

8

Configuring the Oracle Network Environment

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Glauber Soares (glauber.soares@live.com) has a non-transferable license to use this Student Guide.

Objectives

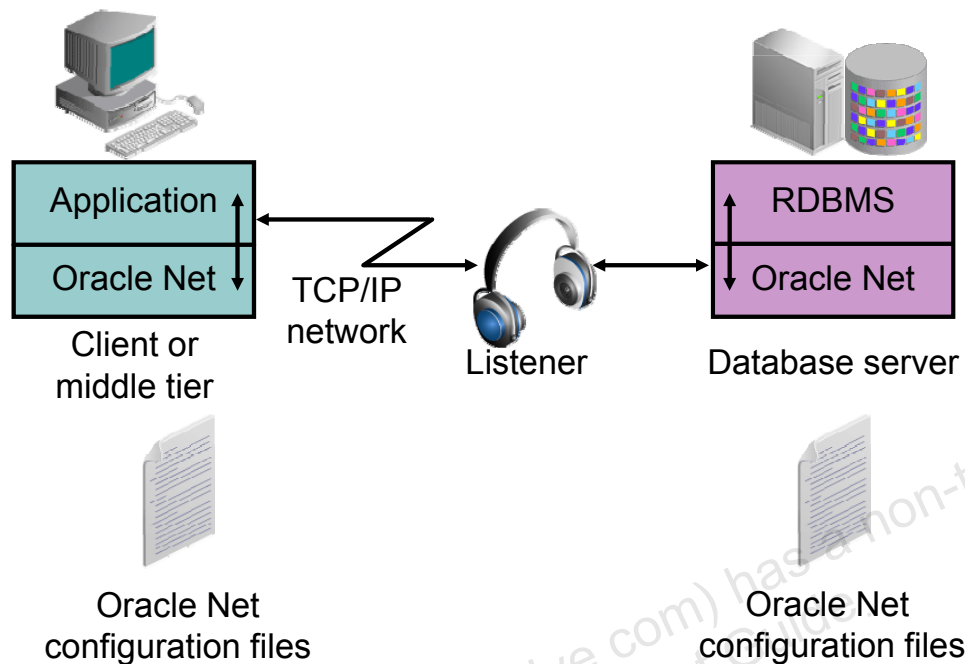
After completing this lesson, you should be able to:

- Use Enterprise Manager Cloud Control and Oracle Net Manager to:
 - Create additional listeners
 - Create Oracle Net Service aliases
 - Control Oracle Net Listener
- Use the Listener Control Utility to manage Oracle Net Listener
- Use `tnsping` to test Oracle Net connectivity
- Identify when to use shared servers and when to use dedicated servers

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Net Services: Overview



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

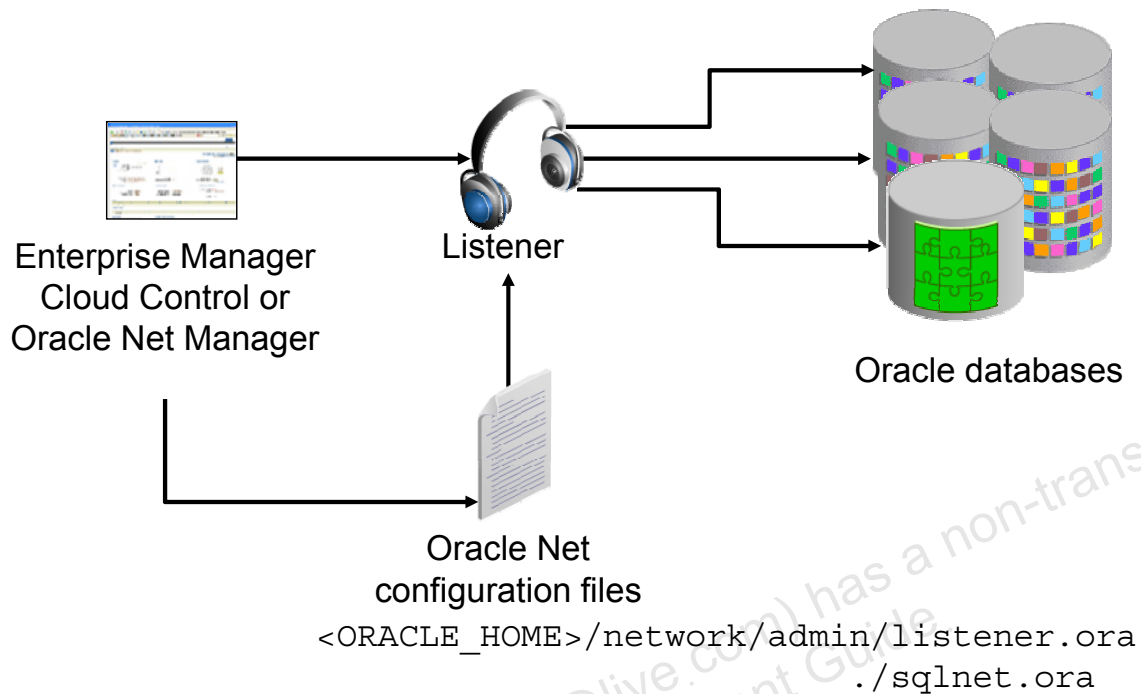
Oracle Net Services enables network connections from a client or middle-tier application to the Oracle server. After a network session is established, Oracle Net acts as the data courier for both the client application and the database server. It is responsible for establishing and maintaining the connection between the client application and database server, as well as exchanging messages between them. Oracle Net (or something that simulates Oracle Net, such as Java Database Connectivity) is located on each computer that needs to talk to the database server.

On the client computer, Oracle Net is a background component for application connections to the database.

On the database server, Oracle Net includes an active process called *Oracle Net Listener*, which is responsible for coordinating connections between the database and external applications.

The most common use of Oracle Net Services is to allow incoming database connections. You can configure additional net services to allow access to external code libraries (EXTPROC) and to connect the Oracle instance to non-Oracle data sources (such as Sybase, Informix, DB2, and SQL Server) through Oracle Heterogeneous Services.

Oracle Net Listener: Overview



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Net Listener (or simply *the listener*) is the gateway to the Oracle instance for all nonlocal user connections. A single listener can service multiple database instances and thousands of client connections.

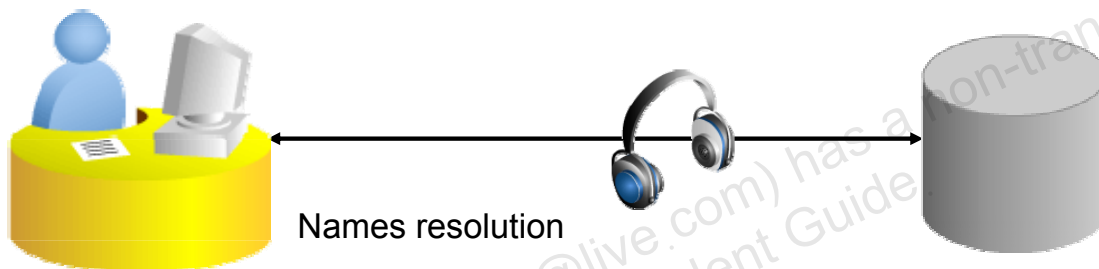
You can use Enterprise Manager Cloud Control or Oracle Net Manager to configure the listener and specify log file locations.

Advanced administrators can also configure Oracle Net Services by manually editing the configuration files, if necessary, with a standard operating system (OS) text editor such as `vi` or `gedit`.

Establishing Oracle Network Connections

To make a client or middle-tier connection, Oracle Net requires the client to know the:

- Host where the listener is running
- Port that the listener is monitoring
- Protocol that the listener is using
- Name of the service that the listener is handling



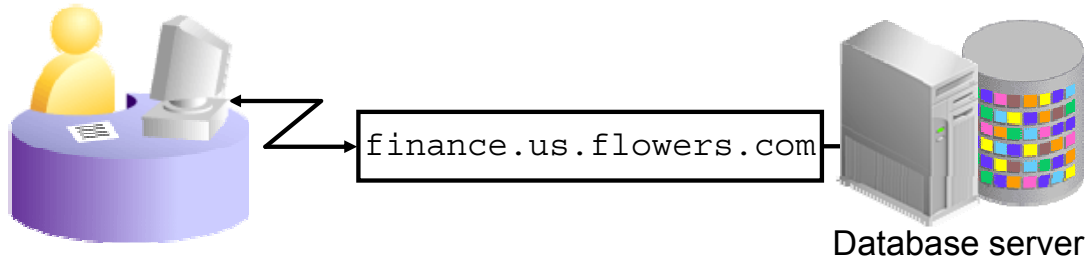
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

For an application to connect to a service through Oracle Net Listener, the application must have information about that service, including the address or host where the listener resides, the protocol that the listener accepts, and the port that the listener monitors. After the listener is located, the final piece of information that the application needs is the name of the service to which it wants to connect.

Oracle Net names resolution is the process of determining this connection information.

Connecting to an Oracle Database



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

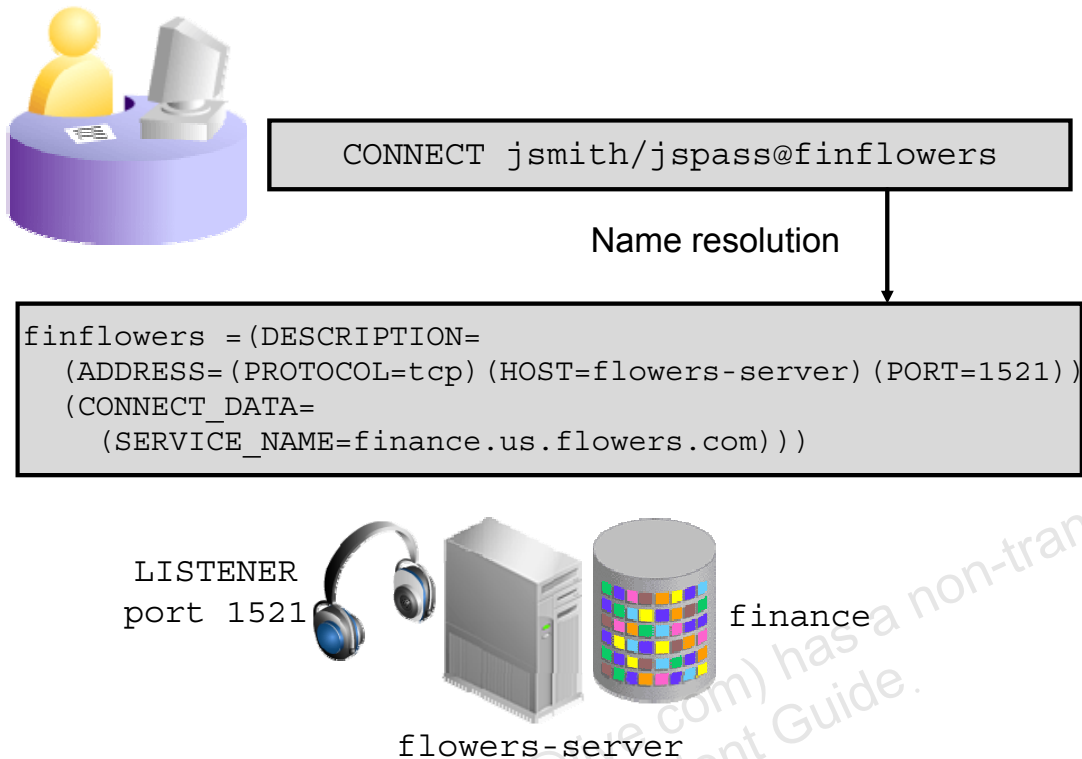
An Oracle database is represented to a client as a service. A database can have one or more services associated with it. Databases are identified by a *service name* that is specified by the `SERVICE_NAMES` parameter in the initialization parameter file. The service name defaults to the global database name, which is a name that comprises the database name (`DB_NAME` parameter value) and the domain name (`DB_DOMAIN` parameter value).

To connect to a database service, clients use a *connect descriptor* that provides the location of the database and the name of the database service. Clients can use the connect descriptor or a name that resolves to the connect descriptor (as discussed later in this lesson).

The following example shows a connect descriptor that enables clients to connect to a database service called `finance.us.flowers.com`.

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

Name Resolution



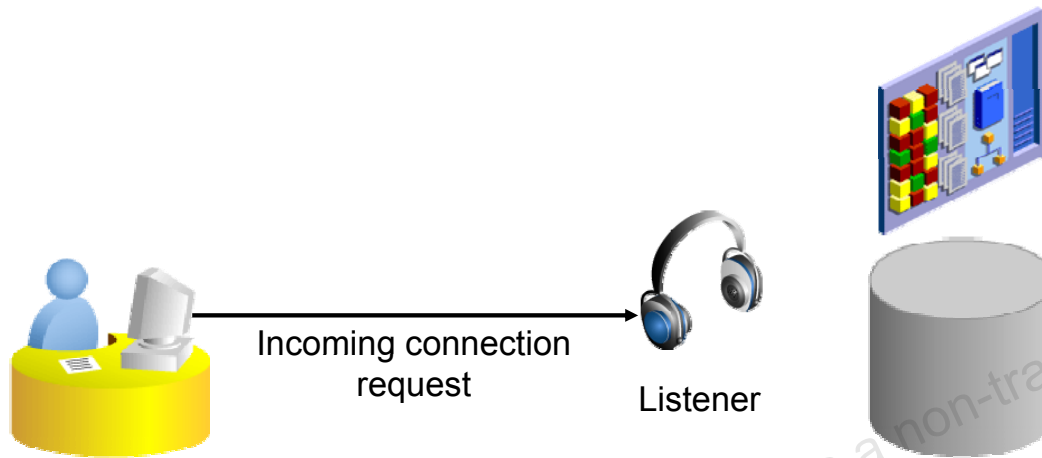
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Users initiate a connection request to the Oracle database by sending a *connect string*. A connect string includes a username and password, along with a *connect identifier*. A connect identifier can be the connect descriptor itself or a *name* that resolves to a connect descriptor. One of the most common connect identifiers is a *net service name*, which is a simple name for a service.

When a net service name is used, connection processing takes place by mapping the net service name to a connect descriptor. The mapping information can be stored in one or more repositories of information and is resolved using a *naming method*.

Establishing a Connection

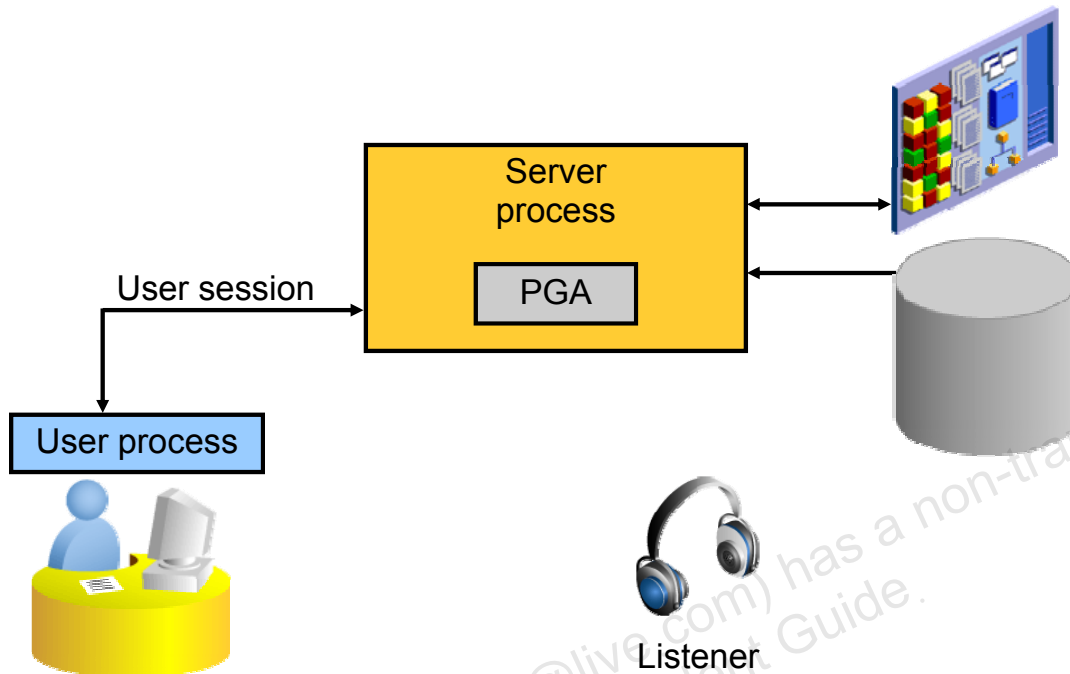
**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After Oracle Net names resolution is complete, a connection request is passed from the user or middle-tier application (hereafter referred to as the *user process*) to the listener. The listener receives a `CONNECT` packet and checks whether that `CONNECT` packet is requesting a valid Oracle Net service name.

If the service name is not requested (as in the case of a `tnsping` request), the listener acknowledges the connect request and does nothing else. If an invalid service name is requested, the listener transmits an error code to the user process.

User Sessions



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If the `CONNECT` packet requests a valid service name, the listener spawns a new process to deal with the connection. This new process is known as the *server process*. The listener connects to the process and passes the initialization information, including the address information for the user process. At this point, the listener no longer deals with the connection and all work is passed to the server process.

The server process checks the user's authentication credentials (usually a password), and if the credentials are valid, a user session is created.

Dedicated server process: With the session established, the server process now acts as the user's agent on the server. The server process is responsible for:

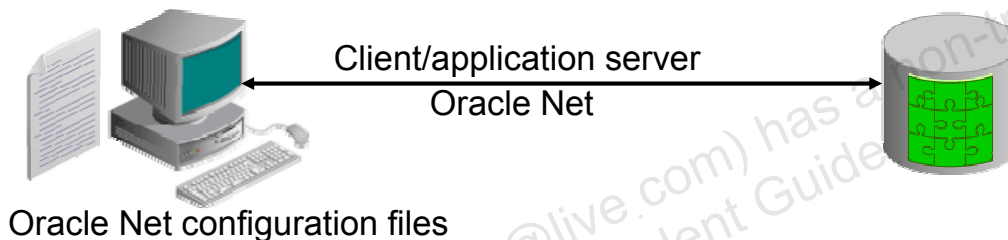
- Parsing and running any SQL statements issued through the application
- Checking the database buffer cache for data blocks required to perform SQL statements
- Reading necessary data blocks from data files on the disk into the database buffer cache portion of the System Global Area (SGA), if the blocks are not already present in the SGA
- Managing all sorting activity. The Sort Area is a memory area that is used to work with sorting; it is contained in a portion of memory that is associated with the Program Global Area (PGA).

- Returning results to the user process in such a way that the application can process the information
- Reading auditing options and reporting user processes to the audit destination

Naming Methods

Oracle Net supports several methods of resolving connection information:

- Easy connect naming: Uses a TCP/IP connect string
- Local naming: Uses a local configuration file
- Directory naming: Uses a centralized LDAP-compliant directory server
- External naming: Uses a supported non-Oracle naming service



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Net provides support for the following naming methods:

- **Easy connect naming:** The easy connect naming method enables clients to connect to an Oracle Database server by using a TCP/IP connect string consisting of a host name and optional port and service name as follows:

```
CONNECT username/password@host[:port] [/service_name]
```

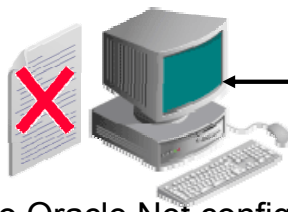
The easy connect naming method requires no configuration.

- **Local naming:** The local naming method stores connect descriptors (identified by their net service name) in a local configuration file named `tnsnames.ora` on the client.
- **Directory naming:** To access a database service, the directory naming method stores connect identifiers in a centralized directory server that is compliant with the Lightweight Directory Access Protocol (LDAP).
- **External naming:** The external naming method stores net service names in a supported non-Oracle naming service. Supported third-party services include:
 - Network Information Service (NIS) External Naming
 - Distributed Computing Environment (DCE) Cell Directory Services (CDS)

Easy Connect

- Is enabled by default
- Requires no client-side configuration
- Supports only TCP/IP (no SSL)
- Offers no support for advanced connection options such as:
 - Connect-time failover
 - Source routing
 - Load balancing

```
SQL> CONNECT hr/hr@db.us.oracle.com:1521/dba11g
```



No Oracle Net configuration files

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With Easy Connect, you supply all information that is required for the Oracle Net connection as part of the connect string. Easy Connect connection strings take the following form:

```
<username>/<password>@<hostname>:<listener port>/<service name>
```

The listener port and service name are optional. If the listener port is not provided, Oracle Net assumes that the default port of 1521 is being used. If the service name is not provided, Oracle Net assumes that the database service name and host name provided in the connect string are identical.

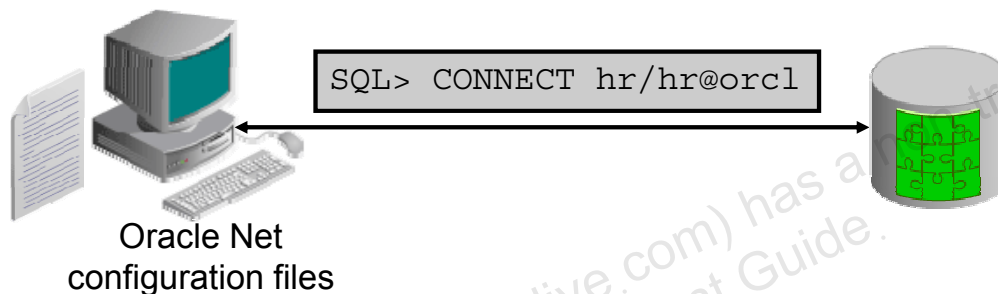
Assuming that the listener uses TCP to listen on port 1521 and the `SERVICE_NAMES=db` and `DB_DOMAIN=us.oracle.com` instance parameters, the connect string shown in the slide can be shortened:

```
SQL> connect hr/hr@db.us.oracle.com
```

Note: The `SERVICE_NAMES` initialization parameter can accept multiple comma-separated values. Only one of those values must be `db` for this scenario to work.

Local Naming

- Requires a client-side names-resolution file
- Supports all Oracle Net protocols
- Supports advanced connection options such as:
 - Connect-time failover
 - Source routing
 - Load balancing



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With local naming, the user supplies an alias for the Oracle Net service. Oracle Net checks the alias against a local list of known services and, if it finds a match, converts the alias into host, protocol, port, and service name.

One advantage of local naming is that the database users need to remember only a short alias rather than the long connect string required by Easy Connect.

The local list of known services is stored in the following text configuration file:

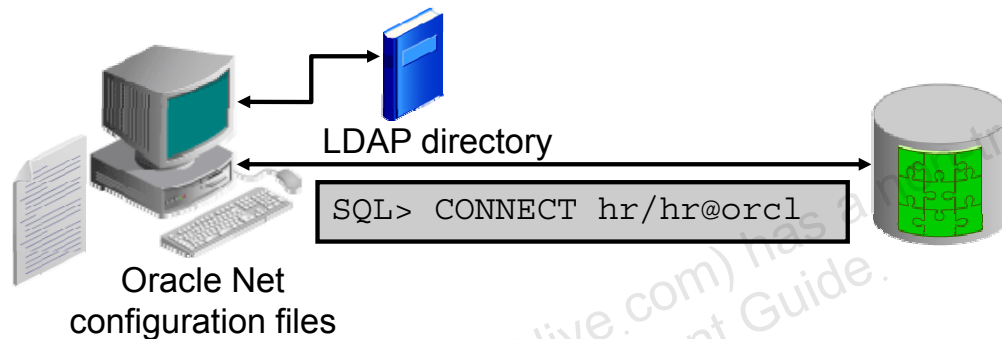
```
<oracle_home>/network/admin/tnsnames.ora
```

This is the default location of the `tnsnames.ora` file, but the file can be located elsewhere using the `TNS_ADMIN` environment variable.

Local naming is appropriate for organizations in which Oracle Net service configurations do not change often.

Directory Naming

- Requires LDAP with Oracle Net names resolution information loaded:
 - Oracle Internet Directory
 - Microsoft Active Directory Services
- Supports all Oracle Net protocols
- Supports advanced connection options



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

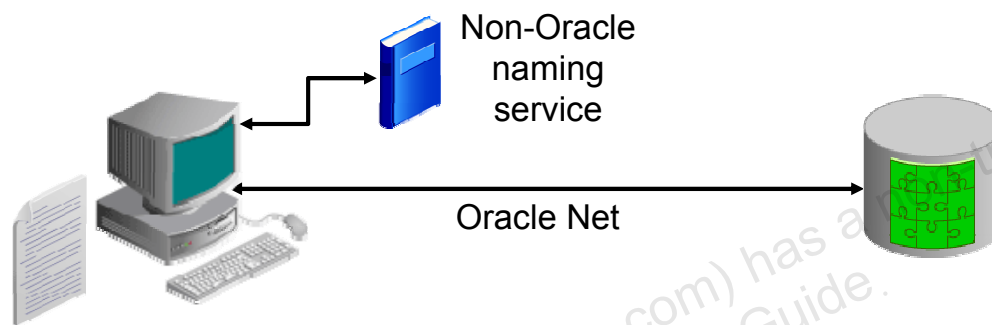
With directory naming, the user supplies an alias for the Oracle Net service. Oracle Net checks the alias against an external list of known services and, if it finds a match, converts the alias into host, protocol, port, and service name. Like local naming, database users need to remember only a short alias.

One advantage of directory naming is that the service name is available for users to connect with as soon as a new service name is added to the LDAP directory. With local naming, the database administrator (DBA) must first distribute updated `tnsnames.ora` files containing the changed service name information before users can connect to new or modified services.

Directory naming is appropriate for organizations in which Oracle Net service configurations change frequently.

External Naming Method

- Uses a supported non-Oracle naming service
- Includes:
 - Network Information Service (NIS) External Naming
 - Distributed Computing Environment (DCE) Cell Directory Services (CDS)

**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The external naming method stores Net Service names in a supported non-Oracle naming service. Supported third-party services include:

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

Conceptually, external naming is similar to directory naming.

Tools for Configuring and Managing Oracle Net Services

- Enterprise Manager Net Services Administration page
- Oracle Net Manager
- Oracle Net Configuration Assistant
- Listener Control Utility

**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Use the following tools and applications to manage your Oracle Network configuration:

- **Enterprise Manager Cloud Control:** Provides an integrated environment for configuring and managing Oracle Net Services. Use Enterprise Manager to configure Oracle Net Services for any Oracle home across multiple file systems and to administer listeners.
- **Oracle Net Manager:** Provides a graphical user interface (GUI) through which you can configure Oracle Net Services for an Oracle home on a local client or a server host.
- **Oracle Net Configuration Assistant:** Launched by Oracle Universal Installer when you install the Oracle software. During a typical database installation, Oracle Net Configuration Assistant automatically configures a listener called LISTENER that has a TCP/IP listening protocol address for the database. If you perform a custom installation, Oracle Net Configuration Assistant prompts you to configure a listener name and protocol address of your choice.
- **Listener Control Utility:** Used to start, stop, and view the status of the listener process

Defining Oracle Net Services Components

Component	Description	File
Listeners	A process that resides on the server whose responsibility is to listen for incoming client connection requests and manage the traffic to the server.	<code>listener.ora</code>
Naming methods	A resolution method used by a client application to resolve a connect identifier to a connect descriptor when attempting to connect to a database service.	
Naming (net service name)	A simple name (connect identifier) for a service that resolves to a connect descriptor to identify the network location and identification of a service.	<code>tnsnames.ora</code> (local configuration)
Profiles	A collection of parameters that specifies preferences for enabling and configuring Oracle Net features on the client or server.	<code>sqlnet.ora</code>



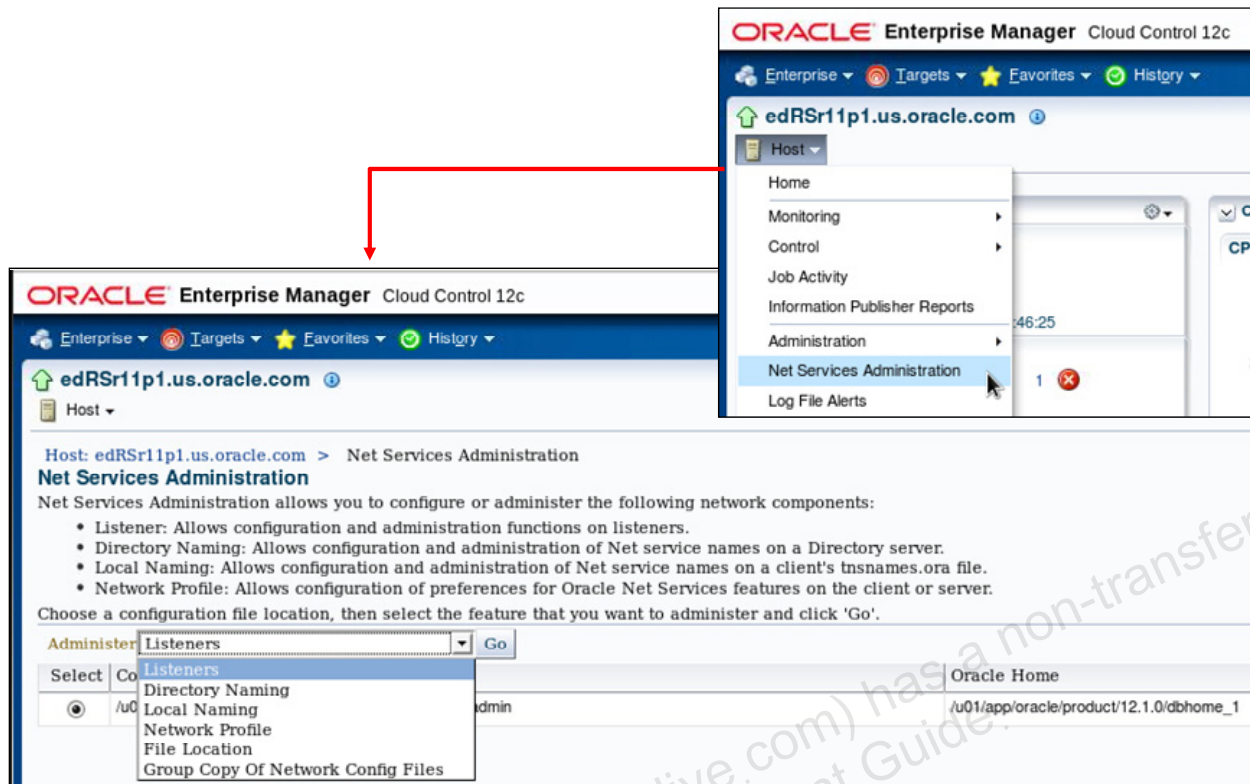
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The following Oracle Net Services components can be configured by using Enterprise Manager Cloud Control and Oracle Net Manager:

- Listener: Configuration of the listener includes specifying the listener name, protocol addresses it is accepting connection requests on, and services (database or non-database service) it is listening for.
- Naming (net service name)
- Naming methods
- Profiles

The Oracle Net Configuration Assistant configures the listener, naming methods, directory server usage, and a local `tnsnames.ora` file during installation of Oracle Database software.

Using Enterprise Manager Cloud Control



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

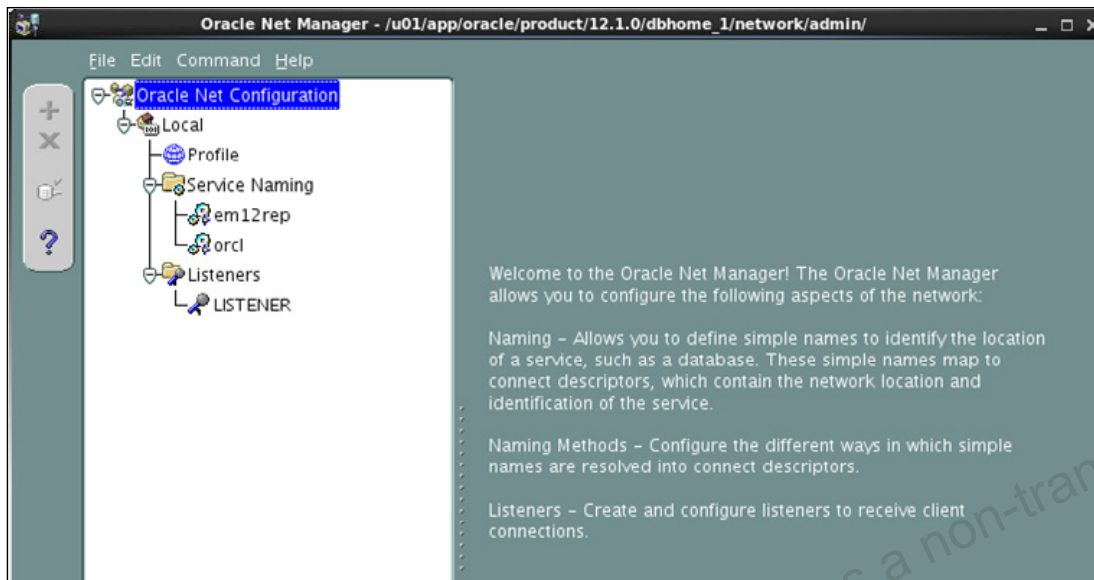
To administer Oracle Net Services in Enterprise Manager Cloud Control, select Net Services Administration in the Host menu for your target host.

The Net Services Administration page enables you to configure Oracle Net Services for any Oracle home across multiple file systems.

It also provides common administration functions for listeners such as starting and stopping a listener, and changing its tracing and logging characteristics. You can also look at a listener's control status report.

You can also specify the `ORACLE_HOME` location where the Oracle Net Services configuration files are stored by selecting "File Location."

Using Oracle Net Manager



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Use Oracle Net Manager to configure Oracle Net Services for an Oracle home on a local client or server host.

You can invoke Oracle Net Manager in the following ways:

- On Linux, enter `netmgr` at the operating system prompt.
- On Microsoft Windows, select:
 - Programs from the Start menu
 - Oracle - `HOME_NAME`
 - Configuration and Migration Tools
 - Net Manager

Using Oracle Net Configuration Assistant

**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Net Configuration Assistant is executed by the Oracle Universal Installer during installation of Oracle Database software. It configures the following basic network components:

- Listener names and protocol addresses
- Naming methods
- Net service names
- Directory server usage: Configures a directory server for directory-enabled features

Using the Listener Control Utility

```
$ lsnrctl
LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 09-JUL-2013 08:47:42

Copyright (c) 1991, 2013, Oracle. All rights reserved.

Welcome to LSNRCTL, type "help" for information.

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*
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Listener Control Utility enables you to control the listener. With `lsnrctl`, you can:

- Start the listener
- Stop the listener
- Check the status of the listener
- Reinitialize the listener from the configuration file parameters
- Dynamically configure many listeners
- Change the listener password

The basic command syntax for this utility is:

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

When the `lsnrctl` command is issued, the command acts on the default listener (named `LISTENER`) unless a different listener name is specified or the `SET CURRENT_LISTENER` command is executed. If the listener name is `LISTENER`, the `listener_name` argument can be omitted. The valid commands for `lsnrctl` are shown in the slide.

Note: The `lsnrctl` utility is located in both the Grid Infrastructure home and the Oracle Database home. It is important to set the environment variables to the appropriate home before using it.

Listener Control Utility Syntax

Commands for the Listener Control Utility can be issued from the command line or from the `lsnrctl` prompt.

- Command-line syntax:

```
$ lsnrctl <command name>
$ lsnrctl start
$ lsnrctl status
```

- Prompt syntax:

```
LSNRCTL> <command name>
LSNRCTL> start
LSNRCTL> status
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `lsnrctl` commands can be issued from within the utility (prompt syntax) or from the command line. The following two commands have the same effect but use command-line syntax and prompt syntax, respectively:

Command-line syntax:

```
$ lsnrctl start
```

Prompt syntax:

```
$ lsnrctl
LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 09-JUL-2013
08:47:42
Copyright (c) 1991, 2013, Oracle. All rights reserved.
Welcome to LSNRCTL, type "help" for information.
LSNRCTL> start
```

The command-line syntax is typically used to execute an individual command or scripted commands. If you plan to execute several consecutive `lsnrctl` commands, the prompt syntax is more efficient. Note that the `listener_name` argument is omitted, and the command would thus affect the listener named `LISTENER`.

Remember that if your listener is named something other than `LISTENER`, you must either include the listener name with the command or use the `SET CURRENT_LISTENER` command. Suppose that your listener is named `custom_lis`. Here are two examples of stopping a listener named `custom_lis` by using prompt syntax:

```
LSNRCTL> stop custom_lis
Connecting to
 (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=host01) (PORT=5521)))
The command completed successfully
```

This produces the same results as the following:

```
LSNRCTL> set cur custom_lis
Current Listener is custom_lis
LSNRCTL> stop
Connecting to
 (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=host01) (PORT=5521)))
The command completed successfully
```

Note: In the preceding syntax, `current_listener` has been abbreviated to `cur`.

Using command-line syntax produces the same results:

```
$ lsnrctl stop custom_lis
LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 09-JUL-
2013 08:47:42
Copyright (c) 1991, 2013, Oracle. All rights reserved.
Connecting to
 (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=host01) (PORT=5521)))
The command completed successfully
```

Advanced Connection Options

Oracle Net supports the following advanced connection options with local and directory naming:

- Connect-time failover
- Load balancing
- Source routing

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, is positioned on the right side of a red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When a database service is accessible by multiple listener protocol addresses, you can specify the order in which the addresses are to be used. The addresses can be chosen randomly or tried sequentially. In cases in which more than one listener is available, such as Oracle Real Application Clusters (RAC) configurations, Oracle Net can take advantage of listener failover and load balancing as well as Oracle Connection Manager source routing.

With *connect-time failover* enabled, the alias has two or more listener addresses listed. If the first address is not available, the second is tried. Oracle Net keeps trying addresses in the listed order until it reaches a listener that is functioning or until all addresses have been tried and failed. Transparent Application Failover (TAF) is a client-side feature that allows clients to reconnect to surviving databases in the event of a database instance failure. Notifications are used by the server to trigger TAF callbacks on the client side.

With *load balancing* enabled, Oracle Net picks an address at random from the list of addresses. The runtime connection load-balancing feature improves connection performance by balancing the number of active connections among multiple dispatchers. In a RAC environment, connection pool load balancing also has the capability to balance the number of active connections among multiple instances.

Source routing is used with Oracle Connection Manager, which serves as a proxy server for Oracle Net traffic, enabling Oracle Net traffic to be routed securely through a firewall. Oracle Net treats the addresses as a list of relays, connecting to the first address and then requesting to be passed from the first to the second until the destination is reached. It differs from failover or load balancing in that all addresses are used each time a connection is made.

Testing Oracle Net Connectivity

The `tnsping` utility that tests Oracle Net service aliases:

- Ensures connectivity between the client and the Oracle Net Listener
- Does not verify that the requested service is available
- Supports Easy Connect Names Resolution:

```
tnsping host01.example.com:1521/orcl
```

- Supports local and directory naming:

```
tnsping orcl
```

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, is positioned on the right side of a red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

`tnsping` is the Oracle Net equivalent of the TCP/IP ping utility. It offers a quick test to verify that the network path to a destination is good. For example, enter `tnsping orcl` in a command-line window.

The utility validates that the host name, port, and protocol reach a listener. It does not actually check whether the listener handles the service name. The `tnsping` utility also reveals the location of the configuration files. In a system with multiple `ORACLE_HOME` locations, this can be helpful.

Comparing Dedicated Server and Shared Server Configurations

- Dedicated server configuration: One server process for each client
- Shared server configuration: A small pool of server processes can serve a large number of clients

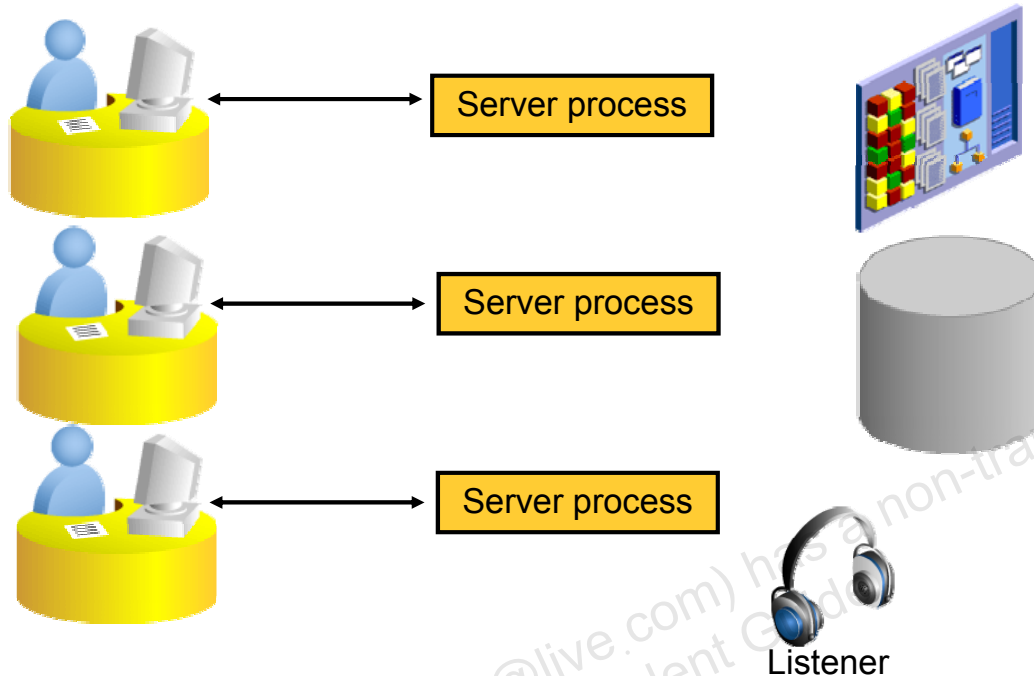
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In a dedicated server configuration a server process handles requests for a single client process. A shared server configuration enables multiple client processes to share a small number of server processes. Detailed information about each configuration follows in this lesson.

User Sessions: Dedicated Server Process

User sessions



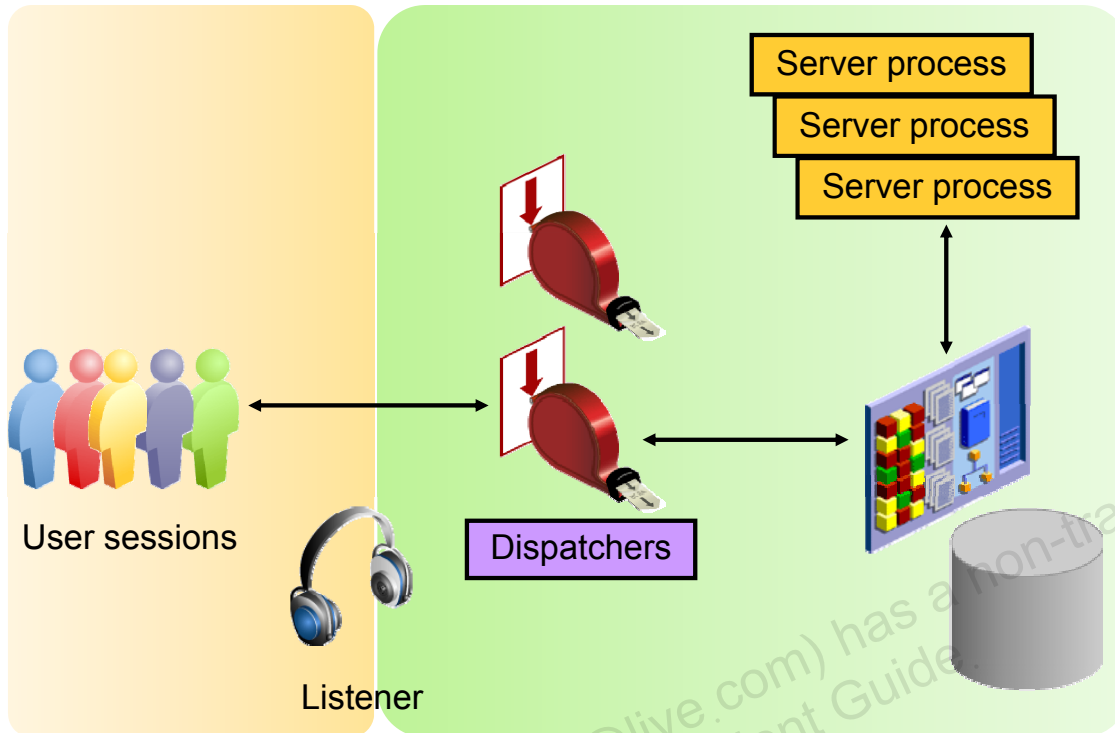
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With dedicated server processes, there is a one-to-one ratio of server processes to user processes. Each server process uses system resources, including CPU cycles and memory. In a heavily loaded system, the memory and CPU resources that are used by dedicated server processes can be prohibitive and can negatively affect the system's scalability. If your system is being negatively affected by the resource demands of the dedicated server architecture, you have the following options:

- Increasing system resources by adding more memory and additional CPU capability
- Using the Oracle Shared Server Process architecture

User Sessions: Shared Server Processes



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Each service that participates in the shared server process architecture has at least one dispatcher process (and usually more). When a connection request arrives, the listener does not spawn a dedicated server process. Instead, the listener maintains a list of dispatchers that are available for each service name, along with the connection load (number of concurrent connections) for each dispatcher.

Connection requests are routed to the lightest loaded dispatcher that is servicing a given service name. Users remain connected to the same dispatcher for the duration of a session.

Unlike dedicated server processes, a single dispatcher can manage hundreds of user sessions.

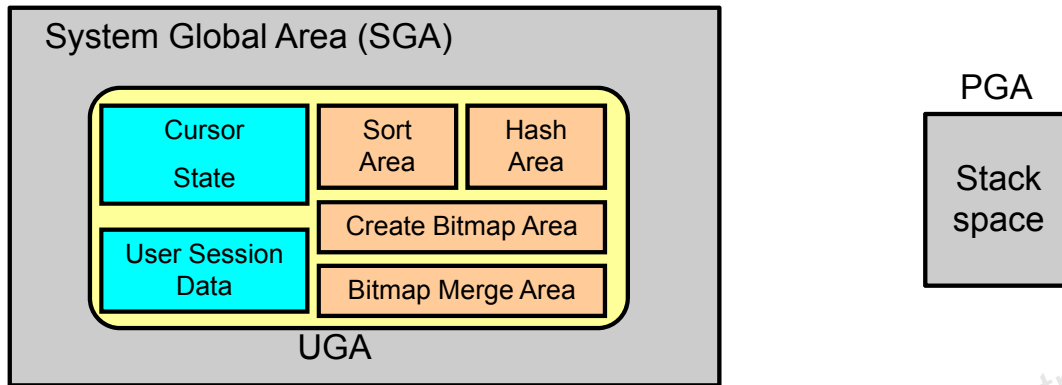
Dispatchers do not actually handle the work of user requests. Instead, they pass user requests to a common queue located in the shared pool portion of the SGA.

Shared server processes take over most of the work of dedicated server processes, pulling requests from the queue and processing them until they are complete.

Because a user session may have requests processed by multiple shared server processes, most of the memory structures that are usually stored in the PGA must be in a shared memory location (by default, in the shared pool). However, if the large pool is configured or if `SGA_TARGET` is set for automatic memory management, these memory structures are stored in the large pool portion of the SGA.

SGA and PGA Usage

Oracle Shared Server: User session data is held in the SGA.



Remember to consider shared server memory requirements when sizing the SGA.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The contents of the SGA and PGA differ when dedicated servers or shared servers are used:

- Text and parsed forms of all SQL statements are stored in the SGA.
- The cursor state contains runtime memory values for the SQL statement, such as rows retrieved.
- User-session data includes security and resource usage information.
- The stack space contains local variables for the process.

Technical Note

The change in the SGA and PGA is transparent to the user; however, if you are supporting multiple users, you need to increase the `LARGE_POOL_SIZE` initialization parameter. Each shared server process must access the data spaces of all sessions so that any server can handle requests from any session. Space is allocated in the SGA for each session's data space. You limit the amount of space that a session can allocate by setting the `PRIVATE_SGA` resource.

Shared Server Configuration Considerations

Certain types of database work must not be performed using shared servers:

- Database administration
- Backup and recovery operations
- Batch processing and bulk load operations
- Data warehouse operations



Dispatcher



Dedicated server
process

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Oracle Shared Server architecture is an efficient process and memory use model, but it is not appropriate for all connections. Because of the common request queue and the fact that many users may share a dispatcher response queue, shared servers do not perform well with operations that must deal with large sets of data, such as warehouse queries or batch processing.

Backup and recovery sessions that use Oracle Recovery Manager (discussed in later lessons) also deal with very large data sets and must use dedicated connections.

Many administration tasks must not (and cannot) be performed by using shared server connections. These include starting up and shutting down the instance, creating tablespaces and data files, maintaining indexes and tables, analyzing statistics, and many other tasks that are commonly performed by the DBA. All DBA sessions must choose dedicated servers.

Configuring Communication Between Databases

- Sending data or messages between sites requires network configuration on both sites.
- You must configure the following:
 - Network connectivity (for example, `tnsnames.ora`)
 - Database links

```
CREATE DATABASE LINK <remote_global_name>  
CONNECT TO <user> IDENTIFIED BY <pwd>  
USING '<connect_string_for_remote_db>';
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A database link is a schema object in one database that enables you to access objects on another database. The other database need not be an Oracle database system. However, to access non-Oracle systems, you must use Oracle Heterogeneous Services.

To create a private database link, you must have the `CREATE DATABASE LINK` system privilege. To create a public database link, you must have the `CREATE PUBLIC DATABASE LINK` system privilege.

When an application uses a database link to access a remote database, Oracle Database establishes a database session in the remote database on behalf of the local request. The `CONNECT TO` clause that is used in creating a database link determines how the connection is established on the remote database. You can create fixed user, current user, and connected user database links. Current user links are available only through the Oracle Advanced Security option. The example in the slide shows the syntax to create a fixed user database link.

After you create a database link, you can use it to refer to tables and views on the other database. In SQL statements, you can refer to a table or view on the other database by appending `@dblink` to the table or view name. You can query a table or view on the other database or use any `INSERT`, `UPDATE`, `DELETE`, or `LOCK TABLE` statement for the table.

Connecting to Another Database

```
REMOTE_ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)
      (HOST = host02.example.com)
      (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl2.example.com)
    )
  )
```

tnsnames.ora

```
CONNECT hr/hr@orcl1;

CREATE DATABASE LINK remote
CONNECT TO HR IDENTIFIED BY HR
USING 'REMOTE_ORCL';

SELECT * FROM employees@remote
```

SQL*Plus

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide shows the `tnsnames.ora` entry that is needed before creating a database link. The example shows a fixed user database link called `REMOTE` that is connecting to the user `HR` by using the connect string `REMOTE_ORCL`. After you create a database link, you can use it to refer to tables and views on the other database.

The description of the view is as follows:

```
SQL> DESC DBA_DB_LINKS
```

Name	Null?	Type
OWNER	NOT NULL	VARCHAR2(30)
DB_LINK	NOT NULL	VARCHAR2(128)
USERNAME		VARCHAR2(30)
HOST		VARCHAR2(2000)
CREATED	NOT NULL	DATE

```
SQL> select owner, db_link, username from dba_db_links;
```

OWNER	DB_LINK	USERNAME
HR	REMOTE.EXAMPLE.COM	HR

Quiz

Which configuration files are used to configure the listener?

- a. listener.ora
- b. listener.conf
- c. tnsnames.ora
- d. tnsnames.conf
- e. sqlnet.ora
- f. sqlnet.conf

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, e

Quiz

When using the shared server process architecture, the UGA portion of the PGA is relocated into the SGA.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Summary

In this lesson, you should have learned how to:

- Use Enterprise Manager to:
 - Create additional listeners
 - Create Oracle Net Service aliases
 - Control the Oracle Net Listener
- Use `tnsping` to test Oracle Net connectivity
- Identify when to use shared servers and when to use dedicated servers

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice 8

- 8-1: Configuring the Oracle Network to Access a Database
- 8-2: Creating an Alternative Listener

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Glauber Soares (glaubersantos@live.com) has a non-transferable license to use this Student Guide.