ORACLE UNIVERSITY

Oracle Database 12*c*: Administration Workshop

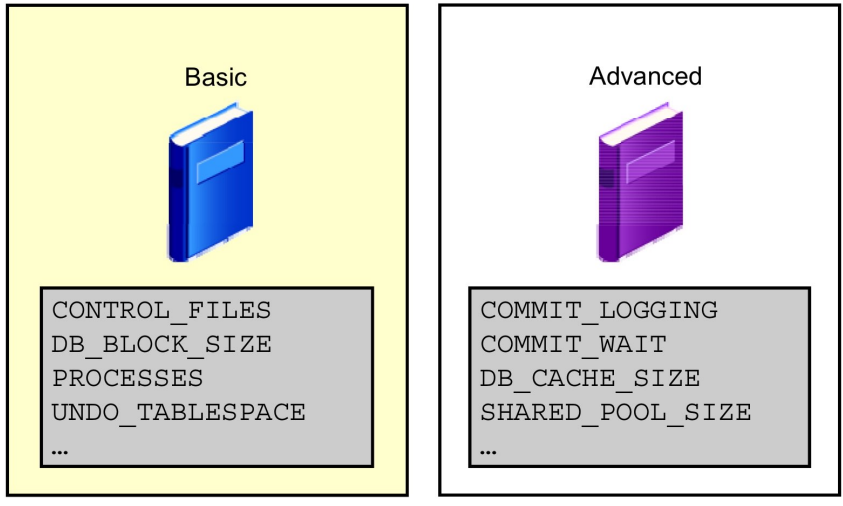
Student Guide – Volume I
D78846GC20
Edition 2.0 | December 2014 | D89298

Learn more from Oracle University at **oracle.com/education/**

4

Managing the Database Instance

Types of Initialization Parameters



Initialization parameters are of two types: basic and advanced.

In the majority of cases, it is necessary to set and tune only the 30 or so basic parameters to get reasonable performance from the database. In rare situations, modification of the advanced parameters may be needed to achieve optimal performance. There are more than 300 advanced parameters.

A basic parameter is defined as one that you are likely to set to keep your database running with good performance. All other parameters are considered to be advanced.

Examples of basic parameters:

- Determining the global database name: `DB_NAME` and `DB_DOMAIN`
- Specifying a fast recovery area and size: `DB_RECOVERY_FILE_DEST` and `DB_RECOVERY_FILE_DEST_SIZE`
- Specifying the total size of all SGA components: `SGA_TARGET`
- Specifying the method of undo space management tablespace: `UNDO_TABLESPACE`
- `COMPATIBLE` initialization parameter and irreversible compatibility

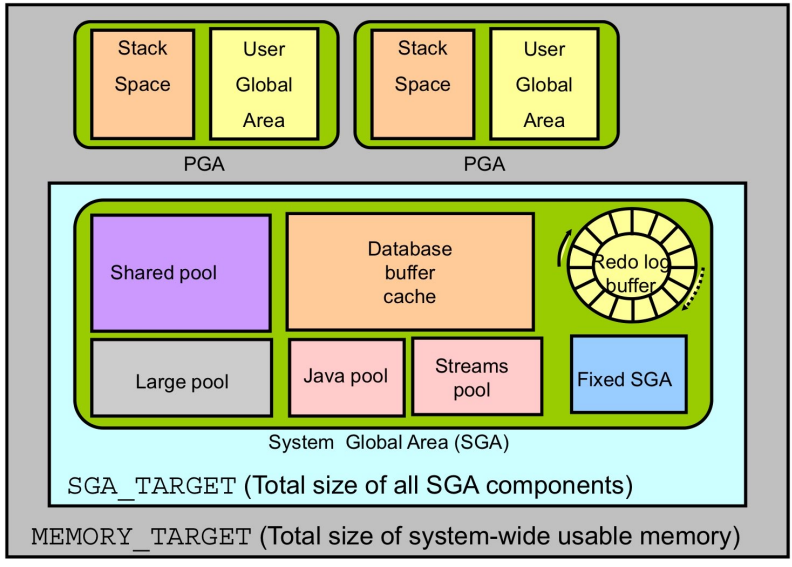
Note: Some of the initialization parameters are listed on the following pages. For a complete list, see the *Oracle Database Reference*.

Initialization Parameters: Examples

Parameter	Specifies
CONTROL_FILES	One or more control file names
DB_FILES	Maximum number of database files
PROCESSES	Maximum number of OS user processes that can simultaneously connect
DB_BLOCK_SIZE	Standard database block size used by all tablespaces
DB_CACHE_SIZE	Size of the standard block buffer cache

- **CONTROL_FILES parameter:** Specifies one or more control file names. Oracle strongly recommends that you multiplex and mirror control files. Range of values: from one to eight file names (with path names). Default value: OS dependent.
- **DB_FILES parameter:** Specifies the maximum number of database files that can be opened for this database. Range of values: OS dependent. Default value: 200.
- **PROCESSES parameter:** Specifies the maximum number of OS user processes that can simultaneously connect to an Oracle server. This value should allow for all background processes and user processes. Range of values: from 6 to an OS-dependent value. Default value: Dynamic and dependent on the number of CPUs.
- **DB_BLOCK_SIZE parameter:** Specifies the size (in bytes) of an Oracle database block. This value is set at database creation and cannot be subsequently changed. This specifies the standard block size for the database. All tablespaces will use this size by default. Range of values: 2048 to 32768 (OS-dependent). Default value: 8192.
- **DB_CACHE_SIZE parameter:** Specifies the size of the default buffer pool. Range of values: At least 4 MB times the number of CPUs (smaller values are automatically rounded up to this value). Default value: 0 if SGA_TARGET is set, otherwise the larger of 48 MB or (4 MB*CPU_COUNT).

Initialization Parameters: Examples



`SGA_TARGET` specifies the total size of all SGA components. If `SGA_TARGET` is specified, the following memory pools are automatically sized:

- Buffer cache (`DB_CACHE_SIZE`)
- Shared pool (`SHARED_POOL_SIZE`)
- Large pool (`LARGE_POOL_SIZE`)
- Java pool (`JAVA_POOL_SIZE`)
- Streams pool (`STREAMS_POOL_SIZE`)

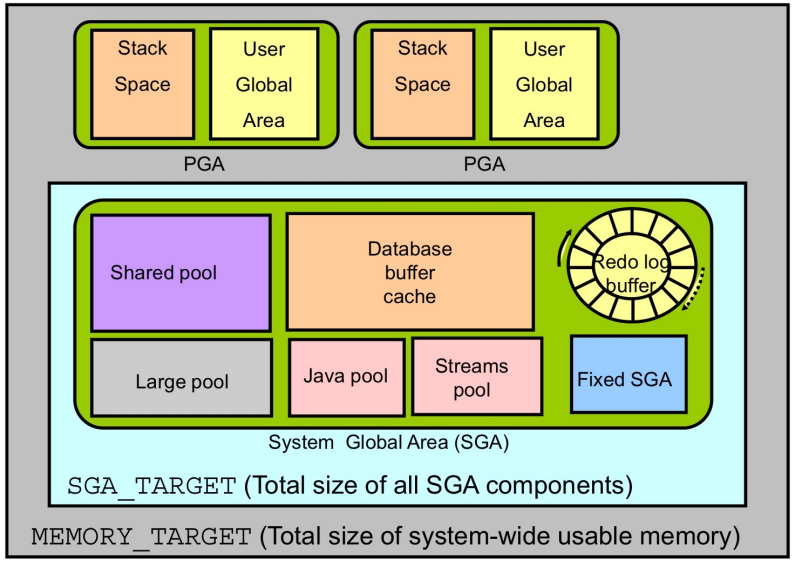
If these automatically tuned memory pools are set to nonzero values, the values are used as minimum levels by Automatic Shared Memory Management (ASMM). You set minimum values if an application component needs a minimum amount of memory to function properly.

The following pools are manually sized components and are not affected by ASMM:

- Log buffer
- Other buffer caches (such as `KEEP` and `RECYCLE`) and other block sizes
- Fixed SGA and other internal allocations

The memory allocated to these pools is deducted from the total available memory for `SGA_TARGET` when ASMM is enabled.

Initialization Parameters: Examples



`SGA_TARGET` specifies the total size of all SGA components. If `SGA_TARGET` is specified, the following memory pools are automatically sized:

- Buffer cache (`DB_CACHE_SIZE`)
- Shared pool (`SHARED_POOL_SIZE`)
- Large pool (`LARGE_POOL_SIZE`)
- Java pool (`JAVA_POOL_SIZE`)
- Streams pool (`STREAMS_POOL_SIZE`)

If these automatically tuned memory pools are set to nonzero values, the values are used as minimum levels by Automatic Shared Memory Management (ASMM). You set minimum values if an application component needs a minimum amount of memory to function properly.

The following pools are manually sized components and are not affected by ASMM:

- Log buffer
- Other buffer caches (such as `KEEP` and `RECYCLE`) and other block sizes
- Fixed SGA and other internal allocations

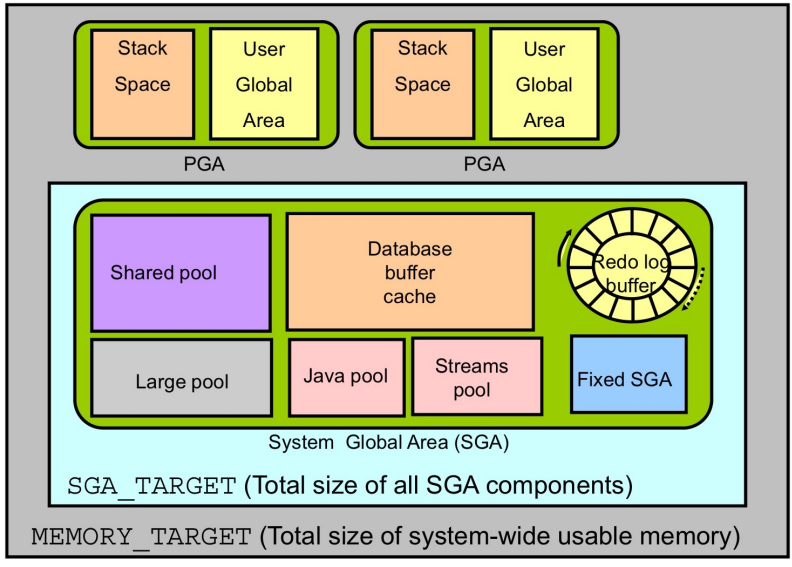
The memory allocated to these pools is deducted from the total available memory for `SGA_TARGET` when ASMM is enabled.

Initialization Parameters: Examples

Parameter	Specifies
PGA_AGGREGATE_TARGET	Amount of PGA memory available to all server processes
SHARED_POOL_SIZE	Size of shared pool (in bytes)
UNDO_MANAGEMENT	Undo space management mode to be used

- **PGA_AGGREGATE_TARGET parameter:** Specifies the amount of Program Global Area (PGA) memory available to all server processes attached to the instance. This memory does not reside in the System Global Area (SGA). The database uses this parameter as a target amount of PGA memory to use. When setting this parameter, subtract the SGA from the total memory on the system that is available to the Oracle instance. The minimum value is 10 MB and the maximum value is (4096 GB – 1). The default is 10 MB or 20% of the size of the SGA, whichever is greater.
- **SHARED_POOL_SIZE parameter:** Specifies the size of the shared pool in bytes. The shared pool contains objects such as shared cursors, stored procedures, control structures, and parallel execution message buffers. Range of values: OS-dependent. Default value: 0 if SGA_TARGET is set, otherwise 128 MB if 64-bit; 48 MB if 32-bit.
- **UNDO_MANAGEMENT parameter:** Specifies the undo space management mode that the system should use. When set to AUTO, the instance is started in automatic undo management mode. Otherwise, it is started in rollback undo mode. In rollback undo mode, undo space is allocated as rollback segments. In automatic undo mode, undo space is allocated as undo tablespaces. Range of values: AUTO or MANUAL. If the UNDO_MANAGEMENT parameter is omitted when the instance is started, the default value AUTO is used.

Initialization Parameters: Examples



`SGA_TARGET` specifies the total size of all SGA components. If `SGA_TARGET` is specified, the following memory pools are automatically sized:

- Buffer cache (`DB_CACHE_SIZE`)
- Shared pool (`SHARED_POOL_SIZE`)
- Large pool (`LARGE_POOL_SIZE`)
- Java pool (`JAVA_POOL_SIZE`)
- Streams pool (`STREAMS_POOL_SIZE`)

If these automatically tuned memory pools are set to nonzero values, the values are used as minimum levels by Automatic Shared Memory Management (ASMM). You set minimum values if an application component needs a minimum amount of memory to function properly.

The following pools are manually sized components and are not affected by ASMM:

- Log buffer
- Other buffer caches (such as `KEEP` and `RECYCLE`) and other block sizes
- Fixed SGA and other internal allocations

The memory allocated to these pools is deducted from the total available memory for `SGA_TARGET` when ASMM is enabled.

Initialization Parameters: Examples

Parameter	Specifies
PGA_AGGREGATE_TARGET	Amount of PGA memory available to all server processes
SHARED_POOL_SIZE	Size of shared pool (in bytes)
UNDO_MANAGEMENT	Undo space management mode to be used

- **PGA_AGGREGATE_TARGET parameter:** Specifies the amount of Program Global Area (PGA) memory available to all server processes attached to the instance. This memory does not reside in the System Global Area (SGA). The database uses this parameter as a target amount of PGA memory to use. When setting this parameter, subtract the SGA from the total memory on the system that is available to the Oracle instance. The minimum value is 10 MB and the maximum value is (4096 GB – 1). The default is 10 MB or 20% of the size of the SGA, whichever is greater.
- **SHARED_POOL_SIZE parameter:** Specifies the size of the shared pool in bytes. The shared pool contains objects such as shared cursors, stored procedures, control structures, and parallel execution message buffers. Range of values: OS-dependent. Default value: 0 if SGA_TARGET is set, otherwise 128 MB if 64-bit; 48 MB if 32-bit.
- **UNDO_MANAGEMENT parameter:** Specifies the undo space management mode that the system should use. When set to AUTO, the instance is started in automatic undo management mode. Otherwise, it is started in rollback undo mode. In rollback undo mode, undo space is allocated as rollback segments. In automatic undo mode, undo space is allocated as undo tablespaces. Range of values: AUTO or MANUAL. If the UNDO_MANAGEMENT parameter is omitted when the instance is started, the default value AUTO is used.

Using SQL*Plus to View Parameters

```
SQL> SELECT name, value FROM v$parameter;
NAME                 VALUE
------------         ----------
lock_name_space
processes             300
sessions              472
timed_statistics      TRUE
timed_os_statistics   0
…

SQL> SHOW PARAMETER SHARED_POOL_SIZE
NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------------------
shared_pool_size                     big integer 0

SQL> show parameter para
NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------------------
cell_offload_parameters              string
fast_start_parallel_rollback         string      LOW
parallel_adaptive_multi_user         boolean     TRUE
parallel_automatic_tuning            boolean     FALSE
…
```


The slide shows examples of using SQL*Plus to view parameters. You can query the `V$PARAMETER` view to find the values of the various parameters. `V$PARAMETER` displays the current parameter values in the current session. You can also use the `SHOW PARAMETER` command with any string to view parameters that contain that string.

The query in the following example is requesting the name and values of the parameters. Use a `WHERE` clause to specify specific parameter names:

```
SQL> SELECT name, value FROM v$parameter
  2  WHERE name LIKE '%pool%';
NAME                          VALUE
--------------------------    ----------
shared_pool_size              0
large_pool_size               0
java_pool_size                0
streams_pool_size             0
shared_pool_reserved_size     15728640
…
9 rows selected.
```

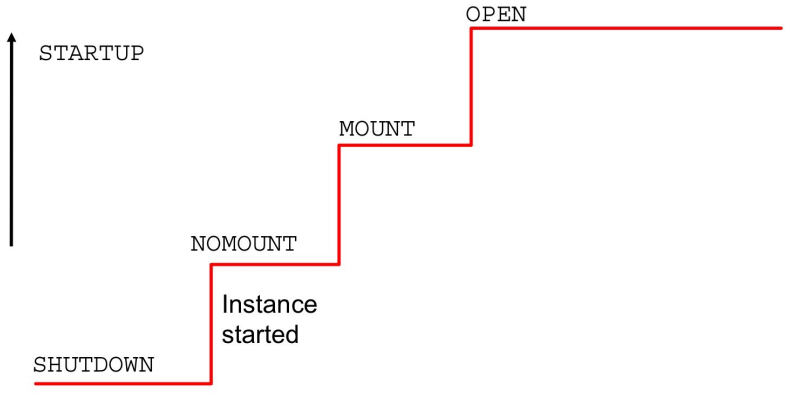
Description of the view:

```
SQL> desc V$parameter
 Name                                      Null?  Type
 ----------------------------------------- -------- -------------
 NUM                                               NUMBER
 NAME                                              VARCHAR2(80)
 TYPE                                              NUMBER
 VALUE                                             VARCHAR2(4000)
 DISPLAY_VALUE                                     VARCHAR2(4000)
 DEFAULT_VALUE                                     VARCHAR2(255)
 ISDEFAULT                                         VARCHAR2(9)
 ISSES_MODIFIABLE                                  VARCHAR2(5)
 ISSYS_MODIFIABLE                                  VARCHAR2(9)
 ISPDB_MODIFIABLE                                  VARCHAR2(5)
 ISINSTANCE_MODIFIABLE                             VARCHAR2(5)
 ISMODIFIED                                        VARCHAR2(10)
 ISADJUSTED                                        VARCHAR2(5)
 ISDEPRECATED                                      VARCHAR2(5)
 ISBASIC                                           VARCHAR2(5)
 DESCRIPTION                                       VARCHAR2(255)
 UPDATE_COMMENT                                    VARCHAR2(255)
 HASH                                              NUMBER
 CON_ID                                            NUMBER
```

The second example shows the use of the SQL*Plus `SHOW PARAMETER` command to view parameter settings. You can also use this command to find all parameters that contain a text string. For example, you can find all parameter names that include the `db` string by using the following command:

```
SQL> show parameter db
NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------
…
db_8k_cache_size                     big integer 0
db_big_table_cache_percent_target    string      0
db_block_buffers                     integer     0
db_block_checking                    string      FALSE
db_block_checksum                    string      TYPICAL
…
```

Starting Up an Oracle Database Instance: NOMOUNT



The database instance and database go through stages as the database is made available for access by users. The database instance is started, the database is mounted, and then the database is opened.

An instance is typically started only in `NOMOUNT` mode during database creation, during re-creation of control files, or in certain backup and recovery scenarios.

When an instance is started, the following takes place:

- Searching `$ORACLE_HOME/dbs` for a file of a particular name in this sequence:
  1. Search for `spfile<SID>.ora`.
  2. If `spfile<SID>.ora` is not found, search for `spfile.ora`.
  3. If `spfile.ora` is not found, search for `init<SID>.ora`.

     This is the file that contains initialization parameters for the instance. Specifying the `PFILE` parameter with `STARTUP` overrides the default behavior.
- Allocating the SGA
- Starting the background processes
- Opening the `alert_<SID>.log` file and the trace files

Note: `SID` is the system ID, which identifies the instance name (for example, `ORCL`).