



CHAPTER 1 Getting Started

- **Welcome to Oracle Database 10g and the “Grid”**
- **Upgrading to Oracle Database 10g**
- **The Database Configuration Assistant**
- **The SYSAUX Tablespace**
- **Automatic Storage Management**
- **Oracle Database 10g Real Application Clusters**



Welcome to *Oracle Database 10g New Features*! This chapter is the place to start in your effort to learn all about Oracle Database 10g. In this chapter, we will first look at the “Grid” and see what it is all about. Then we move on to explain how to upgrade to Oracle Database 10g. The following are the specific topics that are covered:

- The “Grid”
- Upgrading to Oracle Database 10g
- The SYSAUX tablespace
- Automatic Storage Management (ASM)

Welcome to Oracle Database 10g and the “Grid”

Oracle Database 10g. If you are wondering what that *g* is all about, it stands for this rather murky concept of the “Grid.” As I was writing this book, I asked some fellow DBAs what they thought the Grid was. There were those who had some idea, and those who were not so sure. In fact, I did a search in the Oracle Beta documentation set, and found little or no reference to the Grid. I did determine that the final release will contain a book that provides concept, configuration, and administration guidance for the Grid, however.

A new concept and little documentation—what is a DBA to do? Never fear, *Oracle Database 10g New Features* is here! So, what is the Grid? The Grid is all about the synergies that can be achieved by aligning the Oracle technologies with the capabilities of existing and future hardware. As Larry Ellison put it, the Grid is “...capacity on demand made up of low-cost parts.” The Grid enables you to do the following:

- **Leverage components made available to the Grid** Through Oracle Real Application Clusters, Automatic Storage Management (ASM), and Ultra-Large Data Files (all of which are discussed in this book), Oracle allows you to take advantage of your hardware investment in a much more efficient manner.
- **Load-balance across the enterprise** Oracle Real Application Clusters, Oracle Streams, and Oracle distributed database technologies support this concept.

- **Share information regardless of its location** This concept is supported through new transportable tablespace features, Oracle Streams (e.g., Replication), external tables, and distributed SQL queries.
- **Schedule resources across the Grid** The new features of the Oracle scheduler and the Oracle Database Resource Manager help the DBA to effectively take better advantage of the Grid.

The Grid may not be 100 percent there with Oracle Database 10g, but it's certainly a start. Oracle Database 10g is full of new features that more fully align it with the concepts of the Grid. In this book, we will look at these new features, and a number of other new features that make Oracle Database 10g an interesting upgrade indeed!

Upgrading to Oracle Database 10g

Oracle Database 10g provides a fairly easy upgrade path for users of older Oracle Database versions. The following versions can directly be upgraded to Oracle Database 10g:

- Oracle Database 8.0.6
- Oracle Database 8.1.7
- Oracle Database 9.0.1
- Oracle Database 9.2

If your database version is not in the preceding list, then you must first upgrade to one of these versions, after which you can upgrade to Oracle Database 10g.

After you are at a supported upgrade level, you can upgrade to Oracle Database 10g by using any one of the following four upgrade options:

- Use the Oracle Database Upgrade Assistant (DBUA). (This was called the Oracle Data Migration Assistant previously.)
- Perform a manual upgrade.
- Use **exp/imp** to copy the data in your database to a new Oracle Database 10g database.
- Use the SQL*Plus **copy** command or the **create table as select** command to copy the data from your current database to your new Oracle Database 10g database.

4 Oracle Database 10g New Features

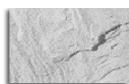


NOTE

Always back up your database before you start your upgrade!

The DBUA

The DBUA is a GUI that is designed for upgrading your Oracle database to Oracle Database 10g. You will have the option of starting the DBUA from the Oracle Universal Installer (OUI) when installing Oracle Database 10g. DBUA guides you through the upgrade of your Oracle database. You can also start the DBUA at any time in stand-alone fashion (from the command line, just enter **dbua**) to upgrade your database. From Windows, you can also start the DBUA from the Start menu (either from the Oracle folder or use Start | Run and click dbua). One nice feature of the DBUA is that it will offer to back up your database for you. This feature does have some limited functionality, because backups to removable media are not supported. DBUA supports both normal database upgrades and Real Application Clusters database upgrades.



CAUTION

*Oracle Database 10g only supports a direct downgrade back to Oracle release 9.2.0.3 or later. You can use **imp/exp** (Oracle's import/export utilities), however, to move the migrated database data to other versions of Oracle Database. I strongly advise that you test this method of downgrading on a nonproduction server first, if you plan to use it.*

Performing Manual Upgrades

Manual upgrades (my personal favorite) allow you to use a series of scripts and utilities to upgrade your database. The summary steps of manual upgrades include

- Develop a test plan to run after your upgrade.
- Back up your database.
- Run the Upgrade Information Tool (UIT), which is a SQL script, `utlu101i.sql`, located in the directory `$ORACLE_HOME/rdbms/admin`. This script analyzes your database before you upgrade it and alerts you to any problems that might endanger the successful upgrade of your database. If you want your output in XML format, you should run `utlu101x.sql`.

- Upgrade the database. Follow the Oracle upgrade instructions for your specific version and operating system. This step includes the creation of the new SYSAUX tablespace, which is new in Oracle Database 10g. (This tablespace is described in detail later in “The SYSAUX Tablespace.”)
- Check the component registry (DBA_REGISTRY) to make sure your upgrade was successful.
- Back up your new Oracle Database 10g database.
- Run your test plan and validate your upgrade.

About the compatible Parameter

Once you have upgraded to Oracle Database 10g, the **compatible** parameter can be set no lower than 9.2.0. Thus, if you are upgrading from 8.0.6, you need to set **compatible** to 9.2.0 before you can open your database under Oracle Database 10g. The Oracle upgrade manual (*Oracle 10g Upgrade Guide*) provides detailed instructions on setting the **compatible** parameter. Once you are satisfied that the database can operate under Oracle Database 10g, you can set the **compatible** parameter to 10.0. Note that, once you set the **compatible** parameter to 10.0, you cannot set it back. This is different than in previous versions of Oracle. Also note that the command **alter database reset compatibility** is now obsolete.

There are a number of other possible upgrade issues that you need to deal with depending on the database features that you are using. I strongly suggest that you carefully review the Oracle Upgrade documentation and that you test your Oracle Database 10g upgrades several times before doing one for real in production.

One final upgrade thought—I suggest that you do not use any of the new Oracle Database 10g features in a production environment until you have tested the feature thoroughly. While Oracle does its best to regression-test new features, there are always a few kinks to be worked out in the beginning. If you find a new feature irresistible (and after you read this book, I hope you do!), then by all means try it out. Test it over and over to make sure it works the way it’s intended, and that it doesn’t have some nasty impacts, like causing performance problems or causing your database to crash. Also, check Oracle MetaLink, and even open an Oracle iTAR, before you use a new feature that will be a prominent part of your design.

Other Upgrade Methods

The use of the Oracle **exp/imp** utilities is supported for migrating your Oracle database data to Oracle Database 10g. You will use the export utility associated with the version of the database you are currently on (e.g., 8.0.6) to create the dump file. Use the **imp** utility from the Oracle Database 10g database to import the dump file created for the

Jonathan Says...

Remember that **sql_trace** (possibly started by a logon trigger that calls the packaged procedure **dbms_support.start_trace**) is your best friend when you are testing. Many of the enhancements and features that appeared in Oracle9i Database were supported by PL/SQL packages and recursive SQL. Expect more of the same approach to appear in Oracle Database 10g.

If you switch on **sql_trace** when testing a feature, you may find out what Oracle Database 10g is doing under the covers to support that feature, and discover that a feature that looks good on paper has a nasty side effect that makes it unsuitable for your production system.

Another little trick for discovering hidden costs when you start to test new data structures is to start with a clean schema, create an example of the new data structure, and then query the **USER_OBJECTS** view to discover what hidden objects Oracle has created to support that structure.

upgrade. The Oracle upgrade manual provides a complete set of instructions on how to perform this type of upgrade.

Finally, you can use the SQL*Plus **copy** command or the SQL **create table as select** command to move your database data to a new Oracle Database 10g instance. Again, the Oracle upgrade manual provides a complete set of instructions on how to perform this type of upgrade.

The Database Configuration Assistant

As an author who admits to hating GUIs, I rather like the Database Configuration Assistant (DBCA). Creation of a small database is a great way to test your new Oracle software install. Oracle Database 10g has added new functionality to the DBCA, including the ability to perform Real Application Cluster database installs and configuration of Automatic Storage Management (a new feature that is described later in this chapter).

The SYSAUX Tablespace

In the previous section you might have noticed that when you upgrade your Oracle database to Oracle Database 10g, you need to create a new tablespace called SYSAUX. Also, when you create your first Oracle Database 10g database, Oracle will create SYSAUX as well. The SYSAUX tablespace is a new tablespace component in Oracle

Database 10g. This section first discusses the SYSAUX tablespace and then reviews some Oracle-supplied procedures that allow you to perform maintenance tasks on the SYSAUX tablespace.

Introducing the SYSAUX Tablespace

The SYSAUX tablespace is a new tablespace that is required in Oracle Database 10g. SYSAUX is a secondary tablespace for storage of a number of database components that were previously stored in the SYSTEM tablespace. The SYSAUX tablespace is created as a locally managed tablespace using automatic segment space management.

Previously, many Oracle features required their own separate tablespaces (such as the RMAN recovery catalog, Ultra Search, Data Mining, XDP, and OLAP). This increases the management responsibility of the DBA. The SYSAUX tablespace consolidates these tablespaces into one location, which becomes the default tablespace for these Oracle features.

When you create an Oracle database, Oracle creates the SYSAUX tablespace for you by default. If you are using Oracle Managed Files (OMF), then the tablespace is created in the appropriate OMF directory. If you use the **sysaux datafile** clause in the **create database** statement, then the SYSAUX tablespace datafile(s) will be created in the location you define. Finally, if no **sysaux datafile** clause is included and OMF is not configured, Oracle creates the SYSAUX tablespace in a default location that is OS-specific. Here is an example of a **create database** statement with the **sysaux datafile** clause in it:

```
CREATE DATABASE my_db
DATAFILE 'c:\oracle\oradata\my_db\my_db_system_01.dbf' SIZE 300m
SYSAUX DATAFILE 'c:\oracle\my_db\my_db_sysaux_01.dbf' SIZE 100m
DEFAULT TEMPORARY TABLESPACE dtemp_tbs tempfile
'c:\oracle\my_db\my_db_temp_01.dbf' SIZE 100m
UNDO TABLESPACE undo_tbs_one DATAFILE
'c:\oracle\my_db\my_db_undo_tbs_one_01.dbf' SIZE 100m;
```

As stated earlier in this chapter, when you migrate to Oracle Database 10g, you need to create the SYSAUX tablespace as a part of that migration. You do this after mounting the database under the new Oracle Database 10g database software. Once you have mounted it, you should open the database in migrate mode with the **startup migrate** command. Once the database is open, you can create the SYSAUX tablespace. Here is the **create tablespace** statement that you would use to perform this operation:

```
CREATE TABLESPACE sysaux
DATAFILE 'c:\oracle\oradata\my_db\my_db_sysaux_01.dbf' SIZE 300m
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
```

8 Oracle Database 10g New Features

The SYSAUX tablespace must be created with the attributes shown in the preceding example. The following restrictions apply to the usage of the SYSAUX tablespace in Oracle Database 10g:

- When migrating to Oracle Database 10g, you can create the SYSAUX tablespace only when the database is open in migrate mode.
- Also, when migrating to Oracle Database 10g, if a tablespace is already named SYSAUX, you need to remove it or rename it while you are in migrate mode.
- Once you have opened your Oracle Database 10g database, you cannot drop the SYSAUX tablespace. If you try, an error will be returned.
- You cannot rename the SYSAUX tablespace during normal database operations.
- The SYSAUX tablespace cannot be transported to other databases via Oracle's transportable tablespace feature.

Once the SYSAUX tablespace is in place and the database has been upgraded, you can add or resize datafiles associated with a SYSAUX tablespace just as you would any other tablespace through the **alter tablespace** command, as shown in this example:

```
ALTER TABLESPACE sysaux ADD DATAFILE  
'd:\oracle\oradata\my_db\my_db_sysaux_01.dbf' SIZE 200M;
```

Managing Occupants of the SYSAUX Tablespace

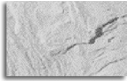
Each set of application tables within the SYSAUX tablespace is known as an *occupant*. Oracle provides some new views to help you monitor space usage of occupants within the SYSAUX tablespace and some new procedures you can use to move the occupant objects in and out of the SYSAUX tablespace.

First, Oracle provides a new view, V\$SYSAUX_OCCUPANTS, to manage occupants in the SYSAUX tablespace. This view allows you to monitor the space usage of occupant application objects in the SYSAUX tablespace, as shown in this example:

```
SELECT occupant_name, space_usage_kbytes FROM v$sysaux_occupants;
```

In this case, Oracle will display the space usage for the occupants, such as the RMAN recovery catalog.

If you determine that you need to move the occupants out of the SYSAUX tablespace, then the MOVE_PROCEDURE column of the V\$SYSAUX_OCCUPANTS view will indicate the procedure that you should use to move the related occupant from the SYSAUX tablespace to another tablespace. This can also be a method of “reorganizing” your component object tables, should that be required.



NOTE

The loss of the SYSAUX tablespace is not fatal to your database. In our testing it appears that the only real impact is that certain functionality related to the occupants of the SYSAUX tablespace is lost.

Automatic Storage Management

Oracle Database 10g introduces Automatic Storage Management (ASM), a service that provides management of disk drives. ASM can be used on a variety of configurations, including Oracle9i Database RAC installations. ASM is an alternative to the use of raw or cooked file systems and is part of Oracle’s overall desire to make management of the Oracle Database 10g easier, overall.

ASM Features

ASM offers a number of features, including:

- Simplified daily administration
- The performance of raw disk I/O for all ASM files
- Compatibility with any type of disk configuration, be it just a bunch of disks (JBOD) or complex Storage Area Network (SAN)
- Use of a specific file-naming convention to name files, enforcing an enterprise-wide file-naming convention
- Prevention of the accidental deletion of files, since there is no file system interface and ASM is solely responsible for file management
- Load balancing of data across all ASM managed disk drives, which helps improve performance by removing disk hot spots
- Dynamic load balancing of disks as usage patterns change and when additional disks are added or removed
- Ability to mirror data on different disks to provide fault tolerance

- Support of vendor-supplied storage offerings and features
- Enhanced scalability over other disk-management techniques

ASM can work in concert with existing databases that use raw or cooked file systems. You can choose to leave existing file systems in place or move the database datafiles to ASM disks. Additionally, new database datafiles can be placed in either ASM disks or on the preexisting file systems. Databases can conceivably contain a mixture of file types, including raw, cooked, OMF, and ASM files (though the management of such a system would be more complex).

ASM makes adding and removing of disks easy. It will rebalance the data on the underlying disks as they are added and removed. This results in the best possible performance for your database.

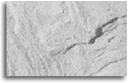
ASM is not a replacement for your existing OS file systems. The disks assigned to ASM will not be visible from the OS. Thus, ASM is not going to be used for Oracle software installs, administrative directories, or the location of parameter files.

The ASM Instance

ASM starts with the ASM instance that is responsible for the management of the various disk groups and the associated files. If ASM is being used by a database then the ASM instance must be started before the Oracle database instance is started. The ASM instance has a very small footprint and does not consume much in the way of resources. The ASM instance will mount the disks, create an extent map, and then pass this information on to the Oracle database instance when it is started. Note that ASM has few real “run-time” responsibilities and that all I/O requests to the ASM file system are handled by the Oracle database itself. The only time that ASM gets involved during run time is if there is some disk configuration change. This would include file removal or creation, or addition or removal of a disk.

To create an ASM instance, create a parameter file called `init_osm.ora`. In the ASM parameter file, you need to set the **instance_type** parameter to a value of `OSM`. The **instance_type** parameter defines the instances as an ASM instance. Generally, other ASM-related parameters take on default values, which can be accepted. Once the parameter file is created, start the ASM instance, using the **startup** command. Shutting down the ASM instance is done via the **shutdown** command (**normal**, **immediate**, **force**, and **abort** commands are supported). Note that this will result in the shutdown of all database instances related to the ASM instance and that the mode used to shut down the ASM instance will be the same mode used to shut down the associated database instances.

Also, the Oracle Database 10g DBCA can be used to create, configure, and start up an ASM instance. If ASM is already in use, the DBCA will allow you to take advantage of it. If ASM is not installed, you will be given an opportunity to install it.

**NOTE**

The DBCA will give you an option to create an ASM instance, if one does not already exist! This includes configuration of the parameter file of that instance.

Setting Up ASM Disks

Once the ASM instance is in place, you are ready to set up disk groups. Much like a logical volume in OS logical volume managers, a disk group is a logical collection of a number of physical disk devices. You create a disk group with the **create diskgroup** command. Within the **create diskgroup** command, you define both the main disk group and the failover disk group that is associated with that main group. Here is an example of the creation of a disk group:



```
Create diskgroup disk_group_001
FAILGROUP fail_group_01 DISK
'/dev/disk1', '/dev/disk2', '/dev/disk3',
FAILGROUP fail_group_02 DISK
'/dev/disk5', '/dev/disk6', '/dev/disk7';
```

ASM offers redundancy of data in the definition of the disk groups. This example creates a disk group with two failure groups. Each failure group has three disks assigned to it. Oracle will mirror the two groups, and will stripe Oracle files across one group, and then mirror that to the other group. Normal redundancy, as you see in this example, is the default, and requires that two failure groups be allocated. High redundancy requires three **failgroup** disk definitions. Another option, external redundancy, does not mirror at all in ASM, and rather leaves redundancy to an external entity (such as the disk array that might be mirrored).

When you create a disk group, you can constrain how much space within the disks are allocated to the disk group. This allows you to keep some space in reserve for later use. You cannot allocate more space to a disk group than already exists.

When the disk group is allocated, ASM will balance database files across the disks of that group, and will rebalance the group when disks are added or removed. All of this can take place without needing to cycle the database, thus availability of the database is not impacted.

**NOTE**

*The **create diskgroup** command is only run from the ASM instance, not from a normal database instance.*

The **alter diskgroup** command is provided to help you manage existing disk groups. The **alter diskgroup** command allows you to add disks to an existing group,

12 Oracle Database 10g New Features

remove disks from an existing disk group, and resize disks in an existing disk group. Also, the **alter diskgroup** command allows you to manually rebalance disks and mount or dismount a given disk group.

You can also create directories within your disk groups by using the **alter diskgroup** command, as shown in this example:

```
Alter diskgroup disk_group_001 add directory  
'+disk_group_001/data_001';
```

You can also rename and drop directories with the **alter diskgroup** command.

Addressing ASM Disks

Once the disk group is assigned, the Oracle DBA need only refer to the disk group when creating a tablespace. Here is an example:

```
Create tablespace data_tbs datafile '+disk_group_001' size 300MB;
```

Note that in this case we referenced the ASM disk group that we created in the previous section. After this command is issued, Oracle will create the datafile in the disk group assigned, using an OMF-compliant naming convention. If you need to manage these files later on (for example, you need to grow the datafile), you can find the files in the DBA_DATA_FILES or V\$DATAFILE views, as you previously did.

ASM and Data Dictionary Views

Several new data dictionary views exist to help you manage ASM. These data dictionary views are available both when connected to the ASM instance as well as to any Oracle Database 10g database. Each view is slightly different in its presentation depending on whether the instance you are looking at is an ASM instance or a database instance (and some views are only used in the ASM instance). Let's quickly look at these views.

V\$ASM_ALIAS

When you are connected to an ASM instance, the V\$ASM_ALIAS view displays a single row for every alias present in every disk group mounted by the ASM instance. This view is not used when connected to a normal database instance (no rows will be displayed).

V\$ASM_CLIENT

When you are connected to an ASM instance, the V\$ASM_CLIENT view displays a single row for every database instance using a disk group managed by the ASM

instance. When you are connected to a normal database instance and the database has open ASM files, the V\$ASM_CLIENT view displays a single row for the ASM instance.

V\$ASM_DISK

When you are connected to an ASM instance, the V\$ASM_DISK view displays a single row for every disk discovered by the ASM instance, including disks that are not part of any disk group. When you are connected to a normal database instance, the V\$ASM_DISK view only displays rows for disks in disk groups in use by the database instance.

V\$ASM_DISKGROUP

When you are connected to an ASM instance, the V\$ASM_DISKGROUP view displays a single row for each disk group discovered by the ASM instance. When you are connected to a database instance, the V\$ASM_DISKGROUP view displays a single row for every ASM disk group mounted by the local ASM instance.

V\$ASM_FILE

When you are connected to an ASM instance, the V\$ASM_FILE view displays a single row for every ASM file in every disk group mounted by the ASM instance. This view is not used when connected to a normal database instance (no rows will be displayed).

V\$ASM_OPERATION

When you are connected to an ASM instance, the V\$ASM_OPERATION view displays a single row for every active ASM long-running operation executing in the ASM instance. This view is not used when connected to a normal database instance (no rows will be displayed).

V\$ASM_TEMPLATE

When you are connected to an ASM instance, the V\$ASM_TEMPLATE view displays a single row for every template present in every disk group mounted by the ASM instance. When you are connected to a database instance, the V\$ASM_TEMPLATE view displays a single row for every ASM disk group mounted by the local ASM instance.

This is just a quick introduction to the power of ASM. A small book probably could be written on just this new feature alone. Still, I hope I have gotten you interested in it!

Oracle Database 10g Real Application Clusters

Oracle9i Database introduced Real Application Clusters (RAC), which was a successor to Oracle Parallel Server. Oracle Database 10g offers several RAC-related improvements, including:

- Portable Clusterware
- Rolling upgrades
- Cluster Ready Services
- Enhancements to the Database Configuration Assistant and the Database Upgrade Assistant
- Enhanced recovery parallelism on multiple CPU systems
- A new parameter, `gcs_server_processes`

Portable Clusterware

First introduced on limited platforms in Oracle9i Database, Portable Clusterware is now available on most Oracle Database 10g platforms. Clusterware provides clustering services for the Oracle Database 10g RAC configuration. Prior to Oracle Database 10g, most RAC configurations required that third-party clusterware software be purchased and installed. Now, in Oracle Database 10g, Oracle offers clusterware that runs on all platforms, eliminating the need for vendor-offered clusterware. Oracle's Portable Clusterware offers a number of features:

- Infiniband high-speed network support
- Simplified client installs
- Simplified cluster installs

Rolling Upgrades

Oracle Database 10g provides for limited rolling upgrades for RAC clusters through the use of the `opatch` utility. These patches must be certified by Oracle to be used within a rolling upgrade scheme. There are a number of restrictions on what patches may be used as rolling upgrade patches, such as the fact that the patch cannot impact

the contents of the database data dictionary and the patch cannot impact RAC internode communications. Rolling upgrades are available for one-off patches only as of this writing, thus patch sets cannot be applied in a rolling fashion.

Cluster Ready Services

Oracle Database 10g introduces Cluster Ready Services (CRS), which provides additional management services to a database cluster, such as node membership, group services, global resource management, and high availability. CRS also interacts with vendor-supplied software in its node management activities.

The Oracle Universal Installer will install CRS on nodes in an `ORACLE_HOME` location that is different from that of the RDBMS software's `ORACLE_HOME` install. As with the normal RDBMS software, you can install CRS on each individual node, or into a common `ORACLE_HOME` location for all the nodes.

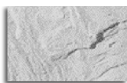
CRS allows you to assign services to specific instances in the cluster, and these services can be configured in such a way as to allow you to manage the workload of the various machines of the cluster. Services are created within the DBCA, the Server Control utility, or the `dbms_services` PL/SQL stored procedure. You can administer services via Oracle Enterprise Manager (OEM) or the Server Control utility.

Oracle Database 10g provides statistics related to CRS within the Automatic Workload Repository (AWR). This allows you to manage cluster-related performance issues. Additionally, AWR provides the ability to monitor any specific performance thresholds that might be exceeded and send the DBA notification alerts based on these thresholds. Also, Oracle Database 10g provide an interface into CRS through the `dbms_service` and `dbms_monitor` PL/SQL procedures.

Database Assistants

Oracle offers new functionality in two of the database assistant GUI programs, the Oracle Database Configuration Assistant (DBCA) and the Oracle Database Upgrade Assistant (DBUA). These assistants are built to make the DBA's job easier, by simplifying database configuration and migration to Oracle Database 10g. The DBCA can also be used to add and remove instances from the cluster.

The DBCA now supports the creation and removal of clustered databases. The DBCA will detect a clustered environment when it is executed and give you the opportunity to create a clustered database if that is your desire. The DBUA allows you to upgrade your previous RAC database to Oracle Database 10g with RAC.



NOTE

Oracle recommends that you create your clustered database with the DBCA in the Oracle Real Application Clusters Deployment and Performance Guide.

The `gcs_server_processes` Parameter

A new parameter, `gcs_server_processes`, is introduced in Oracle Database 10g. This parameter defines the initial number of server processes in Global Cache Service that are started to serve inter-instance traffic. The default value for this parameter is 2, and the value can range from 1 to 20. This parameter can also be set differently for each instance in the cluster.