes from 192.168.0.41: icmp_seq=1 ttl=64 time=0.023 ms
64 bytes from 192.168.0.41: icmp_seq=2 ttl=64 time=0.014 ms
64 bytes from 192.168.0.41: icmp_seq=3 ttl=64 time=0.032 ms
```

Confirm The DNS or host name is configured properly

```
linsrv2:~ # ping linsrv2.terwal.local
PING linsrv2.terwal.local (192.168.0.41) 56(84) bytes of data.
64 bytes from linsrv2.terwal.local (192.168.0.41): icmp_seq=1 ttl=64 time=0.023 ms
64 bytes from linsrv2.terwal.local (192.168.0.41): icmp_seq=2 ttl=64 time=0.016 ms
64 bytes from linsrv2.terwal.local (192.168.0.41): icmp_seq=3 ttl=64 time=0.026 ms
```

Test the TCP/IP setup for the server

```
linsrv2:~ # ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.017 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.026 ms
```

### 5.2. Connecting to a Database

Task 1: Start the Listener

```
lsnrctl
LSNRCTL> STATUS
LSNRCTL> START
```

Task 2: Start the Database

```
oracle@linsrv2:~$ sqlplus /nolog
SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 01:21:31 2016
Copyright (c) 1982, 2016, Oracle. All rights reserved.

SQL> connect sys/change_on_install as sysdba
Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area 496605888 bytes
Fixed Size 2934552 bytes
Variable Size 1029412456 bytes
Database Buffers 3925868544 bytes
Redo Buffers 13840384 bytes
Database mounted.
Database opened.
SQL>
```

```
oracle@linsrv2:~$ sqlplus /nolog
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 28 00:01:04 2016
Copyright (c) 1982, 2016, Oracle. All rights reserved.

SQL> connect system/manager as sysdba
Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area 496605888 bytes
Fixed Size 2934552 bytes
Variable Size 1029412456 bytes
Database Buffers 3925868544 bytes
Redo Buffers 13840384 bytes
Database mounted.
Database opened.
SQL>
```

## 6. Client Connections

The client uses a connect descriptor to specify the database to which it wants to connect. This connect descriptor contains a protocol and a database service name. A database can have multiple service names defined, so a specific service name must be specified for the connect descriptor. In a preconfigured database, there is only one service name, which defaults to the global database name.

The following example shows a connect descriptor that enables clients to connect to a database with service name mydb.us.example.com:

There are several methods to connect to an Oracle database.

### 6.1. From the command line

The general form of connecting an application to a database server from the command line is:

```
oracle@linsrv2:~> sqlplus system@orcl

SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 29 00:11:22 2016

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Enter password:
Last Successful login time: Thu Dec 29 2016 00:08:19 +01:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing opt
ions

SQL> █
```

### 6.2. From a login screen

### 6.3. From a 3GL application

### 6.4. From within SQL\*Plus

```
oracle@linsrv2:~> sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 29 00:13:06 2016

Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> connect sys/oracle as sysdba
Connected.
SQL> █
```