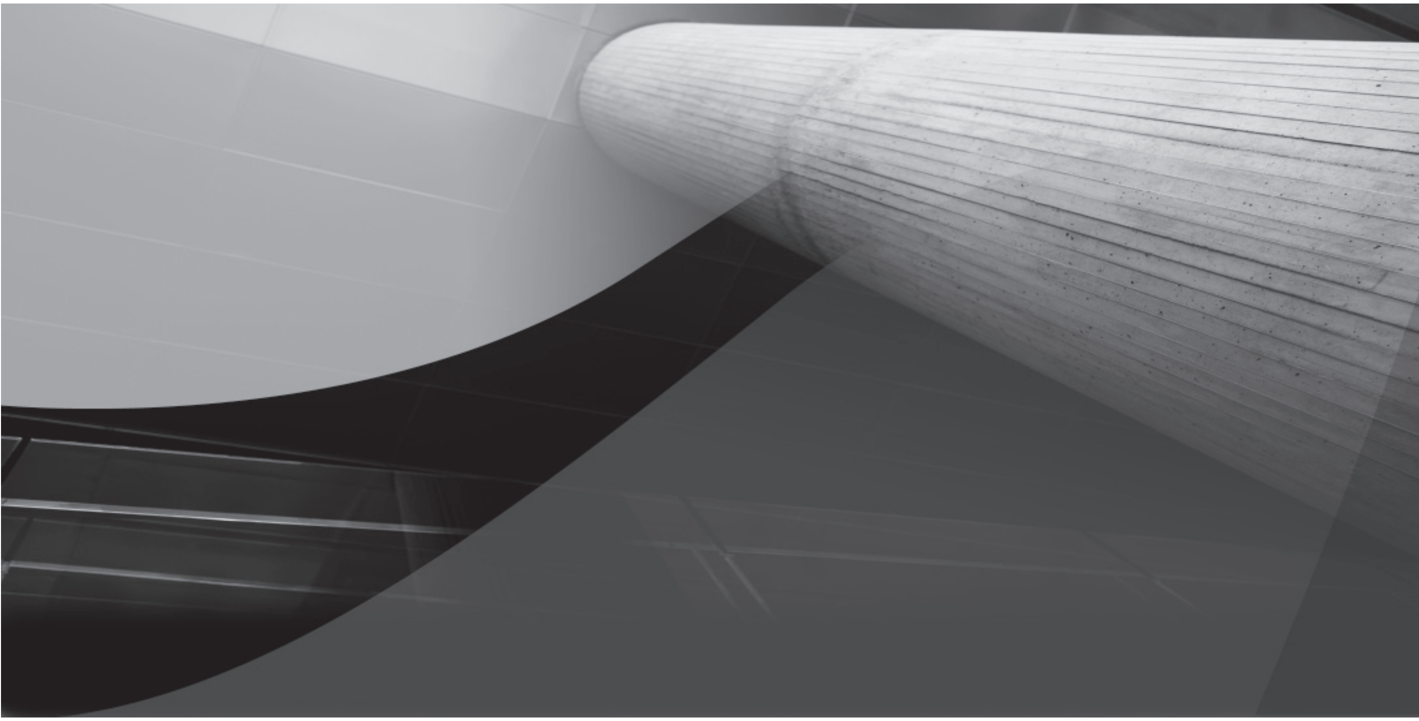

# CHAPTER 2

# Upgrading to Oracle Database 11*g*

If you have previously installed an earlier version of the Oracle database server, you can upgrade your database to Oracle Database 11*g*. Multiple upgrade paths are supported; the right choice for you will depend on factors such as your current Oracle software version and your database size. In this chapter, you will see descriptions of these methods along with guidelines for their use.

If you have not used a version of Oracle prior to Oracle Database 11*g*, you can skip this chapter for now. However, you will likely need to refer to it when you upgrade from Oracle Database 11*g* to a later version or when you migrate data from a different database into your database.

Prior to beginning the upgrade, you should read the Oracle Database 11*g* Installation Guide for your operating system. A successful installation is dependent on a properly configured environment—including operating system patch levels and system parameter settings. Plan to get the installation and upgrade right the first time rather than attempting to restart a partially successful installation. Configure the system to support both the installation of the Oracle software and the creation of a usable starter database.

This chapter assumes that your installation of the Oracle Database 11*g* software (see Chapter 1 and the appendix titled "Installation and Configuration") completed successfully and that you have an Oracle database that uses an earlier version of the Oracle software on the same server. Note that whether you are installing from scratch or upgrading a previous version of the Oracle Database, there are distinct advantages to installing the Oracle Database 11*g* software and creating the database in separate steps. When installing from scratch, you have greater control over initialization parameters, database file locations, memory allocation, and so forth when you create the database in a separate step; when upgrading from a previous release, installing the software first provides you with the Oracle Pre-Upgrade Information Tool that you use against the existing database to alert you to any potential compatibility problems when you upgrade to Oracle Database 11*g*. To upgrade that database, you have four options:

- **Use the Database Upgrade Assistant (DBUA) to guide and perform the upgrade in place.** The old database will become an Oracle 11*g* database during this process. DBUA supports both Oracle Real Application Clusters (RAC) and Automatic Storage Management (ASM); you can launch DBUA as part of the installation process or as a standalone tool after installation. Oracle strongly recommend using DBUA for Oracle Database major releases or patch release upgrades.

- **Perform a manual upgrade of the database.** The old database will become an Oracle 11*g* database during this process. While you have very precise control over every step of the process, this method is more susceptible to error if you miss a step or forget a prerequisite step.

- **Use the Export and Import (or Oracle Data Pump) utilities to move data from an earlier version of Oracle to the Oracle 11*g* database.** Two separate databases will be used—the old database as the source for the export and the new database as the target for the import. If you are upgrading from Oracle Database 10*g*, you will use Oracle Data Pump to move your data from the old database to the new database.

- **Copy data from an earlier version of Oracle to an Oracle 11*g* database.** Two separate databases will be used—the old database as the source for the copy and the new database as the target for the copy. This method is the most straightforward because your migration consists primarily of **create table as select** SQL statements referencing the old and new databases; however, unless your database has very few tables and you aren't concerned with using existing SQL tuning sets, statistics, and so forth, Oracle does not recommend this method for production databases.

Upgrading a database in place—via either the Database Upgrade Assistant or the manual upgrade path—is called a *direct upgrade*. Because a direct upgrade does not involve creating a second database for the one being upgraded, it may complete faster and require less disk space than an indirect upgrade.

**NOTE**
*Direct upgrade of the database to version 11 is only supported if your present database is using one of these releases of Oracle: 9.2.0.4, 10.1.0.2, or 10.2.0.1. If you are using any other release, you will first have to upgrade the database to one of those releases or you will need to use a different upgrade option. Oracle 8.0.6 is only supported for some versions (generally 64-bit), so be sure to check the online certification matrixes at Oracle's Metalink site or in the Oracle Database Upgrade Guide.*

**NOTE**
*Plan your upgrades carefully; you may need to allow time for multiple incremental upgrades (such as from 8.1.7 to 8.1.7.4 to 9.2.0.8) prior to upgrading to Oracle Database 11g.*

# Choosing an Upgrade Method

As described in the previous section, two direct upgrade and two indirect upgrade paths are available. In this section, you will see a more detailed description of the options, followed by usage descriptions.

In general, the direct upgrade paths will perform the upgrade the fastest because they upgrade the database in place. The other methods involve copying data, either to an Export dump file on the file system, across a database link, or via a Data Pump export. For very large databases, the time required to completely re-create the database via the indirect methods may exclude them as viable options.

The first direct method relies on the *Database Upgrade Assistant (DBUA)*. DBUA is an interactive tool that guides you through the upgrade process. DBUA evaluates your present database configuration and recommends modifications that can be implemented during the upgrade process. These recommendations may include the sizing of files and the specifications for the new SYSAUX tablespace if you are upgrading from a version previous to 10*g*. After you accept the recommendations, DBUA performs the upgrade in the background while a progress panel is displayed. DBUA is very similar in approach to the Database Configuration Assistant (DBCA). As discussed in Chapter 1 and the appendix, DBCA is a graphical interface to the steps and parameters required to make the upgrade a success.

The second direct method is called a *manual upgrade*. Whereas DBUA runs scripts in the background, the manual upgrade path involves database administrators running the scripts themselves. The manual upgrade approach gives you a great deal of control, but it also adds to the level of risk in the upgrade because you must perform the steps in the proper order.

You can use the original Export and Import (or Oracle Data Pump Export/Import starting with Oracle Database 10*g*) as an indirect method for upgrading a database. In this method, you export the data from the old version of the database and then import it into a database that uses the new version of the Oracle software. This process may require disk space for multiple copies of the data—in the source database, in the Export dump file, and in the target database. In exchange for these costs, this method gives you great flexibility in choosing which data will be migrated. You can select specific tablespaces, schemas, tables, and rows to be exported.

In the Export/Import and Data Pump methods, the original database is not upgraded; its data is extracted and moved, and the database can then either be deleted or be run in parallel with the new database until testing of the new database has been completed. In the process of performing the export/import, you are selecting and reinserting each row of the database. If the database is very large, the import process may take a long time, impacting your ability to provide the upgraded database to your users in a timely fashion. See Chapter 12 for details on the Export/Import and Data Pump utilities.

**NOTE**
*Depending on the version of the source database, you will need to use a specific version of the Export and Import utilities. See the section "Export and Import Versions to Use" later in this chapter.*

In the data-copying method, you issue a series of **create table as select . . .** or **insert into . . . select** commands that cross database links (see Chapter 16) to retrieve the source data. The tables are created in the Oracle 11*g* database based on queries of data from a separate source database. This method allows you to bring over data incrementally and to limit the rows and columns migrated. However, you will need to be careful that the copied data maintains all the necessary relationships among tables as well as any indexes or constraints. As with the Export/Import method, this method may require a significant amount of time for large databases.

**NOTE**
*If you are changing the operating platform at the same time, you can use transportable tablespaces to move the data from the old database to the new database. For very large databases, this method may be faster than the other data-copying methods. See Chapter 17 for the details on transportable tablespaces.*

Selecting the proper upgrade method requires you to evaluate the technical expertise of your team, the data that is to be migrated, and the allowable downtime for the database during the migration. In general, using DBUA will be the method of choice for very large databases, whereas smaller databases may use an indirect method.

# Before Upgrading

Prior to beginning the migration, you should back up the existing database and database software. If the migration fails for some reason and you are unable to revert the database or software to its earlier version, you will be able to restore your backup and re-create your database.

You should develop and test scripts that will allow you to evaluate the performance and functionality of the database following the upgrade. This evaluation may include the performance of specific database operations or the overall performance of the database under a significant user load.

Prior to executing the upgrade process on a production database, you should attempt the upgrade on a test database so any missing components (such as operating system patches) can be identified and the time required for the upgrade can be measured.

Oracle Database 11*g* includes the Pre-Upgrade Information Tool called utlu111i.sql. This tool is included in the installation files in the directory $ORACLE_HOME/rdbms/admin. Copy this script to a location accessible by the old database, connect to the old database with SYSDBA privileges, and run this tool from a SQL*Plus session similar to the following:

```
SQL> spool upgrade_11g_info.txt
SQL> @utlu111i.sql
SQL> spool off
```

Review the file upgrade_11g_info.txt for adjustments you should make before performing the actual upgrade; these adjustments include increasing the size of tablespaces, removing obsolete initialization parameters, and revoking obsolete roles such as CONNECT. As of Oracle Database 11*g,* the CONNECT role only contains the CREATE SESSION privilege. You need to grant permissions to users with the CONNECT role before upgrading. Here is a query you can use to identify users granted the CONNECT role:

```
SELECT grantee FROM dba_role_privs
     WHERE granted_role = 'CONNECT' and grantee NOT IN (
            'SYS', 'OUTLN', 'SYSTEM', 'CTXSYS', 'DBSNMP',
            'LOGSTDBY_ADMINISTRATOR', 'ORDSYS',
            'ORDPLUGINS',  'OEM_MONITOR', 'WKSYS', 'WKPROXY',
            'WK_TEST', 'WKUSER', 'MDSYS', 'LBACSYS', 'DMSYS',
            'WMSYS',  'OLAPDBA', 'OLAPSVR', 'OLAP_USER',
            'OLAPSYS', 'EXFSYS', 'SYSMAN', 'MDDATA',
            'SI_INFORMTN_SCHEMA', 'XDB', 'ODM');
```

Prior to performing a direct upgrade, you should analyze the data dictionary tables. During the upgrade process to Oracle 11*g,* the data dictionary will be analyzed if it has not been analyzed already, so performing this step in advance will aid the performance of the upgrade. For an Oracle version 10*g* database, you can use this procedure invocation to gather dictionary stats:

```
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS;
```

# Using the Database Upgrade Assistant

You can start the Database Upgrade Assistant (DBUA) via the

```
dbua
```

command (in Unix environments) or by selecting Database Upgrade Assistant from the Oracle Configuration and Migration Tools menu option (in Windows environments). If you are using a Unix environment, you will need to enable an X Window display prior to starting DBUA.

When started, DBUA will display a Welcome screen. At the next screen, select the database you want to upgrade from the list of available databases. You can upgrade only one database at a time.

After you make your selection, the upgrade process begins. DBUA will perform pre-upgrade checks (such as for obsolete initialization parameters or files that are too small). DBUA will then create the SYSAUX tablespace, a standard tablespace in all Oracle 10*g* and 11*g* databases. You can override Oracle's defaults for the location and size parameters for the datafiles used by the SYSAUX tablespace.

DBUA will then prompt you to recompile invalid PL/SQL objects following the upgrade. If you do not recompile these objects after the upgrade, the first user of these objects will be forced to wait while Oracle performs a run-time recompilation.

DBUA will then prompt you to back up the database as part of the upgrade process. If you have already backed up the database prior to starting DBUA, you may elect to skip this step. If you choose to have DBUA back up the database, it will shut down the database and perform an offline backup of the datafiles to the directory location you specify. DBUA will also create a batch file in that directory to automate the restoration of those files to their earlier locations.

The next step is to choose whether to enable Oracle Enterprise Manager (OEM) to manage the database. If you enable the Oracle Management Agent, the upgraded database will automatically be available via OEM.

You will then be asked to finalize the security configuration for the upgraded database. As with the database-creation process, you can specify passwords for each privileged account or you can set a single password to apply to all the OEM user accounts.

Finally, you will be prompted for details on the flash recovery area location (see Chapter 14), the archive log setting, and the network configuration. A final summary screen displays your choices for the upgrade, and the upgrade starts when you accept them. After the upgrade has completed, DBUA will display the Checking Upgrade Results screen, showing the steps performed, the related log files, and the status. The section of the screen titled Password Management allows you to manage the passwords and the locked/unlocked status of accounts in the upgraded database.

If you are not satisfied with the upgrade results, you can choose the Restore option. If you used DBUA to perform the backup, the restoration will be performed automatically; otherwise, you will need to perform the restoration manually.

When you exit DBUA after successfully upgrading the database, DBUA removes the old database's entry in the network listener configuration file, inserts an entry for the upgraded database, and reloads the file.

## Performing a Manual Direct Upgrade

In a manual upgrade, you must perform the steps that DBUA performs. The result will be a direct upgrade of the database in which you are responsible for (and control) each step in the upgrade process.

You should use the Pre-Upgrade Information Tool to analyze the database prior to its upgrade. As I mentioned earlier in this chapter, this tool is provided in a SQL script that is installed with the Oracle Database 11*g* software; you will need to run it against the database to be upgraded. The file, named utlu111i.sql, is located in the $ORACLE_HOME/rdbms/admin subdirectory under the Oracle 11*g* software home directory. You should run that file *in the database to be upgraded* as a SYSDBA-privileged user, spooling the results to a log file. The results will show potential problems that should be addressed prior to the upgrade.

If there are no issues to resolve prior to the upgrade, you should shut down the database and perform an offline backup before continuing with the upgrade process. This ensures that if you have any serious problems with the database upgrade, you can always get back to the state of your old database as of when you started the upgrade process.

Once you have a backup you can restore if needed, you are ready to proceed with the upgrade process. The process is detailed and script-based, so you should consult with the Oracle installation and upgrade documentation for your environment and version. The steps are as follows:

1. Copy configuration files (init.ora, spfile.ora, password file) from their old location to the new Oracle software home directory. By default, the configuration files are found in the /dbs subdirectory on Unix platforms and the \database directory on Windows platforms.
2. Remove obsolete and deprecated initialization parameter from the configuration files identified in the Pre-Upgrade Information Tool. Update any initialization parameters to at least the minimum values specified in the Pre-Upgrade Information Tool report. Use full pathnames in the parameter files.
3. If you are upgrading a cluster database, set the CLUSTER_DATABASE initialization parameter to FALSE. After the upgrade, you must set this initialization parameter back to TRUE.
4. Shut down the instance.
5. If you are using Windows, stop the service associated with the instance and delete the Oracle service at the command prompt. For Oracle 8.0, use the command

   ```
   NET STOP OracleServiceName
   ORADIM –DELETE –SID instance_name
   ```

   Next, create the new Oracle Database 11*g* service using the ORADIM command, as shown here. The variables for this command are shown in the following table.

   ```
   C:\> ORADIM -NEW -SID SID -INTPWD PASSWORD -MAXUSERS USERS
        -STARTMODE AUTO -PFILE ORACLE_HOME\DATABASE\INITSID.ORA
   ```

   | Variable | Description |
   |---|---|
   | *SID* | The name of the SID (instance identifier) of the database you are upgrading. |
   | *PASSWORD* | The password for the new release 11.1 database instance. This is the password for the user connected with SYSDBA privileges. If you do not specify INTPWD, operating system authentication is used and no password is required. |
   | *USERS* | The maximum number of users who can be granted SYSDBA and SYSOPER privileges. |
   | *ORACLE_HOME* | The release 11.1 Oracle home directory. Ensure that you specify the full pathname with the -PFILE option, including the drive letter of the Oracle home directory. |

6. If your operating system is Unix or Linux, make sure the environment variables ORACLE_HOME and PATH point to the new release 11.1 directories, ORACLE_SID is set to the existing database's SID, and the file /etc/oratab points to the new Oracle Database 11*g* home directory. In addition, any server or client-side scripts that set ORACLE_HOME must be changed to point to the new Oracle software home directory.

7. Log into the system as the owner of the Oracle Database 11*g* software.
8. Change your directory to the $ORACLE_HOME/rdbms/admin subdirectory under the Oracle software home directory.
9. Connect to SQL*Plus as a user with SYSDBA privileges.
10. Issue the **startup upgrade** command.
11. Use the **spool** command to log the results of the following steps.
12. Create a SYSAUX tablespace via the **create tablespace** command if you are upgrading from a release prior to 10.1. You should allocate SYSAUX between 500MB and 5GB of disk space, depending on the number of user objects. SYSAUX must be created with the following clauses: **online**, **permanent**, **read write**, **extent management local**, and **segment space management auto**. All those clauses except **segment space management auto** are the defaults. See the output from the Pre-Upgrade Information Tool for suggested sizing of the SYSAUX tablespace. Here's an example:

   ```
   create tablespace SYSAUX
    datafile '/u01/oradata/db1/sysaux01.dbf'
    size 500m reuse
    extent management local
    segment space management auto
    online;
   ```

13. Run the script catupgrd.sql in the 11*g* environment. This script automatically determines which upgrade scripts must be run, runs them, and then shuts down the database.
14. Stop spooling (via **spool off**) and review the spool file for errors. Resolve any problems identified there. Restart the database at the SQL*Plus prompt using the **startup** command.
15. Run the utlu111s.sql file to upgrade Oracle components such as Oracle Text, Oracle Ultra Search, Oracle Application Express, and the Oracle Server itself. You run it as follows:

   ```
   @utlu101s.sql
   ```

16. Oracle will then display the upgrade status of each component. The upgrade elements should all be listed with a status of "VALID."
17. Run the catuppst.sql script, located in $ORACLE_HOME/rdbms/admin to perform upgrade steps that do not require the database to be in UPGRADE mode:

   ```
   @rdbms/admin/catuppst.sql
   ```

18. Run the utlrp.sql script to recompile invalid packages:

   ```
   @utlrp.sql
   ```

19. You can then verify that all packages and classes are valid by using the following SQL:

   ```
   select count(*) from dba_invalid_objects;
   select distinct object_name from dba_invalid_objects;
   ```

20. Exit SQL*Plus.
21. Shut down the database and perform an offline backup of the database; then restart the database. The upgrade is complete.

**NOTE**
*After the upgrade, you should never start your Oracle 11g database with the software from an earlier release.*

# Using Export and Import

Export and Import provide you with an indirect method for the upgrade. You can create an Oracle 11*g* database alongside your existing database and use Export and Import to move data from the old database to the new database. When the movement of the data is complete, you will need to point your applications to connect to the new database instead of the old database. You will also need to update any configuration files, version-specific scripts, and the networking configuration files (tnsnames.ora and listener.ora) to point to the new database.

The advantage to using an Export/Import method is that the existing database is unaffected throughout the upgrade process; however, to ensure that relational integrity remains intact and no new transactions are left behind in the old database, you can run the old database in restricted mode for the duration of the export and upgrade.

## Export and Import Versions to Use

When you create an Export dump file via the Export utility, that file can be imported into all later releases of Oracle. When you create a Data Pump Export dump file, you can only import it into the same or later versions of Data Pump Export. Export dump files are not backward compatible, so if you ever need to revert to an earlier version of Oracle, you will need to carefully select the version of Export and Import used. The following table shows the versions of the Export/Import and Data Pump Export/Import executables you should use when going between versions of Oracle:

| Export From | Import To | Use Export Utility | Use Import Utility |
|---|---|---|---|
| Release 10.2 | Release 11.1 | Data Pump Export 10.2 | Data Pump Import 11.1 |
| Release 10.1 | Release 11.1 | Data Pump Export 10.1 | Data Pump Import 11.1 |
| Release 9.2 | Release 11.1 | Original Export 9.2 | Original Import 11.1 |
| Release 8.1.7 | Release 11.1 | Original Export 8.1.7 | Original Import 11.1 |
| Release 8.0.6 | Release 11.1 | Original Export 8.0.6 | Original Import 11.1 |
| Release 7.3.4 | Release 11.1 | Original Export 7.3.4 | Original Import 11.1 |

Note that when you are exporting in order to downgrade your database release, you should use the older version of the Export utility to minimize compatibility problems. You may still encounter compatibility problems if the newer version of the database uses new features (such as new datatypes) that the old version will not support.

## Performing the Upgrade

Export the data from the source database using the version of the Export utility specified in the prior section. Perform a consistent export or perform the export when the database is not available for updates during and after the export.

**NOTE**
*If you have little free space available, you may back up and delete the existing database at this point and then install Oracle Database 11g software and create a target database for the import. If at all possible, maintain the source and target databases concurrently during the upgrade. The only benefit of having only one database on the server at a time is that they can share the same database name.*

Install the Oracle Database 11*g* software and create the target database. In the target database, pre-create the users and tablespaces needed to store the source data. If the source and target databases will coexist on the server, you need to be careful not to overwrite datafiles from one database with datafiles from the other. The Import utility will attempt to execute the **create tablespace** commands found in the Export dump file, and those commands will include the datafile names from the source database. By default, those commands will fail if the files already exist (although this can be overridden via Import's DESTROY parameter). Pre-create the tablespaces with the proper datafile names to avoid this problem.

**NOTE**
*You can export specific tablespaces, users, tables, and rows.*

Once the database has been prepared, use Import or Data Pump Import to load the data from the Export dump file into the target database. Review the log file for information about objects that did not import successfully. See Chapter 11 for detailed instructions on how to use Data Pump Export and Import.

## Using the Data-Copying Method

The data-copying method requires that the source database and target database coexist. This method is most appropriate when the tables to be migrated are fairly small and few in number. As with the Export/Import method, you must guard against transactions occurring in the source database during and after the extraction of the data. In this method, the data is extracted via queries across database links.

Create the target database using the Oracle Database 11*g* software and then pre-create the tablespaces, users, and tables to be populated with data from the source database. Create database links (see Chapter 16) in the target database that access accounts in the source database. Use the **insert as select** command to move data from the source database to the target.

The data-copying method allows you to bring over just the rows and columns you need; your queries limit the data migrated. You will need to be careful with the relationships between the tables in the source database so that you can re-create them properly in the target database. If you have a long application outage available for performing the upgrade and you need to modify the data structures during the migration, the data-copying method may be appropriate for your needs. Note that this method requires that the data be stored in multiple places at once, thus impacting your storage needs.

To improve the performance of this method, you may consider the following options:

- Disable all indexes and constraints until all the data has been loaded.
- Run multiple data-copying jobs in parallel.

- Use the parallel query option to enhance the performance of individual queries and inserts.
- Use the APPEND hint to enhance the performance of inserts.

As of Oracle 10*g*, you can use cross-platform transportable tablespaces. When transporting tablespaces, you export and import only the metadata for the tablespace, while the datafiles are physically moved to the new platform. For very large databases, the time required to move the datafiles may be significantly shorter than the time required to reinsert the rows. See Chapter 17 for details on the use of transportable tablespaces; see Chapter 8 for additional advice on performance tuning.

## After Upgrading

Following the upgrade, you should double-check the configuration and parameter files related to the database, particularly if the instance name changed in the migration process. These files include

- The tnsnames.ora file
- The listener.ora file
- Programs that may have hard-coded instance names in them

**NOTE**
*You will need to manually reload the modified listener.ora file if you are not using DBUA to perform the upgrade.*

You should review your database initialization parameters to make sure deprecated and obsolete parameters have been removed; these should have been identified during the migration process when you ran the Pre-Upgrade Information Tool utlu111i.sql. Be sure to recompile any programs you have written that rely on the database software libraries.

Once the upgrade has completed, perform the functional and performance tests identified before the upgrade began. If there are issues with the database functionality, attempt to identify any parameter settings or missing objects that may be impacting the test results. If the problem cannot be resolved, you may need to revert to the prior release. If you performed a full backup before starting the upgrade, you should be able to easily revert to the old release with minimal downtime.