Oracle® Database Express Edition

2 Day DBA

10*g* Release 2 (10.2)

B25107-01

October 2005

Beta Draft

ORACLE®

Oracle Database Express Edition 2 Day DBA, 10*g* Release 2 (10.2)

B25107-01

Primary Author: Steve Fogel

Contributing Author: Colin McGregor, Terri Winters, Marcie Young

Contributor: Mark Townsend, Bjørn Engsig, Santanu Datta, Michael Hichwa, Christina Cho, Matt McKerley, Graham Wood, Kant Patel, Ramanujam Srinivasan, Tammy Bednar, Ed Miner, Paul Lo, Roy Swonger, Rae Burns,Valarie Moore, Kathy Rich, Tirthankar Lahiri, Satish Panchumarthy, Srinivas Poovala, Aneesh Khandelwal

Contents

4 Managing Database Memory

5 Managing Database Storage

6 Managing Users and Security

7 Monitoring the Database

8 Viewing Database Configuration Settings

9 Importing and Exporting

10 Backing Up and Recovering the Database

11 Upgrading

Index

Preface

Welcome to *Oracle Database Express Edition 2 Day DBA*.

Audience

Oracle Database Express Edition 2 Day DBA is for anyone who wants to perform common day-to-day administrative tasks with Oracle Database 10g Express Edition. Prior knowledge or experience with managing databases is not required. The only requirement is a basic knowledge of computers.

Documentation Accessibility

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

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

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

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

Related Documentation

For more information, see these Oracle resources:

- *Oracle HTML DB User's Guide*
- *Oracle Database 10g Express Edition 2 Day Developer Guide*

- *Oracle Database SQL Reference*
- *Oracle Database Reference*
- *Oracle Database 10g Express Edition Installation Guide* for your platform

Conventions

The following text conventions are used in this document:

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

1

Getting Started with Oracle Database 10g Express Edition

Oracle Database 10g Express Edition (Oracle Database XE) is a free version of the world's most capable relational database. Oracle Database XE is easy to install, easy to manage, and easy to develop with.

With Oracle Database XE, you use an intuitive browser-based interface to:

- Administer the database
- Create tables, views, and other database objects
- Import, export, and view table data
- Run queries and SQL scripts
- Generate reports

Oracle Database XE includes Oracle HTML DB 2.1, a declarative, graphical development environment for creating database-centric Web applications. In addition to HTML DB 2.1, you can use all the popular Oracle and third-party languages and tools to develop your Oracle Database XE applications.

Oracle Database XE also includes the following command-line utilities:

- SQL*Plus, for entering SQL commands and running scripts
- SQL*Loader, for loading data into the database
- Data Pump and IMP/EXP for data import and export.

This section contains the following topics:

Accessing the Database Home Page

Oracle Database 10g Express Edition (Oracle Database XE) has a browser-based user interface for administering the database, running scripts and queries, building Web-based applications, and more. The starting point for this interface is the Database Home Page.

You can access the Database Home Page from your graphical desktop or by pointing your Web browser to a specific URL.

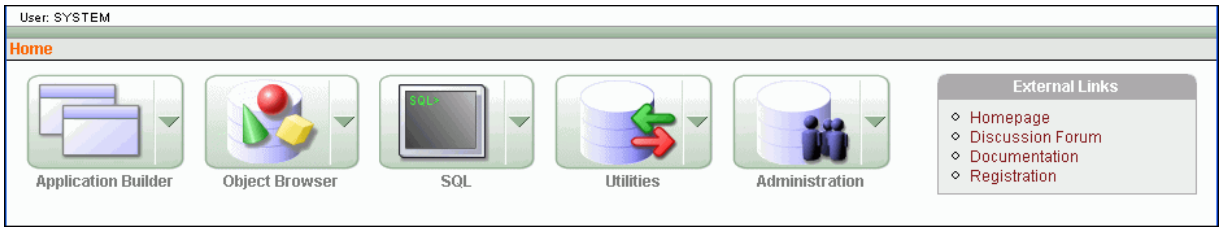
Accessing the Database Home Page from the Graphical Desktop

To access the Database Home Page from the graphical desktop:

1. Do one of the following:
   - On Windows: Click **Start**, point to **Programs**, point to **Oracle Database 10g Express Edition**, and then select **Go to Database Home Page**.
   - On Linux with Gnome: In the Applications menu, point to **Oracle Database 10g Express Edition**, and then select **Go to Database Home Page**.
   - On Linux with KDE: Click the icon for the K Menu, point to **Oracle Database 10g Express Edition**, and then select **Go to Database Home Page**.
2. When the login page appears, log in to the database using a valid database username and password.

   To log in as an administrator, log in with username `SYSTEM`, and supply the password that you specified during installation (Windows platform) or configuration (Linux platform).

   Upon successful login, the Database Home Page appears.



Accessing the Database Home Page with your Web Browser

To access the Database Home Page with your browser:

1. Point your Web browser to the following URL:

   `http://host:port/htmldb`

   where:
   - *`host`* is the host name or IP address of the computer where Oracle Database XE is installed.
   - *`port`* is the TCP port number for HTTP connection requests. Normally has a value of 8080. You may have changed this value during installation (Windows platform) or configuration (Linux platform).

   For example, if you installed Oracle Database XE on a computer with the host name `dbhost.mydomain.com`, and you installed with the default port number, you would access the Database Home Page at this URL:

   `http://dbhost.mydomain.com:8080/htmldb`

Note: If Oracle Database XE and your browser are running on the same computer, you may also be able to use the following URL:

```
http://localhost:port/htmldb
```

An alternative is to use the following URL, which uses the TCP/IP loopback address:

```
http://127.0.0.1:port/htmldb
```

2. When the login page appears, log in to the database using a valid database username and password.

   To log in as an administrator, log in with username `SYSTEM`, and supply the password that you specified during installation (Windows platform) or configuration (Linux platform).

   Upon successful login, the Database Home Page appears.

Getting Help

You can access context sensitive online help in the following two ways:

- For overall help with the current page, click the **Help** icon at the upper right of the page.

  This opens the help window. In addition to viewing the help information specific to the task at hand, you can browse the table of contents that is always displayed in the left pane of the Help window. A search facility in the Help window enables you to search all online help topics.

- For individual data fields on the page, position the cursor over the field label until a question mark appears, and then click the field label.

Navigating the Graphical User Interface

This section describes alternative methods for navigating between pages in the Oracle Database XE graphical user interface. It contains the following topics:

- Navigating Using Icons or Dropdown Menus on page 1-3
- Navigating Using Breadcrumbs on page 1-4

See Also: "Accessing the Database Home Page" on page 1-1

Navigating Using Icons or Dropdown Menus

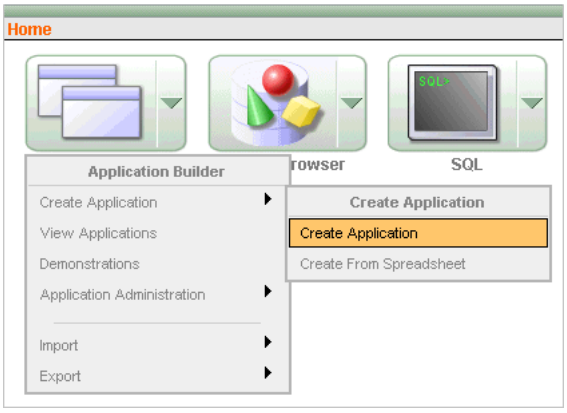
You can navigate the Oracle Database XE graphical user interface by clicking the large icons on the Database Home Page and on other navigation pages. When using these icons, you have two options:

- **Clicking the icon**—Click the icon to go to the page indicated by the icon name.

  You may have to click a number of these icons, descending one page at a time in the page hierarchy, before you reach your destination page.

- **Selecting from the icon's dropdown menu**—Click the down arrow on the right side of the icon to view a dropdown menu, and then select an option from the menu or from any of its submenus.

This is a more direct method of reaching some pages.



Navigating Using Breadcrumbs

Breadcrumbs (also called locator links) appear at the top of every page in the Oracle Database XE graphical user interface. Each breadcrumb "trail" indicates where the current page is in the hierarchy of pages. You can use breadcrumbs to instantly link to the previous page or to any of the pages above the current page in the hierarchy. In the following example, clicking **Home** takes you to the Database Home Page.

Connecting to the Database

This section discusses the following ways to connect to Oracle Database 10g Express Edition (Oracle Database XE):

- Connecting with SQL*Plus and other Oracle utilities
- Connecting with your application

Your application, SQL*Plus, and other Oracle utilities (such as SQL*Loader) are all referred to as **client applications**. Oracle Database XE supports connections between client applications and the database either locally (where the client application and Oracle Database XE are on the same computer), or remotely (where the client application and Oracle Database XE are on different computers) over a TCP/IP network. Any computer running a client application must have Oracle client software installed. Oracle Database XE accepts connections from all of the following types of Oracle client software:

- Oracle Database Express Edition Client (Oracle Database XE Client)

  When you install Oracle Database XE, Oracle Database XE Client is also installed. You can install Oracle Database XE Client separately on remote computers. It is available at `http://www.oracle.com/technology/xe`.

- Instant Client

  Instant Client is available at `http://www.oracle.com/technology/tech/oci/instantclient/instantclient.html`

- Oracle client software for Oracle Database Enterprise Edition or Standard Edition (all supported versions of 8.x, 9.x, and 10.x)

This section contains the following topics:

See Also:

- Chapter 3, "Managing Network Connections" on page 3-1 for information on how Oracle Database XE accepts connection requests over the network.

Connecting with SQL*Plus

This section describes how to connect to Oracle Database XE with SQL*Plus. It includes the following topics:

See Also: "Logging In as an Administrator" on page 6-5

About Local and Remote Connections

The method that you use to connect to Oracle Database XE with SQL*Plus and other Oracle command-line utilities depends on whether you are initiating a **local** connection or a **remote** connection.

Local Connection

Connecting locally means running SQL*Plus (or any other Oracle command-line utility) on the same host where Oracle Database XE is installed (the "Oracle Database XE host") and then initiating a database connection from SQL*Plus (or other utility). To connect locally, you must supply only a database username and password. Examples of local connections include the following:

- Oracle Database XE running on Linux—From the system console of the Oracle Database XE host, in a terminal session, you log in with operating system credentials for that host, and then start SQL*Plus.
- Oracle Database XE running on Linux—From a remote computer, you start a ssh (or telnet) session to the Oracle Database XE host, log in with operating system credentials for the Oracle Database XE host, and then start SQL*Plus on the Oracle Database XE host from the ssh (or Telnet) session.
- Oracle Database XE running on Windows—At the Oracle Database XE host, you open a command window and start SQL*Plus.

Before making a local connection on Linux, you must set environment variables. See "Setting Environment Variables" on page 1-6 for more information.

Remote Connection

Connecting remotely means running SQL*Plus (or any other Oracle command-line utility) on a computer other than the Oracle Database XE host, and then initiating a database connection from SQL*Plus (or other utility). The remote computer must have Oracle client software installed. The client software includes Oracle Net, which is the

Oracle network software layer that enables client applications on one computer to connect to databases on another computer over a network.

To connect remotely, you must supply not just a username and password, but a complete Oracle Net **connect string**. In addition to the database username and password, a connect string includes a host name or host IP address, an optional TCP port number, and possibly a database service name. These additional parameters are required to help Oracle Net find the right host computer and connect to Oracle Database XE.

Before making a remote connection from Linux, you must set environment variables. See "Setting Environment Variables" on page 1-6 for more information.

See Also:

- "Connecting Locally with SQL*Plus" on page 1-8
- "Connecting Remotely with SQL*Plus" on page 1-8

Setting Environment Variables

SQL*Plus and other Oracle utilities retrieve configuration information from operating system environment variables. This section explains how to set these environment variables.

Setting Environment Variables on the Windows Platform On the Windows platform, environment variables are automatically set for you. You need not take any action involving environment variables before initiating a connection.

Although you do not have to set the `ORACLE_HOME` environment variable, it is sometimes helpful to know the location of the Oracle home directory. The Windows default value for `ORACLE_HOME` is:

```
C:\oraclexe\app\oracle\product\10.2.0\server\
```

Setting Environment Variables on the Linux Platform On the Linux platform, before initiating a connection with SQL*Plus or other Oracle Utilities, you must set environment variables. The list of environment variables that you must set depends on the Oracle client software you are using, and whether you are connecting locally or remotely. This section describes only the following two scenarios:

- Connecting locally
- Connecting remotely from Oracle Database Express Edition Client (Oracle Database XE Client).

Table 1–1 on page 1-6 lists the environment variables that you must set for each of these scenarios. Table 1–2 on page 1-7 provides environment variable descriptions and required values.

Table 1–1 Required Linux Environment Variables for Connecting with Oracle Utilities

Connection Type	Required Environment Variables

Local	`ORACLE_SID`
	`ORACLE_HOME`
	`PATH`
	`NLS_LANG`

Remote, using Oracle Database XE Client	`PATH`
	`NLS_LANG`
	`SQLPATH`

Table 1–2 Environment Variable Descriptions and Values for Linux

Variable Name	Description	Required Value
ORACLE_SID	Oracle Instance ID	XE
ORACLE_HOME	Oracle home directory	/usr/lib/oracle/xe/app/oracle/product/10.2.0/server
PATH	Search path for executables. (Must add $ORACLE_HOME/bin to the path.)	For Bourne, Korn, or Bash shell: $ORACLE_HOME/bin:$PATH For C shell: $ORACLE_HOME/bin:${PATH}
NLS_LANG	Locale (language and territory used by client applications and the database; character set used by client applications)	(The desired language, territory, and character set. See *Oracle Database Express Edition Installation and Licensing Guide for Linux* for details.) Defaults to AMERICAN_AMERICA.US7ASCII
SQLPATH	Search path used by SQL*Plus for *.sql scripts. Must include the location of the site profile script, glogin.sql.	$ORACLE_HOME/sqlplus/admin

Example

The following are the Bash shell commands that set the required environment variables for a local connection on a Linux installation in the United States:

```
ORACLE_HOME=/usr/lib/oracle/xe/app/oracle/product/10.2.0/server;export ORACLE_HOME
ORACLE_SID=XE;export ORACLE_SID
PATH=$PATH:$ORACLE_HOME/bin;export PATH
NLS_LANG=AMERICAN_AMERICA.AL32UTF8;export NLS_LANG
```

Environment Variable Scripts

Table 1–3 lists the shell scripts that ship with Oracle Database XE. You can use them to easily set environment variables. The scripts are located in $ORACLE_HOME/bin.

Table 1–3 Shell Scripts for Setting Environment Variables

Script Names	Variables Set by the Scripts
oracle_env.sh oracle_env.csh	ORACLE_SID ORACLE_HOME PATH
nls_lang.sh nls_lang.csh	NLS_LANG

You can invoke these scripts from within dot files so that environment variables are set automatically each time that you log in or start a new terminal session (start a new shell). The following is an example of a command that you can add to the .cshrc file in your home directory:

```
source /usr/lib/oracle/xe/app/oracle/product/10.2.0/server/bin/oracle_env.csh
```

See Also:

- "About Local and Remote Connections" on page 1-5

Connecting Locally with SQL*Plus

Connecting locally means running SQL*Plus and Oracle Database XE on the same computer. There are two ways to start a local connection with SQL*Plus:

- From the graphical desktop
- From a terminal session (Linux) or command window (Windows)

Starting SQL*Plus from the Graphical Desktop

To start SQL*Plus from the graphical desktop and connect locally:

1. Do one of the following:
   - On Windows: Click **Start**, point to **Programs**, point to **Oracle Database 10g Express Edition**, and then select **SQL Command Line**.
   - On Linux with Gnome: In the Applications menu, point to **Oracle Database 10g Express Edition**, and then select **SQL Command Line**.
   - On Linux with KDE: Click the icon for the K Menu, point to **Oracle Database 10g Express Edition**, and then select **SQL Command Line**.

   A SQL*Plus command window opens.
2. At the SQL*Plus prompt, enter the following command:

   ```
   connect username/password
   ```

Starting SQL*Plus from a Terminal Session or Command Window

To start SQL*Plus from a terminal session or command window and connect locally:

1. If not already open, open a terminal session (Linux) or a command window (Windows).
2. If on Linux, set environment variables as described in "Setting Environment Variables on the Linux Platform" on page 1-6.
3. Enter the following command at the operating system prompt:

   ```
   sqlplus /nolog
   ```
4. At the SQL*Plus prompt, enter the following command:

   ```
   connect username/password
   ```

See Also: "Connecting Remotely with SQL*Plus"

Connecting Remotely with SQL*Plus

Connecting remotely means running SQL*Plus on one computer (the remote computer), and then initiating a connection to Oracle Database XE on a different computer.

To initiate a remote connection from SQL*Plus using the Oracle Database XE Client:

1. On the remote computer, start a terminal session (Linux) or open a command window (Windows.)

   If prompted for host credentials, log in to the remote computer.
2. Enter the following command at the operating system prompt:

   ```
   sqlplus /nolog
   ```
3. Enter this command at the SQL*Plus prompt:

```
connect username/password@[//]host[:port][/XE]
```

where:

- `//` is optional
- *`host`* is the host name or IP address of the computer that is running Oracle Database XE.
- *`port`* (optional) is the TCP port number that the Oracle Net listener is listening on. If not specified, the default port number 1521 is assumed.
- `/XE` (optional) provides the name of the database service to connect to. If omitted, Oracle Database XE Client appends a request for the *default database service*, which is configured during installation as `XE`.

Note: If you connect remotely from any Oracle client software other than Oracle Database XE Client, you must include the `/XE`.

See "About Network Connections and the Oracle Net Listener" on page 3-1 for more information.

Examples

In the following examples, Oracle Database XE is running on the host `mydbserver.mydomain.com`.

Example 1 This example initiates a remote connection from Oracle Database XE Client, using the default port number.

```
sqlplus /nolog
connect system/mypassword@mydbserver.mydomain.com
```

Example 2 This example initiates a remote connection from Oracle Database XE Client, using a non-default port number (1522):

```
sqlplus /nolog
connect system/mypassword@mydbserver.mydomain.com:1522
```

Example 3 This example initiates a remote connection from Oracle client software for Oracle Database Enterprise Edition, using the default port number.

```
sqlplus /nolog
connect system/mypassword@mydbserver.mydomain.com/XE
```

Connecting from Your Application

To initiate a connection from your application:

- Provide a connect string as described in the Oracle client software documentation for your development language.

  For example, if you are developing Java applications, consult *Oracle Database Express Edition Java 2 Day Developer Guide* for the correct function calls and required connect string for opening a connection to the database.

  See Also:

  - *Oracle Database Express Edition 2 Day Plus .NET Developer Guide*
  - *Oracle Database Express Edition 2 Day Plus PHP Developer Guide*

2

Starting up and Shutting Down

This section describes how to start up and shut down Oracle Database 10g Express Edition. It contains the following topics:

Starting Up the Database

Oracle Database 10g Express Edition (Oracle Database XE) starts up automatically immediately after installation and after each system reboot. Thus, there is no need to start up the database unless you previously shut it down.

You can start up the database with the graphical desktop or with SQL*Plus. Each of these methods is described in the following sections:

Starting Up the Database Using the Graphical Desktop

This section explains how to start up the database with the graphical desktops that are available in Windows and in the following two Linux windowing managers: KDE and Gnome. If your Linux computer is not running a windowing manager, or is running a windowing manager other than KDE or Gnome, you must start the database with SQL*Plus. See Starting Up the Database Using SQL*Plus on page 2-2 for instructions.

To start up the database using the graphical desktop:

1. Do one of the following:
   - On Windows: Log in to the Oracle Database XE host computer as a Windows administrator—that is, as a user who is a member of the Administrator group.
   - On Linux: Log in to the Oracle Database XE host computer as a user who can be authenticated with operating system authentication—that is, as a user who is a member of an operating system user group shown in Table 6–2 on page 6-5.

     This is normally the user `oraclexe`. See "Operating System Authentication" on page 6-4 for more information.
2. Do one of the following:
   - On Windows: Click **Start**, point to **Programs**, point to **Oracle Database 10g Express Edition**, and then select **Start Database**.

- On Linux with Gnome: In the Applications menu, point to **Oracle Database 10g Express Edition**, and then select **Start Database**.
- On Linux with KDE: Click the icon for the K Menu, point to **Oracle Database 10g Express Edition**, and then select **Start Database**.

Starting Up the Database Using SQL*Plus

You must use a local connection to the database to start it up with SQL*Plus. See "About Local and Remote Connections" on page 1-5 for more information.

To start up the database using SQL*Plus:

1. Log in to the Oracle Database XE host computer as a user that is a member of an operating system user group shown in Table 6–2 on page 6-5.

   On Linux, this is normally the `oraclexe` user, and on Windows, it is normally the user that installed Oracle Database XE. The database can now authenticate you with operating system (OS) authentication. See "Operating System Authentication" on page 6-4 for more information.

2. If not already opened, open a terminal session or command window.
3. Linux platform only: Ensure that environment variables are set properly.

   See "Setting Environment Variables on the Linux Platform" on page 1-6 for details.

4. At the operating system prompt, enter the following command:

   ```
   sqlplus / as sysdba
   ```

   Note that you do not need to supply a database user name and password, because you already provided database administrator credentials to the operating system.

5. At the SQL*Plus prompt, enter the following command:

   ```
   startup
   ```

   If the command is successful, it displays the following output. (System Global Area sizes will vary depending on the amount of physical memory in your Oracle Database XE host computer.)

   ```
   ORACLE instance started.

   Total System Global Area  599785472 bytes
   Fixed Size                  1220804 bytes
   Variable Size             180358972 bytes
   Database Buffers          415236096 bytes
   Redo Buffers                2969600 bytes
   Database mounted.
   Database opened.
   ```

Shutting Down the Database

Oracle Database 10g Express Edition (Oracle Database XE) shuts down automatically when you shut down the computer that hosts it.

Before shutting down Oracle Database XE, it is best to ensure that all users and applications have completed their work and logged out. See "Monitoring Sessions" on page 7-1 for information on how to view active database sessions.

If users or applications are still logged in when you begin a shutdown operation, the shutdown proceeds under the following conditions:

- No new connections are permitted, and no new transactions are allowed to be started.
- Any uncommitted transactions are rolled back.
- All users and applications are immediately disconnected.

You can shut down the database with the graphical desktop or with SQL*Plus. Each of these methods is described in the following sections:

- Shutting Down the Database Using the Graphical Desktop on page 2-3
- Shutting Down the Database Using SQL*Plus on page 2-3

Shutting Down the Database Using the Graphical Desktop

This section explains how to shut down the database with the graphical desktops that are available in Windows and in the following two Linux windowing managers: KDE and Gnome. If your Linux computer is not running a windowing manager, or is running a windowing manager other than KDE or Gnome, you must shut down the database with SQL*Plus.

To shut down the database using the graphical desktop:

1. Do one of the following:
   - On Windows: Log in to the Oracle Database XE host computer as a Windows administrator—that is, as a user who is a member of the Administrator group.
   - On Linux: Log in to the Oracle Database XE host computer as a user who can be authenticated with operating system authentication—that is, as a user who is a member of an operating system user group shown in Table 6–2 on page 6-5.

     This is normally the user `oraclexe`. See "Operating System Authentication" on page 6-4 for more information.
2. Do one of the following:
   - On Windows: Click **Start**, point to **Programs**, point to **Oracle Database 10g Express Edition**, and then select **Stop Database**.
   - On Linux with Gnome: In the Applications menu, point to **Oracle Database 10g Express Edition**, and then select **Stop Database**.
   - On Linux with KDE: Click the icon for the K Menu, point to **Oracle Database 10g Express Edition**, and then select **Stop Database**.

Shutting Down the Database Using SQL*Plus

You must use a local connection to the database to shut it down with SQL*Plus. See "About Local and Remote Connections" on page 1-5 for more information.

To shut down the database using SQL*Plus:

1. Log in to the Oracle Database XE host computer as a user that is a member of an operating system user group shown in Table 6–2 on page 6-5.

   On Linux, this is normally the `oraclexe` user, and on Windows, it is normally the user that installed Oracle Database XE. The database can now authenticate you with operating system (OS) authentication. See "Operating System Authentication" on page 6-4 for more information.
2. If not already opened, open a terminal session or command window.

3. Linux platform only: Ensure that environment variables are set properly.

   See "Setting Environment Variables on the Linux Platform" on page 1-6 for details.

4. At the operating system prompt, enter the following command:

   ```
   sqlplus / as sysdba
   ```

   Note that you do not need to supply a database user name and password, because you already provided database administrator credentials to the operating system.

5. At the SQL*Plus prompt, enter the following command:

   ```
   shutdown immediate
   ```

   Note that this command may take a short while to complete. If the command is successful, it displays the following output:

   ```
   Database closed.
   Database dismounted.
   ORACLE instance shut down.
   ```

   If after a number of minutes the command displays no output and appears to be making no progress, you can attempt the following, in order of preference:

   - Shut down the database using the graphical desktop.

     See "Shutting Down the Database Using the Graphical Desktop" on page 2-3 for instructions.

   - Restart the Oracle Database XE host computer.

     Restarting the host computer has no adverse effect on committed transactions.

   - Enter the `SHUTDOWN ABORT` command.

     Use this command only if other shutdown methods fail or cannot be attempted. The database must go through a recovery process when it starts up after a `SHUTDOWN ABORT` command. See *Oracle Database Administrator's Guide* for information on the `SHUTDOWN ABORT` command.

3

Managing Network Connections

This section explains how to manage network connections to the database. It includes the following topics:

See Also: "Connecting to the Database" on page 1-4

About Network Connections and the Oracle Net Listener

Oracle Database 10g Express Edition (Oracle Database XE) supports remote connections from client applications over the network. The client applications and the database communicate through Oracle Net, which is a software layer that resides both on the remote computer and on the Oracle Database XE host. Oracle Net establishes the connection between the client application and the database, and exchanges messages between them using the TCP/IP protocol. Oracle Net is automatically installed when you install Oracle Database XE and Oracle Database Express Edition Client.

Included with Oracle Net in an Oracle Database XE installation is the **Oracle Net listener**, commonly known as the listener. It is the host process that listens on specific TCP/IP ports for remote connection requests. When the listener receives a valid connection request from a client, it hands off the connection request to the database. The client and the database then communicate directly.

Table 3–1 lists the types of connection requests that the listener handles.

Table 3–1 Types of Connection Requests Handled by the Listener

Connection Request Type	Default TCP Port Number	Used For
Database	1521	Submitting SQL and PL/SQL statements for processing.

Table 3–1 (Cont.) Types of Connection Requests Handled by the Listener

Connection Request Type	Default TCP Port Number	Used For
HTTP	8080	■ Accessing the Database Home Page. See "Accessing the Database Home Page" on page 1-1 for more information. ■ Accessing the XML DB repository. XML DB is the Oracle Database XE feature that provides high-performance, native XML storage and retrieval. Through the XML DB repository, you can access XML data with the HTTP and WebDAV protocols. See *Oracle XML DB Developer's Guide* for more information.

Note: The listener can also handle FTP connection requests for the XML DB repository. For security reasons, FTP requests are disabled when you install Oracle Database XE. See *Oracle XML DB Developer's Guide* for more information.

You can disable or enable all remote connection requests by manually stopping or starting the listener. (The listener is automatically started when you install Oracle Database XE and when you reboot the Oracle Database XE host computer.) You can also leave database connections enabled and disable only HTTP requests. In addition, you can change the TCP port numbers that the listener listens on, either during the Oracle Database XE installation process, or at a later time after installation.

Note: The Windows installation process prompts for the port number for HTTP requests only if the default port number, 8080, is already in use. The Linux configuration script always prompts for HTTP port number.

See Also:

- "Starting and Stopping the Listener" on page 3-3
- "Disabling HTTP Connections" on page 3-7
- "Changing Listener Port Numbers" on page 3-5
- "Connecting to the Database" on page 1-4

Viewing Listener Status

You view listener status to determine if the listener is started and to check listener properties (such as the TCP/IP port numbers that the listener is listening on). You do so with the Listener Control (`lsnrctl`) utility.

To view listener status:

1. Do one of the following:
   - On Linux: Start a terminal session and log in to the Oracle Database XE host with the `oraclexe` user account.
   - On Windows: Log in to the Oracle Database XE host as the user who installed Oracle Database XE, and then open a command window.

2. On Linux, ensure that environment variables are set according to the instructions in "Setting Environment Variables on the Linux Platform" on page 1-6.

3. Enter the following command:

```
lsnrctl status
```

If the listener is not started, the command displays the following error messages:

```
TNS-12541: TNS:no listener
 TNS-12560: TNS:protocol adapter error
  TNS-00511: No listener
```

If the listener is started, the command displays a report that looks something like this:

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=stadh43)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 10.2.0.1.0 - Production
Start Date                27-SEP-2005 09:36:54
Uptime                    1 days 10 hr. 9 min. 36 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Default Service           XE
Listener Parameter File
/usr/lib/oracle/xe/app/oracle/product/10.2.0/server/network/admin/listener.ora
Listener Log File
/usr/lib/oracle/xe/app/oracle/product/10.2.0/server/network/log/listener.log
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stadh43.us.oracle.com)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stadh43.us.oracle.com)(PORT=8080))(Presentation=HTTP)
(Session=RAW))
Services Summary...
Service "PLSExtProc" has 1 instance(s).
  Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
Service "XE" has 1 instance(s).
  Instance "XE", status READY, has 1 handler(s) for this service...
Service "XEXDB" has 1 instance(s).
  Instance "XE", status READY, has 1 handler(s) for this service...
Service "XE_XPT" has 1 instance(s).
  Instance "XE", status READY, has 1 handler(s) for this service...
The command completed successfully
```

Starting and Stopping the Listener

The listener is configured to start automatically when you install Oracle Database 10g Express Edition (Oracle Database XE), and whenever the computer running Oracle Database XE is restarted. The following are reasons why you may want to stop and restart the listener:

- To recover from system errors
- To temporarily block remote connection requests

  You stop the listener to disable remote connection requests, and restart the listener to enable them.

- To change the TCP port number that the listener listens on

  See "Changing Listener Port Numbers" on page 3-5 for more information.

Starting the Listener

To start the listener:

1. Do one of the following:
   - On Linux: Start a terminal session and log in to the Oracle Database XE host with the `oraclexe` user account.
   - On Windows: Log in to the Oracle Database XE host as the user who installed Oracle Database XE, and then open a command window.
2. On Linux, ensure that environment variables are set according to the instructions in "Setting Environment Variables on the Linux Platform" on page 1-6.
3. Enter the following command:

   ```
   lsnrctl start
   ```

   If successful, the command displays the report shown in "Viewing Listener Status" on page 3-2.

   Note: If you stop and then start the listener while the database is running, it may take a minute or so for the database to reregister with the listener and to begin accepting connections. To determine if the database is ready to accept connections, run the `lsnrctl status` command repeatedly until you see the following lines in the report:

   ```
   Service "XE" has 1 instance(s).
     Instance "XE", status READY, has 1 handler(s) for this service...
   ```

   After starting the listener, you can cause the database to immediately register with the listener by doing the following: log in as user `SYSTEM` with SQL*Plus, and enter the following command:

   ```
   ALTER SYSTEM REGISTER;
   ```

Stopping the Listener

To stop the listener:

1. Do one of the following:
   - On Linux: Start a terminal session and log in to the Oracle Database XE host with the `oraclexe` user account.
   - On Windows: Log in to the Oracle Database XE host as the user who installed Oracle Database XE, and then open a command window.
2. On Linux, ensure that environment variables are set according to the instructions in "Setting Environment Variables on the Linux Platform" on page 1-6.
3. Enter the following command:

   ```
   lsnrctl stop
   ```

   The command displays the following output if successful:

   ```
   Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=myhost)(PORT=1521)))
   The command completed successfully
   ```

   The command displays the following output if the listener was already stopped:

   ```
   Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=myhost)(PORT=1521)))
   ```

```
TNS-12541: TNS:no listener
 TNS-12560: TNS:protocol adapter error
  TNS-00511: No listener
   Linux Error: 111: Connection refused
```

See Also: "Viewing Listener Status" on page 3-2

Changing Listener Port Numbers

You would need to change a default listener port number only if there were a port number conflict with another TCP/IP service. You are given the opportunity to change listener port numbers during installation (Windows) or configuration (Linux). This section explains how to change port numbers after installation or configuration. It contains the following topics:

- "Changing the Listener Port Number for Database Connection Requests" on page 3-5
- "Changing the Listener Port Number for HTTP Connection Requests" on page 3-6

See Also: "Disabling HTTP Connections" on page 3-7

Changing the Listener Port Number for Database Connection Requests

If you change the listener port number for database connection requests, you must ensure that all future database connection requests use the new port number. This means that connection requests such as those discussed in "Connecting Remotely with SQL*Plus" on page 1-8 must explicitly include the port number.

For example, if you change the port number for database connection requests to 1522, subsequent SQL*Plus `connect` statements must be similar to the following (assuming a connection from Oracle Database Express Edition Client):

```
connect system/mypassword@mydbserver.mydomain.com:1522
```

Example: Changing Listener Port Number for Database Connection Requests

Assume that your Oracle Database XE host is named `mydbhost.mydomain.com` and that you want to install a new software package on this host that requires TCP port number 1521. Assume also that the port number for that software package is not configurable, and that you must therefore resolve the port number conflict by reconfiguring Oracle Database XE. You decide to change the listener port number for database connection requests to 1522.

To change the listener port number for database connection requests to 1522:

1. Stop the listener.

   See "Starting and Stopping the Listener" on page 3-3 for instructions.

2. Open the file `listener.ora` with a text editor.

   Table 3–2 shows the location of this file on each platform.

Table 3–2 Location of the listener.ora File

Platform	Location
Linux	/usr/lib/oracle/xe/app/oracle/product/10.2.0/server/network/admin/
Windows	c:\oraclexe\app\oracle\product\10.2.0\server\network\admin\

3. Locate the following section of the file:

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = stadh43)(PORT = 1521))
    )
  )
```

4. Change the text `(PORT = 1521)` to `(PORT = 1522)`.
5. Save the modified `listener.ora` file.
6. Start the listener.

   See "Starting and Stopping the Listener" on page 3-3 for instructions.
7. With SQL*Plus, log in to the database as user `SYSTEM`.

   See "Logging In as User SYSTEM" on page 6-6 for instructions.
8. Enter the following two SQL*Plus commands:

```
alter system set local_listener =
"(address=(protocol=tcp)(host=mydbhost.mydomain.com)(port=1522))";

alter system register;
```

9. Exit SQL*Plus and run the `lsnrctl status` command to verify the port number change.

   The new port number should be displayed in the Listening Endpoints Summary section of the status report, and the report should include the following lines:

```
Service "XE" has 1 instance(s).
  Instance "XE", status READY, has 1 handler(s) for this service...
```

Changing the Listener Port Number for HTTP Connection Requests

If you change the listener port number for HTTP connection requests, you must ensure that all future HTTP connection requests use the new port number.

For example, if you change the listener port number for HTTP requests to 8087, you must use the following URL to access the Database Home Page:

```
http://host:8087/htmldb
```

where *host* is the host name or IP address of the computer where Oracle Database XE is installed.

To change the listener port number for HTTP connection requests:

1. Ensure that the listener is started.

   See "Viewing Listener Status" on page 3-2 and "Starting and Stopping the Listener" on page 3-3 for instructions.
2. Start SQL*Plus and connect to the database as user `SYSTEM`.

   See "Connecting Locally with SQL*Plus" on page 1-8 for instructions. You must supply the `SYSTEM` password. You set this password upon installation (Windows) or configuration (Linux) of Oracle Database 10g Express Edition.
3. At the SQL*Plus prompt, enter the following command:

```
exec dbms_xdb.sethttpport(nnnn);
```

where *nnnn* represents the new port number to use for HTTP connection requests. Be certain that you select a port number that is not already in use.

For example, to use port number 8087 for HTTP connection requests, enter this command:

```
exec dbms_xdb.sethttpport(8087);
```

4. Exit SQL*Plus and view listener status to verify the port number change.

   See "Viewing Listener Status" on page 3-2 for instructions. The new port number is displayed in the Listening Endpoints Summary section of the status report.

Disabling HTTP Connections

Under normal circumstances, you would keep HTTP connection requests enabled for Oracle Database 10g Express Edition (Oracle Database XE), because HTTP connections are required to use the Oracle Database XE graphical user interface. However, if you have security concerns about enabling HTTP connections on the Oracle Database XE host, you can disable HTTP connection requests while leaving database connection requests enabled.

To disable HTTP connections to Oracle Database XE:

- Change the listener port number for HTTP connection requests to 0 by connecting to the database with SQL*Plus as user `SYSTEM` and issuing the following command:

  ```
  exec dbms_xdb.sethttpport(0);
  ```

  See "Changing the Listener Port Number for HTTP Connection Requests" on page 3-6 for detailed instructions.

To reenable HTTP connections, change the listener port number for HTTP connection requests to a non-zero number. Be certain that you select a port that is not already in use.

4

Managing Database Memory

This section provides an overview of the Oracle Database 10g Express Edition memory structure and describes how to adjust memory allocation.

The following topics are covered:

- Overview of Memory Management
- Example: Changing SGA and PGA Sizes

Overview of Memory Management

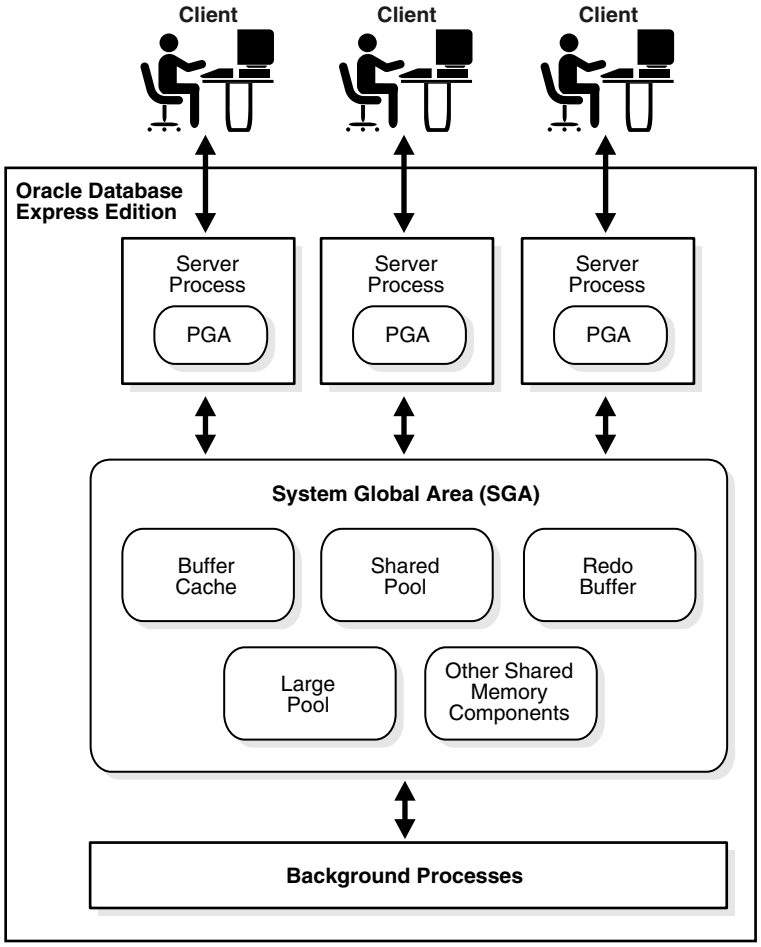
To support database operation, Oracle Database 10g Express Edition (Oracle Database XE) needs to start a set of processes, called background processes, and needs to allocate some memory in the host. The background processes and allocated memory together make up an **Oracle instance**.

There are two types of memory that the Oracle instance allocates:

- System Global Area (SGA)—A shared memory area that contains data buffers and control information for the instance. The SGA is divided into separate buffer areas and data pools. These are described in "System Global Area Components" on page 4-2.
- Program Global Area (PGA)—A memory area used by a single Oracle server process. A server process is a process that services a client's requests. Oracle Database XE creates a new server process whenever it receives a new connection request. Each server process has its own private PGA area. The PGA is used to process SQL statements and to hold logon and other session information.

Figure 4–1 illustrates memory allocation in Oracle Database XE.

Figure 4–1 Memory Allocation in Oracle Database XE



The amount of memory allocated to the SGA and PGA directly affects the performance of your database. The SGA and PGA sizes are configured automatically when you install Oracle Database XE. See "SGA and PGA Sizes" on page 4-3 for information on how these default sizes are calculated, and for a discussion of when you might change them.

See Also:

- "Example: Changing SGA and PGA Sizes" on page 4-3

System Global Area Components

The SGA has several subcomponents, as listed in the following table. Oracle Database XE automatically tunes the individual sizes of these subcomponents for optimal performance.

Component	Description
Buffer cache	The buffer cache is the component of the SGA that acts as the buffer to store any data being queried or modified. All user processes connected to the database share access to the buffer cache. The buffer cache helps avoid repeated access from physical disk, a time consuming operation.

Component	Description
Shared pool	The shared pool caches operational information and code that can be shared among users. For example: ▪ SQL statements are cached so that they can be reused. ▪ Information from the data dictionary, such as user account data, table and index descriptions, and privileges, is cached for quick access and reusability. ▪ Stored procedures are cached for faster access.
Redo log buffer	The redo log buffer improves performance by caching redo information (used for instance recovery) until it can be written at once and at a more opportune time to the physical redo log files that are stored on disk. Redo information and redo log files are discussed in "Redo Log Files" on page 5-2.
Large pool	The large pool is an optional area that is used for buffering large I/O requests for various server processes.

SGA and PGA Sizes

The default sizes for the SGA and PGA are set upon installation, based on the total amount of physical memory in your system.

The only circumstances under which you should need to change SGA and PGA sizes are the following:

- You add physical memory to the computer running Oracle Database XE and want to allocate more to the database

  In this case, increase both the SGA and PGA sizes, maintaining roughly the original ratio of SGA size to PGA size.

- You receive an error due to insufficient memory

  If the error message indicates insufficient memory for an SGA component, increase the SGA size. Examples of such errors include the following:

  ```
  ORA-04031: unable to allocate n bytes of shared memory
  ORA-00379: no free buffers available in buffer pool...
  ```

  If the error message indicates insufficient memory for a process, increase the PGA size. An example of such an error is the following:

  ```
  ORA-04030: out of process memory when trying to allocate n bytes
  ```

  If you are not sure whether the insufficient memory error involves the SGA or PGA, increase both SGA and PGA sizes, maintaining roughly the original ratio of SGA size to PGA size.

For SGA size changes, you must shut down and restart the database for the changes to take effect. For PGA size changes, there is no need to restart the database.

Example: Changing SGA and PGA Sizes

Assume that you just upgraded the computer running Oracle Database 10g Express Edition (Oracle Database XE) from 512 MB to 1 GB of system memory, and that you want to allocate roughly half (256 MB) of the additional memory to the database. Of this 256 MB, you want to add 192 MB to the SGA and 64 MB to the PGA.

To change SGA and PGA sizes:

1. Access the Oracle Database XE graphical user interface.

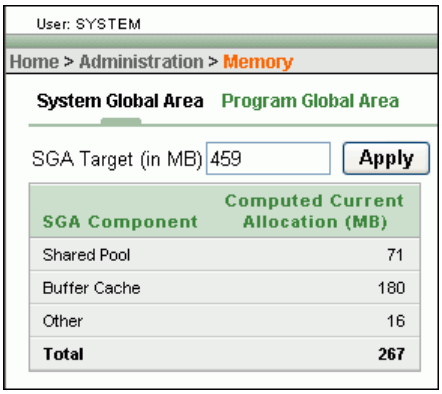
   See "Accessing the Database Home Page" on page 1-1 for instructions.

2. Click the **Administration** icon, and then click the **Memory** icon.

3. If prompted for administrator credentials, enter the `SYSTEM` username and password or another administrator username and password, and then click **Login**.

   See "About Administrative Accounts and Privileges" on page 6-3 for more information.

4. On the Memory page, in the SGA Target field, enter **459**.



   The value 459 is the sum of the current SGA size (267) plus 192.

5. Click **Apply**.

   A confirmation message appears.

6. Click the **Program Global Area** hyperlink to switch to the PGA view of the Memory page.

7. In the Aggregate PGA Target field, enter **169**, and then click **Apply**.

8. At the next convenient time, shut down and restart the database to enable the SGA size changes to take effect.

See Also: "Overview of Memory Management" on page 4-1

5

Managing Database Storage

This section describes the storage structures of your database, and explains how to monitor the amount of storage that is in use and available. It contains the following topics:

- Understanding the Storage Structure of the Database
- Viewing Tablespaces

Understanding the Storage Structure of the Database

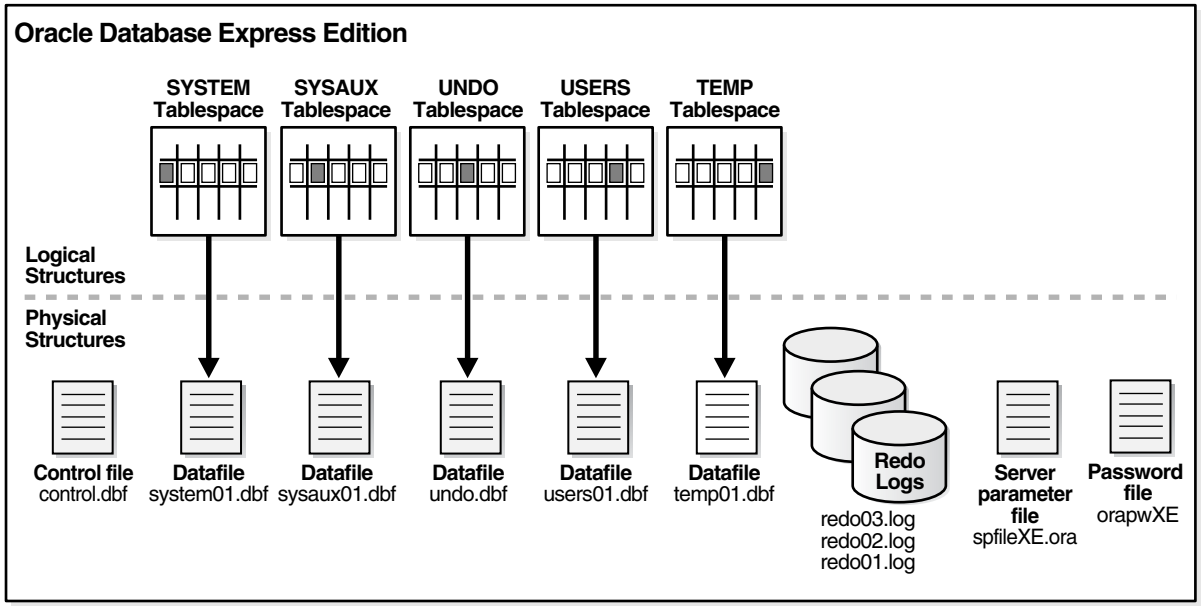
Oracle Database 10g Express Edition (Oracle Database XE) is comprised of both physical and logical storage structures. Physical structures are those that can be seen and operated on from the operating system, such as the physical files that store data on disk. Logical structures such as **tablespaces** are created and recognized by the database only, and are not known to the operating system. A tablespace is a logical grouping of physical datafiles, and is the primary structure by which the database manages storage.

Oracle Database XE completely automates the management of its logical and physical structures. You use the Oracle Database XE graphical user interface to monitor these structures, mostly to understand how much storage your applications have used so far, and how much free storage remains.

The following sections provide a closer look at the database storage structure:

- Control Files
- Datafiles
- Redo Log Files
- Undo
- Tablespaces
- Other Storage Structures

Refer to the following diagram as you review these sections.



Control Files

The control file is a small binary file that tracks the names and locations of the physical components of the database, and that maintains other control information. It is essential to the functioning of the database. For this reason, it is **multiplexed**—that is, multiple (typically two or three) identical copies are created upon installation, and are automatically maintained and kept in sync with each other by the database. If one controlfile becomes lost or damaged, the database can use the other.

Datafiles

Datafiles are the operating system files that hold the data within the database. The data is written to these files in an Oracle proprietary format that cannot be read by programs other than an Oracle database. Tempfiles are a special class of datafiles that are associated only with temporary tablespaces. Temporary tablespaces provide workspaces to help process queries.

Redo Log Files

Every Oracle database has a set of two or more redo log files. The set of redo log files is collectively known as the redo log for the database. A redo log is made up of redo entries, which are also called redo records.

The primary function of the redo log is to record all changes made to data. If a failure prevents modified data from being permanently written to the datafiles, then the changes can be obtained from the redo log, so that work is not lost.

To protect against a failure involving the redo log itself, Oracle allows a multiplexed redo log so that two or more identical copies of the redo log can be maintained on different disks. Multiplexing is accomplished by enabling you to create **groups** of redo log files. A group consists of a redo log file and its multiplexed copies. Each identical copy is said to be a **member** of that group. Each redo log group is defined by a number, such as group 1. The default installation of the Oracle Database XE database has three redo log groups of one member each.

The database log writer process writes redo records from the memory buffer to a redo log group until the group fills up or until you explicitly request a log switch. It then writes to the next group. The log writer performs this action in a circular fashion so that the oldest group is overwritten by the most recent redo records.

With Oracle Database XE, you should not normally need to manually request a log switch. See *Oracle Database Administrator's Guide* for more information.

Undo

When a transaction modifies the database, Oracle Database XE makes a copy of the original data before modifying it. The original copy of the modified data is called undo data. This information is necessary for the following reasons:

- To undo any uncommitted changes made to the database in the event that a rollback operation is necessary. A rollback operation can be the result of a user specifically issuing a rollback statement to undo the changes of a misguided or unintentional transaction, or it can be part of a recovery operation.
- To provide read consistency, which means that each user can get a consistent view of data, even while other uncommitted changes may be occurring against the data. For example, if a user issues a query at 10:00 a.m. and the query lasts for 15 minutes, then the query results should reflect the entire state of the data at 10:00 a.m. regardless of updates or inserts by other users during the query.

  See *Oracle Database Concepts* for a discussion of read consistency.
- To support Flashback Query, which enables you to view or recover older versions of data. See "Viewing and Restoring Historical Data with Flashback Query" on page 10-4 for more information.

Oracle Express automatically manages undo information in an undo tablespace. See "Tablespaces" on page 5-3 for more information.

Tablespaces

A database consists of one or more tablespaces. A tablespace is a logical structure created by and known only to Oracle Database XE. A tablespace consists of one or more physical datafiles or tempfiles.

There are various types of tablespaces, including the following:

- Undo tablespace

  Oracle Database XE transparently creates and manages undo data in this tablespace.
- Permanent tablespaces

  These tablespaces are used to store system and user data. All your application data is stored in a tablespace named `USERS`. By default this tablespace consists of a single datafile that automatically grows ("autoextends") as your applications store more data.
- Temporary tablespaces

  Temporary tablespaces improve the concurrence of multiple sort operations, reduce their overhead, or avoid space management operations altogether. Temporary tablespaces are the most efficient tablespaces for disk sorts. Oracle Database XE automatically manages storage for temporary tablespaces.

Some Tablespaces in the Database

Table 5–1 describes some of the tablespaces included in the database.

Table 5–1 Tablespaces and Descriptions

Tablespace	Description
SYSTEM	This tablespace is automatically created when Oracle Database XE is installed. It contains the data dictionary, which is the central set of tables and views used as a read-only reference for the database. It also contains various tables and views that contain administrative information about the database. These are all contained in the SYS schema, and can only be accessed by user SYS or other administrative users with the required privilege.
SYSAUX	This is an auxiliary tablespace to the SYSTEM tablespace, and is also automatically created upon installation. Some database components and products use this tablespace. The HR sample schema is also stored in the SYSAUX tablespace.
TEMP	This tablespace stores temporary data generated when processing SQL statements. For example, this tablespace is used for sort work space. Every database must have a temporary tablespace that is assigned to users as their temporary tablespace. In Oracle Database XE, the TEMP tablespace is specified as the default temporary tablespace for every user.
UNDO	This is the tablespace used by the database to store undo information.
USERS	This tablespace is used to store permanent user objects and data. In Oracle Database XE, USERS is the assigned default tablespace for all users except the SYS user, which has the default permanent tablespace of SYSTEM.

Other Storage Structures

Other storage structures that exist in Oracle Database XE include the server parameter file and the password file.

Server Parameter File

The server parameter file (SPFILE) contains initialization parameters that Oracle Database XE instance uses at startup to determine the settings and runtime resources for the database. You can view current parameter settings with the Oracle Database XE graphical user interface. See "Viewing Database Initialization Parameters" on page 8-1 for more information.

Password File

Oracle Database XE uses a password file to authenticate a user who is logging in remotely as user SYS. The SYS user can then perform privileged administration functions from a remote workstation. The password file contains the SYS password (encrypted). Whenever you change the password for SYS, the password file is automatically updated.

The password file is automatically created when you install Oracle Database XE.

Note: Under normal circumstances you should never log in to Oracle Database XE as user SYS, either locally or remotely.

Viewing Tablespaces

You can use the Oracle Database XE graphical user interface to view a list of tablespaces in the database, view tablespace properties, and view datafile properties.

To view Oracle Database XE tablespaces:

1. Access the Database Home Page.

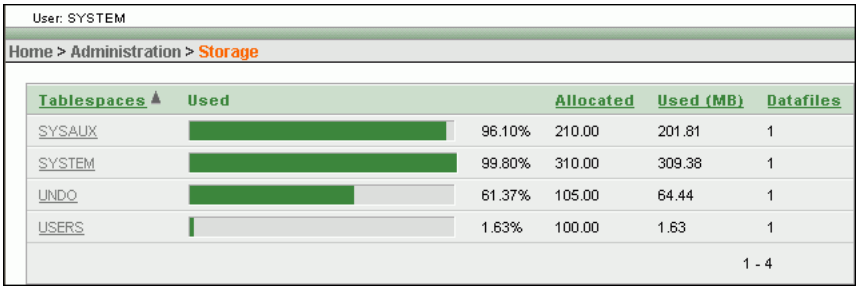
   See "Accessing the Database Home Page" on page 1-1 for instructions.

2. Click the **Administration** icon, and then click the **Storage** icon.

3. If prompted for administrator credentials, enter the `SYSTEM` username and password or another administrator username and password, and then click **Login**.

   See "About Administrative Accounts and Privileges" on page 6-3 for more information.

4. On the Storage page, click a tablespace name to view information on that tablespace's datafiles.



See Also: "Understanding the Storage Structure of the Database" on page 5-1

6

Managing Users and Security

Users access Oracle Database 10g Express Edition through database user accounts. Some of these accounts are automatically created administrative accounts—accounts with database administration privileges. You log in with these accounts to create and manage other user accounts, and to maintain database security.

This section contains the following topics:

About User Accounts

A user account is identified by a username and defines the user's attributes, including the following:

- Password for database authentication
- Privileges and roles
- Default tablespace for database objects
- Default temporary tablespace for query processing
- Tablespace quota

When you create a user, you are also implicitly creating a **schema** for that user. A schema is a logical container for the database objects (such as tables, views, triggers, and so on) that the user creates. The schema name is the same as the username, and can be used to unambiguously refer to objects owned by the user. For example, `HR.EMPLOYEES` refers to the table named `EMPLOYEES` in the `HR` schema. (The `EMPLOYEES` table is owned by `HR`.) The terms "database object" and "schema object" are used interchangeably.

When you drop a user, you must either first drop all the user's schema objects, or use the **cascade** feature of the drop operation, which simultaneously drops a user and all of his schema objects.

This section contains these topics:

- Internal User Accounts on page 6-3

See Also: "Oracle Database 10g Express Edition Predefined User Accounts" on page 6-17

User Privileges and Roles

When creating a user, you grant privileges to enable the user to connect to the database, to run queries and make updates, and to create schema objects. There are two main types of user privileges:

- **System privileges**—A system privilege is the right to perform a particular action, or to perform an action on any schema objects of a particular type. For example, the privileges to create tables and to delete the rows of any table in a database are system privileges.
- **Object privileges**—An object privilege is a right to perform a particular action on a specific schema object. Different object privileges are available for different types of schema objects. The privilege to delete rows from the `DEPARTMENTS` table is an example of an object privilege.

Managing and controlling privileges is made easier by using **roles**, which are named groups of related privileges. You create roles, grant system and object privileges to the roles, and then grant roles to users. Unlike schema objects, roles are not contained in any schema.

Table 6–1 lists three roles that are predefined in Oracle Database 10g Express Edition. You can grant these roles when you create a user with the Oracle Database XE graphical user interface.

Table 6–1 Oracle Database 10g Express Edition Predefined Roles

Role Name	Description
`CONNECT`	Enables a user to connect to the database. Assign this role to any user or application that needs database access.
`RESOURCE`	Enables a user to create schema objects in his own schema. Assign this role only to developers and power users.
`DBA`	Enables a user to perform most administration functions, including creating users and granting privileges; creating and granting roles; creating and dropping schema objects in other users' schemas; and more. Does not include the privileges to start up or shut down the database. The `DBA` role is by default granted to user `SYSTEM`.

See Also:

- *Oracle Database Security Guide* for more information on privileges and roles
- *Oracle Database SQL Reference* for tables of system privileges, object privileges, and predefined roles.
- "Creating Users" on page 6-11
- "Oracle Database 10g Express Edition Predefined User Accounts" on page 6-17
- "About Administrative Accounts and Privileges" on page 6-3

Internal User Accounts

Certain user accounts are created automatically for database administration. Examples are SYS and SYSTEM. Other accounts are automatically created just so that individual Oracle Database XE features or products can have their own schemas. An example is the CTXSYS account, which is used by the Oracle Text product. Oracle Text is used to index the Oracle Database XE online help. The help index is stored in the CTXSYS schema in the database.

These automatically created accounts are called **internal user accounts**. You can view them with the Oracle Database XE graphical user interface, but cannot modify them.

The only internal account that you may log in with is the SYSTEM account. Do not attempt to log in with other internal accounts. See "The SYS and SYSTEM Users" on page 6-3 for more information on the SYSTEM account.

About Administrative Accounts and Privileges

Administrative accounts and privileges enable you to perform administrative functions like managing users, managing database memory, and starting up and shutting down the database.

This section contains the following topics:

See Also:

- "About User Accounts" on page 6-1
- "Logging In as an Administrator" on page 6-5

The SYS and SYSTEM Users

The following two administrative user accounts are automatically created when you install Oracle Database 10g Express Edition (Oracle Database XE). They are both created with the password that you supplied upon installation (Windows operating systems) or upon configuration (Linux operating systems).

- SYSTEM

  This is the user account that you log in with to perform all administrative functions other than starting up and shutting down the database.

  You can log in as SYSTEM with the Oracle Database XE graphical user interface and with SQL*Plus.

- SYS

  All base tables and views for the database data dictionary are stored in the SYS schema. These base tables and views are critical for the operation of Oracle Database XE. To maintain the integrity of the data dictionary, tables in the SYS schema are manipulated only by the database. They should never be modified by any user or database administrator. You must not create any tables in the SYS schema.

  The only way to log in as user SYS is with SQL*Plus, although under normal circumstances, there is no reason to do so. It is not possible to log in as SYS with the Oracle Database XE graphical user interface.

See Also:

- "The SYSDBA System Privilege" on page 6-4
- "Operating System Authentication" on page 6-4
- "Changing Administrative User Passwords" on page 6-7

The SYSDBA System Privilege

SYSDBA is a system privilege that is assigned only to user SYS. It enables SYS to perform high-level administration tasks such as starting up and shutting down the database.

Although under normal circumstances it is not necessary to log in to the database as user SYS, if you want to log in as SYS, you must use SQL*Plus, and you must connect to the database "AS SYSDBA." Connecting AS SYSDBA invokes the SYSDBA privilege. If you omit the AS SYSDBA clause when logging in as user SYS, SQL*Plus rejects the login attempt.

The following example illustrates how to initiate a local connection to the database with the SYSDBA privilege:

```
$ sqlplus /nolog
SQL > connect sys/password as sysdba
```

where *password* is the password for the SYS user account.

Caution: When you connect as user SYS, you have unlimited privileges on data dictionary tables. Be certain that you do not modify any data dictionary tables.

See Also:

- "Operating System Authentication" on page 6-4
- "Changing Administrative User Passwords" on page 6-7
- Chapter 2, "Starting up and Shutting Down" on page 2-1

Operating System Authentication

Operating system authentication (OS authentication) is a way of authenticating users with high-level administrative privileges. If you log in to the Oracle Database XE host computer with a user name that is in a special operating system user group, you are then permitted to connect to the database with the SYSDBA privilege.

OS authentication is needed because there must be a way to identify administrative users even if the database is shut down. A user authenticated in this way can then start up the database. (See "Starting up and Shutting Down" on page 2-1 for more information.)

Connecting to the database with OS authentication is by definition a local connection. For more information about local and remote connections, see "About Local and Remote Connections" on page 1-5.

A user who is authenticated by the database through OS authentication does not need to know the SYS account password.

Table 6–2 lists the operating system user groups whose member users can connect to the database with the SYSDBA privilege.

Table 6–2 Operating System User Groups for OS Authentication

Platform	Operating System User Group Name
Linux	dba
Windows	ORA_DBA

If the OS authentication user group does not already exist, it is automatically created when you install Oracle Database XE. In addition, upon installation on the Linux platform, the user account oraclexe is automatically created and placed in the dba group. Upon installation on the Windows platform, the user performing the installation is automatically added to the ORA_DBA group. On both platforms, you can add other host users to the OS authentication user group to enable them to connect to the database with the SYSDBA privilege.

Caution: Adding other users to the OS authentication user group has security implications, because these users can modify any database object.

See Also:

- "The SYSDBA System Privilege" on page 6-4
- "Logging In as an Administrator" on page 6-5

Logging In as an Administrator

There are three ways to log in to Oracle Database 10g Express Edition (Oracle Database XE) to perform administrative tasks:

- Log in as user SYSTEM
- Log in as a user who has been granted the DBA role
- Connect to the database as SYSDBA

Table 6–3 provides information about each of these login methods.

Table 6–3 Database Administrator Login Methods

Login Method	Permitted In	Notes	See
Log in to the database as user SYSTEM	The Oracle Database XE graphical user interface and SQL*Plus	For routine administration tasks like managing memory and managing users. You must supply the password for the SYSTEM user.	"The SYS and SYSTEM Users" on page 6-3
Log in to the database as a user who has been granted the DBA role	The Oracle Database XE graphical user interface and SQL*Plus	For routine administration tasks like managing users. An administrator must first grant the DBA role to the user.	"User Privileges and Roles" on page 6-2
Connect to the database as SYSDBA	SQL*Plus	For high-level administration tasks like starting up and shutting down the database, and changing the SYS password. You can connect using the SYS user name and password, or using operating system authentication.	"The SYSDBA System Privilege" on page 6-4 and "Operating System Authentication" on page 6-4

This section contains the following topics:

See Also:

Logging In as User SYSTEM

You can log in as user SYSTEM with the Oracle Database XE graphical user interface or with SQL*Plus.

Logging In as User SYSTEM with the Oracle Database XE graphical user interface

To log in to the database as user SYSTEM with the Oracle Database XE graphical user interface:

- Access the Database Home Page, providing the username SYSTEM and the password for the SYSTEM account.

  See "Accessing the Database Home Page" on page 1-1 for instructions.

Note: You set the SYSTEM account password upon installation (Windows) or configuration (Linux).

Logging In as User SYSTEM with SQL*Plus

To log in to the database as user SYSTEM with SQL*Plus:

1. Log in to the Oracle Database XE host computer with any user account.
2. Linux platform only: Ensure that environment variables are set properly.

   See "Setting Environment Variables" on page 1-6 for details.
3. At an operating system command prompt, enter the following command:

   ```
   sqlplus system/password
   ```

   where *password* is the password for the SYSTEM user account.

Note: These instructions establish a local connection to the database. See "Connecting Remotely with SQL*Plus" on page 1-8 for information on connecting to the database remotely.

Logging In as a User with the DBA Role

The procedures for logging in as a user who has been granted the DBA role are the same as those for logging in as user SYSTEM, with the following exceptions:

- When logging in, you must supply the password for this user account.
- An administrator must have previously logged in and granted you the DBA role.

  See "User Privileges and Roles" on page 6-2 for more information.

Connecting to the Database as SYSDBA

You can connect as SYSDBA only with SQL*Plus. You can do so either by supplying the SYS username and password, or by using operating system (OS) authentication.

Connecting as SYSDBA with the SYS Username and Password

To connect as SYSDBA supplying the SYS username and password:

1. Log in to the Oracle Database XE host computer with any user account.
2. Linux platform only: Ensure that environment variables are set properly.

   See "Setting Environment Variables" on page 1-6 for details.
3. At an operating system command prompt, enter the following command:

   ```
   sqlplus sys/password as sysdba
   ```

   where *password* is the password for the SYS user account. You set the SYS account password upon installation (Windows) or configuration (Linux).

> **Note:** These instructions establish a local connection to the database. Do not attempt to connect AS SYSDBA remotely. See "About Local and Remote Connections" on page 1-5 for more information.

Connecting as SYSDBA with OS Authentication

To connect as SYSDBA using OS authentication:

1. Log in to the Oracle Database XE host computer as a user that is a member of an operating system user group shown in Table 6–2 on page 6-5.

   On Linux, this is normally the oraclexe user, and on Windows, it is normally the user that installed Oracle Database XE. The database can now authenticate you with operating system (OS) authentication. See "Operating System Authentication" on page 6-4 for more information.
2. Linux platform only: Ensure that environment variables are set properly.

   See "Setting Environment Variables" on page 1-6 for details.
3. At the operating system prompt, enter the following command:

   ```
   sqlplus / as sysdba
   ```

   Note that you do not need to supply a database user name and password, because you already provided database administrator credentials to the operating system. Remember that when you connect with OS authentication, you are effectively logging in to the database as user SYS.

See Also:

- "Operating System Authentication" on page 6-4

Changing Administrative User Passwords

This section describes how to change the password for database users SYS and SYSTEM using SQL*Plus.

To change the password for user SYS or SYSTEM:

1. Using SQL*Plus, connect to the database as SYSDBA.

See "Connecting to the Database as SYSDBA" on page 6-7 for instructions.

2. Type one of the following SQL*Plus commands:

```
ALTER USER SYS IDENTIFIED BY newpassword;
ALTER USER SYSTEM IDENTIFIED BY newpassword;
```

where *newpassword* is the desired new password.

See Also: "About Administrative Accounts and Privileges" on page 6-3

Managing Database Users

You can use the Oracle Database XE graphical user interface or SQL*Plus to manage database users. This section discusses using the Oracle Database XE graphical user interface, and contains the following topics:

- Viewing Users on page 6-8
- Creating Users on page 6-11
- Altering Users on page 6-13
- Dropping Users on page 6-14
- Locking and Unlocking User Accounts on page 6-15
- Expiring a User Password on page 6-16

See Also: *Oracle Database SQL Reference* and *Oracle Database Security Guide* for information on managing users with SQL*Plus.

Viewing Users

You can view database users with the Oracle Database XE graphical user interface. After viewing a list of users, you can then select an individual user to alter or drop.

To view database users:

1. Access the Database Home Page.

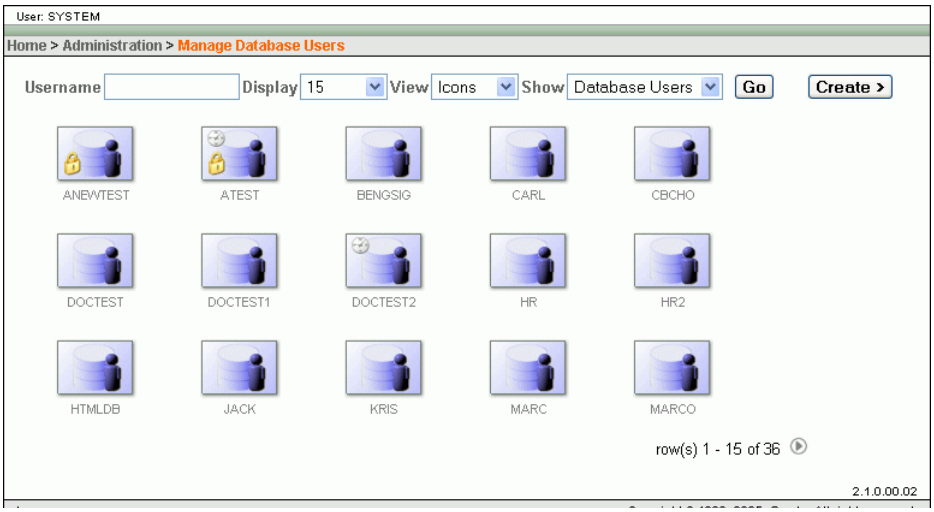
   See "Accessing the Database Home Page" on page 1-1 for instructions.

2. Click the **Administration** icon, and then click the **Users** icon.

Note: If prompted for administrator credentials, enter the `SYSTEM` username and password or another administrator username and password, and then click **Login**.

See "About Administrative Accounts and Privileges" on page 6-3 for more information.

The Manage Database Users page appears.



The large icons indicate account status:

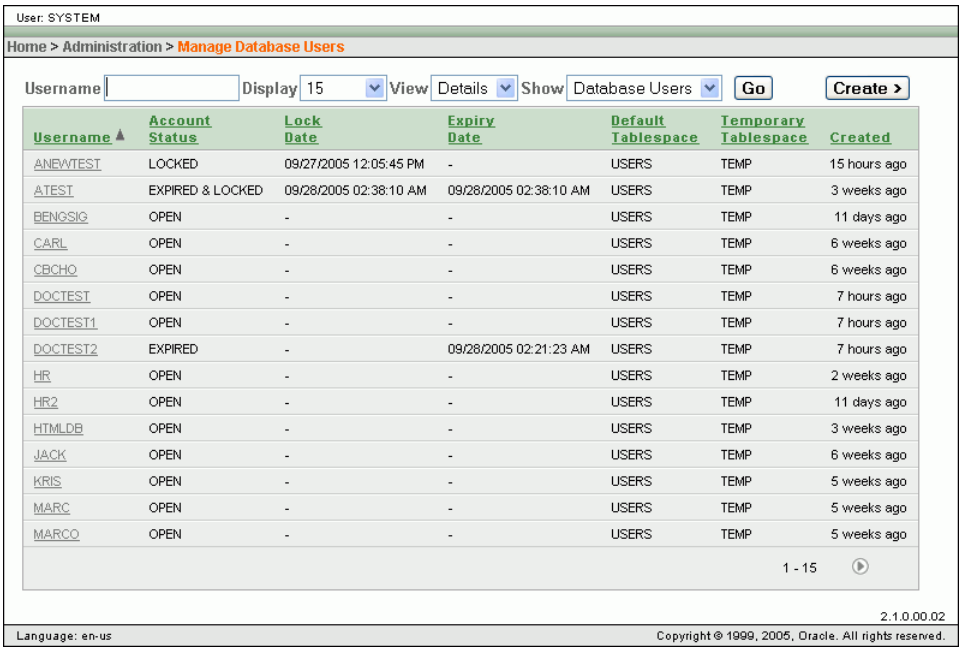
- Large icons that include a small lock icon (see user `ANEWTEST` in the preceding image) indicate locked accounts. Locking an account disables database login for that account.
- Large icons that include a small clock icon (see user `DOCTEST2`) indicate an account with an expired password. The next time that a user logs in with this account, he must choose a new password.
- Large icons that include both a small lock icon and a small clock icon (see user `ATEST`) indicate an account that is both expired and locked.
- Large icons that do not include either small icon indicate an account that is **open**. This means that users are permitted to log in with this account and that they can use the current password for that account.

See "Locking and Unlocking User Accounts" on page 6-15 and "Expiring a User Password" on page 6-16 for more information.

3. (Optional) In the Display dropdown list, select the number of users to display on a single page, and then click **Go**.
4. (Optional) In the Show dropdown list, select **Internal Users** and click **Go** to view a list of internal user accounts.

   You can view these accounts but cannot modify them. See "Internal User Accounts" on page 6-3 for more information.

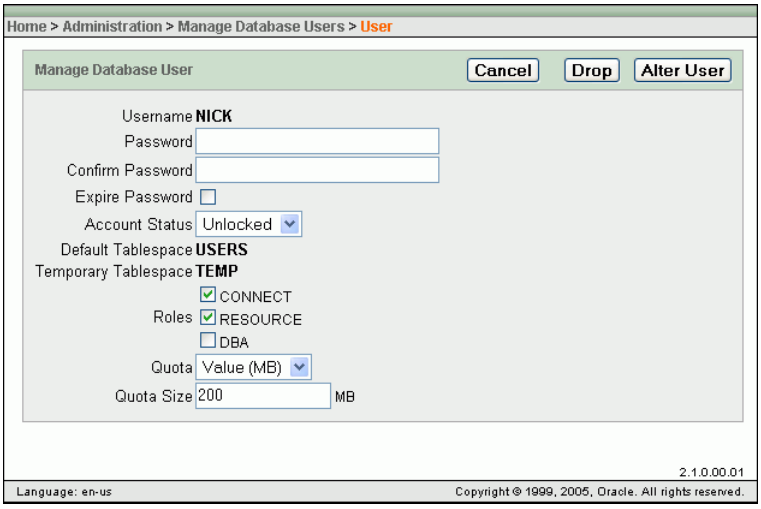
   After viewing internal user accounts, select **Database Users** in the Show dropdown list to return to the page that displays database users.
5. (Optional) In the View dropdown list, select **Details** and then click **Go** to view users as a list instead of as a collection of icons. The list displays user attributes.



6. To view details on a particular user, click the user's icon if in the Icons view, or click the user's hyperlink (under the Username column) if in the Details view.

Note: If you do not see the user who you are looking for, click the right arrow at the bottom of the page to view the next set of users. Continue clicking the right arrow until you find the user of interest. Click the left arrow to return to the previous page. When you find the user, click the user's icon or hyperlink.

The User page appears, with the user's information displayed.



See Also:

- "Creating Users" on page 6-11
- "Altering Users" on page 6-13
- "Dropping Users" on page 6-14
- "Locking and Unlocking User Accounts" on page 6-15
- "Expiring a User Password" on page 6-16

Creating Users

You create users from the Database Users page in the Oracle Database XE graphical user interface. Before creating a user, determine the following:

- Whether or not you want to permit the user to create schema objects. If so, select the RESOURCE role on the Create Database User page.
- Whether or not you want to grant the user DBA privileges. If so, select the DBA role on the Create Database User page.

  DBA privileges include the ability to create schema objects. If you select the DBA role, you do not need to select the RESOURCE role.

  Caution: Granting the DBA role to a user has security implications, because the user can modify objects in other users' schemas.

- How much disk space to allot to users with privileges to create schema objects. You control this value by setting the user's quota.
- Whether or not to create the user with an expired password. When you do this, the password that you assign the user is used only for the user's first login. Upon first login, the user is prompted to select a new password.

See Also:

- "About User Accounts" on page 6-1
- "User Privileges and Roles" on page 6-2 for details on the RESOURCE and DBA roles.

Example: Creating a User

Suppose that you want to create a user account for a database application developer named Nick. Because Nick is a developer, you want to grant him the RESOURCE role so that he can create the schema objects that his applications require. But because Nick is a junior developer, you want to limit the amount of space in the USERS tablespace that his applications can consume. In addition, you want to create his account with an expired password so that he has to choose a password when he first logs in to the database, and you want to assign the temporary password "firesign."

To create the user Nick:

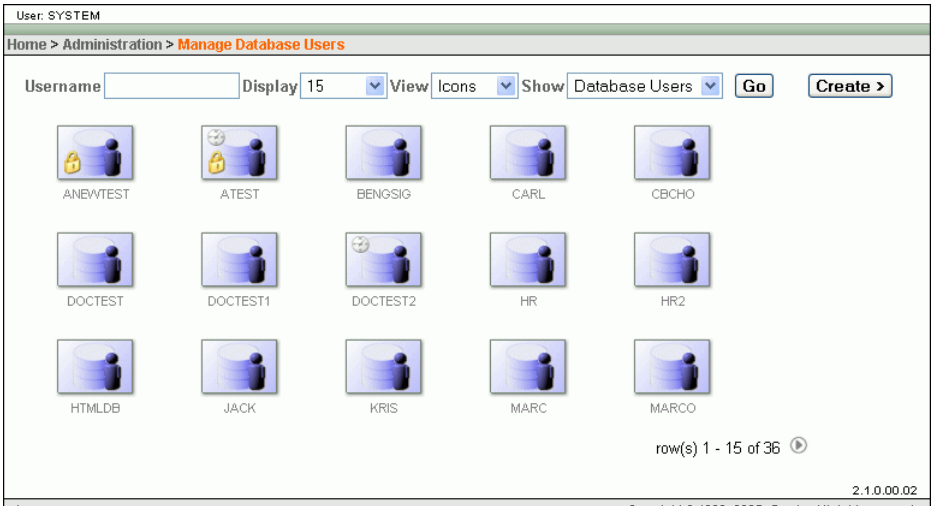
1. Access the Database Home Page.

   See "Accessing the Database Home Page" on page 1-1 for instructions.

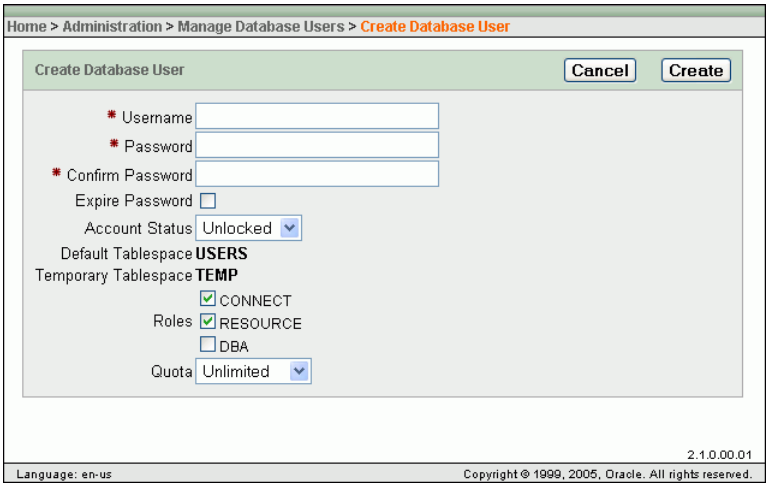
2. Click the **Administration** icon, and then click the **Database Users** icon.
3. If prompted for administrator credentials, enter the SYSTEM username and password or another administrator username and password, and then click **Login**.

See "About Administrative Accounts and Privileges" on page 6-3 for more information.

4. On the Manage Database Users page, click **Create**.



The Create Database User page appears.



5. Enter user information into text fields as follows (all fields are case insensitive):
   - In the Username field, enter **nick**.
   - In the Password and Confirm Password fields, enter **firesign**.
6. Select the **Expire Password** check box.
7. In the Quota dropdown list, select **Value (MB)**.

   A Quota Size text field appears.

8. In the Quota Size field, enter 200.

   This is the maximum amount of space in the `USERS` tablespace, in MB, that Nick's schema objects can consume.

9. Before finishing, note the following:
   - The CONNECT and RESOURCE roles are selected by default. These are the correct settings for Nick.

     See "User Privileges and Roles" on page 6-2 for information about these roles.
   - The DBA role is by default not selected. This is also correct for Nick, because you do not want to give him DBA privileges, which include the ability to create schema objects in other users' schemas, and to create other users.
   - The Account Status dropdown list defaults to Unlocked. This means that the user can log in with this account. Because you want Nick to be able to log in, you accept this default.

     See "Locking and Unlocking User Accounts" on page 6-15 for more information.
10. Click **Create**.

    The Manage Database Users page reappears and displays a confirmation that the user was created.

Altering Users

You can use the Manage Database Users page to alter a user. Altering a user means changing some of his user attributes. You can change all user attributes except the username, default tablespace, and temporary tablespace. If you want to change the username, you must drop the user and recreate him with a different name. (Before you drop the user, ensure that the user's schema objects are either backed up or are no longer needed. See "Dropping Users" on page 6-14 for more information.)

One of the attributes that you can alter is the user password. If you do this, you must either communicate the new password to the user, or request the new password from the user and then enter it. An easier and more secure way to cause a password change is to **expire** the password. When you expire a password, the user is prompted to change his password the next time that he logs in. See "Expiring a User Password" on page 6-16 for more information.

See Also: "Locking and Unlocking User Accounts" on page 6-15

Example: Altering a User

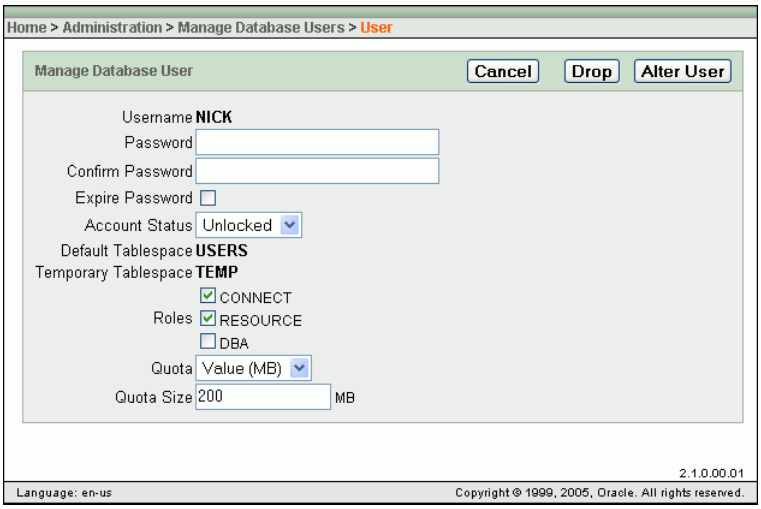
Suppose that user Nick is promoted to senior developer, and that he has shown an interest in helping with routine database administration tasks. You decide to grant the DBA role to Nick and to remove the quota on his user account.

To alter Nick's user account:

1. View the Manage Database Users page that contains the icon or hyperlink for user Nick.

   See "Viewing Users" on page 6-8 for instructions.
2. Click the icon or hyperlink for user Nick.

   The User page appears, with Nick's account information displayed.



3. Select the **DBA** check box to grant the `DBA` role to Nick.
4. In the Quota dropdown list, select **Unlimited**.
5. Click **Alter User** to save your changes.

   The Manage Database Users page reappears and displays a confirmation message that the user was altered.

Dropping Users

Dropping a user removes the user from the database. Before you can drop a user, you must first drop all the user's schema objects. Or, you can use the **cascade** feature of the drop operation, which simultaneously drops a user and all of his schema objects. The following are two alternatives to dropping a user and losing all of the user's schema objects:

- To temporarily deny access to the database for a particular user while preserving the user's schema objects, you can **lock** the user account. See "Locking and Unlocking User Accounts" on page 6-15 for more information.
- To drop a user but retain the data from the user's tables, export the tables first. See Chapter 9, "Importing and Exporting" on page 9-1 for instructions.

> **WARNING: Under no circumstances should you attempt to drop the SYS or SYSTEM users, or any other internal user accounts.**

Example: Dropping a User

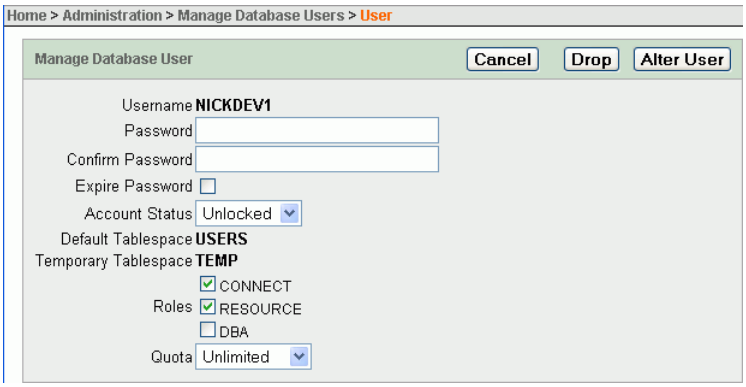
Suppose that one of Nick's projects is canceled and that you want to drop the schema objects for this project. Suppose also that Nick created a new user named `Nickdev1` so that he could have a separate schema in which to store the objects for that project. You want to drop the user `Nickdev1` and all associated schema objects.

To drop user `Nickdev1` and all his owned schema objects:

1. View the Manage Database Users page that contains the icon or hyperlink for `Nickdev1`.

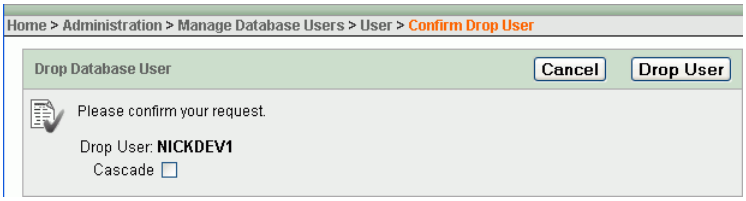
   See "Viewing Users" on page 6-8 for instructions.
2. Click the `Nickdev1` icon or hyperlink.

The User page appears, with Nickdev1's account information displayed.



3. Click **Drop**.

   The Confirm Drop User page appears.



4. Select the **Cascade** check box.

   This indicates that you want to drop the user's schema objects also. If the user has schema objects and you do not select this option, you receive an error message if you attempt to complete the drop operation.

5. Click **Drop User**.

   A confirmation message is displayed.

Locking and Unlocking User Accounts

To temporarily deny access to the database for a particular user, you can lock the user account. If the user then attempts to connect, the database displays an error message and disallows the connection. You can unlock the user account when you want to allow database access again for that user.

Note: Many internal user accounts are locked (or both expired and locked). You should not attempt to log in with these locked user accounts. See "Internal User Accounts" on page 6-3 for more information.

The `HR` user account, which contains a sample schema, is initially expired and locked. You must log in as `SYSTEM`, unlock the account, and assign a password before you can log in as `HR`.

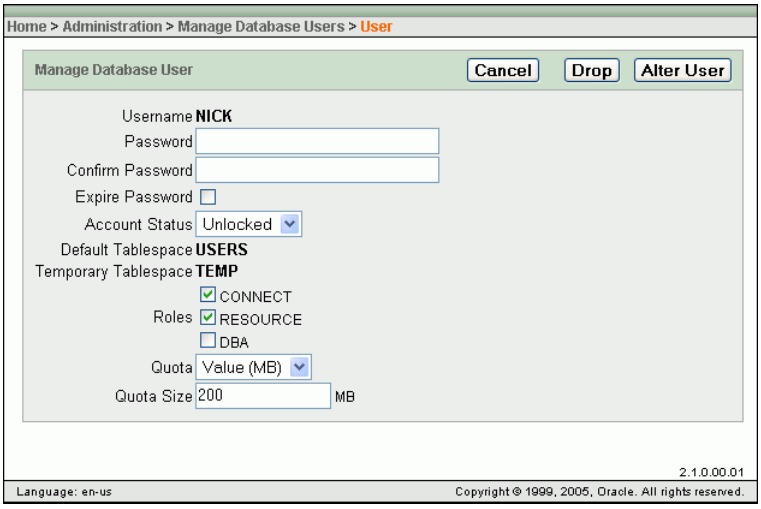
To lock or unlock a user account:

1. View the Manage Database Users page that contains the icon or hyperlink for the user.

   See "Viewing Users" on page 6-8 for instructions.

2. Click the icon or hyperlink for the user.

   The User page appears, with the user account information displayed.



3. Do one of the following:

   - To lock the account, select **Locked** from the Account Status dropdown list.
   - To unlock the account, select **Unlocked** from the Account Status dropdown list.

4. Click **Alter User**.

   The Manage Database Users page reappears and displays a confirmation message. The large icon for user now indicates if the account is locked or unlocked by the presence or absence of a small lock icon.

Expiring a User Password

When you expire a user password, the user is prompted to change his password the next time that he logs in. Reasons to expire a password include the following:

- A user password becomes compromised.
- You have a security policy in place that requires users to change their passwords on a regular basis.
- A user has forgotten his password.

  In this case, you alter the user account, assign a new temporary password, and expire the password. The user then logs in with the temporary password and is prompted to choose a new password.

  See "Altering Users" on page 6-13 for more information.

Example: Expiring a Password

Suppose that user Nick's password becomes compromised, and that you want to assign him a new one. The easiest way to do this is to expire his current password. The next time that Nick logs in with the compromised password, he is prompted to choose a new password.

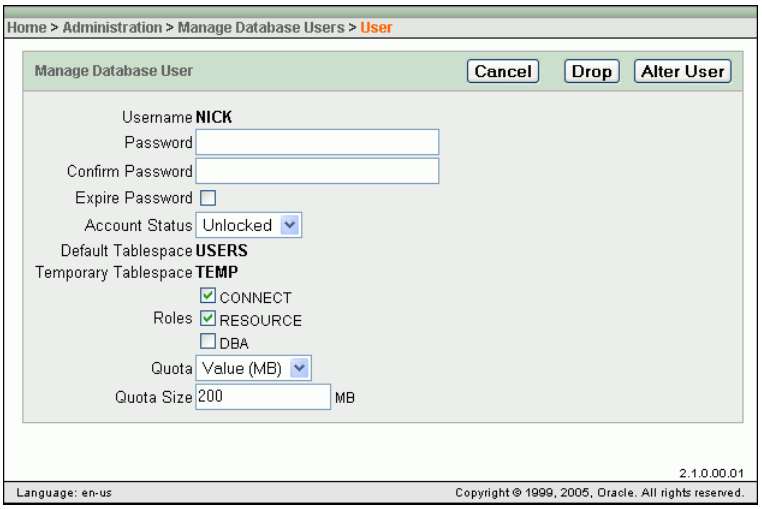
To expire Nick's password:

1. View the Manage Database Users page that contains the icon or hyperlink for user NICK.

   See "Viewing Users" on page 6-8 for instructions.

2. Click the icon or hyperlink for user NICK.

   The User page appears, with Nick's account information displayed.



3. Select the **Expire Password** check box, and then click **Alter User**.

   The Manage Database Users page reappears and displays a confirmation message. The large icon for user NICK now contains a small clock icon.

> **Note:** The presence of a small clock icon in the large icon for a user is the only indication that the user's password is expired. When you view the Users page for a user with an expired password, the Expire Password check box is not checked.

Oracle Database 10g Express Edition Predefined User Accounts

Table 6–4 lists the Oracle Database 10g Express Edition predefined user accounts. Many of these accounts are internal accounts. You must not drop internal accounts, and with the exception of SYSTEM, you must not attempt to log in with an internal account.

Table 6–4 Oracle Database 10g Express Edition Predefined User Accounts

User Account Name	Purpose
ANONYMOUS	Internal.
CTXSYS	Internal.
DBSNMP	Internal.
DIP	Internal.
FLOWS *version*	Internal.
FLOWS_FILES	Internal.
HR	For the HR sample schema. This account is initially expired and locked.

Table 6–4 (Cont.) Oracle Database 10g Express Edition Predefined User Accounts

User Account Name	Purpose
OUTLN	Internal.
SYS	Owns the data dictionary base tables and views. The account password is set upon installation (Windows) or configuration (Linux).
SYSTEM	Log in with this account to perform routine database administration. The account password is set upon installation (Windows) or configuration (Linux).
TSMSYS	Internal.
XDB	Internal.

See Also:

- "About User Accounts" on page 6-1
- "About Administrative Accounts and Privileges" on page 6-3

7

Monitoring the Database

As an administrator, you can monitor the activities of the database and its users. You can use this information for tuning, troubleshooting, and more.

This section contains the following topics:

Monitoring Sessions

You can use the Oracle Database XE graphical user interface to monitor the current database sessions. Monitoring sessions is an easy way to know who is currently logged in to the database, and in some cases, to know what activities users are engaged in. You can view all sessions or just active sessions. The default view shows active sessions only.

To monitor sessions:

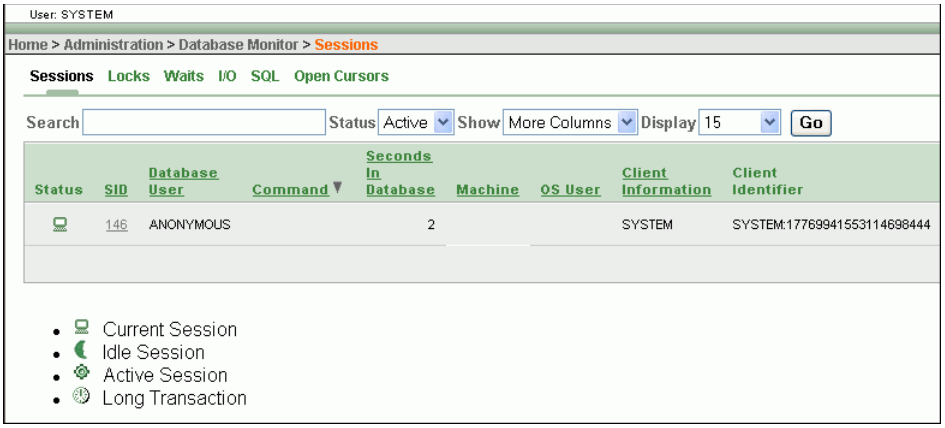
1. Log in to the Oracle Database XE graphical user interface as user `SYSTEM`.

   See "Logging In as User SYSTEM" on page 6-6 for instructions.

2. On the Database Home Page, click **Administration**, and then click **Monitor**.

3. On the Database Monitor page, click **Sessions**.

   The Sessions page appears and displays the current active sessions.



Note that the session marked with the Current Session icon under the Status column is your session.

4. (Optional) In the Status dropdown list, select **All**, and then click **Go**.

   The page now shows all sessions, including SQL*Plus sessions.

Monitoring Long Operations

This section to be provided in the production release of the documentation.

Viewing System Statistics

This section to be provided in the production release of the documentation.

Viewing the Top SQL Statements

This section to be provided in the production release of the documentation.

8

Viewing Database Configuration Settings

This section to be provided in the production release of the documentation.

Viewing Database Initialization Parameters

This section to be provided in the production release of the documentation.

Viewing Database Version, Options, and Features

This section to be provided in the production release of the documentation.

9

Importing and Exporting

This section describes how to import and export data with Oracle Database 10g Express Edition. It contains the following topics:

Choosing the Right Import/Export Option

Oracle Database 10g Express Edition (Oracle Database XE) provides a number of powerful options for importing and exporting data. Table 9–1 provides a summary of these options.

Table 9–1 Summary of Oracle Database XE Import/Export Options

Feature or Utility	Description
Data Load/Unload wizards in the Oracle Database XE graphical user interface	▪ Easy to use graphical interface ▪ Imports/exports from and to external text files (delimited fields) or XML files ▪ Imports/exports tables only. One table at a time ▪ Access only to schema of logged in user ▪ No data filtering
Data Pump utility	▪ Command line interface ▪ Exports and imports from one Oracle database to another (proprietary binary format) ▪ Imports/exports all schema object types. Multiple schema objects simultaneously ▪ Imports/exports entire database, entire schema, multiple schemas, multiple tablespaces, or multiple tables ▪ Powerful data filtering capabilities ▪ High speed ▪ Does not support XMLType data

Table 9–1 (Cont.) Summary of Oracle Database XE Import/Export Options

Feature or Utility	Description
IMP and EXP utilities	■ Command line interface ■ Exports and imports from one Oracle database to another (proprietary binary format) ■ Supports XMLType data ■ Capabilities similar to Data Pump. Data Pump is preferred unless you must import or export XMLType data
SQL*Loader utility	■ Command line interface ■ Bulk-loads data into the database from external files ■ Supports numerous input formats, including delimited, fixed record, variable record, and stream ■ Loads multiple tables simultaneously ■ Powerful data filtering capabilities

Table 9–2 provides a number of import/export scenarios and suggests the appropriate option to use for each.

Table 9–2 Import/Export Scenarios and Recommended Options

Import/Export Scenario	Recommended Option
You have fewer than 10 tables to import, the data is in spreadsheets or tab- or comma-delimited text files, and there are no complex data types (such as objects or multi-valued fields)	Data Load/Unload wizards in the Oracle Database XE graphical user interface
You have to import data that is not delimited. The records are fixed length, and field definitions depend on column positions	SQL*Loader
You have tab-delimited text data to import, and there are more than 10 tables	SQL*Loader
You have text data to import, and you want to import only records that meet certain selection criteria (for example, only records for employees in department number 3001)	SQL*Loader
You want to import or export an entire schema from or to another Oracle database. There are no XMLTypes in any of the data	Data Pump
You want to import or export data from or to another Oracle database. The data contains XMLTypes	IMP and EXP

See Also: *Oracle Database Utilities* for more information on Data Pump, IMP, EXP, and SQL*Loader

Importing and Exporting with Wizards

The Data Load/Unload wizards of the Oracle Database XE graphical user interface enable you to easily import and export delimited text data to and from the database. The step-by-step wizards have the following features:

- You can import or export XML files or delimited-field text files (such as comma-delimited (.csv) or tab-delimited files).
- You can import by copying and pasting from a spreadsheet.

- You can omit (skip) columns when importing or exporting
- You can import into an existing table or create a new table from the import data.
- When importing into a new table, the primary key can be taken from the data or generated from a new or existing Oracle sequence.
- When importing into a new table, column names can be taken from the import data.
- Each time you import from a file, file details are saved in a Text Data Load Repository. You can access these files from within the repository at any time.

Limitations include the following:

- Imports and exports table data only. Does not import and export other kinds of schema objects.
- You can import and export to and from your own schema only. This is also true for users with administrator privileges.
- You can import or export only a single table at a time.
- There are no data type limitations for Unload to Text, Unload to XML, or Load XML Data. However, Load Text Data and Load Spreadsheet Data support only the following data types: `NUMBER`, `DATE`, `VARCHAR2`, `CLOB`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Example: Exporting with the Unload Wizard - Tab-Delimited Text File

Suppose that you want to export the `REGIONS` table, which is part of the HR sample schema, so that it can be used in another application. Suppose also that you want to create a tab-delimited text file, and you want to save the data in a file called `regions.txt`.

To export the `REGIONS` table:

1. Log in to the Oracle Database XE graphical user interface as the `HR` user.

   See "Accessing the Database Home Page" on page 1-1 for information on getting logged in.

   Note: An administrator must first unlock the `HR` account and assign a password. See "Logging In as an Administrator" on page 6-5 and "Locking and Unlocking User Accounts" on page 6-15 for instructions.

2. On the Database Home Page, click the **Utilities** icon, and then click the **Data Load/Unload icon**.
3. On the Data Load/Unload page, click the **Unload icon**, and then click the **Unload to Text icon**.

   The Unload to Text page appears, showing the "Schema" wizard step. This wizard step displays a Schema dropdown list, in which `HR` is selected. Because you can export from your own schema only, you cannot change this selection.
4. Click **Next**.

   The "Table Name" wizard step appears.
5. From the Table dropdown list, select **`REGIONS`**, and then click **Next**.

   The "Columns" wizard step appears.

6. Select all columns and click **Next**. (You could have also selected a subset of columns. Deselected columns are excluded from the export.)

   The "Options" wizard step appears.

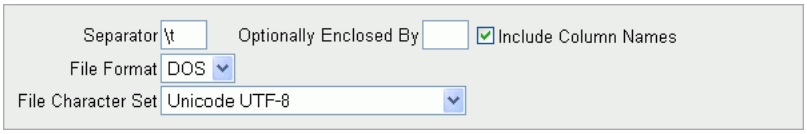
7. Complete the following steps:

   a. In the Separator field, remove the comma if present, and enter a backslash and a lower case T (`\t`) to indicate that you want the tab character to be the field delimiter. (You can use any character as the delimiter.)

   b. Select the **Include Column Names** check box.

      This causes the first row exported to be the column names, rather than the first row of data. You can use this first row to set column names when you import.

   c. In the File Character Set dropdown list, select **Unicode UTF-8**.



8. Click **Unload Data**.

   A Save As window appears, with the filename `regions.txt` filled in. Depending on your browser, another window may precede the Save As window, asking you if you want to save or open the file. If so, take the option to save the file to disk.

9. Save the file `regions.txt` to the Desktop.

10. (Optional) Open `regions.txt` with a text editor or spreadsheet application to verify that the `REGIONS` table was exported properly.

Example: Importing with the Load Wizard - Tab-Delimited Text File

Suppose that your application calls for a `REGIONS` table, where each row contains a region number and a region name. Suppose also that you previously exported region data from a desktop database system into a tab-delimited text file named `regions.txt`.

You want to use the region number field in each record as a business key but not as the primary key, and you therefore decide to have the Load wizard generate a numeric primary key for each imported record.

> **Note:** You can complete the following steps with the `regions.txt` file that you create in "Example: Exporting with the Unload Wizard - Tab-Delimited Text File" on page 9-3.

To import the `REGIONS` table:

1. Log in to the Oracle Database XE graphical user interface as any user other than `SYSTEM` or `HR`.

   See "Accessing the Database Home Page" on page 1-1 for information on getting logged in. If no database user other than `SYSTEM` or `HR` exists, create it. See "Creating Users" on page 6-11 for instructions.

2. On the Database Home Page, click the **Utilities** icon, and then click the **Data Load/Unload icon**.

3. On the Data Load/Unload page, click the **Load icon**, and then click the **Load Text Data icon**.

   The Load Data page appears, showing the "Target and Method" wizard step.

4. Under the Load To heading, select **New table**, and under the Load From heading, select **Upload file (comma separated or tab delimited)**.

5. Click **Next**.

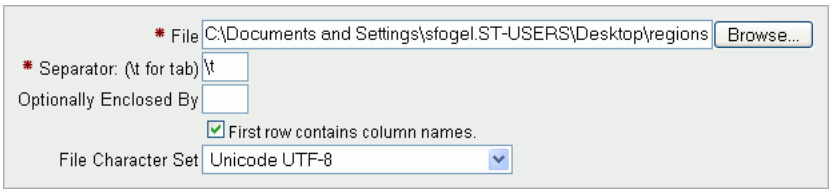
   The "File Details" wizard step appears.

6. Complete the following steps:

   a. Click **Browse**, and select the `regions.txt` file from the Desktop.

   b. In the Separator field, replace the comma with a backslash and a lower case T (`\t`) to indicate that the field delimiter is a tab character.

   c. In the File Character Set dropdown list, select **Unicode UTF-8**.

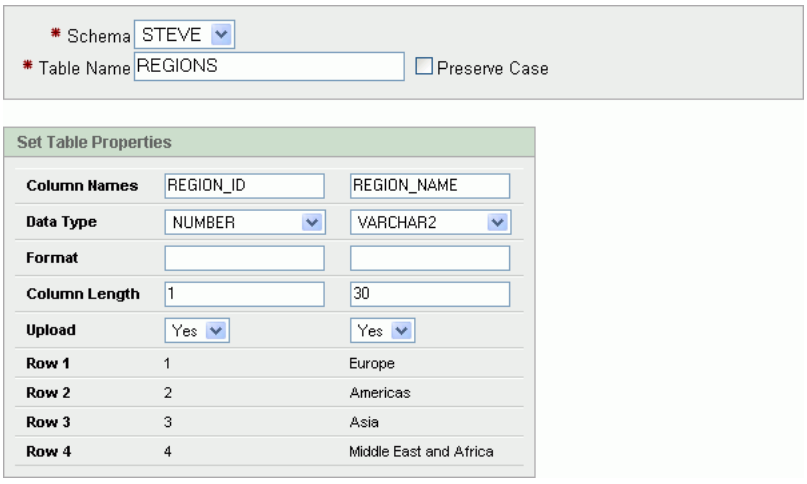


   d. Click **Next**.

   The "Table Properties" wizard step appears.

7. Complete the following steps:

   a. In the Table Name field, enter `REGIONS`.



   b. Accept the default (`Yes`) in all **Upload** dropdown lists.

Setting Upload to `No` excludes the column from the import operation.

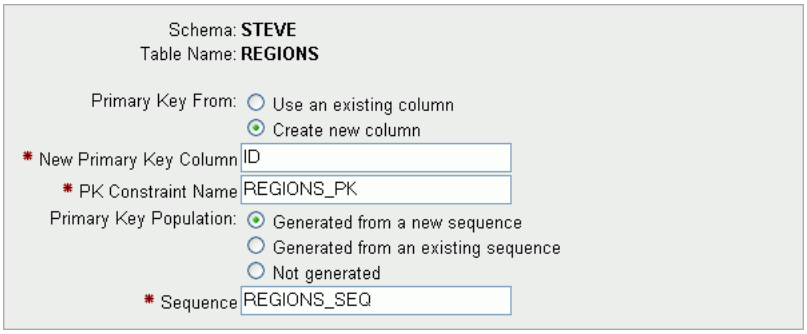
c. Click **Next**.

The "Primary Key" wizard step appears.

8. Complete the following steps:

   a. Next to the Primary Key From label, select **Create New Column**.

   b. Next to the Primary Key Population label, select **Generated from a new sequence**.

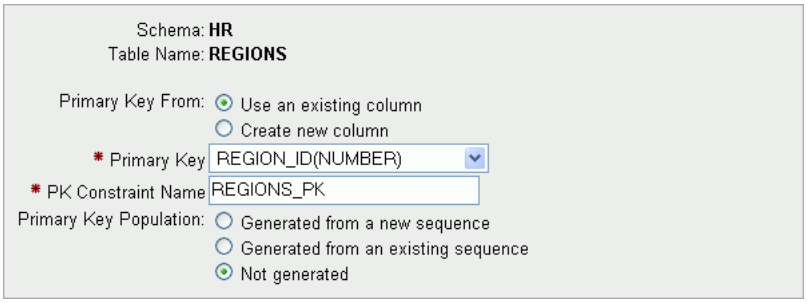


These selections cause Oracle Database XE to:

- Create an additional table column called `ID`, which is used as the primary key for `REGIONS`.
- Create a new sequence called `REGIONS_SEQ`.
- Use the values from the sequence to populate the `ID` field as each new row is added.

If you did not want to create a new primary key, and wanted to instead use the existing `REGION_ID` field as the primary key, you would do the following:

- Select **Use an existing column**.
- In the Primary Key dropdown list, select **`REGION_ID(NUMBER)`**.
- Select **Not Generated**



9. Click the **Load Data button**.

   The import proceeds, and when it is complete, the Text Data Load Repository page appears, showing the `regions.txt` file at the top of the list of imported files.

10. Check the import status by looking under the `Succeeded` and `Failed` columns for the `regions.txt` file.

    The numbers in these columns indicate the number of rows that were successfully imported or that caused an error.

Importing and Exporting with Oracle Utilities

This section to be provided in the production release of the documentation.

Importing and Exporting with Data Pump

This section to be provided in the production release of the documentation.

Exporting and Exporting with IMP and EXP

This section to be provided in the production release of the documentation.

Loading Data with SQL*Loader

This section to be provided in the production release of the documentation.

10

Backing Up and Recovering the Database

This section discusses backup and recovery of the entire database, and recovery of data from individual schema objects. It includes the following topics:

About the Database Backup Facility

Oracle Database 10g Express Edition (Oracle Database XE) includes a facility to back up and restore the entire database. Oracle recommends that you back up your database frequently.

The Oracle Database XE backup facility uses the Data Pump utility to back up the entire database to the **Flash Recovery Area**. The Flash Recovery Area is a directory on the Oracle Database XE host that is used to store backup and other offline files. Table 10–1 shows the location of the Flash Recovery Area on each platform.

Table 10–1 Flash Recovery Area Locations

Platform	Location
Linux	`/usr/lib/oracle/xe/app/oracle/flash_recovery_area/`
Windows	`c:\oraclexe\app\oracle\flash_recovery_area\`

You back up and restore the database by running automated scripts that provide the necessary sequence of commands to Data Pump. These scripts are available as menu choices on your graphical desktop.

The supplied scripts back up and restore the entire database. You cannot use them to do a partial or incremental backup. When you create a new backup (a new set of backup files), only the previous backup is retained. Any earlier backup files are deleted. That is, at any time, only the two most recent sets of backup files are available for restoring the database. Therefore, if you want to retain more generations of database backups, you must copy the backup files from the Flash Recovery Area to another location.

Backup Files

A database backup operation writes the following files into the Flash Recovery Area:

```
OXE_BackupLog_current.dmp
OXE_BackupLog_current.log
```

The files for the previous backup are renamed as follows:

```
OXE_BackupLog_old.dmp
OXE_BackupLog_old.log
```

Online Backup

The Oracle Database XE backup facility performs an *online* backup. This means that the database is running during the backup process.

Backing up the Database

To back up the database:

1. Do one of the following:
   - On Windows: Log in to the Oracle Database XE host computer as a Windows administrator—that is, as a user who is a member of the Administrator group.
   - On Linux: Log in to the Oracle Database XE host computer as a user who can be authenticated with operating system authentication—that is, as a user who is a member of an operating system user group shown in Table 6–2 on page 6-5.

     This is normally the user `oraclexe`. See "Operating System Authentication" on page 6-4 for more information.
2. Ensure that the database is running.

   An easy way to do this is to try to access the Database Home Page. See "Accessing the Database Home Page" on page 1-1 for instructions.
3. Ensure that all users are logged out of the database.

   To do so, view all sessions (not just active sessions) on the Sessions page. See "Monitoring Sessions" on page 7-1 for instructions.
4. Do one of the following:
   - On Windows: Click **Start**, point to **Programs**, point to **Oracle Database 10g Express Edition**, and then select **Backup Database**.
   - On Linux with Gnome: In the Applications menu, point to **Oracle Database 10g Express Edition**, and then select **Backup Database**.
   - On Linux with KDE: Click the icon for the K Menu, point to **Oracle Database 10g Express Edition**, and then select **Backup Database**.

See Also:

- "About the Database Backup Facility" on page 10-1

Restoring the Database

When you restore the database, you always automatically restore from the most recent backup.

WARNING: Restoring the database completely overwrites all current data in the database.

To restore the database:

1. Do one of the following:
   - On Windows: Log in to the Oracle Database XE host computer as a Windows administrator—that is, as a user who is a member of the Administrator group.
   - On Linux: Log in to the Oracle Database XE host computer as a user who can be authenticated with operating system authentication—that is, as a user who is a member of an operating system user group shown in Table 6–2 on page 6-5.

     This is normally the user `oraclexe`. See "Operating System Authentication" on page 6-4 for more information.
2. Ensure that the database is running.

   An easy way to do this is to try to access the Database Home Page. See "Accessing the Database Home Page" on page 1-1 for instructions.
3. Ensure that all users are logged out of the database.

   To do so, view all session (not just active sessions) on the Sessions page. See "Monitoring Sessions" on page 7-1 for instructions.
4. Do one of the following:
   - On Windows: Click **Start**, point to **Programs**, point to **Oracle Database 10g Express Edition**, and then select **Restore Database**.
   - On Linux with Gnome: In the Applications menu, point to **Oracle Database 10g Express Edition**, and then select **Restore Database**.
   - On Linux with KDE: Click the icon for the K Menu, point to **Oracle Database 10g Express Edition**, and then select **Restore Database**.

See Also: "Backing up the Database" on page 10-2

Recovering Dropped Tables

When you drop a table, the database does not immediately remove the space associated with the table. The database renames the table and places it and any dependent objects in a **recycle bin**, where, in case the table was dropped in error, it can be recovered at a later time.

Recycle Bin

The recycle bin is actually a data dictionary table containing information about dropped objects. Dropped tables and any dependent objects such as indexes, constraints, nested tables, and the likes are not removed and still occupy space until you purge them from the recycle bin or until they are purged by the database because of tablespace space constraints. Each user can be thought of as having his own recycle bin, because unless a user has the SYSDBA privilege, the only objects that the user has access to in the recycle bin are those that the user owns.

You can use the Oracle Database XE graphical user interface to view the contents of the recycle bin, restore dropped tables and dependent objects from the recycle bin, and purge the recycle bin.

> **See Also:** *Oracle Database Administrator's Guide* to learn more about the recycle bin.

Viewing Recycle Bin Contents

This section to be provided in the production release of the documentation.

Restoring Tables from the Recycle Bin

This section to be provided in the production release of the documentation.

Purging the Recycle Bin

This section to be provided in the production release of the documentation.

Viewing and Restoring Historical Data with Flashback Query

The Flashback Query feature of Oracle Database 10g Express Edition (Oracle Database XE) enables you to view data at a point in time in the past. You can then reconstruct lost data that was deleted or changed by accident.

About Flashback Query

When you write a Flashback Query, you add a clause to the `SELECT` statement that specifies either a time or a system change number (SCN). The query then uses the committed data from the corresponding time. Flashback Query does not change any data; it queries only. It is up to you to analyze the historical data and then construct and issue DML statements to restore data.

Flashback Query retrieves historical data by applying undo as needed. The length of time that you can flash back therefore depends on the amount of undo that is available. For more information on Flashback Query, see the discussion of the `AS OF` clause for the `SELECT` statement in *Oracle Database SQL Reference*.

Example: Recovering Data with Flashback Query

This example uses a Flashback Query to examine the state of a table at a previous time. Suppose that you discover at 12:30 PM that the row for employee Chung was deleted from the `employees` table. You also know that at 9:30AM the data for Chung was correctly stored in the database. You can use a Flashback Query to examine the contents of the table at 9:30 to find out what data was lost. If appropriate, you can then re-insert the lost data.

Example 10–1 retrieves the state of the record for `Chung` at 9:30AM, April 4, 2005:

Example 10–1 Retrieving a Row with Flashback Query

```
SELECT * FROM employees AS OF TIMESTAMP
   TO_TIMESTAMP('2005-04-04 09:30:00', 'YYYY-MM-DD HH:MI:SS')
   WHERE last_name = 'Chung';
```

The update in Example 10–2 restores Chung's information to the `employees` table:

Example 10–2 Reinserting a Row After a Flashback Query

```
INSERT INTO employees
    (SELECT * FROM employees AS OF TIMESTAMP
     TO_TIMESTAMP('2005-04-04 09:30:00', 'YYYY-MM-DD HH:MI:SS')
     WHERE last_name = 'Chung');
```

Tips for Using Flashback Query

Keep the following in mind when using a Flashback Query (`SELECT ... AS OF`):

- You can specify or omit the `AS OF` clause for each table in the query and specify different times for different tables. Use an `AS OF` clause in a query to perform DDL operations (such as creating and truncating tables) or DML operations (such as inserting and deleting) in the same session as the query.
- To use the results of a Flashback Query in a DDL or DML statement that affects the current state of the database, use an `AS OF` clause inside an `INSERT` or `CREATE TABLE AS SELECT` statement.

11

Upgrading

This section describes how to upgrade Oracle Database 10g Express Edition to Oracle Database Standard Edition or Oracle Database Enterprise Edition. It includes the following topics:

About Upgrading

You can upgrade Oracle Database 10g Express Edition (Oracle Database XE) to the same patch release of Oracle Database Standard Edition (SE) or Oracle Database Enterprise Edition (EE). For example, if you have version 10.2.0.2 of Oracle Database XE, you can upgrade to version 10.2.0.2 of Oracle Database SE or EE.

Note: It is not permitted to downgrade a database from SE or EE to Oracle Database XE.

Upgrading to Oracle Database Standard Edition or Enterprise Edition

This section to be provided in the next release of this document.

Index

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W