CHAPTER 5

Configuring Net8

✦ ✦ ✦ ✦

In This Chapter

✦ ✦ ✦ ✦

It's a networked world out there, and Oracle is smack dab in the middle of it. Not only do client-server applications hit Oracle databases, but databases also communicate with each other, and application servers execute database queries on behalf of Internet and intranet clients.

In this chapter, you'll learn about Net8, which is Oracle's networking software. You'll learn about the various Net8 components, both on the server and on the client. You'll also learn some of the more common ways to configure Net8 server and client software. Finally, you'll read about some useful troubleshooting techniques that you can use to resolve Net8 connectivity problems.

Describing the Net8 Oracle Networking Software

Net8 is Oracle's networking software. Its purpose is to provide a common communication protocol for all Oracle software to use. Oracle clients can use Net8 to communicate with database servers; servers use Net8 to communicate with other servers. From an application development standpoint, Net8 provides a common interface that works the same regardless of the underlying networking technology or hardware platforms being used.

Note

Oracle's networking software used to be called SQL*Net. With the release of Oracle8, the name was changed to Net8.

Figure 5-1 shows a simplified representation of how Net8 fits into the client/server picture. As you can see, requests from a client software application are passed to Net8. Net8 then transmits those requests to the database server using a low-level networking protocol such as TCP/IP or SPX.

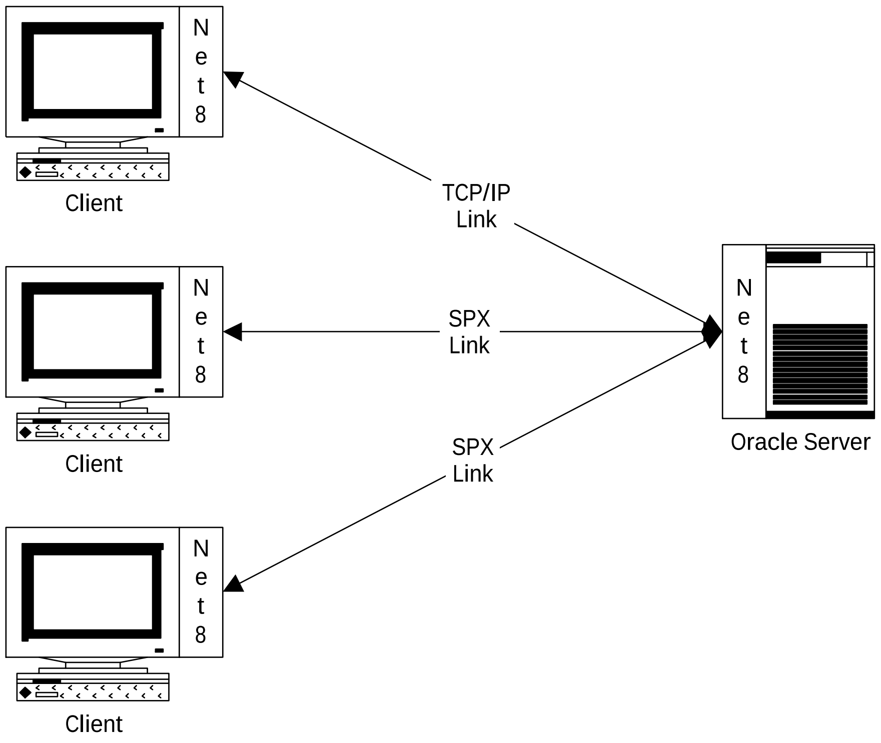


Figure 5-1: Net8 enables client applications to communicate with an Oracle database.

Because Net8 works the same from an application point of view, regardless of the underlying networking protocol being used, it is said to be *network transparent.* To move an application from one network environment to another, all you have to do is use the appropriate Net8 protocol adapter for that environment.

Net8 is also location transparent and operating-system transparent. It is *location transparent* because application programs do not need to know the name of the server to which they are connecting. Net8 is *operating-system transparent* because it works the same regardless of whether it is running on Windows NT, UNIX, Novell, or any other operating system that Oracle supports.

Net8 server components

The Net8 server component that you will be most concerned with is Net8 listener. The listener is a process that runs on a database server to monitor the network for incoming database connection requests. You interact with the listener by using a program called the Listener Control program.

In addition to the listener is the Net8 Assistant, a Java program that gives you an easy-to-use GUI interface for setting and modifying a variety of Net8 parameters. Oracle servers usually have the Net8 Client components, such as Net8 Easy Config and `tnsping`, installed as well. You'll read more about these client components later in this chapter.

The Net8 listener

The Net8 listener's job is to monitor the network, or *listen,* for requests to connect to one of the databases on a database server. When an incoming request is detected, the listener validates that request, logs the client on to the database, and hands the client off to a server process, or possibly a dispatcher process. Figure 5-2 illustrates this sequence of events.

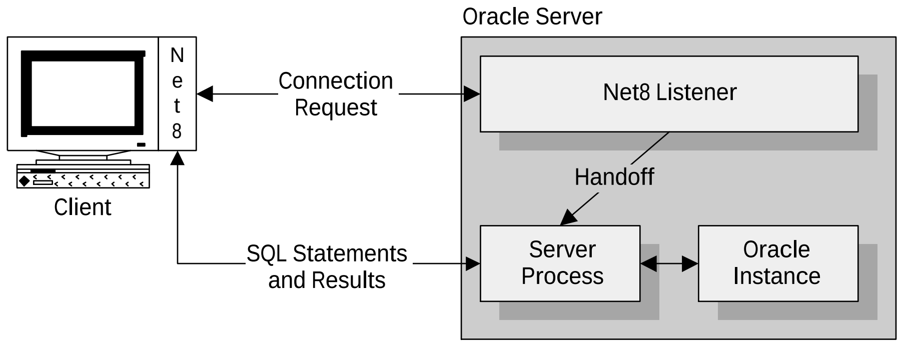


Figure 5-2: Net8 Listener accepts incoming connection requests.

Once the connection has been made and the remote user has been logged on to the database, the listener has no further role to play. All further communication takes place between the client and either a server process or a dispatcher process. However, the listener does listen for other connection requests.

Listener Control

The Listener Control program is your primary means of interacting with the Net8 listener. You can use the Listener Control program to stop and start the listener, check the status of the listener, turn on tracing, or set one of several options. The screen output shown in Listing 5.1 demonstrates Listener Control being used to *bounce*, that is, to stop and start, a listener:

Listing 5-1: Using the Listener Control program

```
C:\>lsnrctl

LSNRCTL for 32-bit Windows: Version 8.1.5.0.0 - Production on 25-JUL-99 18:11:50


(c) Copyright 1998 Oracle Corporation.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> set password bonk
The command completed successfully
LSNRCTL> stop
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0)))
The command completed successfully
LSNRCTL> start
Starting tnslsnr: please wait...

Service OracleOraHome81TNSListener start pending.
Service OracleOraHome81TNSListener started.
TNSLSNR for 32-bit Windows: Version 8.1.5.0.0 - Production
System parameter file is E:\ORACLE\ORA81\NETWORK\ADMIN\listener.ora
Log messages written to E:\Oracle\Ora81\network\log\listener.log
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=jonathan)(PORT=1521)))
Listening on: (DESCRIPTION=(PROTOCOL_STACK=(PRESENTATION=GIOP)(SESSION=RAW))(ADD
RESS=(PROTOCOL=TCP)(HOST=jonathan)(PORT=2481)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for 32-bit Windows: Version 8.1.5.0.0 - Produc
tion
Start Date                25-JUL-99 18:12:00
Uptime                    0 days 0 hr. 0 min. 2 sec
Trace Level               off
Security                  ON
SNMP                      OFF
Listener Parameter File   E:\ORACLE\ORA81\NETWORK\ADMIN\listener.ora
Listener Log File         E:\Oracle\Ora81\network\log\listener.log
```

```
Services Summary...
  PLSExtProc          has 1 service handler(s)
  JONATHAN            has 1 service handler(s)
The command completed successfully
LSNRCTL>
```

Cross-Reference

The most common Listener Control commands are described later in this chapter in the section "Configuring Net8 on the Server."

Multiple Listeners

In Figure 5-2 and in the previous example, the server was assumed to have only one Net8 listener process running; however, you can have multiple listener processes running at once. You might do this to ensure the complete separation of production and test databases. Having separate listeners means that you can take the listener for one database down without affecting the other database. There's an example later in this chapter showing you how to configure multiple listeners.

Net8 Assistants

In addition to the listener and the Listener Control program, a server may have two assistants: Net8 Assistant and Net8 Configuration Assistant. Oracle uses the term assistant to describe what Microsoft calls a wizard. An *assistant* is a program that helps you perform a complex task by gathering information from you in a user-friendly manner, and then it carries out the task for you.

Net8 Assistant is a nice GUI interface that you can use to configure the Net8 software on your server. You perform most Net8 configuration by editing various text files, some of which use some rather arcane syntax. Net8 Assistant lets you fill in forms and dialog boxes and then uses that information to edit the text files for you. Net8 Configuration Assistant walks you through some of the common Net8 configuration tasks that you need to perform after first installing Oracle8i software on a server.

Net8 Client components

A PC running Oracle client software will typically have the following Net8 components installed:

- Net8 Client
- Net8 Easy Config

Releases of Oracle prior to 8.1 also included protocol adapters as part of the Net8 Client software. You would pick which protocol adapters to install based on the network protocols in use at your site. With Oracle8i, the protocol adapters have been bundled into the basic Net8 Client software. You can no longer install them separately.

Net8 Client

For the most part, Net8 Client software consists of dynamic link libraries (DLLs on Windows NT) that are used by programs written to interact with an Oracle database. There's no process, like the listener, always running in the background. There's no stand-alone *program* that you can run from the Start menu.

From a configuration standpoint, the most significant part of the Net8 Client software installation is the two configuration files named sqlnet.ora and tnsnames.ora. Correctly configuring these files is the key to being able to connect to an Oracle database running on a server.

Net8 Easy Config

Net8 Easy Config is an assistant that puts a user-friendly face on the task of editing your local tnsnames.ora file. The tnsnames.ora file is a critical Net8 configuration file. You use it to define the remote databases to which you will be able to connect. The syntax used in the tnsnames.ora file is tough to follow and involves multiple levels of nested parentheses. If you edit this file by hand, it's easy to make a mistake, perhaps by leaving out a parenthesis. Oracle created Net8 Easy Config to simplify the process of editing the tnsnames.ora file and to reduce the risk of error.

Note The version of Net8 Easy Config that shipped with Oracle 8.1.5 is the first one that truly merits "easy" as part of its name. Previous versions didn't deal well with tnsnames.ora files that had been edited by hand. In release 8.1.5, Net8 Easy Config can not only handle manually edited tnsnames.ora files, but it also reformats them in a standard way so that they are easy to read.

Configuring Net8 on the Server

Configuring Net8 on a database server generally involves editing a text file named listener.ora. After editing listener.ora, you will need to use the Listener Control program to stop and restart the listener so that your changes can take effect.

As an alternative to editing the listener.ora file, you may be able to use Net8 Assistant. It provides a GUI interface that you can use to edit the listener.ora file. In your listener.ora file, you need to specify the following:

- ✦ The number of listeners that you want to run
- ✦ The protocols that you want to support
- ✦ The databases for which you will accept connections

In addition, you may want to configure the tnsnames.ora file on a server just as you would on a client so that you can access databases on other servers.

Using the Listener Control program

You use the Listener Control program to control and monitor the operation of the Net8 listeners on a server. Use Listener Control to perform the following functions:

- Start a listener
- Stop a listener
- Check the status of a listener

The next few sections show you how to start the Listener Control program and how to use it to perform each of the above tasks.

Starting Listener Control

Listener Control is not a GUI utility. It's a command-line-based utility, and it must be started from the command prompt. The command used to start the Listener Control program is `lsnrctl`. Here is an example:

```
C:\>lsnrctl

LSNRCTL for 32-bit Windows: Version 8.1.5.0.0 - Production on 25-JUL-99 18:11:50


(c) Copyright 1998 Oracle Corporation.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL>
```

Tip

Remember that "control" is abbreviated in the command as "ctl" and not as "ctrl." You may find yourself occasionally typing "lsnrctrl" instead.

Starting a Listener

For remote clients to connect to a database on a server, a listener process must be running on that server. You use the `start` command to start a listener, and the syntax looks like this:

```
  start [listener_name]
```

The syntax is described as follows:

- `start` — The command to start a listener.
- *`listener_name`* — The name of the listener that you want to start. This defaults to "listener."

Listing 5-2 shows a listener being started.

Listing 5-2: Starting a listener

```
LSNRCTL> start listener
Starting tnslsnr: please wait...

Service OracleOraHome81TNSListener start pending.
Service OracleOraHome81TNSListener started.
TNSLSNR for 32-bit Windows: Version 8.1.5.0.0 - Production
System parameter file is E:\ORACLE\ORA81\NETWORK\ADMIN\listener.ora
Log messages written to E:\Oracle\Ora81\network\log\listener.log
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=jonathan)(PORT=1521)))
Listening on: (DESCRIPTION=(PROTOCOL_STACK=(PRESENTATION=GIOP)(SESSION=RAW))(ADD
RESS=(PROTOCOL=TCP)(HOST=jonathan)(PORT=2481)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for 32-bit Windows: Version 8.1.5.0.0 - Produc
tion
Start Date                25-JUL-99 18:12:00
Uptime                    0 days 0 hr. 0 min. 2 sec
Trace Level               off
Security                  ON
SNMP                      OFF
Listener Parameter File   E:\ORACLE\ORA81\NETWORK\ADMIN\listener.ora
Listener Log File         E:\Oracle\Ora81\network\log\listener.log
Services Summary...
  PLSExtProc            has 1 service handler(s)
  JONATHAN              has 1 service handler(s)
The command completed successfully
LSNRCTL>
```

Once you've started a listener, remote clients will be able to connect to any databases served by that listener.

Stopping a Listener

You use the `stop` command to stop a listener. The syntax looks like this:

```
stop [listener_name]
```

The syntax is described as follows:

- ✦ `stop` — The command to stop a listener.
- ✦ *`listener_name`* — The name of the listener that you want to stop. This defaults to "listener."

The following example shows a listener being stopped:

```
LSNRCTL> stop listener
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0)))
The command completed successfully
LSNRCTL>
```

Stopping a listener prevents remote clients from connecting to databases served by that listener.

Checking a Listener's Status

You use the `status` command to check the status of a listener. It is often used when troubleshooting connectivity problems to verify that the listener is running. The syntax looks like this:

```
status [listener_name]
```

The syntax is described as follows:

- ✦ `status`—The command used to get a listener's status.
- ✦ *`listener_name`*—The name of the listener that you are interested in. This defaults to "listener."

Listing 5-3 shows how you would check the status of the default listener.

Listing 5-3: **Checking a listener's status**

```
LSNRCTL> status listener
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for 32-bit Windows: Version 8.1.5.0.0 - Produc
tion
Start Date                25-JUL-99 20:23:22
Uptime                    0 days 0 hr. 1 min. 46 sec
Trace Level               off
Security                  ON
SNMP                      OFF
Listener Parameter File   E:\ORACLE\ORA81\NETWORK\ADMIN\listener.ora
Listener Log File         E:\Oracle\Ora81\network\log\listener.log
Services Summary...
Service "PLSExtProc"            has 1 instances.
    Instance "PLSExtProc"                has 0 handlers.
Service "jonathan.gennick"               has 1 instances.
    Instance "JONATHAN"          has 0 handlers.
The command completed successfully
LSNRCTL>
```

The `status` command gives you several pieces of useful information. It tells you where the listener parameter file, `listener.ora`, is located. It tells you when the listener was started and how long it has been running. It also gives you a list of services, usually databases, for which the listener is listening. Take a look at the services summary section of the previous status output:

```
Services Summary...
Service "PLSExtProc"            has 1 instances.
    Instance "PLSExtProc"                has 0 handlers.
Service "jonathan.gennick"               has 1 instances.
    Instance "JONATHAN"          has 0 handlers.
The command completed successfully
```

The first service is `PLSExtProc`. This isn't a database service; `PLSExtProc` is a service that allows PL/SQL code running on Windows NT servers to make calls to external DLL routines. The second service is named `jonathan.gennick`, and it is a database service. This tells you that this listener will handle incoming connection requests for the database named `jonathan` in the `gennick` domain.

Setting a Password

You have the option of password-protecting your listeners to prevent unauthorized people from starting them, stopping them, or otherwise affecting their operation. If you have password-protected a listener, you will need to use the `set password` command to enter the appropriate password before you can operate that listener. The syntax for the `set password` command looks like this:

```
set password [password]
```

✦ `set password` — The command for setting the listener password.

✦ *`password`* — The password that you want to set. If you omit this argument, you will be prompted for the password.

The following example shows the `set password` command being used:

```
LSNRCTL> set password
Password:
The command completed successfully
LSNRCTL>
```

Note that the password was not passed as an argument. It could be, but in this case the argument was omitted so that Listener Control would prompt for the password. This is because Listener Control doesn't echo the characters typed in response to the prompt. This prevents people from looking over your shoulder and spotting the password as you type it.

Note You password-protect a listener by placing a `PASSWORDS_LISTENER` entry into your listener.ora file.

Locating the listener.ora file

The listener.ora file is a text file containing a number of settings that control the operation of the listener and that tell the listener which databases to listen for. Finding this file can sometimes be a challenge. The default location depends on the operating system that you are using. On Windows NT running Oracle8i, you'll find the listener.ora file in the following subdirectory underneath the Oracle software directory:

```
network\admin
```

Note

The 8.0.*x* releases of Oracle on Windows NT used net8\admin for the directory. The 7.*x.x* releases used network\admin.

On UNIX systems, the default location for `listener.ora` will be one of the following directories:

```
/var/opt/oracle
/etc
$ORACLE_HOME/network/admin
```

The directory used as the default location on UNIX systems can vary from one vendor's version of UNIX to another. If you have any doubts about the location on your system, consult the operating-system-specific documentation that came with your Oracle8i software distribution.

If you don't want to use the default listener.ora location, you can specify some other directory as the location. The `TNS_ADMIN` environment variable is used for that purpose.

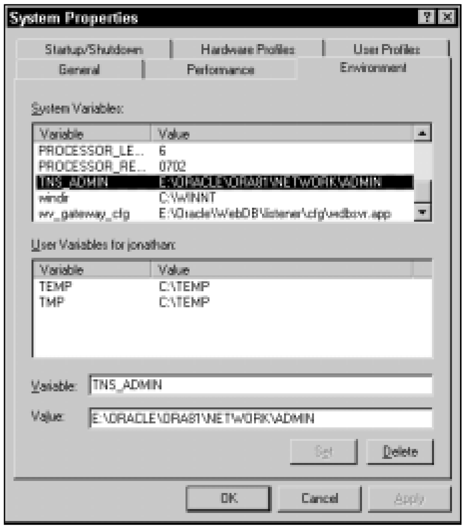
Using the TNS_ADMIN environment variable

If you don't like Oracle's default location for your Net8 files, you can use the `TNS_ADMIN` environment variable to point to whatever directory you want. Just make sure that you copy or move your Net8 configuration files (listener.ora, tnsnames.ora, sqlnet.ora, and so on) to the location that you have specified.

If you're running Windows NT, you have a choice between specifying `TNS_ADMIN` as an environment variable or as a registry entry. To set it as an environment variable, open the System control panel, click the Environment tab, and create a new variable named `TNS_ADMIN`. Figure 5-3 shows the `TNS_ADMIN` variable in the Windows NT System Properties dialog box. In this case, Oracle will look in the e:\oracle\ora81\network\admin directory for any Net8 configuration files.

If you prefer to specify `TNS_ADMIN` as a registry setting, then be aware that it is Oracle home-specific. For example, the registry entries controlling the operation of Oracle Enterprise Manager can be in the following key on a Windows NT system:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME2
```



System Properties

System Variables:

Variable	Value
PROCESSOR_LE...	6
PROCESSOR_RE...	0702
TNS_ADMIN	E:\ORACLE\ORA81\NETWORK\ADMIN
windir	C:\WINNT
wv_gateway_cfg	E:\Oracle\WebDB\listener\cfg\wdbsvr.app

User Variables for jonathan:

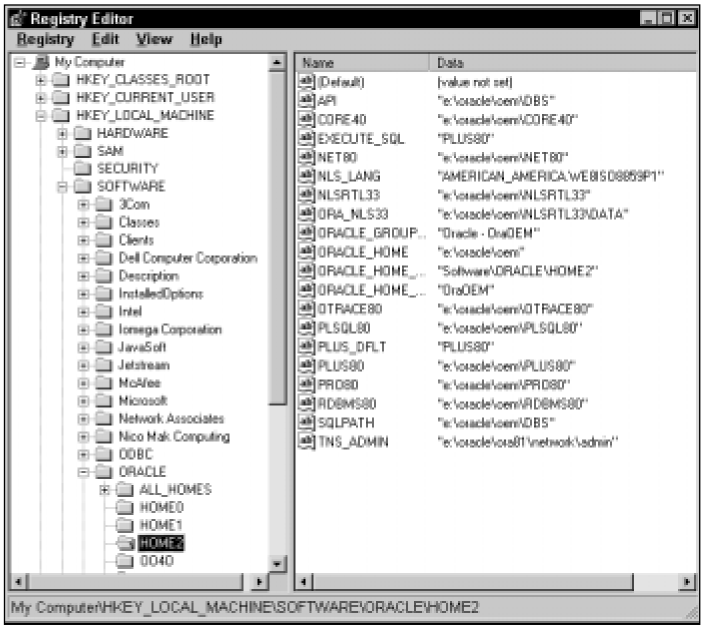
Variable	Value
TEMP	C:\TEMP
TMP	C:\TEMP

Variable: TNS_ADMIN

Value: E:\ORACLE\ORA81\NETWORK\ADMIN

Figure 5-3: The TNS_ADMIN environment variable pointing to the directory containing the Net8 configuration files

Underneath that key in our example, a `TNS_ADMIN` variable is pointing to the network\admin directory for the database software. That way, both Enterprise Manager and the database share the same Net8 configuration files. (See Figure 5-4.)



Registry Editor

```
My Computer
  HKEY_CLASSES_ROOT
  HKEY_CURRENT_USER
  HKEY_LOCAL_MACHINE
    HARDWARE
    SAM
    SECURITY
    SOFTWARE
      3Com
      Classes
      Clients
      Dell Computer Corporation
      Description
      InstalledOptions
      Intel
      Iomega Corporation
      JavaSoft
      Jetstream
      McAfee
      Microsoft
      Network Associates
      Nico Mak Computing
      ODBC
      ORACLE
        ALL_HOMES
        HOME0
        HOME1
        HOME2
        0040
```

Name	Data
(Default)	(value not set)
API	"e:\oracle\oem\DBS"
CORE40	"e:\oracle\oem\CORE40"
EXECUTE_SQL	"PLUS80"
NET80	"e:\oracle\oem\NET80"
NLS_LANG	"AMERICAN_AMERICA.WE8ISO8859P1"
NLSRTL33	"e:\oracle\oem\NLSRTL33"
ORA_NLS33	"e:\oracle\oem\NLSRTL33\DATA"
ORACLE_GROUP...	"Oracle - OraOEM"
ORACLE_HOME	"e:\oracle\oem"
ORACLE_HOME_...	"Software\ORACLE\HOME2"
ORACLE_HOME_...	"OraOEM"
OTRACE80	"e:\oracle\oem\OTRACE80"
PLSQL80	"e:\oracle\oem\PLSQL80"
PLUS_DFLT	"PLUS80"
PLUS80	"e:\oracle\oem\PLUS80"
PRO80	"e:\oracle\oem\PRO80"
RDBMS80	"e:\oracle\oem\RDBMS80"
SQLPATH	"e:\oracle\oem\DBS"
TNS_ADMIN	"e:\oracle\ora81\network\admin"

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME2

Figure 5-4: A TNS_ADMIN entry for Oracle Enterprise Manager

With the release of Oracle8i, Oracle has moved toward a model of installing each product in its own Oracle home. So if you have installed Oracle8i, WebDB, and Enterprise Manager, you will have three Oracle homes on your server. In a case like that, `TNS_ADMIN` provides a handy mechanism to point all those products to the same set of Net8 configuration files.

Looking at the listener.ora syntax

The `listener.ora` file defines one or more Net8 listeners that monitor the network for incoming database connection requests. For each listener, the file contains the following:

- Listener address information
- A list of Oracle instances for which the listener is listening
- Optional control parameters

The default installation of Oracle results in one listener being defined. The name of that listener is `LISTENER`, which tends to confuse some people more then help them because of the way the listener name is worked into the listener.ora syntax.

Listener Names and listener.ora Parameters

If you look in the `listener.ora` file created by a fresh install of the Oracle server software, you'll see two entries that look like this:

```
LISTENER =
  (
  ...
  )

SID_LIST_LISTENER =
  (
  ...
  )
```

The first entry defines the protocols that the listener recognizes, as well as the addresses on which the listener listens. The second entry contains a list of instances for which the listener will accept connections. In each case, the listener's name forms part of the keyword that begins each entry. The listener name in this example is LISTENER.

If you wanted to, you could define a second listener. To do that, just pick a name and place two corresponding entries in the `listener.ora` file. For example, suppose that you want a second listener named `PRODUCTION_LISTENER`. The entries for that listener would look like this:

```
PRODUCTION_LISTENER =
  (
  ...
  )

SID_LIST_PRODUCTION_LISTENER =
  (
  ...
  )
```

This naming convention — that is, using the listener name as part of the keyword — is used for all entries in the listener.ora file. You can see some examples later, in the section "Studying Listener.ora Examples."

The Listener Address Section

The listener address entry defines the protocols that your listener will recognize and the addresses (ports for TCP/IP) that the listener monitors. The syntax for this section is shown in Listing 5-4.

Listing 5-4: **Syntax for the listener address**

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS =
        (PROTOCOL = TCP)(HOST = oak.gennick.com)(PORT = 1521))
      (PROTOCOL_STACK =
         (PRESENTATION = TTC)
         (SESSION = NS)
      )
    )
   (DESCRIPTION =
      (ADDRESS =
        (PROTOCOL = TCP)(HOST = oak.gennick.com)(PORT = 2481))
      (PROTOCOL_STACK =
         (PRESENTATION = GIOP)
         (SESSION = RAW)
      )
    )
  )
```

Whoever dreamed up this syntax must have had a fetish for parentheses! It's easy to get a bit cross-eyed trying to make the parentheses match up when editing one of these files by hand. Here are descriptions of each of the syntax elements shown in the example:

`DESCRIPTION_LIST`	Identifies list protocol addresses, together with their respective protocol stack specifications.
`DESCRIPTION`	Identifies a specific protocol address and protocol stack specification.
`ADDRESS`	Defines the address used by a listener for a specific protocol. The format for this parameter varies with the protocol being used. See Table 5-1.
`PROTOCOL_STACK`	Specifies the presentation layer and session layers to use. There are only two choices: `(PRESENTATION = TTC) (SESSION = NS)`, which is the default, is used for Net8 connections to the database; `(PRESENTATION = GIOP) (SESSION = RAW)` allows connections to Java code using the General Inter-orb Protocol (GIOP).

Consider the following points about the syntax used in listener.ora, as well as in the other Net8 configuration files that you will read about later in this chapter:

✦ The `listener.ora` file keywords are not case sensitive. The values used with those keywords usually aren't either, but they may be, depending on the specific operating system and network protocol being used.

Note

Although SQL*Net is not case sensitive, it turns out that the Intelligent Agent used by OEM requires that the keywords all be in uppercase. You will not be able to use some of the OEM options (tuning, change management) if these are not included correctly.

✦ The spacing and line breaks don't have to be exactly as shown. The format you see here is that produced by Oracle's Net8 Assistant.

✦ The specific order of elements within an enclosing element isn't important. If you are specifying a TCP/IP address, for example, it would be just as acceptable to use `(PROTOCOL = TCP)(HOST = oak.gennick.com) (PORT = 1521))` as it would be to use `(HOST = oak.gennick.com) (PROTOCOL = TCP) (PORT = 1521))`.

This TCP/IP address tells the listener that the database server is named `oak` and that the listener should monitor TCP/IP port 1521 for incoming database connection requests. Table 5-1 shows the address entry format for TCP/IP and all the other protocols supported by Net8.

Table 5-1
Listener.ora Address Entry Formats

Protocol	***ADDRESS Entry Syntax***	***Notes***

TCP/IP	(ADDRESS =
	  (PROTOCOL = TCP)
	  (HOST = *host_name*)
	  (PORT = *port_number*)
	)
		The port number 1521 is generally 2481 is used for connections to Java routines in the database.
		The server's IP address may be used in place of the host name.
		The port number in the listener.ora file must match that used in the tnsnames.ora file on client machines.

IPC	(ADDRESS =
	  (PROTOCOL = IPC)
	  (KEY = *service_name*)
	)
		IPC is used to connect to remote procedures, such as those in a Windows DLL file.

Named Pipes	(ADDRESS =
	  (PROTOCOL = NMP)
	  (SERVER = *server_name*)
	  (PIPE = *pipe_name*)
	)
		The pipe name may be any arbitrary value, but it must match an entry in a client's tnsnames.ora file for a connection to be made.

LU6.2	(ADDRESS =
	  (PROTOCOL = LU62)
	  (LU_NAME = *server_name*)
	  (LOCAL_LU =
	  local_lu_alias)
	  (LOCAL_LU_NAME =
	  local_lu_name)
	  (MODE = *log_mode_entry*)
	  (PARTNER_LU_NAME =
	  server_name)
	  (PARTNER_LU_
	  LOCAL_ALIAS =
	    partner_lu_alias)
	  (TP_NAME = *transaction_*
	  program_name)
	)
		You can use LLU, LLU_NAME, MDN, PLU, and PLU_LA in place of LU_NAME, LOCAL_LU_NAME, MODE, z PARTNER_LU_NAME, and PARTNER_LU_LOCAL_ ALIAS, respectively.
		You either have to use an alias or a name, but not both. For example, you have to choose between LU_NAME and LOCAL_LU.
		Local names cannot be used with local aliases.

SPX	(ADDRESS =
	  (PROTOCOL = SPX)
	  (SERVICE = *service_name*)
	
	)
		SPX service names are arbitrary, but the name used in the tnsnames.ora file on client PCs must match the name used on the server.

The SID_List Entry

The SID_LIST entry for a listener contains a list of Oracle database instances for which the listener will accept connections. The SID_LIST entry also controls the number of dedicated server processes that the listener will prestart to have one ready and waiting when a user connects. The syntax for the SID_LIST entry is shown in Listing 5-5.

Listing 5-5: **The syntax for the SID_LIST entry**

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = bible_db.oak.gennick.com)
      (SID_NAME = BIBDB)
      (PROGRAM_NAME = extproc)
      (ORACLE_HOME = /oracle/ora81)
      (PRESPAWN_MAX = 99)
      (PRESPAWN_LIST =
        (PRESPAWN_DESC =
          (PROTOCOL = TCP)
          (POOL_SIZE = 10)
          (TIMEOUT = 5)
        )
      )
    )
  )
```

The following list provides descriptions for each of the elements in the SID_LIST:

SID_LIST_LISTENER	Introduces the SID list entry for the listener named LISTENER. If your listener is named PRODUCTION_LISTENER, then you can use SID_LIST_PRODUCTION_LISTENER.
SID_LIST	Identifies a list of database instances associated with the listener in question. You need only one of these entries.
SID_DESC	Defines one database instance to the listener. You can have as many SID_DESC entries as you have databases. A number of entries falling underneath SID_DESC further define how the listener services a particular database.

GLOBAL_DBNAME	Specifies the global database name, including the domain, identifying a database. This must match the value of the SERVICE_NAMES parameter in the database's initialization file. This entry is optional. Use it if you want to reference the database by its global name when connecting to it.
SID_NAME	Specifies the instance name. This value should match the value of the INSTANCE_NAME parameter in the database's initialization file. This entry is optional, but you need to specify at least one SID_NAME or GLOBAL_DBNAME.
PROGRAM_NAME	Associates the SID_DESC entry with an executable program name.
ORACLE_HOME	Specifies the path of the Oracle home directory for the database in question.
PRESPAWN_MAX	Specifies the maximum number of prespawned dedicated server processes that the listener is allowed to create. A *prespawned server process* is one that is created in anticipation of a future user connection. This parameter must not be less than the sum of the POOL_SIZE parameters for the SID_LIST. This parameter is ignored in Windows NT because prespawned dedicated server processes can't be created on that operating system.
PRESPAWN_LIST	Introduces a list of prespawned dedicated server processes for a SID_DESC entry.
PRESPAWN_DESC	Defines a number of dedicated server processes to be prespawned for a particular Net8 protocol. You may have as many PRESPAWN_DESC entries as you desire.
PROTOCOL	Specifies the Net8 protocol for which you want to prespawn the dedicated server processes. The processes are protocol-specific, so if you are supporting multiple protocols, you will need to use multiple PRESPAWN_DESC entries to configure prespawned processes for those protocols. Note, however, that you don't need to create prespawned processes for each protocol that you support.
POOL_SIZE	Specifies the number of *unused* dedicated server processes that you want to keep on hand for possible user connections. Set this value to match what you expect the average number of connection attempts to be at any given time.

TIMEOUT	Applies to dedicated server processes that have been used, and from which a user has disconnected. The timeout controls the number of minutes that the process will remain in the pool waiting for a new connection before being terminated.

Control Parameters

In addition to the listener address information and the SID list, the listener.ora file may optionally contain any of several miscellaneous entries. These are referred to as control parameters, and they are described in Table 5-2.

Table 5-2
Listener.ora Control Parameters

Parameter Name and Example	***Description***

`CONNECT_TIMEOUT`
`CONNECT_TIMEOUT_ LISTENER = 10`
	Default: 10 seconds
	Specifies that the listener will wait up to 10 seconds for a valid connection request after a connection has been initiated.

`LOG_DIRECTORY`
`LOG_DIRECTORY_ LISTENER = $ORACLE_HOME/ network/log`
	Default: `$ORACLE_HOME/network/log`
	Tells the listener to write the listener log file to the `$ORACLE_HOME/network/log` directory.

`LOG_FILE`
`LOG_FILE_LISTENER = LISTENER.LOG`
	Default: `LISTENER.LOG`
	Tells the listener that the listener log file should be named `LISTENER.LOG`.

`PASSWORDS`
`PASSWORDS_LISTENER = (oracle)`
`PASSWORDS_LISTENER = (oracle,big_secret, little_secret)`
	Default: none
	Specifies one or more unencrypted passwords that must be supplied by the DBA (using Listener Control's `SET PASSWORD` command) before Listener Control can be used to control the listener. The first example specifies a password of "oracle." The second example specifies several passwords, any one of which may be used.

Continued

Table 5-2 *(continued)*

Parameter Name and Example	***Description***

`STARTUP_WAITTIME`
`STARTUP_WAITTIME_LISTENER = 5`
	Default: 0 seconds
	Tells the listener to sleep five seconds before responding to the first `status` command from the Listener Control program.

`TRACE_DIRECTORY`
`TRACE_DIRECTORY_LISTENER = $ORACLE_HOME/network/trace`
	Default: `$ORACLE_HOME/network/trace`
	Default: `$ORACLE_HOME/network/trace`
	Tells the listener to write trace files into the `$ORACLE_HOME/network/trace` directory.

`TRACE_FILE`
`TRACE_FILE_LISTENER = LISTENER.TRC`
	Default: `LISTENER.TRC`
	Controls the name to be used for a listener trace file.

`TRACE_LEVEL`
`TRACE_LEVEL_LISTENER = SUPPORT`
	Default: `OFF`
	Controls the type of information written to a listener trace file. The `OFF` value means that no trace information will be generated. Values of `USER`, `ADMIN`, and `SUPPORT` result in trace files containing increasing levels of detail.

`USE_PLUG_AND_PLAY`
`USE_PLUG_AND_PLAY_LISTENER = ON`
	Default: `OFF`
	Tells the listener whether it should automatically register itself with an Oracle Names server. Valid values are `OFF` and `ON`.

Pay attention to the fact that each control parameter has the listener name attached to it. The examples in Table 5-2 all used the default listener name `LISTENER`. If you had two listeners defined, named `LISTENER` and `PRODUCTION_LISTENER`, you could specify parameters for each by qualifying the parameter name with the listener name. Here is an example that sets a password for each of those listeners:

```
PASSWORDS_LISTENER = (oracle)
PASSWORDS_PRODUCTION_LISTENER = (big_secret)
```

In this example, `oracle` is the password for the default listener, while `big_secret` is the password to the production listener named `PRODUCTION_LISTENER`.

Studying listener.ora examples

Sometimes it's easier to learn by example, and where Net8 is concerned, that's often the case. This section contains three sample listener.ora files that demonstrate some common configurations. To start with, you will see a basic, single-database configuration. Next, you will see how to modify that configuration to handle multiple databases. Finally, you will see how to configure multiple listeners and split several databases between them. These examples also use some of the control parameters.

One Listener, One Database

The following example in Listing 5-6 shows a listener.ora file that is pretty close to the default that you get when you first install Oracle. It defines one listener and gives it the default name of `LISTENER`. Only one protocol is supported, TCP/IP, and one database is in the SID list.

Listing 5-6: A listener.ora file supporting one protocol and one listener

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
      (PROTOCOL_STACK =
        (PRESENTATION = TTC)
        (SESSION = NS)
      )
    )
    (DESCRIPTION =
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = OAK.GENNICK.COM)
        (PORT = 1521))
      (PROTOCOL_STACK =
        (PRESENTATION = TTC)
        (SESSION = NS)
      )
    )
    (DESCRIPTION =
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = OAK.GENNICK.COM)
        (PORT = 2481))
      (PROTOCOL_STACK =
        (PRESENTATION = GIOP)
        (SESSION = RAW)
      )
    )
```

Continued

Listing 5-6: *(continued)*

```
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = E:\Oracle\Ora81)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = BIBLE_DB.OAK.GENNICK.COM)
      (ORACLE_HOME = E:\Oracle\Ora81)
      (SID_NAME = BIBDB)
    )
  )
```

In this example, the `DESCRIPTION` entry referencing port 2481 supports connections to Oracle8i's internal Java engine. If you weren't using any Java features, you could omit that entry. The entry referring to `EXTPROC0` exists to allow PL/SQL to call external DLL routines. It too could be omitted.

One Listener, Two Databases

This example, shown in Listing 5-7, is similar to the previous example, but it eliminates support for Java and external DLLs, and it listens for two databases instead of just one. It also specifies a password `MUNISING` for the listener.

Listing 5-7: **A listener.ora file supporting two databases**

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = OAK.GENNICK.COM)
        (PORT = 1521))
      )
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = BIBLE_DB.OAK.GENNICK.COM)
      (ORACLE_HOME = E:\Oracle\Ora81)
      (SID_NAME = BIBDB)
    )
(SID_DESC =
      (GLOBAL_DBNAME = PROD)
      (ORACLE_HOME = E:\Oracle\Ora81)
      (SID_NAME = PROD)
    )
)

PASSWORDS_LISTENER = (MUNISING)
```

In this example, the `PASSWORDS_LISTENER` entry defines a password for the listener. There are also two instances shown in the `SID_LIST` entry. One instance is named PROD, while the other is named BIBDB.

Two Listeners, Two Databases, and Two Protocols

The example shown in Listing 5-8 configures two listeners. One is listening for two databases, the other for one. The production listener supports only the TCP/IP protocol, while the development listener supports both TCP/IP and SPX. The listeners each have a password. The production listener is running in trace mode, generating a trace file for Oracle support to use in debugging a connectivity problem. The development listener isn't running in trace mode.

Listing 5-8: **A listener.ora file supporting two listeners, two protocols, and two databases**

```
DEVELOPMENT_LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = OAK.GENNICK.COM)
        (PORT = 1521))
      )
      (ADDRESS =
        (PROTOCOL = SPX)
        (SERVICE = OAK)
       )
    )
  )

PRODUCTION_LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = OAK.GENNICK.COM)
```

Continued

Listing 5-8: *(continued)*

```
          (PORT = 1522))
        )
      )
    )

  SID_LIST_DEVELOPMENT_LISTENER =
    (SID_LIST =
      (SID_DESC =
        (GLOBAL_DBNAME = BIBLE_DB_TEST.OAK.GENNICK.COM)
        (ORACLE_HOME = E:\Oracle\Ora81)
        (SID_NAME = TEST)
      )
      (SID_DESC =
        (GLOBAL_DBNAME = BIBLE_DB_DEV.OAK.GENNICK.COM)
        (ORACLE_HOME = E:\Oracle\Ora81)
        (SID_NAME = DEV)
      )
  )

  SID_LIST_PRODUCTION_LISTENER =
    (SID_LIST =
      (SID_DESC =
        (GLOBAL_DBNAME = BIBLE_DB_PROD.OAK.GENNICK.COM)
        (ORACLE_HOME = E:\Oracle\Ora81)
        (SID_NAME = PROD)
      )
  )

  TRACE_DIRECTORY_DEVELOPMENT_LISTENER = c:\trace
  TRACE_DIRECTORY_PRODUCTION_LISTENER = c:\trace
  TRACE_FILE_DEVELOPMENT_LISTENER = DEV_LISTENER.TRC
  TRACE_FILE_PRODUCTION_LISTENER = PROD_LISTENER.TRC
  TRACE_LEVEL_DEVELOPMENT_LISTENER = OFF
  TRACE_LEVEL_PRODUCTION_LISTENER = SUPPORT
  PASSWORDS_DEVELOPMENT_LISTENER = (MUNISING)
  PASSWORDS_PRODUCTION_LISTENER = (MARQUETTE)
```

Notice that this listing defines two listeners. One is named DEVELOPMENT_LISTENER, and the other is named PRODUCTION_LISTENER. The entries defining these two listeners are the first two in the file. Notice how the listener names are made part of all the other parameter names in the file. The password for the listener named PRODUCTION_LISTENER is defined by the `PASSWORDS_PRODUCTION_LISTENER` entry. The password for the listener named DEVELOPMENT_LISTENER is defined by the `PASSWORDS_DEVELOPMENT_LISTENER` entry. This naming convention is used for all the other entries as well.

Using Oracle's Net8 Assistant

Having looked at the listener.ora syntax, you can see how easily you might make a mistake when editing this file by hand. It's particularly easy to end up with mismatched parentheses; in fact, Oracle support has historically been deluged with calls related to missing parentheses or other syntax errors in the Net8 configuration files.

Partly to combat the listener.ora syntax problems, and partly just to simplify life for DBAs in a world where everything is point and click, Oracle created a GUI-based program named Net8 Assistant for editing the Net8 configuration files. In Oracle8i, Net8 Assistant is a Java utility that allows you to configure general Net8 parameters in the sqlnet.ora file, Net8 service names in the tnsnames file, and listener parameters in the listener.ora file. Listener parameters are divided into these four categories:

- General parameters
- Listening locations
- Database services
- Other services

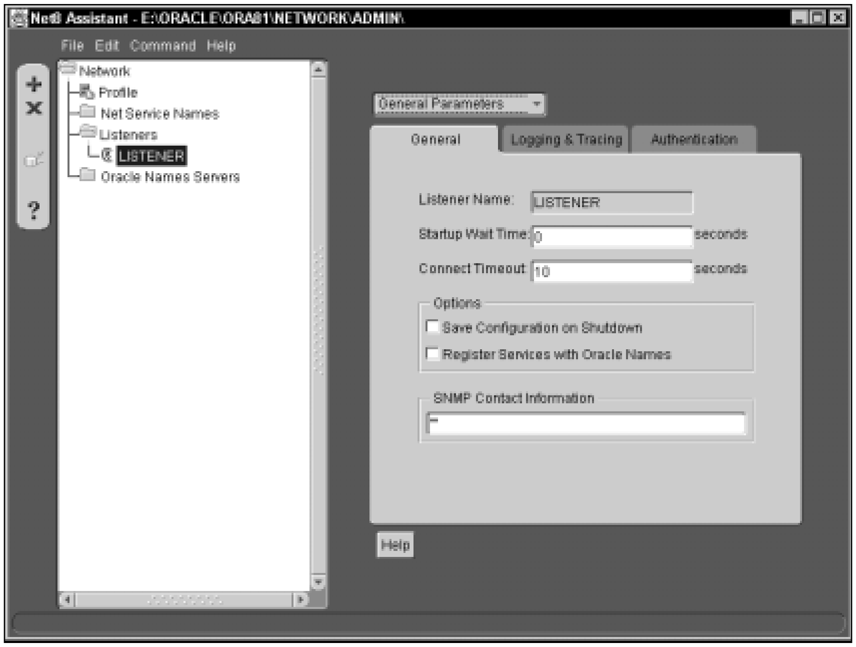
Figure 5-5 shows the General Parameters window for the default listener. The general parameters correspond to the control parameters that you read about earlier, and they are divided into three tabs named General, Logging & Tracing, and Authentication.

Editing these parameters is fairly intuitive — Net8 Assistant works like any other Windows application and comes complete with drop-down list boxes, radio button fields, check boxes, and text boxes. The Listening Locations window, shown in Figure 5-6, corresponds to the listener address section of the listener.ora file, and it allows you to specify the protocols that the listener will recognize. Figure 5-6 shows the address entry for the TCP/IP protocol.

The Database Services window, shown in Figure 5-7, corresponds to the SID list section of the listener.ora file. It allows you to define the databases for which the listener in question will accept connection requests.

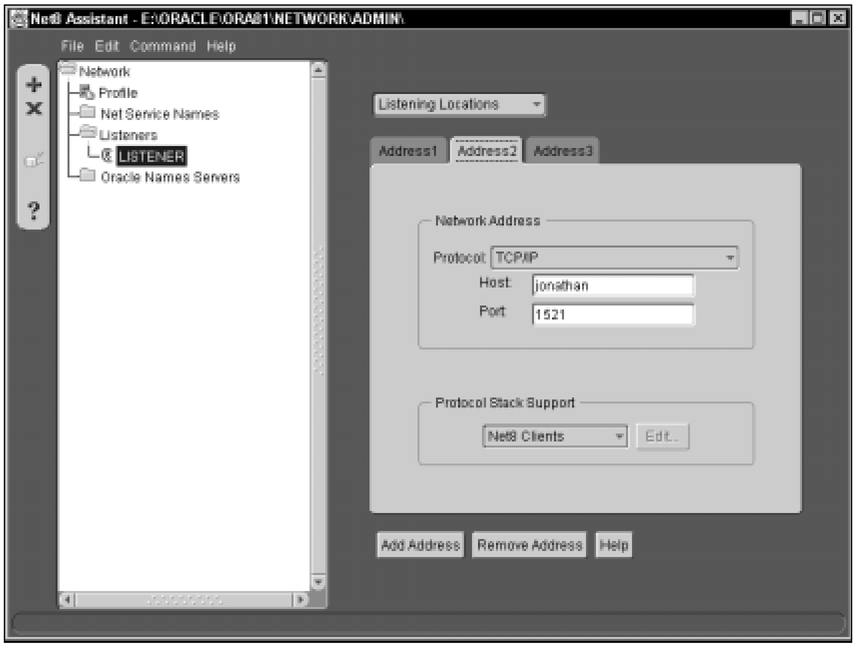
You use the Other Services window, shown in Figure 5-8, to define nondatabase services, such as the `PLSExtProc` service, that are used when you make calls from PL/SQL to external DLL routines.

To run Net8 Assistant on a Windows NT system, go to the Start menu, point to Oracle, point to OraHome81, point to Network Administration program group, and select the Net8 Assistant icon. Make whatever changes you want and then use the File Save menu option to write out a new listener.ora file. You may have to bounce the listener, that is, stop and restart the listener, for some changes to take effect.



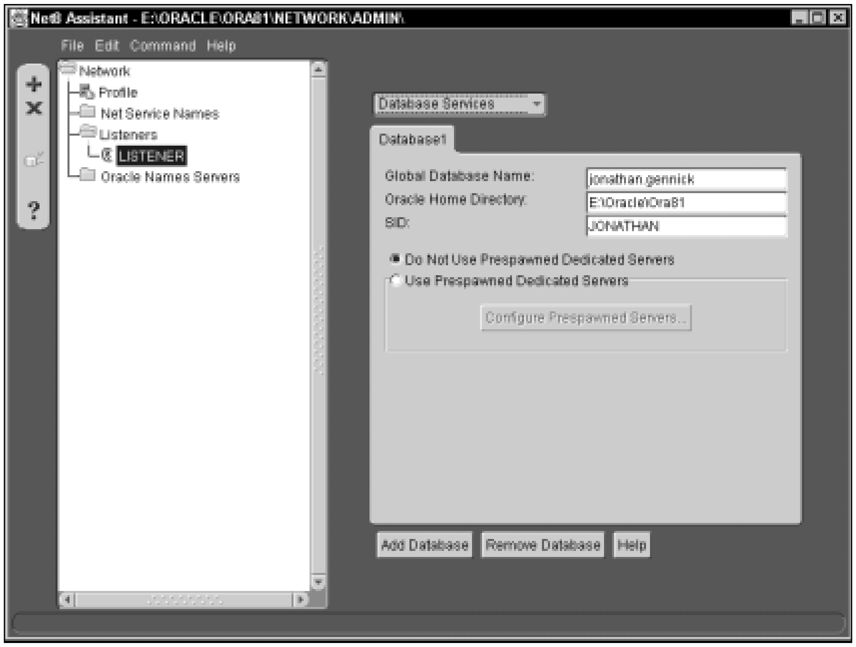
Listener Name: LISTENER
Startup Wait Time: 0 seconds
Connect Timeout: 10 seconds
Options
Save Configuration on Shutdown
Register Services with Oracle Names
SNMP Contact Information

Figure 5-5: General parameters for the default listener



Network Address
Protocol: TCP/IP
Host: jonathan
Port: 1521
Protocol Stack Support
Net8 Clients

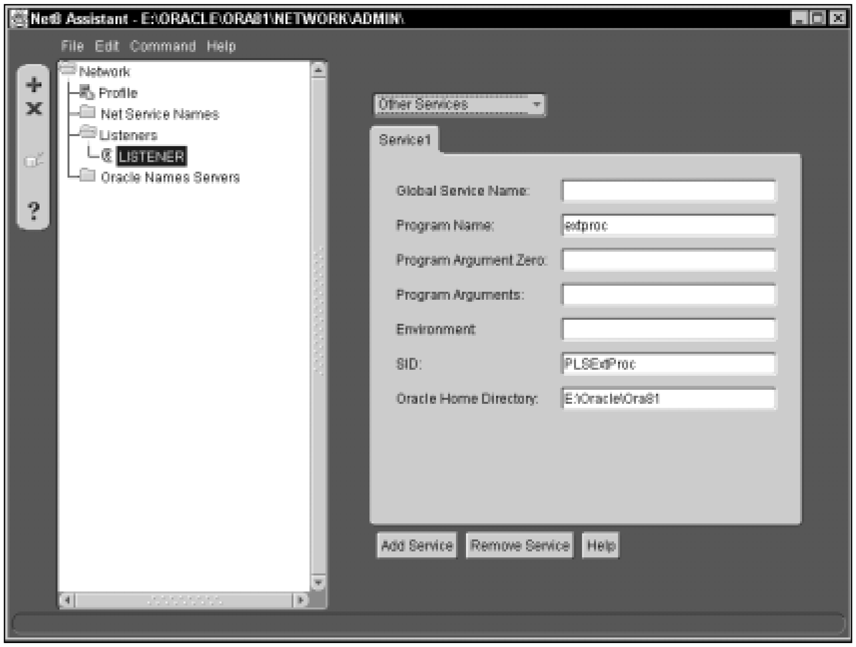
Figure 5-6: Listener locations for the default listener



Global Database Name: jonathan.gennick
Oracle Home Directory: E:\Oracle\Ora81
SID: JONATHAN

● Do Not Use Prespawned Dedicated Servers
○ Use Prespawned Dedicated Servers

Figure 5-7: Database services for the default listener



Global Service Name:
Program Name: extproc
Program Argument Zero:
Program Arguments:
Environment:
SID: PLSExtProc
Oracle Home Directory: E:\Oracle\Ora81

Figure 5-8: Other services for the default listener

Tip

Take a little time to become familiar with Net8 Assistant. Compare what you see on its screens with the information in your listener.ora file. Add database services or listeners using Net8 Assistant, and observe how it changes your listener.ora file.

Configuring Net8 on the Client

Configuring Net8 on a client involves editing two text files named sqlnet.ora and tnsnames.ora. As with the listener.ora file, these files — especially the tnsnames. ora file — tend to be burdened with multiple sets of nested parentheses. On a client, you place entries in the sqlnet.ora file to configure some general items such as the following:

- The name and location of the Net8 log file
- The order in which naming methods are used to try and resolve Net8 service names
- Preferred names servers
- The name and location of trace files
- The level of tracing, if any, that you want

If you are using the local naming method, you also place entries in a file named `tnsnames.ora` that define the following:

- Net8 service names
- The databases that those service names refer to
- The servers on which those databases reside
- The protocol (TCP/IP, SPX, and so on) used to communicate with those databases

The next few sections discuss how to locate the client configuration files, show you the syntax to use in these files, and provide some examples of typical configurations.

The client configuration files

The same considerations apply to finding the client configuration files as apply to finding the listener.ora file on the server. On Windows NT machines, you will find these files in:

```
c:\Oracle\Ora81\network\admin
```

Note that the `c:\Oracle\Ora81` directory represents the Oracle home directory that you created when you installed the software. If you choose a different Oracle home location, look in the `network\admin` directory underneath that Oracle home instead.

On UNIX platforms, the exact location of the Net8 configuration files depends on the operating system being used. Look in the following locations:

```
/var/opt/oracle (e.g. Unix System V)
/etc
$ORACLE_HOME/network/admin (e.g. HP-UX)
```

As on the server, you can set the `TNS_ADMIN` environment variable on the client to specify any directory you like as the location for the Net8 configuration files.

The sqlnet.ora syntax

The sqlnet.ora file contains parameters that control how Net8 operates. Far more parameters exist (many of which are related to the use of Oracle Names or to the different authentication methods that Oracle supports) than can be described in one chapter. Table 5-3 summarizes the most common among these parameters.

Table 5-3
Common sqlnet.ora Parameters

Parameter Name and Example	*Description*

`NAMES.DEFAULT_DOMAIN`
 	
`NAMES.DEFAULT_DOMAIN = idg.com`
	Default: `null`
	Specifies the default domain to append to unqualified Net8 service names. In this example, if a client attempts to connect to a service named `prod`, Net8 will automatically translate that to `prod.idg.com`.

`NAMES.DIRECTORY_PATH`
 	
`NAMES.DIRECTORY_PATH = (TNSNAMES, ONAMES)`
	Default: `(TNSNAMES, ONAMES, HOSTNAME)`
	Specifies the naming methods that Oracle uses and the order in which Oracle uses them when trying to resolve a Net8 service name. In this example, Oracle will first look in the `tnsnames.ora` file, then look for an Oracle Names server, and finally attempt to use host naming to match the service name with a database.

Continued

Table 5-3 *(continued)*

Parameter Name and Example	***Description***

`NAMES.INITIAL_`
`RETRY_TIMEOUT`
 	
`NAMES.INITIAL_`
`RETRY_TIMEOUT = 15`
	Default: 15 seconds
	Specifies the time, in seconds, that a client will wait for a request from one Oracle Names server before trying the next server in the preferred servers list. The valid range for this value is from 1 to 600 seconds.

`NAMES.PREFERRED_SERVERS`
 	
`NAMES.PREFERRED_SERVERS ]`
`  (ADDRESS_LIST =`
`     (ADDRESS =`
`       (PROTOCOL = TCP)`
`       (HOST = jonathan)`
`       (KEY = 1575))`
`     (ADDRESS =`
`        (PROTOCOL = IPC)`
`       (KEY = n01))`
`     ...`
`   )`
	Default: none
	Defines a list of Oracle Names servers in the order in which they will be used, in an attempt to resolve a Net8 service name.

`SQLNET.EXPIRE_TIME`
 	
`SQLNET.EXPIRE_TIME = 10`
	Default: 0 minutes
	Sets the time interval between probes to see if a session is still alive. Sessions that don't respond to the probe are assumed to represent dropped user sessions and are terminated.

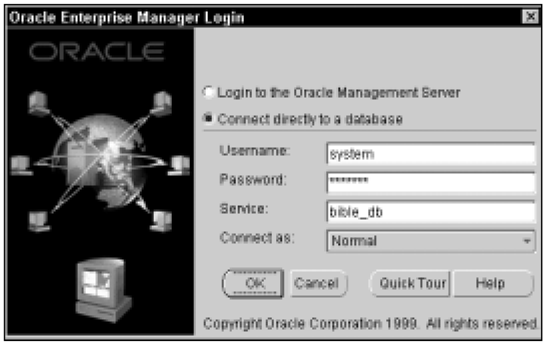
`TRACE_DIRECTORY_CLIENT`
 	
`TRACE_DIRECTORY_CLIENT =`
`c:\trace`
	Default: `$ORACLE_HOME/network/trace`
	Specifies the directory to which Net8 client trace files are to be written.

`TRACE_FILE_CLIENT`
 	
`TRACE_FILE_CLIENT =`
`sqlnet.trc`
	Default: `SQLNET.TRC`
	Specifies the name to use for client trace files.

`TRACE_UNIQUE_CLIENT`
 	
`TRACE_UNIQUE_CLIENT = ON`
	Default: `ON`
	
	Controls whether each client Net8 trace file is uniquely identified. Valid values are `ON` and `OFF`. If `ON` is used, then any Net8 trace files will have the process ID number appended to the file name.

The tnsnames.ora syntax

The tnsnames.ora file resides on a client and contains entries that resolve Net8 service names to specific databases on specific servers. Take a look at Figure 5-9, which shows the Oracle Enterprise Manager Login window for SQLPlus Worksheet, where a user logs on to a database named `bible_db`.


Oracle Enterprise Manager Login
ORACLE
Login to the Oracle Management Server
Connect directly to a database
OK
Cancel
Quick Tour
Help

Username: system
Password: ********
Service: bible_db
Connect as: Normal

Copyright Oracle Corporation 1999. All rights reserved.

Figure 5-9: Using SQLPlus Worksheet to log on to the `bible_db` database

Just what database does the service name `bible_db` refer to, and where is it? Oracle needs a way to determine this, and there are several methods to choose from. One method is referred to as *local naming* and consists of looking up the service name in a text file that contains the details of where the database for that service is. The name of this text file is tnsnames.ora.

The syntax for a basic entry in the tnsnames.ora file looks like this:

```
BIBLE_DB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = OAK.GENNICK.COM)
        (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = BIBLE_DB.OAK.GENNICK.COM)
      (SID = BIBDB)
    )
  )
```

The following list describes each of the syntax elements shown in this example:

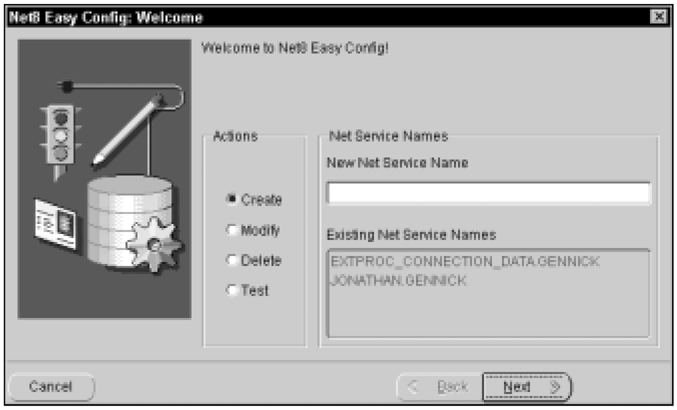
`BIBLE_DB`	Represents the Net8 service name. You can use simple names like this, or you can specify a name and a domain using Internet-style notation — for example, `BIBLE_DB.OAK.GENNICK.COM`. If you don't specify a domain, Net8 will use the value from the `NAMES.DEFAULT_DOMAIN` parameter in your sqlnet.ora file.
`DESCRIPTION`	Defines both a listener address and a database to which you want to connect via that listener.
`ADDRESS_LIST`	Defines a list of listener addresses for the Net8 service.
`ADDRESS`	Specifies the host address used to contact the listener. The format matches that used for the `ADDRESS` entry in the `listener.ora` file, and it varies depending on the network protocol being used. See Table 5-1 for details.
`CONNECT_DATA`	Specifies the database to which you want to connect when using the Net8 service name.
`SERVICE_NAME`	Defines the service name of an Oracle8i database. This is valid only for Oracle releases 8.1.*x* and above. The service name must match a name in the `SERVICE_NAMES` parameter in the database's parameter file. Typically, the service name will match the global name and domain of the database.
`SID`	Specifies the name of the specific Oracle instance that you want to connect to when using this Net8 service name. The `SID` is used when connecting to a database running under Oracle release 8.0.*x* or prior.

You may see other parameters in a tnsnames.ora file that involve load balancing and failover, but those are outside the scope of this book. The parameters described in this section are those used most commonly.

Net8 Easy Config

Net8 Easy Config is a network administration utility that ships with Oracle8i that allows you to easily add new entries to your tnsnames.ora file. It also allows you to modify or delete existing entries. Net8 Easy Config is a wizard-like tool that prompts you for the information required for the network protocol you are using and that tests the new or modified connection before making the change permanent.

If you are running Windows NT, you can find Net8 Easy Config in the Network Administration folder under the Oracle software program group. Click Start, point to Programs, point to Oracle – OraHome81, point to Network Administration, and select Net8 Easy Config. Figure 5-10 shows the opening page.

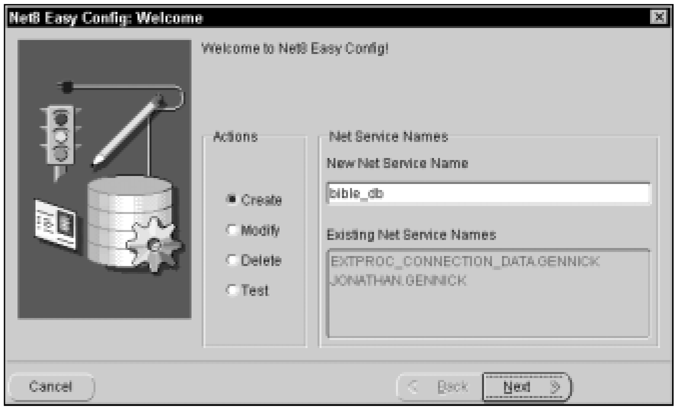


EXTPROC_CONNECTION_DATA.GENNICK
JONATHAN.GENNICK

Figure 5-10: Net8 Easy Config's opening wizard page

The radio buttons on the left allow you to choose whether you want to create a new entry, modify or delete an existing entry, or simply test an existing entry. The text box at the bottom shows you a list of the existing entries in your tnsnames.ora file. Let's assume that you are going to add a new entry.

Your first job would be to think up a new service name. It can be anything you like that fits your naming conventions and that makes sense to your users. Next, click the Create radio button. After that, type the new Net8 service name into the text box, as shown in Figure 5-11.

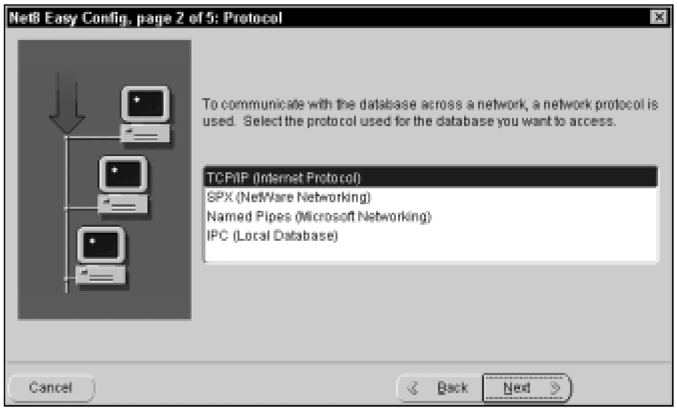


New Net Service Name: bible_db

EXTPROC_CONNECTION_DATA.GENNICK
JONATHAN.GENNICK

Figure 5-11: Creating a new Net8 service name

After entering the new name, as shown in Figure 5-11, click the Next button. You will see the page shown in Figure 5-12. Your task now is to select the network protocol to use for this new entry.

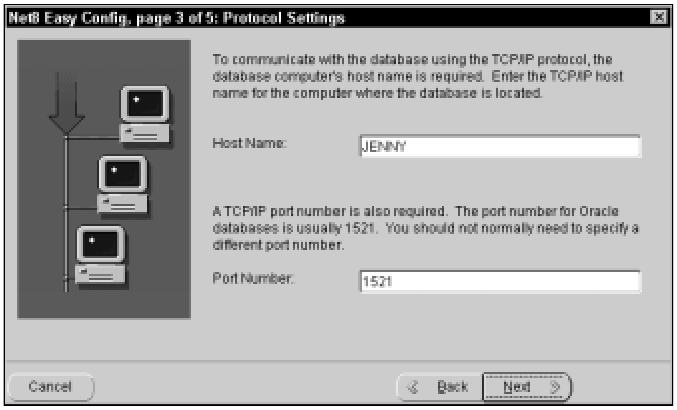


To communicate with the database across a network, a network protocol is used. Select the protocol used for the database you want to access.

TCP/IP (Internet Protocol)
SPX (NetWare Networking)
Named Pipes (Microsoft Networking)
IPC (Local Database)

Figure 5-12: Selecting the network protocol for a service name

By far, the most commonly used protocol is TCP/IP, so let's assume that you've selected that and clicked Next. You should now see the page shown in Figure 5-13. Your job here is to specify the host name of the database server and the port number that the listener is monitoring.



To communicate with the database using the TCP/IP protocol, the database computer's host name is required. Enter the TCP/IP host name for the computer where the database is located.

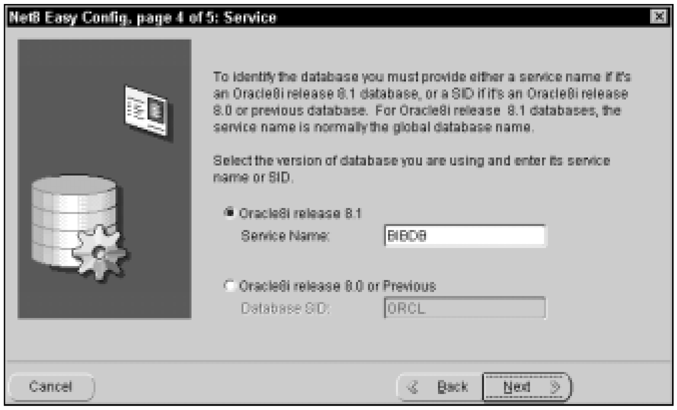
Host Name: JENNY

A TCP/IP port number is also required. The port number for Oracle databases is usually 1521. You should not normally need to specify a different port number.

Port Number: 1521

Figure 5-13: Entering the host name and port number for a service name

In this example, the host name JENNY is used. The port number has been left at the default of 1521. The port number here must match the port number specified in the listener.ora file on the server in question. If you are uncertain, 1521 is a good bet to start with because it's the default, and most people don't change it. Click Next, and you will see the page shown in Figure 5-14.


Net8 Easy Config, page 4 of 5: Service

To identify the database you must provide either a service name if it's an Oracle8i release 8.1 database, or a SID if it's an Oracle8i release 8.0 or previous database. For Oracle8i release 8.1 databases, the service name is normally the global database name.

Select the version of database you are using and enter its service name or SID.

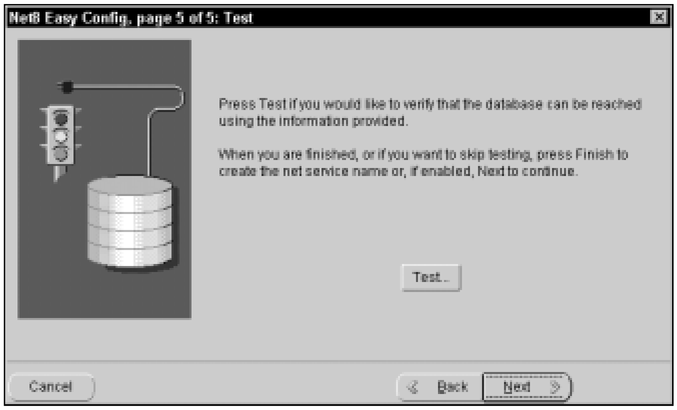
● Oracle8i release 8.1
Service Name: BIBDB

○ Oracle8i release 8.0 or Previous
Database SID: ORCL

Figure 5-14: Choosing the Oracle database for a service name

Now you need to choose the database instance or database service on the remote host to which you want to connect. If you are running a release of Oracle that precedes the 8i release, then your only choice here is to specify the SID name of an instance on the server. If you are running release 8.1.*x* or greater, you should specify a database service name. The database service name here should match up with a `GLOBAL_DBNAME` entry in your server's listener.ora file.

The next step is to test the new connection. Click the Next button, and you will see the page shown in Figure 5-15.

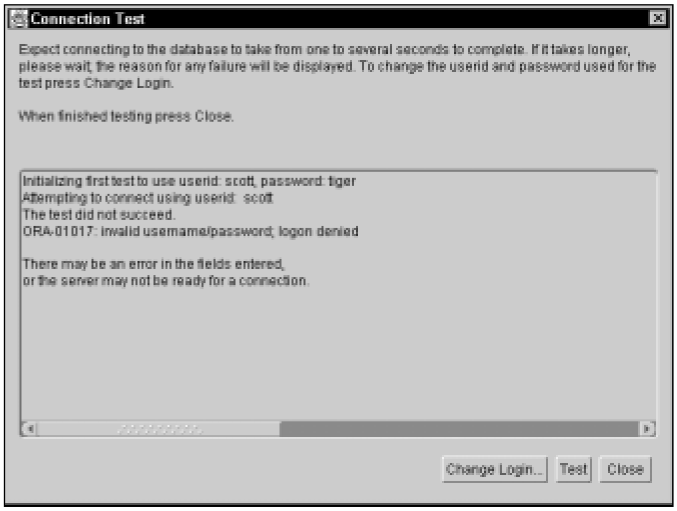

Net8 Easy Config, page 5 of 5: Test

Press Test if you would like to verify that the database can be reached using the information provided.

When you are finished, or if you want to skip testing, press Finish to create the net service name or, if enabled, Next to continue.

Figure 5-15: Preparing to test a new Net8 service name

Click the Test button, and Net8 Easy Config will attempt to connect to the remote database using the SCOTT/TIGER login. The SCOTT user is a demo user, and the test will fail if you have deleted that user from your database, as many DBAs do. Figure 5-16 shows you the page that you will see if an error occurs.



Expect connecting to the database to take from one to several seconds to complete. If it takes longer, please wait; the reason for any failure will be displayed. To change the userid and password used for the test press Change Login.

When finished testing press Close.

```
Initializing first test to use userid: scott, password: tiger
Attempting to connect using userid:  scott
The test did not succeed.
ORA-01017: invalid username/password; logon denied

There may be an error in the fields entered,
or the server may not be ready for a connection.
```

Figure 5-16: A new Net8 service name that fails the connection test

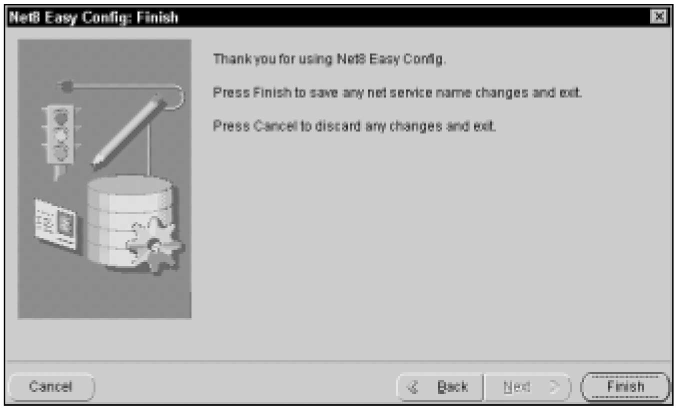
If the test failed because SCOTT/TIGER wasn't a valid login, click the Change Login button to enter a different username and password.

Note After entering a new username and password, you must remember to click the Test button to retry the test. Otherwise, Net8 Easy Config won't respond.

When you're done testing, click the Close button to close the test page, and click the Next button to continue with Net8 Easy Config. The last page that you will see is shown in Figure 5-17.

This is where you finalize your changes. You have one last chance to click the Back button and review or change your new entry. If you're satisfied with it, click the Finish button to have Net8 Easy Config write your changes out to your tnsnames.ora file.

Note If you need to add two or more databases, you have to invoke the program for each database. You have to *finish* the program to save the data for the first database before going back.



Thank you for using Net8 Easy Config.

Press Finish to save any net service name changes and exit.

Press Cancel to discard any changes and exit.

Figure 5-17: Net8 Easy Config's final page

Troubleshooting Net8

When Net8 works, it works very well. When it doesn't work, troubleshooting the problem can be a frustrating experience. A few techniques and utilities can be helpful, though, and they are discussed in the last part of this chapter.

Examining types of problems

The most common Net8-related problems you may encounter have the following causes:

- ✦ Net8 Listener isn't running.
- ✦ The address information in the tnsnames.ora file is incorrect.
- ✦ The service name being used isn't correct.
- ✦ The default domain specified in the sqlnet.ora file doesn't match the domain in the tnsnames.ora file for the service that you are trying to use.

The trick to solving most connectivity problems is to start from the physical connection and work your way up through each layer of networking until you find exactly where the problem occurs. When you troubleshoot a connectivity problem, you should generally check the following items in the order they are listed:

1. The physical LAN connection
2. Connectivity to the server using the underlying network protocol: TCP/IP, for example
3. Connectivity to the listener
4. Connectivity to the database

Checking the physical connection is as simple as making sure that the network cable is plugged into the PC and that the other end of the cable is plugged into the wall. Only occasionally is this the problem, but still it's worth a try. You'll feel pretty silly after you spend an hour searching for the problem only to finally find that someone accidentally kicked the network cable loose.

Using ping and tnsping

Testing the connectivity between client and server often involves the use of two utilities known as `ping` and `tnsping`. With TCP/IP networks, which are the most common, you can use the `ping` utility to verify connectivity between a client and a server. For testing the Net8 connectivity, Oracle supplies a utility analogous to `ping` named `tnsping`. The `tnsping` utility tells you whether the remote Net8 listener process can be contacted.

Pinging a Server

The TCP/IP `ping` utility verifies that a given host can be contacted via the network. The `ping` command is simple. It takes the host name as an argument and reports back on whether the host could be contacted. Here is an example:

```
ping jonathan.gennick
```

The output from this example is as follows:

```
Pinging jonathan.gennick [10.11.49.241] with 32 bytes of data:

Reply from 10.11.49.241: bytes=32 time<10ms TTL=128
Reply from 10.11.49.241: bytes=32 time<10ms TTL=128
Reply from 10.11.49.241: bytes=32 time<10ms TTL=128
Reply from 10.11.49.241: bytes=32 time<10ms TTL=128
```

If `ping` replies like this, then all is good. The target host is reachable, and you should move on up to the next level. If `ping` fails to contact the remote host, then you need to find out why. The following are some areas to look at:

- ✥ Is the remote server up and running?
- ✥ Did you supply `ping` with the correct host name?
- ✥ Is the IP address that the `ping` command reported correct?
- ✥ Is `ping` able to resolve the server name to an IP address?
- ✥ Can you `ping` using the IP address but not the server name?

IP address problems almost always require that you work together with your system or network administrators to resolve them. Many networks will use Dynamic Host Configuration Protocol (DHCP) or something similar to translate a host name into an IP address. If you can `ping` using an IP address but not using the host name, then your DHCP server might be down, or it might not be configured properly.

Note

Some networks still use hosts files on individual PCs to equate IP addresses with host names. If your network is one of these, you should verify that your hosts file contains the correct address for the server you are trying to reach.

The `ping` command will usually report back the IP address of the server that you are pinging. If that IP address isn't correct, or if `ping` is unable to translate the server name to an IP address, then you should contact your network administrator and work with him or her to resolve the problem.

Tnspinging a Listener

The `tnsping` utility is similar to `ping` and is provided by Oracle to help resolve Net8 connectivity problems. It verifies that the listener on the remote server can be contacted from the client. The `tnsping` utility has one advantage over `ping`: It is not network-specific. Even if you aren't using TCP/IP as your network protocol, you can still use `tnsping` to test connectivity to the server.

To run `tnsping`, you issue the `tnsping` command and supply a Net8 service name as an argument to that command. For example:

```
tnsping jonathan.gennick
```

generates the following output:

```
TNS Ping Utility for 32-bit Windows: Version 8.1.5.0.0 - Production on 25-JUL-99
 21:09:53

(c) Copyright 1997 Oracle Corporation.  All rights reserved.

Attempting to contact (ADDRESS=(PROTOCOL=TCP)(HOST=jonathan)(PORT=1521))
OK (60 msec)

C:\>
```

The `tnsping` utility will report back the address entry for the server that it is trying to contact, and it will tell you whether the listener could be contacted. If the listener could be contacted, you should check the following:

✦ Is the listener up and running?

✦ Can you `ping` the server (TCP/IP), or otherwise verify connectivity to the server?

✦ Is the port number in your `tnsnames.ora` file correct? The `tnsping` utility will report the port number to you. Make sure it matches the port number in the `listener.ora` file on the server.

If `tnsping` is successful and the listener could be contacted, then you should be able to log on to your database.

Tip The SQL*Plus utility is convenient to use to test your ability to log on to a database.

If, even after `tnsping`ing the listener, you still can't log on to the database, check the following:

- ✦ Is the database instance running?
- ✦ Is the database open?
- ✦ Are you using the correct username and password?
- ✦ Does the user that you are logging on have `CREATE SESSION` privileges?

If you've performed all the tests up to this point and still haven't isolated a connectivity problem, you have reached the point where no pat answers exist. You have reached a point in the troubleshooting process that calls for a good understanding of Net8, the underlying network protocols, the Oracle database software, and networking in general. You will need a healthy amount of intuition and experience to guide your next steps. Now is the time for you to consult with other professionals and consider calling Oracle support for help as well.

Summary

In this chapter, you learned:

- ✦ Net8 enables connectivity between an Oracle database and a client, or between two Oracle databases. Net8 supports multiple physical protocols, and provides location and protocol transparency to clients.
- ✦ The task of configuring Net8 on the server consists largely of editing the listener.ora file and adding entries that define the databases available on that server, that specify the number of listeners to start, and that associate each database with a listener.
- ✦ The task of configuring Net8 on a client consists mainly of adding entries to the sqlnet.ora and tnsnames.ora files. Oracle provides two utilities to make this task easier. Net8 Easy Config is a wizard-like assistant that automates the task of adding a new Net8 service name to the listener.ora file. Net8 Assistant is a GUI-based utility allowing you to change sqlnet.ora settings, as well as edit entries in listener.ora.
- ✦ The Listener Control program is used on the server to start and stop the Net8 listeners. Listener Control is also used to report the current status of the various listeners.

- The best way to troubleshoot Net8 connection problems is to work your way up through each layer of networking, beginning with the physical layer. If your physical connection is good, check the connectivity via the network protocol that you are using. Lastly, check the Net8 connectivity, to be sure that you can reach the listener on the server to which you are trying to connect.
- The `ping` utility is a widely available operating system utility that allows you to check TCP/IP connectivity between a client and a server. The `tnsping` utility is an Oracle-supplied utility that allows you to check the connectivity between a Net8 client and a Net8 listener running on a server.

✥ ✥ ✥