Initialization Parameters

This chapter contains detailed descriptions (in alphabetical order) of the database initialization parameters.

This chapter contains the following topics:

- Uses of Initialization Parameters
- Basic Initialization Parameters
- Parameter Files
- Changing Parameter Values in a Parameter File
- Reading the Parameter Descriptions
- Initialization Parameter Descriptions

2.1 Uses of Initialization Parameters

Initialization parameters are used to perform various functions.

For example, initialization parameters are used to:

- Set limits for the entire database
- · Set user or process limits
- Set limits on database resources
- Affect performance (these are called variable parameters)

Variable parameters are of particular interest to database administrators, because these parameters are used primarily to improve database performance.

Database administrators can use initialization parameters to:

- Optimize performance by adjusting memory structures, such as the number of database buffers in memory
- Set database-wide defaults, such as the amount of space initially allocated for a context area when it is created
- Set database limits, such as the maximum number of database users
- Specify names of files or directories required by the database

Many initialization parameters can be fine-tuned to improve database performance. Other parameters should never be altered or should be altered only under the supervision of Oracle Support Services.

All initialization parameters are optional. Oracle has a default value for each parameter. This value may be operating system-dependent, depending on the parameter.

2.1.1 Types of Initialization Parameters

The Oracle database server has derived parameters, operating system-dependent parameters, and variable parameters.

- Derived Parameters
- Operating System-Dependent Parameters
- Variable Parameters (these can be dynamic parameters or any of the preceding ones)

2.1.1.1 Derived Parameters

Some initialization parameters are **derived**, meaning that their values are calculated from the values of other parameters. Normally, you should not alter values for derived parameters, but if you do, then the value you specify will override the calculated value.

For example, the default value of the SESSIONS parameter is derived from the value of the PROCESSES parameter. If the value of PROCESSES changes, then the default value of SESSIONS changes as well, unless you override it with a specified value.

2.1.1.2 Operating System-Dependent Parameters

The valid values or value ranges of some initialization parameters depend upon the host operating system. For example, the parameter <code>DB_BLOCK_BUFFERS</code> indicates the number of data buffers in main memory, and its maximum value depends on the operating system. The size of those buffers, set by <code>DB_BLOCK_SIZE</code>, has an operating system-dependent default value.



Your operating system-specific Oracle documentation for more information on operating system-dependent Oracle parameters and operating system parameters

2.1.1.3 Variable Parameters

The variable initialization parameters offer the most potential for improving system performance. Some variable parameters set capacity limits but do not affect performance. For example, when the value of <code>OPEN_CURSORS</code> is 10, a user process attempting to open its eleventh cursor receives an error. Other variable parameters affect performance but do not impose absolute limits. For example, reducing the value of <code>DB_BLOCK_BUFFERS</code> does not prevent work even though it may slow down performance.

Increasing the values of variable parameters may improve your system's performance, but increasing most parameters also increases the system global area (SGA) size. A larger SGA can improve database performance up to a point. In virtual memory operating systems, an SGA that is too large can degrade performance if it is swapped in and out of memory. Operating system parameters that control virtual memory working areas should be set with the SGA size in mind. The operating system configuration can also limit the maximum size of the SGA.



2.2 Basic Initialization Parameters

Most databases should only need to have the database basic initialization parameters set to run properly and efficiently.

Oracle advises you to become familiar with the basic parameters and only use other parameters when directed to by feature documentation or in special circumstances. The following is a list of the database basic initialization parameters:

```
CLUSTER DATABASE
COMPATIBLE
CONTROL FILES
DB BLOCK SIZE
DB CREATE FILE DEST
DB CREATE ONLINE LOG DEST n
DB DOMAIN
DB NAME
DB_RECOVERY_FILE_DEST
DB RECOVERY FILE DEST SIZE
DB UNIQUE NAME
INSTANCE NUMBER
LDAP DIRECTORY SYSAUTH
LOG ARCHIVE DEST n
LOG ARCHIVE DEST STATE n
NLS DATE LANGUAGE
NLS TERRITORY
OPEN_CURSORS
PGA AGGREGATE TARGET
PROCESSES
REMOTE LISTENER
REMOTE LOGIN PASSWORDFILE
SESSIONS
SGA TARGET
SHARED SERVERS
STAR_TRANSFORMATION_ENABLED
UNDO TABLESPACE
```

2.3 Parameter Files

A **parameter file** is a file that contains a list of initialization parameters and a value for each parameter. You specify initialization parameters in a parameter file that reflect your particular installation.

Oracle supports the following two types of parameter files:

- Server Parameter Files
- Initialization Parameter Files

2.3.1 Server Parameter Files

A server parameter file is a binary file that acts as a repository for initialization parameters.

The server parameter file can reside on the computer where the Oracle database server executes. Initialization parameters stored in a server parameter file are persistent, in that any changes made to the parameters while an instance is running can persist across instance shutdown and startup.



Oracle Database Administrator's Guide for an introduction to and detailed information about managing and using server parameter files and initialization parameter files

2.3.2 Initialization Parameter Files

An **initialization parameter file** is a text file that contains a list of initialization parameters.



See "About the Character Set of Parameter Values" for more information about the character set to use when specifying values in an initialization parameter file.

The following are sample entries in an initialization parameter file:

```
PROCESSES = 100
OPEN_LINKS = 12
GLOBAL NAMES = true
```

The name of the initialization parameter file varies depending on the operating system. For example, it can be in mixed case or lowercase, or it can have a logical name or a variation of the name init.ora. Also supplied is an initdw.ora file, which contains suggested parameter settings for data warehouses and data marts. The database administrator can choose a different file name for the initialization parameter file.

Refer to your operating system-specific Oracle documentation for the default locations and filenames for initialization parameter files on your operating system. The initialization parameter file is read by the client-side tool used to start the server (such as SQL*Plus).

Sample initialization parameter files are provided on the Oracle distribution medium for each operating system. A sample file is sufficient for initial use, but you will probably want to modify the file to tune the database for best performance. Any changes will take effect after you completely shut down and restart the instance.

2.3.2.1 About the Character Set of Parameter Values

Only characters from the default character set of the database platform are supported in parameter values. For all platforms except IBM z/OS and Fujitsu BS2000 the default platform character set is US7ASCII (7-bit ASCII).

The database behavior is undefined when parameter values with unsupported characters are specified. When using the ALTER SYSTEM statement to set a parameter value in a server parameter file, make sure only supported characters are used. The database does not always report an error when a value with unsupported characters is specified.



The initialization parameter file is a client-side file. When used, it is located on the host on which you start SQL*Plus to start up a database instance. The initialization parameter file must be written in the client character set as specified by the NLS_LANG client setting. However, parameter values should not contain characters not supported by the default character set of the database.

2.3.2.2 Specifying Values in an Initialization Parameter File

This section describes several aspects of setting parameter values in an initialization parameter file.

2.3.2.2.1 Rules Governing Initialization Parameter Files

This section describes the rules that govern the specification of parameters in initialization parameter files.

- An initialization parameter file should contain only parameters and comments. A pound sign (#) starts a comment line. The rest of the line is ignored.
- Comments must use the same character set as parameter values.
- You can specify parameters in any order.
- Case (upper or lower) in filenames is significant only if case is significant on the host operating system.
- To enter several parameters on one line, use spaces between parameter names and values, as in the following example:

```
PROCESSES = 100 CPU COUNT = 1 OPEN CURSORS = 10
```

- Some parameters, such as ROLLBACK_SEGMENTS, accept multiple values. Any of the following represent valid syntax.
 - Enter multiple values enclosed in parentheses and separated by commas. For example:

```
ROLLBACK SEGMENTS = (SEG1, SEG2, SEG3, SEG4, SEG5)
```

Enter multiple values without parentheses and commas. For example:

```
ROLLBACK SEGMENTS = SEG1 SEG2 SEG3 SEG4 SEG5
```

Enter multiple values, one per line. For example:

```
ROLLBACK_SEGMENTS = SEG1
ROLLBACK_SEGMENTS = SEG2
ROLLBACK_SEGMENTS = SEG3
ROLLBACK_SEGMENTS = SEG4
ROLLBACK_SEGMENTS = SEG5
```

If you enter values for one parameter on multiple lines, then the entries must be on consecutive lines. If they are not, then the first entry will not be processed properly. For example, in the following entry the setting for SEG3 and SEG4 will override the setting for SEG1 and SEG2:

```
ROLLBACK_SEGMENTS = SEG1 SEG2
OPEN_CURSORS = 10
ROLLBACK SEGMENTS = SEG3 SEG4
```

 A backslash (\), also known as an escape character, indicates continuation of the parameter specification. If a backslash continues a line, then the continued line must have no leading spaces. For example:



```
ROLLBACK_SEGMENTS = (SEG1, SEG2, \
SEG3, SEG4, SEG5)
```

- You can use the IFILE initialization parameter to embed the contents of another initialization parameter file into the current initialization parameter file.
- Enclose in quotation marks any parameter values that contain spaces or tabs. You can use either single or double quotation marks unless otherwise indicated. For example:

```
NLS TERRITORY = 'CZECH REPUBLIC'
```



Listing parameters in alphabetical order in the initialization parameter file can help you to find them and can help ensure that each parameter is specified only once.

Enclose in quotation marks any parameter value that contains a special character.

See Also:

- Your operating system-specific Oracle documentation for more information on initialization parameter files
- "IFILE"

2.3.2.2.2 Using Expressions for Parameter Values

You can specify an expression as the value for an initialization parameter.

The optimal value for an initialization parameter can depend on environmental characteristics, such as system configurations, run-time decisions, and the values of other parameters. In previous releases, you were required to specify an absolute value when setting an initialization parameter, which meant that maintaining its optimal value required manual oversight and adjustments. Starting with Oracle Database 21c, you can specify an expression as the value for some initialization parameters, which enables the database to automatically adjust the parameter value in response to environmental changes.

Specifying an expression for a parameter value is also useful in Oracle Real Application Cluster (Oracle RAC) environments. You can specify the same expression for a parameter value across all Oracle RAC instances, but the expression will be evaluated on a per-instance basis, utilizing the environmental characteristics of the current instance. This allows the database to fine-tune the parameter value for each Oracle RAC instance.

You can specify expressions only for initialization parameters of type integer or big integer.

Initialization parameter expressions can contain the following constructs:

Integer values

```
OLAP PAGE POOL SIZE = 1073741824
```



Decimal values, but only when they are part of a numeric operation

```
CPU COUNT = 8 * 0.6
```

If the operation evaluates to a decimal value, then the result is truncated to an integer value. The above example would set CPU COUNT to 4.

3. An integer value followed by an indicator that the integer is expressed in kilobytes (k or K), megabytes (m or M), gigabytes (g or G), terabytes (t or T), petabytes (p or P), or exabytes (e or E)

```
OLAP_PAGE_POOL_SIZE = 900m
MEMORY TARGET = 2G
```

4. Other parameter names

```
JOB QUEUE PROCESSES = PROCESSES
```

5. The binary operators for multiplication (*), division (/), modulo (%), addition (+), and subtraction (-)

```
SHARED_SERVERS = MAX_SHARED_SERVERS / 2
DATA GUARD MAX LONGIO TIME = DATA GUARD MAX IO TIME + 10 * 3
```

Operator precedence: the multiplication, division, and modulo operators are evaluated first, from left to right, and then the addition and subtraction operators are evaluated from left to right.

If the expression evaluates to a decimal value, then the result is truncated to an integer value.

6. Parentheses, which allow you to override operator precedence

```
SHARED_SERVERS = (MAX_SHARED_SERVERS - 1) / 2
DATA_GUARD_MAX_LONGIO_TIME = (DATA_GUARD_MAX_IO_TIME + 10) * 3
```

7. The MIN and MAX functions. These functions take two operands and return the minimum and maximum value, respectively.

```
AQ_TM_PROCESSES = MIN(40, PROCESSES * .1)
SESSIONS = MAX(200, PROCESSES * 1.5)
```

8. Environment variable values, which must be preceded by the dollar sign (\$)

```
CPU COUNT = $SYSTEM CPU/5
```

You can specify an expression when using any of the following methods to set an initialization parameter:

- In a server parameter file (SPFILE)
- In a text initialization parameter file, for example, initORACLE SID.ora.

Note that you cannot use expressions in a client-side text initialization parameter file (PFILE).



• In the SQL statements ALTER SESSION SET and ALTER SYSTEM SET. In this case you must enclose the expression in single quotation marks. For example:

```
ALTER SESSION SET AQ_TM_PROCESSES = 'MIN(40, PROCESSES * .1)';
ALTER SYSTEM SET JOB QUEUE PROCESSES = 'PROCESSES' SCOPE=BOTH;
```

2.3.2.2.3 Using Special Characters in Parameter Values

If a parameter value contains a special character, then the special character must be preceded by a backslash or the entire parameter value must be enclosed in quotation marks.

For example, you can specify special characters using either of the following:

```
DB_DOMAIN = 'JAPAN.ACME#.COM'

DB_DOMAIN = JAPAN.ACME\#.COM
```

Table 2-1 lists the special characters you can use in initialization parameter files.

Table 2-1 Special Characters in Initialization Parameter Files

Character	Name	Description
#	Number sign	Comment
(Left parenthesis	Start list of values
)	Right parenthesis	End list of values
II .	Double quotation mark	Start or end of quoted string
T	Single quotation mark	Start or end of quoted string
=	Equal sign	Separator of keyword and values
,	Comma	Separator of elements
-	Minus sign	Precedes UNIX-style keywords
\	Backslash	Escape character

If a special character must be treated literally in an initialization parameter file, then it must either be preceded by the backslash character, or the entire string containing the special character must be enclosed in quotation marks.

2.3.2.2.4 Using the Escape Character

In an initialization parameter file, a backslash (\) can also signify a line continuation. If the backslash is followed by an alphanumeric character, then the backslash is treated as a normal character in the input.

If it is not followed by an alphanumeric character, then the backslash is treated either as a backslash or as a continuation character.



"Rules Governing Initialization Parameter Files"



2.3.2.2.5 Using Quotation Marks

Quotation marks can be nested in two ways in an initialization parameter file.

The first method is to double the quotation marks in the nested string. For example:

```
NLS DATE FORMAT = '''Today is'' MM/DD/YYYY'
```

The second method is to alternate single and double quotation marks. For example:

```
NLS DATE FORMAT = '"Today is" MM/DD/YYYY'
```

2.4 Changing Parameter Values in a Parameter File

You can change the value of a parameter in a parameter file in several ways.

- By editing an initialization parameter file
 In most cases, the new value takes effect the next time you start an instance of the database.
- By issuing an ALTER SYSTEM SET ... SCOPE=SPFILE statement to update a server parameter file
- By issuing an ALTER SYSTEM RESET statement to clear an initialization parameter value.



Oracle Database Administrator's Guide for more information about using the ALTER SYSTEM RESET statement to clear initialization parameter values

2.4.1 Parameters by Functional Category

This section lists initialization parameters by their functional category.

ANSI Compliance

BLANK_TRIMMING

Backup and Restore

BACKUP_TAPE_IO_SLAVES
DATA_TRANSFER_CACHE_SIZE
CLONEDB
CLONEDB_DIR
RECYCLEBIN
TAPE_ASYNCH_IO

BFILEs

SESSION_MAX_OPEN_FILES

Buffer Cache and I/O

```
CLIENT_PREFETCH_ROWS
CLIENT_RESULT_CACHE_LAG
CLIENT_RESULT_CACHE_SIZE
```

```
DB nK CACHE SIZE
DB_BIG_TABLE_CACHE_PERCENT_TARGET
DB BLOCK BUFFERS
DB_BLOCK_SIZE
DB CACHE ADVICE
DB_CACHE_SIZE
DB_FILE_MULTIBLOCK_READ_COUNT
DB_KEEP_CACHE_SIZE
DB_RECYCLE_CACHE_SIZE
DB WRITER PROCESSES
DBWR IO SLAVES
DISK ASYNCH IO
DNFS_BATCH_SIZE
FILESYSTEMIO OPTIONS
IORM_LIMIT_POLICY
READ ONLY
READ_ONLY_OPEN_DELAYED
RESULT_CACHE_AUTO_BLOCKLIST
RESULT_CACHE_EXECUTION_THRESHOLD
RESULT_CACHE_INTEGRITY
RESULT CACHE MAX RESULT
RESULT_CACHE_MAX_SIZE
RESULT_CACHE_MAX_TEMP_RESULT
RESULT_CACHE_MAX_TEMP_SIZE
RESULT CACHE MODE
```

Cursors and Library Cache

CURSOR_BIND_CAPTURE_DESTINATION
CURSOR_INVALIDATION
CURSOR_SHARING
CURSOR_SPACE_FOR_TIME
OPEN_CURSORS
SESSION_CACHED_CURSORS

Database/Instance Identification

DB_DOMAIN
DB_NAME
INSTANCE_NAME

Diagnostics and Statistics

BACKGROUND_CORE_DUMP
BACKGROUND_DUMP_DEST
CLIENT_STATISTICS_LEVEL
CORE_DUMP_DEST
DB_BLOCK_CHECKING
DB_BLOCK_CHECKSUM
DIAGNOSTIC_DEST
DIAGNOSTICS_CONTROL
ERROR_MESSAGE_DETAILS
EVENT
HEAT_MAP

MAX_DUMP_FILE_SIZE
SHADOW_CORE_DUMP
SQL_ERROR_MITIGATION
SQL_HISTORY_ENABLED
STATISTICS_LEVEL
TIMED_OS_STATISTICS
TIMED_STATISTICS
TRACE_ENABLED
TRACEFILE_IDENTIFIER
USER_DUMP_DEST

Distributed, Replication

COMMIT_POINT_STRENGTH
DISTRIBUTED_LOCK_TIMEOUT
ENABLE_GOLDENGATE_REPLICATION
GLOBAL_NAMES
HS_AUTOREGISTER
OPEN_LINKS
OPEN_LINKS_PER_INSTANCE
REPLICATION_DEPENDENCY_TRACKING

File Locations, Names, and Sizes

AUDIT_FILE_DEST
BACKGROUND_CORE_DUMP
BACKGROUND_DUMP_DEST
CONTROL_FILES
CORE_DUMP_DEST
DB_CREATE_FILE_DEST
DB_CREATE_ONLINE_LOG_DEST_n
DB_FILES
DB_RECOVERY_FILE_DEST
DB_RECOVERY_FILE_DEST_SIZE
FILE_MAPPING
IFILE
LOG_ARCHIVE_DEST_n
SPFILE

Globalization

NLS_CALENDAR
NLS_COMP
NLS_CURRENCY
NLS_DATE_FORMAT
NLS_DATE_LANGUAGE
NLS_DUAL_CURRENCY
NLS_ISO_CURRENCY
NLS_LANGUAGE
NLS_LENGTH_SEMANTICS
NLS_NCHAR_CONV_EXCP
NLS_NUMERIC_CHARACTERS
NLS_SORT
NLS_TERRITORY

```
NLS_TIMESTAMP_FORMAT
NLS_TIMESTAMP_TZ_FORMAT
TIMEZONE VERSION UPGRADE ONLINE
```

Java

JAVA_JIT_ENABLED
JAVA_MAX_SESSIONSPACE_SIZE
JAVA_POOL_SIZE
JAVA_SOFT_SESSIONSPACE_LIMIT

Job Queues

JOB_QUEUE_PROCESSES

License Limits

LICENSE_MAX_SESSIONS
LICENSE_MAX_USERS
LICENSE SESSIONS WARNING

Memory

INMEMORY_ADG_ENABLED INMEMORY AUTOMATIC LEVEL INMEMORY_CLAUSE_DEFAULT INMEMORY_DEEP_VECTORIZATION INMEMORY EXPRESSIONS USAGE INMEMORY_FORCE INMEMORY MAX POPULATE SERVERS INMEMORY OPTIMIZED ARITHMETIC INMEMORY OPTIMIZED DATE INMEMORY_QUERY INMEMORY_SIZE INMEMORY_TRICKLE_REPOPULATE_SERVERS_PERCENT INMEMORY_VIRTUAL_COLUMNS MEMOPTIMIZE_POOL_SIZE MEMOPTIMIZE WRITE AREA SIZE MEMOPTIMIZE WRITES MEMORY_MAX_SIZE MEMORY SIZE MEMORY_MAX_TARGET **MEMORY TARGET** PMEM_FILESTORE

Miscellaneous

ALLOW_ROWID_COLUMN_TYPE

AQ_TM_PROCESSES

ASM_IO_PROCESSES

ASM_PREFERRED_READ_FAILURE_GROUPS

AWR_SNAPSHOT_TIME_OFFSET

COMPATIBLE

DB_INDEX_COMPRESSION_INHERITANCE

ENABLE_DNFS_DISPATCHER

FIXED_DATE

GROUP_BY_POSITION_ENABLED

IGNORE SESSION SET PARAM ERRORS INSTANCE ABORT DELAY TIME JSON BEHAVIOR JSON EXPRESSION CHECK LDAP_DIRECTORY_SYSAUTH MAX COLUMNS MAX_DATAPUMP_JOBS_PER_PDB MAX_DATAPUMP_PARALLEL_PER_JOB MAX_STRING_SIZE MLE PROG LANGUAGES PRIVATE_TEMP_TABLE_PREFIX SPATIAL VECTOR ACCELERATION TIME_AT_DBTIMEZONE

UNIFORM_LOG_TIMESTAMP_FORMAT

XML_CLIENT_SIDE_DECODING

XML DB EVENTS

XML PARAMS

Multitenant Architecture

AUTOTASK MAX ACTIVE PDBS AWR_PDB_AUTOFLUSH_ENABLED

COMMON_USER_PREFIX

CONTAINER DATA

CONTAINERS_PARALLEL_DEGREE

CPU MIN COUNT

DB PERFORMANCE PROFILE

DBNEST ENABLE

DBNEST_PDB_FS_CONF

DEFAULT SHARING

ENABLE_AUTOMATIC_MAINTENANCE_PDB

ENABLED_PDBS_ON_STANDBY

EXTERNAL_KEYSTORE_CREDENTIAL_LOCATION

HYBRID_READ_ONLY

MAX IOPS

MAX_MBPS

MAX PDBS

NONCDB_COMPATIBLE

PDB_FILE_NAME_CONVERT

PDB_LOCKDOWN

PDB OS CREDENTIAL

SGA_MIN_SIZE

Networking

CONNECTION BROKERS DRCP_CONNECTION_LIMIT DRCP DEDICATED OPT ENABLE_PER_PDB_DRCP FORWARD LISTENER

LOCAL LISTENER

MAX AUTH SERVERS

MIN_AUTH_SERVERS

OFS_THREADS
REMOTE_LISTENER
SERVICE NAMES

Objects and LOBs

OBJECT_CACHE_MAX_SIZE_PERCENT OBJECT CACHE OPTIMAL SIZE

OLAP

OLAP_PAGE_POOL_SIZE

Optimizer

APPROX FOR AGGREGATION APPROX_FOR_COUNT_DISTINCT APPROX FOR PERCENTILE OPTIMIZER_ADAPTIVE_PLANS OPTIMIZER ADAPTIVE REPORTING ONLY OPTIMIZER_ADAPTIVE_STATISTICS OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES OPTIMIZER CAPTURE SQL QUARANTINE OPTIMIZER DYNAMIC SAMPLING OPTIMIZER FEATURES ENABLE OPTIMIZER_IGNORE_HINTS OPTIMIZER_IGNORE_PARALLEL_HINTS OPTIMIZER_INDEX_CACHING OPTIMIZER INDEX COST ADJ OPTIMIZER_INMEMORY_AWARE OPTIMIZER MODE OPTIMIZER_REAL_TIME_STATISTICS OPTIMIZER SESSION TYPE OPTIMIZER_USE_PENDING_STATISTICS OPTIMIZER_USE_SQL_PLAN_BASELINES OPTIMIZER_USE_SQL_QUARANTINE PLSQL_FUNCTION_DYNAMIC_STATS QUERY REWRITE ENABLED QUERY_REWRITE_INTEGRITY STAR TRANSFORMATION ENABLED

Oracle RAC

ACTIVE_INSTANCE_COUNT CLUSTER_DATABASE CLUSTER_INTERCONNECTS INSTANCE_NUMBER PARALLEL_INSTANCE_GROUP THREAD

Parallel Execution

PARALLEL_ADAPTIVE_MULTI_USER
PARALLEL_EXECUTION_MESSAGE_SIZE
PARALLEL_MAX_SERVERS
PARALLEL_MIN_DEGREE
PARALLEL MIN PERCENT

PARALLEL_MIN_SERVERS
PARALLEL THREADS PER CPU

PL/SQL

LONG_MODULE_ACTION
PLSQL_IMPLICIT_CONVERSION_BOOL
PLSQL_V2_COMPATIBILITY
REMOTE_DEPENDENCIES_MODE
SESSION_EXIT_ON_PACKAGE_STATE_ERROR

PL/SQL Compiler

NLS_LENGTH_SEMANTICS
PERMIT_92_WRAP_FORMAT
PLSCOPE_SETTINGS
PLSQL_CCFLAGS
PLSQL_CODE_TYPE
PLSQL_DEBUG
PLSQL_OPTIMIZE_LEVEL
PLSQL_WARNINGS
SQL_TRANSPILER

Redo Logs, Archiving, and Recovery

ADG ACCOUNT INFO TRACKING ADG REDIRECT DML CONTROL_FILE_RECORD_KEEP_TIME DATA GUARD MAX IO TIME DATA_GUARD_MAX_LONGIO_TIME DATA GUARD SYNC LATENCY DB CREATE ONLINE LOG DEST n DB_FLASHBACK_LOG_DEST DB_FLASHBACK_LOG_DEST_SIZE DB_RECOVERY_FILE_DEST DB_RECOVERY_FILE_DEST_SIZE DB_UNRECOVERABLE_SCN_TRACKING FAST START MTTR TARGET LOG_ARCHIVE_CONFIG LOG ARCHIVE DEST n LOG ARCHIVE DEST STATE n LOG ARCHIVE DUPLEX DEST LOG ARCHIVE FORMAT LOG_ARCHIVE_MAX_PROCESSES LOG_ARCHIVE_MIN_SUCCEED_DEST LOG_ARCHIVE_TRACE LOG BUFFER LOG_CHECKPOINT_INTERVAL LOG CHECKPOINT TIMEOUT LOG CHECKPOINTS TO ALERT RECOVERY PARALLELISM REDO_TRANSPORT_USER REMOTE_RECOVERY_FILE_DEST

Resource Manager

MAX_IDLE_BLOCKER_TIME
MAX_IDLE_TIME
PROCESSOR_GROUP_NAME
RESOURCE_LIMIT
RESOURCE_MANAGER_CPU_ALLOCATION
RESOURCE_MANAGER_CPU_SCOPE
RESOURCE_MANAGER_PLAN
SGA_MIN_SIZE
STANDBY_DB_PRESERVE_STATES

Security and Auditing

ALLOW_GLOBAL_DBLINKS ALLOW_GROUP_ACCESS_TO_SGA ALLOW_LEGACY_RECO_PROTOCOL AUDIT_FILE_DEST **AUDIT SYS OPERATIONS** AUDIT_SYSLOG_LEVEL **AUDIT TRAIL** BLOCKCHAIN_TABLE_MAX_NO_DROP BLOCKCHAIN_TABLE_RETENTION_THRESHOLD COMMIT_LOGGING COMMIT_WAIT DBFIPS 140 ENCRYPT_NEW_TABLESPACES HEARTBEAT BATCH SIZE IDENTITY_PROVIDER_CONFIG IDENTITY PROVIDER TYPE LOB_SIGNATURE_ENABLE MANDATORY_USER_PROFILE OS_AUTHENT_PREFIX OS ROLES OUTBOUND DBLINK PROTOCOLS PKCS11_LIBRARY_LOCATION RDBMS SERVER DN REMOTE_LOGIN_PASSWORDFILE REMOTE OS ROLES SEC_MAX_FAILED_LOGIN_ATTEMPTS SEC_PROTOCOL_ERROR_FURTHER_ACTION SEC_PROTOCOL_ERROR_TRACE_ACTION SEC RETURN SERVER RELEASE BANNER **SQL92 SECURITY** TABLESPACE ENCRYPTION TABLESPACE ENCRYPTION DEFAULT ALGORITHM TDE CONFIGURATION TDE_KEY_CACHE UNIFIED_AUDIT_COMMON_SYSTEMLOG UNIFIED_AUDIT_SYSTEMLOG WALLET_ROOT

Sessions and Processes

CPU COUNT

CPU_MIN_COUNT
PROCESSES
SESSIONS
THREADED_EXECUTION
USE_DEDICATED_BROKER

SGA Memory

DB nK CACHE SIZE DB CACHE SIZE HI_SHARED_MEMORY_ADDRESS JAVA POOL SIZE LARGE_POOL_SIZE LOCK_SGA OLAP PAGE POOL SIZE PRE_PAGE_SGA SGA MAX SIZE SGA MIN SIZE SGA TARGET SHARED_MEMORY_ADDRESS SHARED_POOL_RESERVED_SIZE SHARED_POOL_SIZE STREAMS_POOL_SIZE USE_LARGE_PAGES

Sharding

MULTISHARD_QUERY_DATA_CONSISTENCY
OPTIMIZER_CROSS_SHARD_RESILIENCY
SHARD_ENABLE_RAFT_FOLLOWER_READ
SHARD_RAFT_LOGFILE_SIZE
SHRD_DUPL_TABLE_REFRESH_RATE

Shared Server Architecture

CIRCUITS
DISPATCHERS
MAX_DISPATCHERS
MAX_SHARED_SERVERS
SHARED_SERVER_SESSIONS
SHARED_SERVERS

Standby Database

ARCHIVE_LAG_TARGET
DB_FILE_NAME_CONVERT
DB_RECOVERY_AUTO_REKEY
DB_UNIQUE_NAME
DG_BROKER_CONFIG_FILEn
DG_BROKER_START
ENABLE_IMC_WITH_MIRA
FAL_CLIENT
FAL_SERVER
LOG_FILE_NAME_CONVERT
STANDBY_FILE_MANAGEMENT

```
STANDBY_PDB_SOURCE_FILE_DBLINK
STANDBY_PDB_SOURCE_FILE_DIRECTORY
```

Temporary Sort Space

BITMAP_MERGE_AREA_SIZE
CREATE_BITMAP_AREA_SIZE
HASH_AREA_SIZE
PGA_AGGREGATE_LIMIT
PGA_AGGREGATE_TARGET
SORT_AREA_RETAINED_SIZE
SORT_AREA_SIZE
WORKAREA_SIZE_POLICY

Transactions

COMMIT_LOGGING
COMMIT_WAIT
DB_LOST_WRITE_PROTECT
DDL_LOCK_TIMEOUT
DML_LOCKS
FAST_START_PARALLEL_ROLLBACK
GLOBAL_TXN_PROCESSES
MAX_SAGA_DURATION
PRIORITY_TXNS_HIGH_WAIT_TARGET
PRIORITY_TXNS_MEDIUM_WAIT_TARGET
PRIORITY_TXNS_MODE
SAGA_HIST_RETENTION
TRANSACTION_RECOVERY
TRANSACTIONS
TXN_PRIORITY

Undo Management

RESUMABLE_TIMEOUT
ROLLBACK_SEGMENTS
TEMP_UNDO_ENABLED
TRANSACTIONS_PER_ROLLBACK_SEGMENT
UNDO_MANAGEMENT
UNDO_RETENTION
UNDO_TABLESPACE

2.4.2 Modifiable Parameters

You can modify some initialization parameters using the ALTER SESSION or ALTER SYSTEM statements while an instance is running.

Use the following syntax to modify initialization parameters:

```
ALTER SESSION SET parameter_name = value
ALTER SYSTEM SET parameter_name = value [DEFERRED]
```

Whenever a parameter is modified using the ALTER SYSTEM statement, the Oracle Database records the statement that modifies the parameter in the alert log.

The ALTER SESSION statement changes the value of the specified parameter for the duration of the session that invokes the statement. The value of the parameter does not change for other sessions in the instance. The value of the following initialization parameters can be changed with ALTER SESSION:

ALLOW_ROWID_COLUMN_TYPE APPROX FOR AGGREGATION APPROX_FOR_COUNT_DISTINCT APPROX_FOR_PERCENTILE ASM DISKSTRING ASM_POWER_LIMIT CLIENT_PREFETCH_ROWS COMMIT_LOGGING COMMIT_WAIT COMMIT_WRITE CONTAINER DATA CONTAINERS_PARALLEL_DEGREE CREATE STORED OUTLINES CURSOR_BIND_CAPTURE_DESTINATION CURSOR_INVALIDATION CURSOR_SHARING DB_CREATE_FILE_DEST DB_CREATE_ONLINE_LOG_DEST_n DB_FILE_MULTIBLOCK_READ_COUNT DB_FILE_NAME_CONVERT DB_INDEX_COMPRESSION_INHERITANCE DB SECUREFILE DB_UNRECOVERABLE_SCN_TRACKING DDL LOCK TIMEOUT DEFAULT_SHARING DEFERRED_SEGMENT_CREATION DIAGNOSTICS_CONTROL DST_UPGRADE_INSERT_CONV **ENABLE DDL LOGGING** ERROR_MESSAGE_DETAILS **GLOBAL NAMES** GROUP_BY_POSITION_ENABLED HASH AREA SIZE **HEAT MAP** IGNORE_SESSION_SET_PARAM_ERRORS INMEMORY_CLAUSE_DEFAULT INMEMORY_DEEP_VECTORIZATION INMEMORY_OPTIMIZED_DATE INMEMORY_QUERY JAVA JIT ENABLED JSON BEHAVIOR JSON EXPRESSION CHECK LOG_ARCHIVE_DEST_n LOG_ARCHIVE_DEST_STATE_n LOG_ARCHIVE_MIN_SUCCEED_DEST MAX_DUMP_FILE_SIZE

MEMOPTIMIZE WRITES

MLE PROG LANGUAGES

MULTISHARD QUERY DATA CONSISTENCY

NLS CALENDAR

NLS COMP

NLS CURRENCY

NLS_DATE_FORMAT

NLS DATE LANGUAGE

NLS_DUAL_CURRENCY

NLS ISO CURRENCY

NLS LANGUAGE

NLS LENGTH SEMANTICS

NLS_NCHAR_CONV_EXCP

NLS_NUMERIC_CHARACTERS

NLS_SORT

NLS TERRITORY

NLS_TIMESTAMP_FORMAT

NLS_TIMESTAMP_TZ_FORMAT

OBJECT_CACHE_MAX_SIZE_PERCENT

OBJECT CACHE OPTIMAL SIZE

OLAP_PAGE_POOL_SIZE

OPTIMIZER_ADAPTIVE_PLANS

OPTIMIZER_ADAPTIVE_REPORTING_ONLY

OPTIMIZER_ADAPTIVE_STATISTICS

OPTIMIZER CAPTURE SQL PLAN BASELINES

OPTIMIZER_CAPTURE_SQL_QUARANTINE

OPTIMIZER_CROSS_SHARD_RESILIENCY

OPTIMIZER DYNAMIC SAMPLING

OPTIMIZER_FEATURES_ENABLE

OPTIMIZER IGNORE HINTS

OPTIMIZER_IGNORE_PARALLEL_HINTS

OPTIMIZER_INDEX_CACHING

OPTIMIZER_INDEX_COST_ADJ

OPTIMIZER INMEMORY AWARE

OPTIMIZER_MODE

OPTIMIZER REAL TIME STATISTICS

OPTIMIZER SESSION TYPE

OPTIMIZER USE INVISIBLE INDEXES

OPTIMIZER USE PENDING STATISTICS

OPTIMIZER_USE_SQL_PLAN_BASELINES

OPTIMIZER_USE_SQL_QUARANTINE

PARALLEL_DEGREE_LIMIT

PARALLEL DEGREE POLICY

PARALLEL_FORCE_LOCAL

PARALLEL INSTANCE GROUP

PARALLEL MIN DEGREE

PARALLEL MIN PERCENT

PARALLEL_MIN_TIME_THRESHOLD

PDB FILE NAME CONVERT

PDB LOCKDOWN

```
PLSCOPE SETTINGS
PLSQL CCFLAGS
PLSOL CODE TYPE
PLSQL DEBUG
PLSQL FUNCTION DYNAMIC STATS
PLSQL_IMPLICIT_CONVERSION_BOOL
PLSQL_OPTIMIZE_LEVEL
PLSQL V2 COMPATIBILITY
PLSQL WARNINGS
QUERY REWRITE ENABLED
QUERY_REWRITE_INTEGRITY
READ ONLY
RECYCLEBIN
REMOTE DEPENDENCIES MODE
RESULT_CACHE_MODE
RESULT CACHE REMOTE EXPIRATION
RESUMABLE TIMEOUT
SESSION_CACHED_CURSORS
SESSION EXIT ON PACKAGE STATE ERROR
SHARD ENABLE RAFT FOLLOWER READ
SHRD DUPL TABLE REFRESH RATE
SKIP UNUSABLE INDEXES
SMTP_OUT_SERVER
SORT_AREA_RETAINED_SIZE
SORT AREA SIZE
SPATIAL_VECTOR_ACCELERATION
SQL ERROR MITIGATION
SQL_HISTORY_ENABLED
SQL TRACE
SQL TRANSPILER
SQLTUNE CATEGORY
STAR TRANSFORMATION ENABLED
STATISTICS_LEVEL
TDE KEY CACHE
TEMP_UNDO_ENABLED
TIMED OS STATISTICS
TIMED STATISTICS
TRACEFILE IDENTIFIER
TXN PRIORITY
WORKAREA_SIZE_POLICY
XML_CLIENT_SIDE_DECODING
XML_DB_EVENTS
```

The ALTER SYSTEM statement without the DEFERRED keyword modifies the global value of the parameter for all sessions in the instance, for the duration of the instance (until the database is shut down). The value of the following initialization parameters can be changed with ALTER SYSTEM:

```
ADG_REDIRECT_DML
ALLOW_GLOBAL_DBLINKS
```

XML PARAMS

ALLOW ROWID COLUMN TYPE

APPROX FOR AGGREGATION

APPROX FOR COUNT DISTINCT

APPROX FOR PERCENTILE

AQ TM PROCESSES

ARCHIVE_LAG_TARGET

ASM_DISKGROUPS

ASM DISKSTRING

ASM_IO_PROCESSES

ASM POWER LIMIT

ASM PREFERRED READ FAILURE GROUPS

AUTOTASK MAX ACTIVE PDBS

AWR PDB AUTOFLUSH ENABLED

AWR_SNAPSHOT_TIME_OFFSET

BACKGROUND_CORE_DUMP

BACKGROUND DUMP DEST

BLOCKCHAIN_TABLE_MAX_NO_DROP

BLOCKCHAIN_TABLE_RETENTION_THRESHOLD

CIRCUITS

CLIENT PREFETCH ROWS

COMMIT LOGGING

COMMIT_WAIT

COMMIT_WRITE

CONNECTION_BROKERS

CONTAINERS PARALLEL DEGREE

CONTROL_FILE_RECORD_KEEP_TIME

CONTROL_MANAGEMENT_PACK_ACCESS

CORE_DUMP_DEST

CPU COUNT

CPU MIN COUNT

CREATE_STORED_OUTLINES

CURSOR BIND CAPTURE DESTINATION

CURSOR_INVALIDATION

CURSOR SHARING

DATA_GUARD_MAX_IO_TIME

DATA GUARD MAX LONGIO TIME

DATA GUARD SYNC LATENCY

DATA TRANSFER CACHE SIZE

DB nK CACHE SIZE

DB_BIG_TABLE_CACHE_PERCENT_TARGET

DB_BLOCK_CHECKING

DB_BLOCK_CHECKSUM

DB CACHE ADVICE

DB_CACHE_SIZE

DB CREATE FILE DEST

DB CREATE ONLINE LOG DEST n

DB FILE MULTIBLOCK READ COUNT

DB_FILE_NAME_CONVERT

DB FLASH CACHE FILE

DB_FLASH_CACHE_SIZE



DB FLASHBACK LOG DEST

DB FLASHBACK LOG DEST SIZE

DB FLASHBACK RETENTION TARGET

DB_INDEX_COMPRESSION_INHERITANCE

DB_KEEP_CACHE_SIZE

DB_LOST_WRITE_PROTECT

DB_PERFORMANCE_PROFILE

DB_RECOVERY_AUTO_REKEY

DB_RECOVERY_FILE_DEST

DB RECOVERY FILE DEST SIZE

DB_RECYCLE_CACHE_SIZE

DB_SECUREFILE

DB_UNRECOVERABLE_SCN_TRACKING

DDL_LOCK_TIMEOUT

DEFAULT_SHARING

DEFERRED SEGMENT CREATION

DG_BROKER_CONFIG_FILEn

DG_BROKER_START

DIAGNOSTIC DEST

DISPATCHERS

DISTRIBUTED LOCK TIMEOUT

DRCP_CONNECTION_LIMIT

DRCP DEDICATED OPT

DST_UPGRADE_INSERT_CONV

ENABLE AUTOMATIC MAINTENANCE PDB

ENABLE DDL LOGGING

ENABLE GOLDENGATE REPLICATION

ENABLE_IMC_WITH_MIRA

ENABLED_PDBS_ON_STANDBY

ENCRYPT_NEW_TABLESPACES

ERROR MESSAGE DETAILS

FAL_CLIENT

FAL_SERVER

FAST START MTTR TARGET

FAST_START_PARALLEL_ROLLBACK

FILE MAPPING

FIXED DATE

FORWARD LISTENER

GLOBAL NAMES

GLOBAL_TXN_PROCESSES

GROUP_BY_POSITION_ENABLED

HEARTBEAT_BATCH_SIZE

HEAT MAP

HS_AUTOREGISTER

HYBRID READ ONLY

IDENTITY PROVIDER CONFIG

IDENTITY PROVIDER TYPE

IGNORE_SESSION_SET_PARAM_ERRORS

INMEMORY ADG ENABLED

INMEMORY_AUTOMATIC_LEVEL

INMEMORY CLAUSE DEFAULT

INMEMORY DEEP VECTORIZATION

INMEMORY EXPRESSIONS USAGE

INMEMORY FORCE

INMEMORY_MAX_POPULATE_SERVERS

INMEMORY_OPTIMIZED_ARITHMETIC

INMEMORY_OPTIMIZED_DATE

INMEMORY_QUERY

INMEMORY_SIZE

INMEMORY TRICKLE REPOPULATE SERVERS PERCENT

INMEMORY_VIRTUAL_COLUMNS

IORM LIMIT POLICY

JAVA_JIT_ENABLED

JAVA POOL SIZE

JOB_QUEUE_PROCESSES

JSON EXPRESSION CHECK

LARGE POOL SIZE

LDAP_DIRECTORY_ACCESS

LICENSE MAX SESSIONS

LICENSE MAX USERS

LICENSE SESSIONS WARNING

LISTENER_NETWORKS

LOB_SIGNATURE_ENABLE

LOCAL_LISTENER

LOG ARCHIVE CONFIG

LOG_ARCHIVE_DEST

LOG ARCHIVE DEST n

LOG ARCHIVE DEST STATE n

LOG ARCHIVE DUPLEX DEST

LOG_ARCHIVE_MAX_PROCESSES

LOG_ARCHIVE_MIN_SUCCEED_DEST

LOG_ARCHIVE_TRACE

LOG_CHECKPOINT_INTERVAL

LOG_CHECKPOINT_TIMEOUT

LOG_CHECKPOINTS_TO_ALERT

LOG FILE NAME CONVERT

LONG MODULE ACTION

MANDATORY USER PROFILE

MAX AUTH SERVERS

MAX_DATAPUMP_JOBS_PER_PDB

MAX_DATAPUMP_PARALLEL_PER_JOB

MAX_DISPATCHERS

MAX DUMP FILE SIZE

MAX_IDLE_BLOCKER_TIME

MAX IDLE TIME

MAX IOPS

MAX_MBPS

MAX_PDBS

MAX SAGA DURATION

MAX_SHARED_SERVERS

MEMOPTIMIZE POOL SIZE

MEMOPTIMIZE WRITE AREA SIZE

MEMORY SIZE

MEMORY TARGET

MIN AUTH SERVERS

MLE_PROG_LANGUAGES

MULTISHARD_QUERY_DATA_CONSISTENCY

NLS LENGTH SEMANTICS

NLS_NCHAR_CONV_EXCP

OFS THREADS

OPEN CURSORS

OPTIMIZER ADAPTIVE PLANS

OPTIMIZER_ADAPTIVE_REPORTING_ONLY

OPTIMIZER_ADAPTIVE_STATISTICS

OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES

OPTIMIZER CAPTURE SQL QUARANTINE

OPTIMIZER_CROSS_SHARD_RESILIENCY

OPTIMIZER_DYNAMIC_SAMPLING

OPTIMIZER FEATURES ENABLE

OPTIMIZER_IGNORE_HINTS

OPTIMIZER_IGNORE_PARALLEL_HINTS

OPTIMIZER_INDEX_CACHING

OPTIMIZER_INDEX_COST_ADJ

OPTIMIZER_INMEMORY_AWARE

OPTIMIZER MODE

OPTIMIZER_REAL_TIME_STATISTICS

OPTIMIZER_USE_INVISIBLE_INDEXES

OPTIMIZER USE PENDING STATISTICS

OPTIMIZER USE SQL PLAN BASELINES

OPTIMIZER USE SQL QUARANTINE

OUTBOUND_DBLINK_PROTOCOLS

PARALLEL ADAPTIVE MULTI USER

PARALLEL_DEGREE_LIMIT

PARALLEL DEGREE POLICY

PARALLEL FORCE LOCAL

PARALLEL INSTANCE GROUP

PARALLEL MAX SERVERS

PARALLEL MIN DEGREE

PARALLEL MIN SERVERS

PARALLEL_MIN_TIME_THRESHOLD

PARALLEL_SERVERS_TARGET

PARALLEL_THREADS_PER_CPU

PDB FILE NAME CONVERT

PDB LOCKDOWN

PGA AGGREGATE LIMIT

PGA AGGREGATE TARGET

PKCS11 LIBRARY LOCATION

PLSCOPE_SETTINGS

PLSQL CCFLAGS

PLSQL_CODE_TYPE



PLSQL DEBUG

PLSQL IMPLICIT CONVERSION BOOL

PLSOL OPTIMIZE LEVEL

PLSQL V2 COMPATIBILITY

PLSQL WARNINGS

PRIORITY_TXNS_HIGH_WAIT_TARGET

PRIORITY_TXNS_MEDIUM_WAIT_TARGET

PRIORITY_TXNS_MODE

PROCESSES

QUERY REWRITE ENABLED

QUERY REWRITE INTEGRITY

RECOVERY_PARALLELISM

REDO TRANSPORT USER

REMOTE_DEPENDENCIES_MODE

REMOTE_LISTENER

REMOTE RECOVERY FILE DEST

RESOURCE LIMIT

RESOURCE_MANAGER_CPU_ALLOCATION

RESOURCE MANAGER PLAN

RESULT_CACHE_AUTO_BLOCKLIST

RESULT_CACHE_EXECUTION_THRESHOLD

RESULT_CACHE_INTEGRITY

RESULT_CACHE_MAX_RESULT

RESULT_CACHE_MAX_SIZE

RESULT CACHE MAX TEMP RESULT

RESULT_CACHE_MAX_TEMP_SIZE

RESULT_CACHE_MODE

RESULT CACHE REMOTE EXPIRATION

RESUMABLE TIMEOUT

SAGA HIST RETENTION

SEC_PROTOCOL_ERROR_FURTHER_ACTION

SEC_PROTOCOL_ERROR_TRACE_ACTION

SERVICE_NAMES

SESSION EXIT ON PACKAGE STATE ERROR

SGA_MIN_SIZE

SGA TARGET

SHADOW CORE DUMP

SHARD_ENABLE_RAFT_FOLLOWER_READ

SHARD RAFT LOGFILE SIZE

SHARED_POOL_SIZE

SHARED_SERVER_SESSIONS

SHARED_SERVERS

SHRD DUPL TABLE REFRESH RATE

SKIP_UNUSABLE_INDEXES

SMTP OUT SERVER

SPATIAL_VECTOR_ACCELERATION

SPFILE

SQL_ERROR_MITIGATION

SQL_HISTORY_ENABLED

SQL_TRACE

```
SQL_TRANSPILER
SQLTUNE CATEGORY
STANDBY FILE_MANAGEMENT
STANDBY PDB SOURCE FILE DBLINK
STANDBY PDB SOURCE FILE DIRECTORY
STAR TRANSFORMATION ENABLED
STATISTICS_LEVEL
STREAMS POOL SIZE
TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM
TDE_CONFIGURATION
TDE KEY CACHE
TEMP UNDO ENABLED
THREAD
TIMED_OS_STATISTICS
TIMED_STATISTICS
TIMEZONE VERSION UPGRADE ONLINE
TRACE ENABLED
TRANSACTION_RECOVERY
UNDO RETENTION
UNDO TABLESPACE
UNIFORM LOG TIMESTAMP FORMAT
USE DEDICATED BROKER
USER_DUMP_DEST
WORKAREA_SIZE_POLICY
XML CLIENT SIDE DECODING
XML DB EVENTS
XML PARAMS
```

The ALTER SYSTEM ... DEFERRED statement does not modify the global value of the parameter for existing sessions, but the value will be modified for future sessions that connect to the database. The value of the following initialization parameters can be changed with ALTER SYSTEM ... DEFERRED:

```
AUDIT_FILE_DEST
BACKUP_TAPE_IO_SLAVES
CLIENT_STATISTICS_LEVEL
OBJECT_CACHE_MAX_SIZE_PERCENT
OBJECT_CACHE_OPTIMAL_SIZE
OLAP_PAGE_POOL_SIZE
PRIVATE_TEMP_TABLE_PREFIX
RECYCLEBIN
SESSION_CACHED_CURSORS
SORT_AREA_RETAINED_SIZE
SORT_AREA_SIZE
```

2.4.3 Displaying Current Parameter Values

You can use the SQL*Plus SHOW PARAMETERS statement to see the current settings for initialization parameters.

For example:

SQL> SHOW PARAMETERS

This statement displays all parameters in alphabetical order, along with their current values.

Enter the following text string to display all parameters having BLOCK in their names:

SQL> SHOW PARAMETERS BLOCK

You can use the SPOOL command to write the output to a file.

2.4.4 Parameters You Should Not Specify in the Parameter File

You should not specify these types of parameters in your parameter files:

- Parameters that you never alter unless instructed to do so by Oracle to resolve a problem
- Derived parameters, which normally do not need altering because their values are calculated automatically by the Oracle database server

2.4.5 When Parameters Are Set Incorrectly

When a parameter is set incorrectly, the effect can be different for different parameters, or based on how low or high the parameter is set.

Some parameters have a minimum setting below which an Oracle instance will not start. For other parameters, setting the value too low or too high may cause Oracle to perform badly, but it will still run. Also, Oracle may convert some values outside the acceptable range to usable levels.

If a parameter value is too low or too high, or you have reached the maximum for some resource, then Oracle returns an error. Frequently, you can wait a short while and retry the operation when the system is not as busy. If a message occurs repeatedly, then you should shut down the instance, adjust the relevant parameter, and restart the instance.

2.5 Reading the Parameter Descriptions

This section describes the properties that are documented in the initialization parameter descriptions.

PARAMETER NAME

Property	Description	
Parameter type	Specifies the type of the parameter:	
	 A Boolean parameter accepts either true or false as its value. 	
	 A string parameter accepts any sequence of characters as its value, subject to the syntax for the parameter. 	
	 An integer parameter accepts a 4-byte value that can range from 0 to 2³² 1. 	
	 A parameter file parameter accepts an initialization parameter file specification as its value. 	
	 A big integer parameter accepts an 8-byte value that can range from 0 to 2⁶⁴ - 1. You specify a value for a big integer as an integer together with an optional modifier such as K, M, or G, which respectively denotes kilobytes, megabytes, or gigabytes. 	
	For example, 1000, 100 KB, 50 MB and 2 GB are valid specifications for big integers.	



Property	Description
Syntax	For string and big integer parameters, specifies the valid syntax for specifying the parameter.
Default value	Specifies the value that the parameter assumes if a value is not explicitly specified.
	For most initialization parameters, the value that is documented for the Default value property is obtained by querying the DEFAULT_VALUE column in the V\$PARAMETER view for the parameter.
Modifiable	Specifies whether the parameter can be changed for the current session (by an ALTER SESSION statement) or for all sessions in the current instance (by an ALTER SYSTEM statement):
	 ALTER SESSION overrides the instance-wide setting of the parameter for the current session only. You can restore the instance-wide setting for that session only by issuing another ALTER SESSION statement.
	 ALTER SYSTEM can be used to change the value in the server parameter file (SPFILE) of any initialization parameter. Such a change takes effect only in subsequent instances. The parameter descriptions indicate only those parameters that can be modified for the current instance.
Modifiable in a PDB	Specifies whether the parameter can be modified in a PDB (Yes) or not (No)
	For most initialization parameters, the value that is documented for the Modifiable in a PDB property is obtained by querying the ISPDB_MODIFIABLE column in the V\$PARAMETER view for the parameter.
Range of values	Specifies the valid range of values that this parameter can assume, shown as a minimum and maximum value. Not applicable to all parameters.
Basic	Specifies whether the parameter is a basic initialization parameter or not
Oracle RAC	Specifies how the values for this parameter must be specified for multiple instances in an Oracle Real Application Clusters environment. Not applicable to all parameters.

For each parameter, paragraphs following these details further describe the parameter and the effects of different settings.



2.6 Initialization Parameter Descriptions

The remainder of this chapter describes the initialization parameters in alphabetical order.

Initialization parameter values apply to the entire database, not to an individual user, unless otherwise specified.



Parameters that have become obsolete are not documented.

See Also:

- Oracle Database Upgrade Guide for information about obsolete parameters
- Your system release bulletins or other operating system-specific Oracle documentation

2.7 ACTIVE_INSTANCE_COUNT

ACTIVE_INSTANCE_COUNT enables you to designate one instance in a two-instance cluster as the primary instance and the other instance as the secondary instance. This parameter has no functionality in a cluster with more than two instances.

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	No
Modifiable in a PDB	No
Range of values	1 or >= the number of instances in the cluster. (Values other than 1 have no effect on the active or standby status of any instances.)
Basic	No
Oracle RAC	You must set this parameter for every instance, and multiple instances must have the same value.



The ACTIVE_INSTANCE_COUNT parameter is deprecated. It is retained for backward compatibility only.

When you set this parameter to 1, the first instance you start up becomes the primary instance and accepts client connections. The second instance starts up as a secondary instance and can accept client connections only if the first instance fails. In such an event, the secondary instance becomes the primary instance.

When the failed instance can once again be started up, it starts up as the secondary instance, and will not accept client connections unless the current primary instance fails.



This parameter functions only in a cluster with only two instances.



2.8 ADG_ACCOUNT_INFO_TRACKING

ADG_ACCOUNT_INFO_TRACKING controls login attempts of users on Oracle Active Data Guard standby databases. It extends the control of user account security information.

Property	Description
Parameter type	String
Syntax	ADG_ACCOUNT_INFO_TRACKING = { LOCAL GLOBAL }
Default value	LOCAL
Modifiable	No
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value must be used on all instances.

Setting ADG_ACCOUNT_INFO_TRACKING to LOCAL (the default value) continues to enforce the default behavior, by maintaining a local copy of users account information in the standby's inmemory view. Login failures are only tracked locally on a per database basis, and login is denied when the failure maximum is reached.

Setting the parameter to GLOBAL triggers a more secure behavior, by maintaining a single global copy of users account information across all Data Guard databases (primary and standby). Login failures across all databases in the Data Guard configuration count towards the maximum count and logins anywhere will be denied when the count is reached. This setting improves security against login attacks across a production database and all Active Data Guard standby databases.

2.9 ADG REDIRECT DML

Use <code>ADG_REDIRECT_DML</code> to enable or disable automatic redirection of DML operations from a standby to the primary in Oracle Active Data Guard environments.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
	Can be modified at the session level with a special ALTER SESSION command. See details below.
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	Different instances can use different values.

Automatic redirection of DML operations to the primary can be configured at the system level or the session level. The session level setting overrides the system level setting.

 To enable automatic redirection of DML operations to the primary at the system level, set ADG REDIRECT DML to true.

- To disable automatic redirection of DML operations to the primary at the system level, set ADG REDIRECT DML to false.
- To enable automatic redirection of DML operations from a standby to the primary for the current session, issue the following SQL statement:

```
ALTER SESSION ENABLE ADG_REDIRECT_DML;
```

 To disable automatic redirection of DML operations from a standby to the primary for the current session, issue the following SQL statement:

ALTER SESSION DISABLE ADG REDIRECT DML;



Oracle Data Guard Concepts and Administration for more information about automatic redirection of DML operations

2.10 ALLOW_GLOBAL_DBLINKS

ALLOW_GLOBAL_DBLINKS specifies whether LDAP lookup for database links is allowed for the database.

Property	Description
Parameter type	Boolean
Syntax	ALLOW_GLOBAL_DBLINKS = { FALSE TRUE }
Default value	FALSE
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	The same value must be used on all instances.

The following values can be set:

- FALSE: LDAP lookup for a database link's definition is not allowed for the database.
- TRUE: LDAP lookup for a database link's definition is allowed for the database.

2.11 ALLOW_GROUP_ACCESS_TO_SGA

ALLOW GROUP ACCESS TO SGA controls group access to shared memory on UNIX platforms.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No



Property	Description
Range of values	true false
Basic	No
Oracle RAC	The same value must be used on all instances.

The default value is false, which means that database shared memory is created with owner access only. In Oracle Database releases prior to Oracle Database 12c Release 2 (12.2.0.1), database shared memory was created with owner and group access.

When this parameter is set to true, database shared memory is created with owner and group access. This behavior grants permissions to DBAs to manage shared memory outside the database, but also allows DBAs to read and write to shared memory, which may not be desirable for certain installations.

2.12 ALLOW_LEGACY_RECO_PROTOCOL

ALLOW_LEGACY_RECO_PROTOCOL specifies whether the distributed transactions recovery process RECO uses a legacy recovery protocol or an upgraded recovery protocol.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	All instances must use the same value.

By default, ALLOW_LEGACY_RECO_PROTOCOL is set to true, which instructs RECO to use the legacy recovery protocol available in releases prior to Oracle Database 23ai.

Setting this parameter to false instructs RECO to use an upgraded recovery protocol. In this case, all databases involved in distributed transactions must run Oracle Database 23ai or later. If you set this parameter to false and a pre-Oracle Database 23ai database is involved in a distributed transaction, the distributed transaction recovery will fail. To ensure that distributed transaction recovery will complete successfully, the ALLOW_LEGACY_RECO_PROTOCOL must remain set to true until all databases are upgraded to at least Oracle Database 23ai.



This parameter is available starting with Oracle Database 23ai.

See Also:

Oracle Database Administrator's Guide for more information about this parameter

2.13 ALLOW_ROWID_COLUMN_TYPE

ALLOW_ROWID_COLUMN_TYPE determines whether table columns of data type ROWID are allowed to be created.



This parameter is valid only for Oracle Autonomous Database on dedicated Exadata infrastructure. On other types of Oracle databases, the value of this parameter is ignored and ROWID columns are always allowed to be created.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Multiple instances should use the same value

Values:

false

Columns of data type ROWID are not allowed.

You cannot specify ROWID columns when creating tables, nor can you add ROWID columns to existing tables. If the value of this parameter is changed from true to false at a time when ROWID columns exist in the database, then those columns are allowed to remain. However, Oracle recommends that you leave this parameter set to its default value of false and avoid introducing ROWID columns into Oracle Autonomous Database on dedicated Exadata infrastructure.

true

Columns of data type ROWID are allowed.

If you choose this setting, be aware that ROWID columns are incompatible with rolling upgrade operations and other internal operations that physically move table rows. At a minimum, database activities involving ROWID columns should be suspended during upgrades. Applications using ROWID columns should introduce correctness validation to mitigate against logical errors in the application if a row relocates.



2.14 APPROX_FOR_AGGREGATION

APPROX_FOR_AGGREGATION replaces exact query processing for aggregation queries with approximate query processing.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Different instances can use different values.

Data analysis applications heavily use aggregate function and analytic function queries. Aggregation functions and analytic functions require sorting of large volumes of data, and exact query answering requires lots of memory, and can be time consuming. With approximate query processing, the results of aggregate function and analytic function queries are returned much faster than with exact query processing. Approximate query processing is useful for situations where a tolerable amount of error is acceptable in order to obtain faster query results.

The values that can be set are:

• false:

Approximate query processing is not used for aggregation queries and analytic queries.

true: Approximate query processing is used for aggregation queries and analytic queries.

See Also:

- "APPROX_FOR_COUNT_DISTINCT"
- "APPROX FOR PERCENTILE"

2.15 APPROX_FOR_COUNT_DISTINCT

APPROX_FOR_COUNT_DISTINCT automatically replaces COUNT (DISTINCT expr) queries with APPROX COUNT DISTINCT queries.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false



Property	Description	
Basic	No	
Oracle RAC	Different instances can use different values.	

Query results for APPROX_COUNT_DISTINCT queries are returned faster than the equivalent COUNT (DISTINCT expr) queries. APPROX_COUNT_DISTINCT queries are useful for situations where a tolerable amount of error is acceptable in order to obtain faster query results than with a COUNT (DISTINCT expr) query.

See Also:

- "APPROX_FOR_AGGREGATION"
- "APPROX_FOR_PERCENTILE"

2.16 APPROX_FOR_PERCENTILE

APPROX_FOR_PERCENTILE converts exact percentile functions to their approximate percentile function counterparts.

Property	Description	
Parameter type	String	
Syntax	APPROX_FOR_PERCENTILE = { NONE PERCENTILE_CONT PERCENTILE_DISC PERCENTILE_DISC PERCENTILE_DISC DETERMINISTIC ALL ALL DETERMINISTIC }	
Default value	none	
Modifiable	ALTER SESSION, ALTER SYSTEM	
Modifiable in a PDB	Yes	
Basic	No	
Oracle RAC	Different instances can use different values.	

Approximate percentile function queries are faster than their exact percentile function query counterparts, so they can be useful in situations where a tolerable amount of error is acceptable in order to obtain faster query results.

The values that can be set are:

NONE:

No queries are converted. This is the default value.

PERCENTILE CONT:

Converts PERCENTILE CONT queries to APPROX PERCENTILE queries.

PERCENTILE CONT DETERMINISTIC:

Converts PERCENTILE CONT queries to APPROX PERCENTILE DETERMINSTIC queries.

PERCENTILE DISC:

Converts PERCENTILE DISC queries to APPROX PERCENTILE queries.

PERCENTILE DISC DETERMINISTIC:

Converts PERCENTILE DISC queries to APPROX PERCENTILE DETERMINISTIC queries.

ALL:

Converts both PERCENTILE_CONT queries and PERCENTILE_DISC queries to APPROX PERCENTILE queries.

• ALL DETERMINISTIC:

Converts both PERCENTILE_CONT and PERCENTILE_DISC queries to APPROX_PERCENTILE DETERMINISTIC queries.

See Also:

- "APPROX_FOR_AGGREGATION"
- "APPROX_FOR_COUNT_DISTINCT"
- Oracle Database SQL Language Reference for information about APPROX_MEDIAN aggregate functions
- Oracle Database SQL Language Reference for information about APPROX PERCENTILE aggregate functions

2.17 AQ_TM_PROCESSES

 $AQ_TM_PROCESSES$ controls time monitoring on queue messages and controls processing of messages with delay and expiration properties specified.

Property	Description
Parameter type	Integer
Default value	1
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 40
Basic	No

You do not need to specify a value for this parameter because Oracle Database automatically determines the number of processes and autotunes them, as necessary. Therefore, Oracle highly recommends that you leave the $AQ_TM_PROCESSES$ parameter unspecified and let the system autotune.

The default value for AQ_TM_PROCESSES is used if the client does not explicitly set a value for the parameter in the init.ora file or using the ALTER SYSTEM statement.



If you want to disable the Queue Monitor Coordinator, then you must set ${\tt AQ_TM_PROCESSES}$ to 0 in your parameter file. Oracle strongly recommends that you do NOT set ${\tt AQ_TM_PROCESSES}$ to 0.

See Also:

Oracle Database Advanced Queuing User's Guide for more information about this parameter

2.18 ARCHIVE_LAG_TARGET

ARCHIVE LAG TARGET forces a log switch after the specified amount of time elapses.

Property	Description
Parameter type	Integer
Default value	0 (disabled)
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 or any integer in [60, 7200]
Basic	No
Oracle RAC	Multiple instances should use the same value

A 0 value disables the time-based thread advance feature; otherwise, the value represents the number of seconds. Values larger than 7200 seconds are not of much use in maintaining a reasonable lag in the standby database. The typical, or recommended value is 1800 (30 minutes). Extremely low values can result in frequent log switches, which could degrade performance; such values can also make the archiver process too busy to archive the continuously generated logs.

See Also:

Oracle Database Administrator's Guide for more information about setting this parameter

2.19 ASM_DISKGROUPS

ASM_DISKGROUPS specifies a list of disk group names that an Oracle ASM instance mounts at startup when the ALTER DISKGROUP ALL MOUNT statement is issued.

Property	Description
Parameter type	String
Syntax	ASM_DISKGROUPS = diskgroup [, diskgroup]
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Comma-separated list of strings, up to 30 characters
Basic	No
Oracle RAC	Multiple instances can have different values.



This parameter may only be specified in an Oracle Automatic Storage Management (Oracle ASM) instance.

The Oracle ASM instance startup process executes ALTER DISKGROUP ALL MOUNT unless the NOMOUNT startup option is specified.

The ASM_DISKGROUPS parameter is dynamic. If you are using a server parameter file (SPFILE), then you do not have to manually alter the value of ASM_DISKGROUPS in most situations. Oracle ASM automatically adds a disk group to the parameter when the disk group is successfully created or mounted. Oracle ASM also automatically removes a disk group from the parameter when the disk group is dropped. However, the SPFILE is not updated on a manual dismount.

Issuing the ALTER DISKGROUP...ALL MOUNT or ALTER DISKGROUP...ALL DISMOUNT command does not affect the value of this parameter.

Supporting Up to 511 Disk Groups for an Oracle ASM Instance

In Oracle Database 12c Release 1 or later, Oracle ASM configurations support up to 511 disk groups. Oracle ASM configurations with Oracle Database releases before 12c Release 1 can only support up to 63 disk groups.



Oracle Automatic Storage Management Administrator's Guide for more information about and examples of using this parameter

2.20 ASM_DISKSTRING

ASM_DISKSTRING specifies an operating system-dependent value used by Oracle ASM to limit the set of disks considered for discovery.

Property	Description
Parameter type	String



Property	Description
Syntax	ASM_DISKSTRING = discovery_string [, discovery_string]
Default value	The null string; Oracle Automatic Storage Management discovery finds all disks in an operating system-specific location to which the Oracle Automatic Storage Management instance has read/write access.
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Multiple instances can have different values. Different nodes might see the same disks under different names; however, each instance must be able to use its <code>ASM_DISKSTRING</code> to discover the same physical media as the other nodes in the cluster.

Note:

This parameter may only be specified in an Oracle Automatic Storage Management (Oracle ASM) instance.

When a new disk is added to a disk group, each Oracle ASM instance that has the disk group mounted must be able to discover the new disk using the value of ASM DISKSTRING.

In most cases, the default value will be sufficient. Using a more restrictive value may reduce the time required for Oracle ASM to perform discovery, and thus improve disk group mount time or the time for adding a disk to a disk group. A "?" at the beginning of the string gets expanded to the Oracle home directory. Depending on the operating system, wildcard characters can be used. It may be necessary to dynamically change ASM_DISKSTRING before adding a disk so that the new disk will be discovered.

An attempt to dynamically modify $ASM_DISKSTRING$ will be rejected and the old value retained if the new value cannot be used to discover a disk that is in a disk group that is already mounted.

See Also:

Oracle Automatic Storage Management Administrator's Guide for more information and examples of using this parameter

2.21 ASM_IO_PROCESSES

ASM_IO_PROCESSES specifies the number of I/O worker processes to be started in an Oracle IOServer instance.

Property	Description
Parameter type	Integer
Default value	20
Modifiable	ALTER SYSTEM



Property	Description
Modifiable in a PDB	No
Range of values	1 – 32
Basic	No
Oracle RAC	Multiple instances can have different values.

This parameter is applicable only in an Oracle IOServer instance, which runs out of an Oracle Grid Infrastructure home.

The default value should work in most cases. However, under heavy I/O load, there may be some delays associated with posting processes out of I/O waits. In this case, a slightly higher value than the default may be appropriate.

2.22 ASM_POWER_LIMIT

 ${\tt ASM_POWER_LIMIT}$ specifies the maximum power on an Oracle ASM instance for disk rebalancing.

Property	Description
Parameter type	Integer
Default value	1
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 to 11 ¹
Basic	No
Oracle RAC	Multiple instances can have different values.

¹ Beginning with Oracle Database 11*g* Release 2 (11.2.0.2), if the COMPATIBLE.ASM disk group attribute is set to 11.2.0.2 or higher, then the range of values is 0 to 1024.

Note:

- This parameter may only be specified in an Oracle Automatic Storage Management (Oracle ASM) instance.
- In an Oracle Exadata or Oracle Data Appliance environment, you cannot set this parameter to 0.

The higher the limit, the faster rebalancing will complete. Lower values will take longer, but consume fewer processing and I/O resources.

If the POWER clause of a rebalance operation is not specified, then the default power will be the value of ASM POWER LIMIT.

See Also:

Oracle Automatic Storage Management Administrator's Guide for more information about using this parameter

2.23 ASM_PREFERRED_READ_FAILURE_GROUPS

ASM_PREFERRED_READ_FAILURE_GROUPS specifies the failure groups that contain preferred read disks. Preferred disks are instance specific.

Property	Description
Parameter type	String
Syntax	ASM_PREFERRED_READ_FAILURE_GROUPS = \diskgroup_name>. <failure_group_name>,'</failure_group_name>
Default value	NULL
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	The value is different on different nodes

Note:

The ASM_PREFERRED_READ_FAILURE_GROUPS initialization parameter is deprecated in Oracle Database 12c Release 2 (12.2.0.1) and may be desupported in a future release. It is replaced by the PREFERRED_READ.ENABLED disk group attribute.

See Oracle Automatic Storage Management Administrator's Guide for more information about the PREFERRED READ. ENABLED disk group attribute.

To reset the parameter value to null, issue the following statement, which updates the server parameter file (SP file), then reboot Oracle ASM so that the change takes effect:

alter system reset asm preferred read failure groups;

See Also:

Oracle Automatic Storage Management Administrator's Guide for more information about this parameter



2.24 AUDIT FILE DEST

AUDIT_FILE_DEST specifies the operating system directory into which the audit trail is written when the AUDIT_TRAIL initialization parameter is set to os, xml, or xml,extended.



This parameter is deprecated and applies only to traditional auditing. Oracle strongly recommends against the use of this parameter. If you attempt to set this parameter, you will incur the following errors:

ORA-32004: obsolete or deprecated parameter(s) specified for *string* instance
ORA-32006: AUDIT_FILE_DEST initialization parameter has been deprecated

Traditional auditing is desupported starting in Oracle Database 23ai. Though traditional auditing is desupported, any current traditional audit settings that you have will still be honored. However, you cannot create new traditional audit settings. You can delete existing traditional audit settings. See *Oracle Database Security Guide* for more information about how this desupport works.

Property	Description
Parameter type	String
Syntax	AUDIT_FILE_DEST = 'directory'
Default value	The first default value is:
	ORACLE_BASE/admin/ORACLE_SID/adump
	The second default value, which is used if the first default value does not exist or is unusable, is:
	ORACLE_HOME/rdbms/audit
	Both of these default values are for UNIX systems. Other platforms may have different defaults.
	In a multitenant container database (CDB), both of these default values will be appended with the GUID of the pluggable database (PDB) to store audit records that belong to the PDB. For example, if the PDB's GUID is 03E1F908EE04252CE053B280E80AAAA3, the first default directory will be:
	ORACLE_BASE/admin/ORACLE_SID/adump/ 03E1F908EE04252CE053B280E80AAAA3
	You can use the V\$CONTAINERS view to query a PDB's GUID.
Modifiable	ALTER SYSTEM DEFERRED
Modifiable in a PDB	No
Basic	No

If you do not explicitly set a value for this parameter in the initialization file, then at least one of the two default operating system directories mentioned in the preceding table must exist and be writeable. Otherwise, the instance will fail to start.

If you explicitly set a value for this parameter in the initialization file, then the directory you specify must exist and be writeable. Otherwise, the instance will fail to start.

The audit records will be written in XML format if the AUDIT_TRAIL initialization parameter is set to xml or xml, extended. It is also the location to which mandatory auditing information is written and, if so specified by the AUDIT_SYS_OPERATIONS initialization parameter, audit records for user SYS.

In a multitenant container database (CDB), the scope of the settings for this initialization parameter is the CDB. Although the audit trail is provided per pluggable database (PDB) in a CDB, this initialization parameter cannot be configured for individual PDBs.

See Also:

- Oracle Multitenant Administrator's Guide for conceptual information about CDBs and PDBs
- Oracle Multitenant Administrator's Guide for information about managing CDBs and PDBs
- "V\$CONTAINERS"
- "V\$PDBS"

2.25 AUDIT_SYS_OPERATIONS

AUDIT_SYS_OPERATIONS enables or disables the auditing of directly issued user SQL statements with SYS authorization. These include SQL statements directly issued by users when connected with the SYSASM, SYSBACKUP, SYSDBA, SYSDG, SYSKM, or SYSOPER privileges, as well as SQL statements that have been executed with SYS authorization using the PL/SQL package DBMS SYS SQL.

Note:

This parameter is deprecated and applies only to traditional auditing. Oracle strongly recommends against the use of this parameter. If you attempt to set this parameter, you will incur the following errors:

```
ORA-32004: obsolete or deprecated parameter(s) specified for string instance
ORA-32006: AUDIT_SYS_OPERATIONS initialization parameter has been deprecated
```

Traditional auditing is desupported starting in Oracle Database 23ai. Though traditional auditing is desupported, any current traditional audit settings that you have will still be honored. However, you cannot create new traditional audit settings. You can delete existing traditional audit settings. See *Oracle Database Security Guide* for more information about how this desupport works.

Property	Description
Parameter type	Boolean
Default value	FALSE
Modifiable	No



Property	Description
Modifiable in a PDB	No
Range of values	TRUE FALSE
Basic	No

The audit records are written to the operating system's audit trail. The audit records will be written in XML format if the AUDIT TRAIL initialization parameter is set to xml or xml, extended

On UNIX platforms, if the AUDIT_SYSLOG_LEVEL parameter has also been set, then it overrides the AUDIT_TRAIL parameter and SYS audit records are written to the system audit log using the SYSLOG utility.

In a CDB, the scope of the settings for this initialization parameter is the CDB. Although the audit trail is provided per PDB in a CDB, this initialization parameter cannot be configured for individual PDBs.

2.26 AUDIT SYSLOG LEVEL

AUDIT_SYSLOG_LEVEL allows SYS and standard OS audit records to be written to the system audit log using the SYSLOG utility.

Note:

This parameter is deprecated and applies only to traditional auditing. Oracle strongly recommends against the use of this parameter. If you attempt to set this parameter, you will incur the following errors:

```
ORA-32004: obsolete or deprecated parameter(s) specified for string instance
ORA-32006: AUDIT_SYSLOG_LEVEL initialization parameter has been deprecated
```

Traditional auditing is desupported starting in Oracle Database 23ai. Though traditional auditing is desupported, any current traditional audit settings that you have will still be honored. However, you cannot create new traditional audit settings. You can delete existing traditional audit settings. See *Oracle Database Security Guide* for more information about how this desupport works.

Property	Description
Parameter type	String
Syntax	AUDIT_SYSLOG_LEVEL = 'facility_clause.priority_clause'
Syntax	facility_clause::=
	{ USER LOCAL[0 1 2 3 4 5 6 7] SYSLOG DAEMON KERN MAIL AUTH LPR NEWS UUCP CRON }
Syntax	<pre>priority_clause::=</pre>
	{ NOTICE INFO DEBUG WARNING ERR CRIT ALERT EMERG }
Default value	There is no default value.



Property	Description
Modifiable	No
Modifiable in a PDB	No
Basic	No
Examples	AUDIT_SYSLOG_LEVEL = 'KERN.EMERG'; AUDIT_SYSLOG_LEVEL = 'LOCAL1.WARNING';

If you use this parameter, it is best to assign a file corresponding to every combination of facility and priority (especially KERN.EMERG) in syslog.conf. Sometimes these are assigned to print to the console in the default syslog.conf file. This can become annoying and will be useless as audit logs. Also, if you use this parameter, it is best to set the maximum length of syslog messages in the system to 512 bytes.

Note:

Audit records written to the system audit log could get truncated to 512 bytes, and different parts of the same audit record may not be joined to get the original complete audit record.

See Also:

Oracle Database Security Guide for information about configuring syslog auditing

If AUDIT_SYSLOG_LEVEL is set and SYS auditing is enabled (AUDIT_SYS_OPERATIONS = TRUE), then SYS audit records are written to the system audit log. If AUDIT_SYSLOG_LEVEL is set and standard audit records are being sent to the operating system (AUDIT_TRAIL = \circ s), then standard audit records are written to the system audit log.

In a CDB, the scope of the settings for this initialization parameter is the CDB. Although the audit trail is provided per PDB in a CDB, this initialization parameter cannot be configured for individual PDBs.



2.27 AUDIT_TRAIL

AUDIT TRAIL enables or disables database auditing.



This parameter is deprecated and applies only to traditional auditing. Oracle strongly recommends against the use of this parameter. If you attempt to set this parameter, you will incur the following errors:

```
ORA-32004: obsolete or deprecated parameter(s) specified for string instance
ORA-32006: AUDIT TRAIL initialization parameter has been deprecated
```

Traditional auditing is desupported starting in Oracle Database 23ai. Though traditional auditing is desupported, any current traditional audit settings that you have will still be honored. However, you cannot create new traditional audit settings. You can delete existing traditional audit settings. See *Oracle Database Security Guide* for more information about how this desupport works.

Property	Description
Parameter type	String
Syntax	AUDIT_TRAIL = { none os db [, extended] xml [, extended] }
Default value	none
Modifiable	No
Modifiable in a PDB	No
Basic	No

Values

none

Disables standard auditing. This value is the default if the AUDIT_TRAIL parameter was not set in the initialization parameter file or if you created the database using a method other than Database Configuration Assistant. If you created the database using Database Configuration Assistant, then the default is db.

• 08

Directs all audit records to an operating system file. Oracle recommends that you use the os setting, particularly if you are using an ultra-secure database configuration.

• db

Directs audit records to the database audit trail (the SYS.AUD\$ table), except for records that are always written to the operating system audit trail. Use this setting for a general database for manageability.

If the database was started in read-only mode with ${\tt AUDIT_TRAIL}$ set to db, then Oracle Database internally sets ${\tt AUDIT_TRAIL}$ to os. Check the alert log for details.

db, extended

Performs all actions of AUDIT_TRAIL=db, and also populates the SQL bind and SQL text CLOB-type columns of the SYS.AUD\$ table, when available. These two columns are populated only when this parameter is specified. When standard auditing is used with DB, EXTENDED, then virtual private database (VPD) predicates and policy names are also populated in the SYS.AUD\$ table.

If the database was started in read-only mode with AUDIT_TRAIL set to db, extended, then Oracle Database internally sets AUDIT TRAIL to os. Check the alert log for details.

xml

Writes to the operating system audit record file in XML format. Records all elements of the AuditRecord node except Sql Text and Sql Bind to the operating system XML audit file.

xml, extended

Performs all actions of AUDIT_TRAIL=xml, and includes SQL text and SQL bind information in the audit trail.

You can use the SQL AUDIT statement to set auditing options regardless of the setting of this parameter.

In a CDB, the scope of the settings for this initialization parameter is the CDB. Although the audit trail is provided per PDB in a CDB, this initialization parameter cannot be configured for individual PDBs.

Examples

The following statement sets the db, extended value for the AUDIT_TRAIL parameter. The new value takes effect after the database is restarted.

```
SQL> alter system set AUDIT_TRAIL=db, extended scope=spfile;
System altered.
```

The following statement sets the xml, extended value for the AUDIT_TRAIL parameter. The new value takes effect after the database is restarted.

```
SQL> alter system set AUDIT_TRAIL=xml, extended scope=spfile;
System altered.
```

The following statement sets the db value for the AUDIT_TRAIL parameter. The new value takes effect after the database is restarted.

```
SQL> alter system set AUDIT_TRAIL=db scope=spfile;
System altered.
```



See Also:

- Oracle Database Security Guide for information about configuring unified audit policies
- Oracle Database Upgrade Guide to learn more about traditional non-unified auditing

2.28 AUTOTASK_MAX_ACTIVE_PDBS

AUTOTASK_MAX_ACTIVE_PDBS enables you to specify the maximum number of PDBs that can schedule automated maintenance tasks at the same time (during a maintenance window).

Property	Description
Parameter type	Integer
Default value	2
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 to the number of PDBs in the CDB
Basic	No
Oracle RAC	The same value should be used on all instances.

This parameter only affects PDBs. The CDB\$ROOT container (CDB root) for a CDB can always schedule and run maintenance tasks during a maintenance window.

The default value is 2. Therefore, by default, two PDBs and the CDB root can run tasks at the same time during a maintenance window.

This parameter can be set only in the CDB root, not in a PDB.

See Also:

- "ENABLE_AUTOMATIC_MAINTENANCE_PDB" for information on disabling or enabling the running of automated maintenance tasks for specific PDBs or for all the PDBs in CDB
- Oracle Database Administrator's Guide for more information about managing automated database maintenance tasks

2.29 AWR_PDB_AUTOFLUSH_ENABLED

AWR_PDB_AUTOFLUSH_ENABLED enables you to specify whether to enable or disable automatic Automatic Workload Repository (AWR) snapshots for all the PDBs in a CDB or for individual PDBs in a CDB.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	The same value must be used on all instances.



The value of this parameter in CDB\$ROOT (the root of a CDB) has no effect in the root. Automatic AWR snapshots are always enabled in the root, regardless of the setting of this parameter.

The default value of AWR_PDB_AUTOFLUSH_ENABLED is true. Thus, by default, automatic AWR snapshots are enabled for all the PDBs in a CDB.

When you change the value of AWR_PDB_AUTOFLUSH_ENABLED in the CDB root, the new value takes effect in all the PDBs in the CDB.

Therefore, if you change the value of AWR_PDB_AUTOFLUSH_ENABLED in the CDB root to false, the value of AWR_PDB_AUTOFLUSH_ENABLED is also changed to false in all of the PDBs, so that automatic AWR snapshots are disabled for all the PDBs.

You can also change the value of AWR_PDB_AUTOFLUSH_ENABLED in any of the individual PDBs in a CDB, and the value that is set for each individual PDB will be honored. This enables you to enable or disable automatic AWR snapshots for individual PDBs.

When a new PDB is created, or a PDB from a previous database release is upgraded to the current database release, automatic AWR snapshots are enabled or disabled for the PDB based on the current value of AWR PDB AUTOFLUSH ENABLED in the root.

See Also:

- Oracle Database Performance Tuning Guide for more information about Automatic Workload Repository (AWR)
- Oracle Database Performance Tuning Guide for more information about AWR snapshots



2.30 AWR_PDB_MAX_PARALLEL_SLAVES

AWR_PDB_MAX_PARALLEL_SLAVES enables a DBA to allocate the correct amount of resources to enable quick and timely Automatic Workload Repository (AWR) flushes for multitenant container databases (CDBs).

Property	Description
Parameter type	Integer
Default value	10
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	1 – 30
Basic	No
Oracle RAC	Different instances can use different values.

Use AWR_PDB_MAX_PARALLEL_SLAVES to control the amount of resources dedicated to AWR snapshot flushing in pluggable databases (PDBs). With this parameter, a DBA can modify the maximum number of MMON worker processes (Mnnn background processes) that can concurrently be used to handle AWR flush operations for the entire CDB.

For CDBs with a large number of PDBs enabled to create automatic AWR snapshots, a DBA can increase this parameter to enable timely AWR flushing. For CDBs with a small number of PDBs enabled, the value of this parameter can be decreased to reduce the concurrency and the chance of performance spikes.

Note that this parameter is set in the root of a CDB and determines the maximum degree of parallelism used to create AWR automatic snapshots for PDBs.

See Also:

- "AWR_PDB_AUTOFLUSH_ENABLED" for information about enabling the automatic creation of AWR PDB snapshots
- Oracle Database Performance Tuning Guide for an introduction to AWR
- "Background Processes" for more information about the MMON and Mnnn background processes

2.31 AWR_SNAPSHOT_TIME_OFFSET

AWR_SNAPSHOT_TIME_OFFSET specifies an offset for the Automatic Workload Repository (AWR) snapshot start time.

Property	Description
Parameter type	Integer
Default value	There is no offset by default.



Property	Description
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 - 3599, or the special value 1000000
Basic	No
Oracle RAC	Multiple instances should use the same value

AWR snapshots normally start at the top of the hour (12:00, 1:00, 2:00, and so on). This parameter allows DBAs to specify an offset for the AWR snapshot start time.

This is a useful parameter to avoid CPU spikes from multiple instances all starting their AWR snapshots at the same time. If you have a large system with many instances on it (like many Exadata installations), and you are experiencing such CPU spikes, this parameter can be very useful.

The parameter is specified in seconds. Normally, you set it to a value less than 3600. If you set the special value 1000000 (1,000,000), you get an automatic mode, in which the offset is based on the database name.

The automatic mode is an effective way of getting a reasonable distribution of offset times when you have a very large number of instances running on the same node.

2.32 BACKGROUND CORE DUMP

BACKGROUND_CORE_DUMP specifies whether Oracle includes the SGA in the core file for Oracle background processes.

Property	Description
Parameter type	String
Syntax	BACKGROUND_CORE_DUMP = { partial full }
Default value	partial
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

Values

partial

Oracle does not include the SGA in the core dump.

full

Oracle includes the SGA in the core dump.



2.33 BACKGROUND_DUMP_DEST

BACKGROUND_DUMP_DEST specifies the pathname (directory or disc) where debugging trace files for the background processes (LGWR, DBWn, and so on) are written during Oracle operations.

Property	Description
Parameter type	String
Syntax	BACKGROUND_DUMP_DEST = { pathname directory }
Default value	Operating system-dependent
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Any valid local path, directory, or disk
Basic	No

Note:

The <code>BACKGROUND_DUMP_DEST</code> parameter is deprecated in Oracle Database 12c Release 1 (12.1.0.1).

An **alert log** in the directory specified by <code>BACKGROUND_DUMP_DEST</code> logs significant database events and messages. Anything that affects the database instance or global database is recorded here. The alert log is a normal text file. Its file name is operating system-dependent. For platforms that support multiple instances, it takes the form <code>alert_sid.log</code>, where <code>sid</code> is the system identifier. This file grows slowly, but without limit, so you might want to delete it periodically. You can delete the file even when the database is running.

Note:

This parameter is ignored by the diagnosability infrastructure introduced in Oracle Database 11g Release 1 (11.1), which places trace and core files in a location controlled by the <code>DIAGNOSTIC</code> <code>DEST</code> initialization parameter.

See Also:

- Oracle Database Administrator's Guide for more information on the DIAGNOSTIC_DEST initialization parameter
- "USER_DUMP_DEST" for information on setting a destination for server process trace files



2.34 BACKUP_TAPE_IO_SLAVES

BACKUP_TAPE_IO_SLAVES specifies whether I/O server processes are used by Recovery Manager to back up, copy, or restore data to tape.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM DEFERRED
Modifiable in a PDB	No
Range of values	true false
Basic	No

When the value is set to true, Oracle uses an I/O server process to write to or read from a tape device. When the value is set to false (the default), Oracle does not use I/O server process for backups. Instead, the shadow process engaged in the backup accesses the tape device.



You cannot perform duplexed backups unless you enable this parameter. Otherwise, Oracle returns an error. When this parameter is enabled, Recovery Manager will configure as many server processes as needed for the number of backup copies requested.

See Also:

- Oracle Database Backup and Recovery User's Guide for more information on duplexed backups
- "DBWR IO SLAVES"

2.35 BITMAP MERGE AREA SIZE

 $\verb|BITMAP_MERGE_AREA_SIZE| specifies the amount of memory Oracle uses to merge bitmaps retrieved from a range scan of the index.$

Property	Description
Parameter type	Integer
Default value	1048576 (1 MB)
Modifiable	No
Modifiable in a PDB	Yes



Property	Description
Range of values	Operating system-dependent
Basic	No

Note:

Oracle does not recommend using the <code>BITMAP_MERGE_AREA_SIZE</code> parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting <code>PGA_AGGREGATE_TARGET</code> instead. <code>BITMAP_MERGE_AREA_SIZE</code> is retained for backward compatibility.

BITMAP_MERGE_AREA_SIZE is relevant only for systems containing bitmap indexes. A larger value usually improves performance, because the bitmap segments must be sorted before being merged into a single bitmap.

See Also:

Oracle Database SQL Tuning Guide for more information on using bitmap indexes for performance

2.36 BLANK TRIMMING

BLANK TRIMMING specifies the data assignment semantics of character datatypes.

Droporty	Description
Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

Values

TRUE

Allows the data assignment of a source character string or variable to a destination character column or variable even though the source length is longer than the destination length. In this case, however, the additional length over the destination length must be all blanks, else an exception condition is raised. This value complies with the semantics of SQL-92 Transitional Level and above.

• FALSE

Disallows the data assignment if the source length is longer than the destination length and reverts to SQL92 Entry Level semantics.



Oracle Database Globalization Support Guide for more information on how using this parameter can help prevent data truncation issues during character set migration

2.37 BLOCKCHAIN TABLE MAX NO DROP

BLOCKCHAIN_TABLE_MAX_NO_DROP lets you control the maximum amount of idle time that can be specified when creating a blockchain table or an immutable table. The idle time represents the length of time the table must be inactive before it can be dropped. When explicitly set by a user, the only value allowed for this parameter is 0.

Property	Description
Parameter type	Integer
Default value	None
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	None or 0
Basic	No
Oracle RAC	The same value must be used on all instances.

When creating a blockchain table or an immutable table with the SQL statement CREATE TABLE, you can specify the clause NO DROP UNTIL number DAYS IDLE. This clause specifies the number of days the table must be inactive before it can be dropped, that is, the number of days that must pass after the most recent row insertion before the table can be dropped.

• If BLOCKCHAIN_TABLE_MAX_NO_DROP is set to 0, then when a user creates a blockchain table or an immutable table, and specifies the NO DROP UNTIL number DAYS IDLE clause, the value of number must be 0:

```
CREATE BLOCKCHAIN TABLE ... NO DROP UNTIL 0 DAYS IDLE ... CREATE IMMUTABLE TABLE ... NO DROP UNTIL 0 DAYS IDLE ...
```

This allows the table to be dropped at any time, regardless of how long the table has been inactive.

• If BLOCKCHAIN_TABLE_MAX_NO_DROP is not set, then when a user creates a blockchain table or an immutable table, and specifies the NO DROP UNTIL number DAYS IDLE clause, any number value can be specified for number.

See Also:

Oracle Database SQL Language Reference for information about the CREATE TABLE statement



2.38 BLOCKCHAIN_TABLE_RETENTION_THRESHOLD

BLOCKCHAIN_TABLE_RETENTION_THRESHOLD lets you control the maximum amount of idle time that can be specified when a database user without the TABLE RETENTION system privilege creates or alters a blockchain table or an immutable table.

Property	Description
Parameter type	Integer
Default value	16
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 365,000
Basic	No
Oracle RAC	The same value must be used on all instances.

The idle time for a non-empty blockchain table or a non-empty immutable table represents the length of time (in days) the table must be inactive before it can be dropped.



This parameter is available starting with Oracle Database 23ai.

2.39 CIRCUITS

CIRCUITS specifies the total number of virtual circuits that are available for inbound and outbound network sessions.

Property	Description
Parameter type	Integer
Default value	4294967295
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

It is one of several parameters that contribute to the total SGA requirements of an instance.

You should not specify a value for this parameter unless you want to limit the number of virtual circuits.



See Also:

- Oracle Database Concepts for more information on memory structures
- Oracle Database Concepts for more information on processes

2.40 CLIENT_PREFETCH_ROWS

Use CLIENT_PREFETCH_ROWS to enable clients to reduce the number of roundtrips required while fetching rows of a query result set.

Property	Description
Parameter type	Integer
Default value	0 (only client-side settings apply)
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to UB4MAXVAL (usually 4294967295)
	UB4MAXVAL is defined in the oratypes.h header file, which is found in the public directory. Its value may vary according to the operating system you are using.
Basic	No

CLIENT_PREFETCH_ROWS specifies the number of rows to be prefetched by the Oracle client driver, without making any changes to the client application. The client driver buffers the prefetched rows after each successful query execution and for each subsequent fetch request sent to the database.

This parameter applies only to clients that use Oracle Call Interface (OCI) to connect to the database.

This parameter applies only with Oracle Instant Client/Oracle Database Client 19.17 (or later) and 21.8 (or later), for all platforms.

The CLIENT_PREFETCH_ROWS parameter can be set with ALTER SESSION or ALTER SYSTEM. If the parameter value changes using ALTER SESSION, the new value becomes effective for that specific session on subsequent resultset fetches. If the parameter value changes using ALTER SYSTEM, the new value takes effect for the statements that run on connections created after the ALTER SYSTEM command.

For example, if <code>CLIENT_PREFETCH_ROWS</code> is set to 100 and a client application asks to fetch 10 rows, a total of 110 rows are returned to the client driver. The first 10 rows out of the 110 rows are given to the application, and the client driver internally buffers the remaining 100 rows. The next 10 row fetches from the client application, each with 10 rows per fetch iteration can be fulfilled from the 100 rows that are internally buffered by the client driver. This process reduces the number of required network roundtrips to and from the database. In this example, on the 11th fetch, a new network roundtrip is incurred and the database returns the next batch of 110 rows, as long as the result set is not exhausted, and the cycle repeats.

Notes for setting CLIENT_PREFETCH_ROWS:



- When CLIENT_PREFETCH_ROWS is set to a non-zero value, its value takes precedence over the default OCI_ATTR_PREFETCH_ROWS value for prefetch row count.
- If the OCI_ATTR_PREFETCH_ROWS value is set to a non-default value, then the CLIENT PREFETCH ROWS value is ignored for the prefetch row count.
- Using CLIENT PREFETCH ROWS with OCIAttrSet():

OCI_ATTR_PREFETCH_ROWS sets the number of top-level rows to be prefetched. The default value is 1 row. However, if <code>CLIENT_PREFETCH_ROWS</code> is set, the number of top-level rows to be prefetched is determined by the following precedence

- If you set the OCI_ATTR_PREFETCH_ROWS attribute using OCIAttrSet() function or
 oraaccess.xml as the value '1', then the database initialization parameter
 CLIENT_PREFETCH_ROWS value takes precedence and determines the number of top-level rows to be prefetched.
- 2. If you set the OCI_ATTR_PREFETCH_ROWS attribute using OCIAttrSet() function or oraccess.xml as the value 'x' other than 1, then 'x' number of top-level rows will be prefetched, and the database initialization parameter CLIENT_PREFETCH_ROWS is ignored.
- 3. If you do not set an OCI_ATTR_PREFETCH_ROWS value using OCIAttrSet() or oraccess.xml, then the database initialization parameter CLIENT_PREFETCH_ROWS value takes precedence and determines the number of top-level rows to be prefetched.
- Using CLIENT PREFETCH ROWS with OCIAttrGet():

The function <code>OCIAttrGet()</code> returns the effective prefetch row value set from <code>OCI_ATTR_PREFETCH_ROWS</code>, <code>oraaccess.xml</code> and the database initialization parameter <code>CLIENT_PREFETCH_ROWS</code>. If the <code>OCI_ATTR_PREFETCH_MEMORY</code> value is set, the value returned by <code>OCIAttrGet()</code> might not be the final prefetch rows value and may be restricted to the maximum number of rows allowed by the memory value specified by the <code>OCI_ATTR_PREFETCH_MEMORY</code> attribute.

Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

Oracle Call Interface Developer's Guide for more information about fetching results and setting the prefetch count, as well as the <code>OCI_ATTR_PREFETCH_MEMORY</code> and <code>OCI_ATTR_PREFETCH_ROWS</code> attributes, and the <code>OCIAttrGet()</code> and <code>OCIAttrSet()</code> functions

2.41 CLIENT RESULT CACHE_LAG

CLIENT_RESULT_CACHE_LAG specifies the maximum time (in milliseconds) since the last round trip to the server, before which the OCI client query execute makes a round trip to get any database changes related to the queries cached on the client.

Property	Description
Parameter type	Big integer
Syntax	CLIENT_RESULT_CACHE_LAG = integer
Default value	3000
Modifiable	No
Modifiable in a PDB	Yes
Range of values	0 to operating system-dependent
Basic	No

See Also:

Oracle Call Interface Developer's Guide for more information about the client query cache feature

2.42 CLIENT_RESULT_CACHE_SIZE

CLIENT_RESULT_CACHE_SIZE specifies the maximum size of the client per-process result set cache (in bytes).

Property	Description
Parameter type	Big integer
Syntax	CLIENT_RESULT_CACHE_SIZE = integer [K M G]
Default value	0
Modifiable	No
Modifiable in a PDB	Yes
Range of values	0 to operating system-dependent
Basic	No

All OCI client processes inherit this maximum size. Setting a nonzero value enables the client query cache feature. This can be overridden by the client configuration parameter OCI_RESULT_CACHE_MAX_SIZE.

See Also:

Oracle Call Interface Developer's Guide for more information about the client query cache feature

2.43 CLIENT_STATISTICS_LEVEL

CLIENT_STATISTICS_LEVEL controls whether database clients report network statistics to the database.

This parameter applies only to clients that use Oracle Call Interface (OCI) to connect to the database.

Property	Description
Parameter type	String
Syntax	CLIENT_STATISTICS_LEVEL = { TYPICAL OFF }
Default value	TYPICAL
Modifiable	ALTER SYSTEM DEFERRED
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value must be used on all instances.

Database clients sometimes have performance issues when interacting with the database over a network (LAN or WAN). You can use this parameter to enable clients to collect network statistics and periodically send them to the database, where the statistics are displayed in AWR and V\$ views. These statistics include TCP and SQL*Net statistics. They complement existing database statistics and can be used to analyze, troubleshoot, and tune potential network performance or configuration issues.

Possible values:

TYPICAL

This setting enables clients to collect network statistics and periodically send them to the database.

• OFF

This setting disables this feature. Clients will not send network statistics to the database.

When you modify this parameter, the new setting applies only to subsequent new connections to database clients.



Note:

To view a complete listing of the statistics that are collected and reported when this feature is enabled, issue the following SQL statement:

```
SELECT name FROM V$STATNAME
  WHERE name LIKE 'Client%' ORDER BY name;
NAME
Client Advertised Receive Window
Client Advertised Send Window
Client Data Segments In
Client Data Segments Out
Client Lost Packets
Client Path Maximum Transmission Unit (MTU)
Client Send Congestion Window
Client Time (usec) Busy Sending Data
Client Time (usec) Busy Sending Data under Congestion
Client Time (usec) Last Ack Received
Client Time (usec) Last Ack Sent
Client Time (usec) Last Data Received
Client Time (usec) Last Data Sent
Client Time (usec) Limited by Receive Window
Client Time (usec) Limited by Send Buffer
Client Time (usec) Round Trip Time
Client Time (usec) Round Trip Time Variance
Client Total Bytes Acked
Client Total Bytes Received
Client Total Number of Retransmitted Packets
```

2.44 CLONEDB

CLONEDB should be set on Direct NFS Client CloneDB databases. When this parameter is set, the CloneDB database uses the database backup as the backing store for the datafiles.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	The same value should be set for all instances.

See Also:

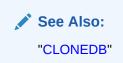
- See *Oracle Database Administrator's Guide* for more information about cloning databases on network attached storage (NAS).
- "CLONEDB_DIR"

2.45 CLONEDB_DIR

 ${\tt CLONEDB_DIR\ sets\ the\ directory\ path\ where\ CloneDB\ bitmap\ files\ should\ be\ created\ and\ accessed.}$

Property	Description
Parameter type	String
Syntax	CLONEDB_DIR = string
Default value	\$ORACLE_HOME/dbs
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	In an Oracle RAC environment, this parameter should be set to a shared location that is accessible from all the instances.

By default the CloneDB bitmap file is created under the <code>\$ORACLE_HOME/dbs</code> directory. This directory may not be in a shared location in an Oracle RAC environment, and therefore this parameter is provided to identify a shared location where CloneDB specific files can be created.



2.46 CLUSTER_DATABASE

CLUSTER DATABASE is an Oracle RAC parameter that specifies whether Oracle RAC is enabled.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	Yes
Oracle RAC	For all instances, the value must be set to true.





Oracle Real Application Clusters Administration and Deployment Guide for an introduction to Oracle RAC

2.47 CLUSTER_INTERCONNECTS

CLUSTER_INTERCONNECTS can be used in Oracle Real Application Clusters environments to indicate cluster interconnects available for use for the database traffic.

Property	Description
Parameter type	String
Syntax	CLUSTER_INTERCONNECTS = ifn [: ifn]
Default value	There is no default value.
Modifiable	No
Modifiable in a PDB	No
Range of values	One or more IP addresses, separated by colons
Basic	No

Use this parameter to override the default interconnect configured for the database traffic, which is stored in the cluster registry. This procedure also may be useful with Data Warehouse systems that have reduced availability requirements and high interconnect bandwidth demands.

CLUSTER INTERCONNECTS specifically overrides the following:

- Network classifications stored by oifcfg in the OCR.
- The default interconnect chosen by Oracle.

If you want to load-balance the interconnect, then Oracle recommends that you use link-bonding at the operating system level, even if you have two databases on the same server, so that multiple interconnects use the same address. Note that multiple private addresses provide load balancing, but do not provide failover unless bonded. If you specify multiple addresses in init.ora using <code>CLUSTER_INTERCONNECTS</code>, instead of bonding multiple addresses at the operating system level, then typically availability is reduced, because each network interface card failure will take down that instance.

Refer to your vendor documentation for information about bonding interfaces. Some vendor bonding architectures may require the use of this parameter.

If you have multiple database instances on Oracle Real Application Clusters nodes and want to use a specific interface for each instance, then you can set the <code>CLUSTER_INTERCONNECTS</code> initialization parameter to the IP address for each database instance. For example:

```
hr1.init.ora.cluster_interconnects="192.0.2.111"
oltp3.init.ora.cluster_interconnects="192.0.2.112"
```

If the Oracle RAC interconnect is configured to run on a different interface than the Oracle Clusterware interconnect, then this configuration can cause reduced availability, as failovers or instance evictions can be delayed if the Oracle RAC interconnect fails while the Oracle Clusterware NIC remains up.



See Also:

Oracle Real Application Clusters Administration and Deployment Guide for additional information about using CLUSTER INTERCONNECTS

2.48 COMMIT_LOGGING

COMMIT LOGGING is an advanced parameter used to control how redo is batched by Log Writer.

Property	Description
Parameter type	String
Syntax	COMMIT_LOGGING = { IMMEDIATE BATCH }
Default value	There is no default value.
Modifiable	Yes (at both session-level and system-level)
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Each instance may have its own setting

If COMMIT_LOGGING is altered after setting COMMIT_WAIT to FORCE_WAIT, then the FORCE_WAIT option is no longer valid.

2.49 COMMIT_POINT_STRENGTH

COMMIT_POINT_STRENGTH specifies a value that determines the **commit point site** in a distributed transaction.

Property	Description
Parameter type	Integer
Default value	1
Modifiable	No
Modifiable in a PDB	Yes
Range of values	0 to 255
Basic	No

Note:

This parameter is relevant only in distributed database systems.

The node in the transaction with the highest value for <code>COMMIT_POINT_STRENGTH</code> will be the commit point site.

The commit point site of a database should reflect the amount of critical shared data in the database. For example, a database on a mainframe computer typically shares more data

among users than one on a personal computer. Therefore, <code>COMMIT_POINT_STRENGTH</code> should be set to a higher value for the mainframe computer.

The commit point site stores information about the status of transactions. Other computers in a distributed transaction require this information during Oracle's two-phase commit, so it is desirable to have machines that are always available as commit point sites. Therefore, set COMMIT_POINT_STRENGTH to a higher value on your more available machines.

See Also:

- Oracle Database Concepts and Oracle Database Administrator's Guide for information on two-phase commit
- Your operating system-specific Oracle documentation for the default value of this parameter

2.50 COMMIT_WAIT

COMMIT_WAIT is an advanced parameter used to control when the redo for a commit is flushed to the redo logs.

Property	Description
Parameter type	String
Syntax	COMMIT_WAIT = { NOWAIT WAIT FORCE_WAIT }
Default value	There is no default value.
Modifiable	Yes (at both session-level and system-level)
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Each instance may have its own setting

Be aware that the NOWAIT option can cause a failure that occurs after the database receives the commit message, but before the redo log records are written. This can falsely indicate to a transaction that its changes are persistent. Also, it can violate the durability of ACID (Atomicity, Consistency, Isolation, Durability) transactions if the database shuts down unexpectedly.

If the parameter is set to <code>FORCE_WAIT</code>, the default behavior (immediate flushing of the redo log buffer with wait) is used. If this is a system setting, the session level and transaction level options will be ignored. If this is a session level setting, the transaction level options will be ignored. If <code>COMMIT_WAIT</code> is altered after it has been set to <code>FORCE_WAIT</code>, then the <code>FORCE_WAIT</code> option is no longer valid.

2.51 COMMIT_WRITE

COMMIT_WRITE is an advanced parameter used to control how redo for transaction commits is written to the redo logs.

Property	Description
Parameter type	String



Property	Description
Syntax	COMMIT_WRITE = '{IMMEDIATE BATCH}, {WAIT NOWAIT}'
Default value	If this parameter is not explicitly specified, then database commit behavior defaults to writing commit records to disk before control is returned to the client.
	If only IMMEDIATE or BATCH is specified, but not WAIT or NOWAIT, then WAIT mode is assumed.
	If only WAIT or NOWAIT is specified, but not IMMEDIATE or BATCH, then IMMEDIATE mode is assumed.
	Be aware that the NOWAIT option can cause a failure that occurs after the database receives the commit message, but before the redo log records are written. This can falsely indicate to a transaction that its changes are persistent. Also, it can violate the durability of ACID (Atomicity, Consistency, Isolation, Durability) transactions if the database shuts down unexpectedly.
Modifiable	Yes (at both session-level and system-level). Values supplied for COMMIT_WRITE in an ALTER SYSTEM or ALTER SESSION statement must be separated by a comma.
Modifiable in a PDB	Yes
Range of values	Single-quoted, comma-separated list of either IMMEDIATE or BATCH, and either WAIT or NOWAIT.
Basic	No
Oracle RAC	Each instance may have its own setting

The IMMEDIATE and BATCH options control how redo is batched by Log Writer. The WAIT and NOWAIT options control when the redo for a commit is flushed to the redo logs.



The ${\tt COMMIT_WRITE}$ parameter is deprecated. It is retained for backward compatibility only. It is replaced by the ${\tt COMMIT_LOGGING}$ and ${\tt COMMIT_WAIT}$ parameters.

2.52 COMMON_USER_PREFIX

COMMON_USER_PREFIX specifies a prefix that the names of common users, roles, and profiles in a multitenant container database (CDB) must start with.

Property	Description
Parameter type	String
Syntax	COMMON_USER_PREFIX = prefix
Default value	In a CDB root, $\mathbb{C}\#\#$ is the default value. In an application root, the empty string is the default value.
Modifiable	No
Modifiable in a PDB	No
	However, you can modify this parameter in an application root.
Basic	No

Property	Description
Oracle RAC	Multiple instances must have the same value

Names of local users, roles, and profiles must not start with the same prefix. A case-insensitive comparison of the prefixes for user, role, and profile names to the value of this parameter is done.

When a value is set for <code>COMMON_USER_PREFIX</code>, Oracle will require that the names of common users, roles, and profiles start with the string assigned to this parameter.

The names of users, roles, and profiles created using this prefix must be valid names.

Regardless of the value of the COMMON_USER_PREFIX parameter, the names of local users, roles, and profiles cannot start with C##. Similarly, the names of application common users and local users in an application PDB cannot start with C##.



Caution:

You can change the value of the <code>COMMON_USER_PREFIX</code> parameter, but do so only with great care.

If COMMON_USER_PREFIX is set to an empty string, Oracle will not enforce any restrictions on the names of common or local users, roles, and profiles.

Setting this parameter to an empty string will result in no restrictions being placed on names of common and local users, roles, and profiles, which could lead to conflicts between the names of local and common users, roles, and profiles when a PDB is plugged into a different CDB, or when opening a PDB that was closed when a common user was created.

Note:

COMMON_USER_PREFIX can be set in the context of an application container.

If you set <code>COMMON_USER_PREFIX</code> in the application root to a non-null value, then the application common user should start with that prefix, and a local user created in an application PDB cannot start with that prefix.

By default, COMMON USER PREFIX is the empty string in an application root.

Note:

When you query the V\$PARAMETER or V\$SYSTEM_PARAMETER view, a value of NONE is returned for the DEFAULT VALUE column for the COMMON USER PREFIX parameter.

However, the database enforces a default value of C## for a CDB root and a default value of the empty string for an application root.



See Also:

- Oracle Database SQL Language Reference for information about valid user names
- Oracle Database SQL Language Reference for information about valid role names
- Oracle Database SQL Language Reference for information about valid profile names
- Oracle Multitenant Administrator's Guide for more information about the multitenant architecture

2.53 COMPATIBLE

COMPATIBLE enables you to use a new release of Oracle while ensuring the ability to downgrade the database to an earlier release.

Property	Description
Parameter type	String
Syntax	COMPATIBLE = release_number
Default value	Oracle Database 23ai (initial release): 23.4.0
	Oracle Database 23ai, Release Update 23.5: 23.5.0
	Oracle Database 23ai, Release Updates 23.6 and later: 23.6.0
Modifiable	No
Modifiable in a PDB	No
Range of values	19.0.0 to default release
	Values must be specified as at least three decimal numbers with each number separated by a dot, such as 23.4.0 or 23.4.0.0
Basic	Yes
Oracle RAC	Multiple instances must have the same value.

The COMPATIBLE parameter specifies the Oracle version number that the database disk format must be compatible with. The database can be downgraded to the version specified in the COMPATIBLE parameter or any later version.

Setting COMPATIBLE ensures that new features do not write data formats or structures to disk that are not compatible with the earlier release, preventing a future downgrade. Features that require a higher value of COMPATIBLE to work correctly may be restricted or disabled to ensure downgrades are possible.

Typically, users keep COMPATIBLE unchanged when upgrading their Oracle software. After upgrade, users will run the new release of the Oracle software for a few weeks to ensure that the new release is working correctly. Afterwards, users can choose to update COMPATIBLE to the latest version to take advantage of the new features.

The following Release Update guidelines apply:

For Oracle Database 23ai:



The value of the COMPATIBLE parameter should be changed for an RU only under the following circumstance: You are running the Oracle Database 23ai initial release and you have *explicitly* set COMPATIBLE to any value (including the default value of 23.4.0). You then apply a Release Update. In this case:

- If you apply Release Update 23.5, then you must set the value of COMPATIBLE to 23.5.0 in order to use all of the features of Oracle Database 23ai, including the features introduced in 23.5.
- If you apply Release Update 23.6 or later, then you must set the value of COMPATIBLE to 23.6.0 in order to use all of the features of Oracle Database 23ai, including the features introduced in 23.6, such as AI Vector Search features.

The value of the COMPATIBLE parameter should not be changed for an RUR in Oracle Database 23ai.

For Oracle Database 21c, Oracle Database 19c, and Oracle Database 18c:

The value of the COMPATIBLE parameter should not be changed for a Release Update (RU) or Release Update Revision (RUR). For example, assume you are running Oracle Database 21c and the value of COMPATIBLE is 21.0.0. You then apply Oracle Database 21c, Release Update 21.4. Do not set the value of COMPATIBLE to 21.4.0; leave it set to 21.0.0.

Notes:

- The value of the COMPATIBLE parameter can be increased to a higher version, but it can never be decreased to a lower version.
- When using a standby database, this parameter must have the same value on both the primary and standby databases.

See Also:

Oracle Database Upgrade Guide for more information on setting this parameter

2.54 CONNECTION_BROKERS

CONNECTION_BROKERS is used to specify connection broker types, the number of connection brokers of each type, and the maximum number of connections per broker.

Property	Description
Parameter type	String
Syntax	<pre>CONNECTION_BROKERS = broker_description[,]</pre>
Syntax	broker_description::=
	((type_clause)(brokers_clause)[(broker_options)])



Property	Description
Syntax	type_clause::=
	TYPE={ DEDICATED EMON POOLED }
	Indicates the broker type. This is a mandatory parameter.
Syntax	brokers_clause::=
	BROKERS=integer
	Indicates the number of brokers of the specified type. This is a mandatory parameter. The specified integer value must be greater than or equal to 0. If set to 0, brokers will not accept new connections, but existing connections will continue as normal.
Syntax	broker_options::=
	CONNECTIONS=integer
	The maximum number of connections per broker. This is an optional parameter. If the specified integer value is greater than the maximum allowed for the platform, the maximum allowed for the platform is used.
Default value	The default value for the initialization parameter will contain the <code>DEDICATED</code> and <code>EMON</code> broker types. Certain types of brokers are only started when certain features are enabled, so their configuration will only apply if the broker is needed.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	The same value should be set on all instances

The connection brokers are used in these cases:

- DEDICATED brokers are used by the multiprocess and multithreaded Oracle features. See Oracle Database Concepts for more information about these features.
 - DEDICATED brokers are also used when the USE_DEDICATED_BROKER initialization parameter is set to true. See "USE_DEDICATED_BROKER" for more information about the USE_DEDICATED_BROKER initialization parameter.
- EMON brokers are used when there is a client registered for notification over 12c AQ queues. It is not on by default. See Oracle Database Advanced Queuing User's Guide for more information about AQ queues.
- POOLED brokers are used whenever a Database Resident Connection Pooling (DRCP) pool is active. See Oracle Database Concepts for more information about DRCP pooling.
 - The POOLED broker does not appear in the parameter by default. In this case, the DRCP fields are used. If you specify the POOLED broker in the parameter, then that specification will override the DRCP fields, and further attempts to set the DRCP fields will then throw an error. Oracle recommends that the parameter be used instead of the DRCP fields.

To make it easier to modify the list, individual entries are indexed using the TYPE so it is not necessary to re-specify the entire list. For example, to adjust the number of DEDICATED brokers:

```
alter system set connection_brokers = '((TYPE=DEDICATED) (BROKERS=2))'
```



2.55 CONTAINER_DATA

Use CONTAINER_DATA to control whether queries on extended data-linked objects from within a PDB return rows from both the root and the PDB, or rows from only the PDB.

Property	Description
Parameter type	String
Syntax	CONTAINER_DATA = { ALL CURRENT CURRENT_DICTIONARY }
Default value	ALL
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different instances can use different values.

This parameter controls the behavior of queries on the following types of objects:

Extended data-linked application common objects

This type of object stores shared data in an application root, and also allows application PDBs to store local data that is unique to each application PDB.

When this type of object is queried from an application root, only the shared data in the application root is returned.

When this type of object is queried from an application PDB, the value of the CONTAINER DATA initialization parameter controls the query result as follows:

- ALL: The query returns both the shared data in the application root and the local data pertaining to the application PDB
- CURRENT: The query returns only the local data pertaining to the application PDB
- CURRENT_DICTIONARY: The query returns both the shared data in the application root and the local data pertaining to the application PDB
- Extended data-linked Oracle-supplied data dictionary objects

This type of object stores data pertaining to the CDB root, as well as data pertaining to individual PDBs.

When this type of object is queried from the CDB root, only data pertaining to the CDB root is returned.

When this type of object is queried from a PDB, the value of the ${\tt CONTAINER_DATA}$ initialization parameter controls the query result as follows:

- ALL: The query returns data pertaining to both the CDB root and the PDB
- CURRENT: The query returns only data pertaining to the PDB
- CURRENT DICTIONARY: The query returns only data pertaining to the PDB



2.56 CONTAINERS_PARALLEL_DEGREE

CONTAINERS_PARALLEL_DEGREE can be used to control the degree of parallelism of a query involving containers().

Property	Description
Parameter type	Integer
Default value	65535
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	Cannot be set to a value lower than 2 or higher than 65535.
Basic	No
Oracle RAC	Different values can be set on different instances.

The value of CONTAINERS_PARALLEL_DEGREE, if set, will override the default DOP for a containers() query.

By default, a containers() query uses a degree of parallelism equal to (1 + number of open PDBs) in the case of CDB root and (1 + number of open application PDBs) in the case of application root.

If the value of <code>CONTAINERS_PARALLEL_DEGREE</code> is lower than 65535, then this value is used as the degree of parallelism of a query involving containers(). Otherwise (when the value is 65535), the default degree of parallelism is (1 + number of open PDBs) or (1 + number of open application PDBs) as described above.

See Also:

- Oracle Multitenant Administrator's Guide for more information about CDB roots
- Oracle Multitenant Administrator's Guide for more information about application roots

2.57 CONTROL_FILE_RECORD_KEEP_TIME

CONTROL_FILE_RECORD_KEEP_TIME specifies the minimum number of days before a reusable record in the control file can be reused.

Description
Integer
7 (days)
ALTER SYSTEM
No
0 to 365 (days)
Ι Γ



Property	Description
Basic	No

In the event a new record must be added to a reusable section and the oldest record has not aged enough, the record section expands. If this parameter is set to 0, then reusable sections never expand, and records are reused as needed.

If the number of reusable records in the control file exceeds the circular reuse record limit UB4MAXVAL, then reusable records will be overwritten even if CONTROL_FILE_RECORD_KEEP_TIME has not elapsed. UB4MAXVAL is defined in the oratypes.h header file, which is found in the public directory. Its value may vary according to the operating system you are using.

Note:

This parameter applies only to records in the control file that are circularly reusable (such as archive log records and various backup records). It does not apply to records such as data file, tablespace, and redo thread records, which are never reused unless the corresponding object is dropped from the tablespace.

See Also:

- "CONTROL FILES"
- Oracle Database Backup and Recovery User's Guide for more information about this parameter

2.58 CONTROL FILES

CONTROL FILES specifies one or more names of control files, separated by commas.

Property	Description
Parameter type	String
Syntax	CONTROL_FILES = filename [, filename]
	Note: The control file name can be an OMF (Oracle Managed Files) name. This occurs when the control file is re-created using the CREATE CONTROLFILE REUSE statement.
Default value	Operating system-dependent
Modifiable	No
Modifiable in a PDB	No
Range of values	1 to 8 filenames
Basic	Yes
Oracle RAC	Multiple instances must have the same value.



Every database has a **control file**, which contains entries that describe the structure of the database (such as its name, the timestamp of its creation, and the names and locations of its data files and redo files).

Oracle recommends that you multiplex multiple control files on different devices or mirror the file at the operating system level.

See Also:

Oracle Database Administrator's Guide for information about specifying control files at database creation

2.59 CONTROL_MANAGEMENT_PACK_ACCESS

CONTROL_MANAGEMENT_PACK_ACCESS specifies which of the Server Manageability Packs should be active.

Property	Description
Parameter type	String
Syntax	<pre>CONTROL_MANAGEMENT_PACK_ACCESS = { NONE DIAGNOSTIC DIAGNOSTIC+TUNING }</pre>
Default value	Enterprise Edition: DIAGNOSTIC+TUNING
	Oracle Database Free: DIAGNOSTIC+TUNING
	All other editions: NONE
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

The following packs are available:

- The DIAGNOSTIC pack includes AWR, ADDM, and so on.
- The TUNING pack includes SQL Tuning Advisor, SQLAccess Advisor, and so on.

A license for DIAGNOSTIC is required for enabling the TUNING pack.

Values

NONE

Neither pack is available.

DIAGNOSTIC

Only the DIAGNOSTIC pack is available.

DIAGNOSTIC+TUNING

Both packs are available.



See Also:

Oracle Database Licensing Information User Manual

2.60 CORE_DUMP_DEST

CORE DUMP DEST specifies the directory where Oracle dumps core files.

Property	Description
Parameter type	String
Syntax	CORE_DUMP_DEST = directory
Default value	ORACLE_HOME/DBS
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

CORE DUMP DEST is primarily a UNIX parameter and may not be supported on your platform.

✓ Note:

This parameter is ignored by the new diagnosability infrastructure introduced in Oracle Database 11g Release 1 (11.1), which places trace and core files in a location controlled by the ${\tt DIAGNOSTIC_DEST}$ initialization parameter.

See Also:

"DIAGNOSTIC_DEST"

2.61 CPU_COUNT

CPU COUNT specifies the number of CPUs available for Oracle Database to use.

Property	Description
Parameter type	String
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to unlimited
Basic	No
Oracle RAC	The same value should be used on all instances.

On CPUs with multiple CPU threads, it specifies the total number of available CPU threads. Various components of Oracle Database are configured based on the number of CPUs, such as the Optimizer, Parallel Query, and Resource Manager.

If CPU_COUNT is set to 0 (its default setting), then Oracle Database continuously monitors the number of CPUs reported by the operating system and uses the current count. If CPU_COUNT is set to a value other than 0, then Oracle Database will use this count rather than the actual number of CPUs, thus disabling dynamic CPU reconfiguration.

When Resource Manager is managing CPU (RESOURCE_MANAGER_PLAN is set), then the database's CPU utilization is limited to CPU_COUNT CPU threads. This feature is called Instance Caging. If Resource Manager is enabled at the CDB level, then the PDB's CPU utilization is limited to the PDB's CPU COUNT.

Note:

Setting CPU_COUNT to a value greater than the current number of CPUs results in an error. However, if CPU_COUNT is set to a value greater than the current number of CPUs in the initialization parameter file, then CPU_COUNT is capped to the current number of CPUs.

Note:

When a value is not explicitly set for CPU_COUNT, the maximum default value for CPU_COUNT is 2 for an Oracle ASM Proxy instance, 4 for an Oracle ASM instance, and 8 for an Oracle IOServer instance.

See Also:

- "CPU MIN COUNT"
- Oracle Database VLDB and Partitioning Guide for information about how CPU_COUNT is used to determine the default degree of parallelism for a single instance or Oracle RAC configuration when the PARALLEL clause is specified but no degree of parallelism is listed
- Oracle Database Administrator's Guide for an example of how CPU resources are allocated if you enable instance caging and set a maximum utilization limit in a resource plan

2.62 CPU_MIN_COUNT

CPU_MIN_COUNT specifies the minimum number of CPUs required by a pluggable database (PDB) at any given time.

Property	Description
Parameter type	String



Property	Description
Syntax	CPU_MIN_COUNT = 'value'
Default value	The value of CPU_COUNT
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of Values	Decimal values from 0.1 through 0.95, where the decimal value is a multiple of 0.05
	Integer values from 1 up to and including the value of ${\tt CPU_COUNT}$
Basic	No
Oracle RAC	The same value should be used on all instances.

This parameter specifies the minimum number of CPUs required by a PDB at any given time. For multi-threaded CPUs, this number corresponds to CPU threads, not CPU cores.

You can set this parameter at the CDB level, and for each individual PDB. This enables you to control each PDBs minimum share of CPU utilization within a CDB. If the sum of the CPU_MIN_COUNT values across all open PDBs in a CDB is equal to the value of CPU_MIN_COUNT for the CDB, then the CDB instance is considered full. If the sum exceeds the value of CPU_MIN_COUNT for the CDB, then the CDB instance is over-provisioned. Oracle does not prevent you from over-provisioning a CDB.

Resource Manager is enabled at the CDB level by setting the RESOURCE_MANAGER_PLAN at the root level to the name of a CDB resource plan. If the CDB resource plan has no configured CPU directives, that is, the SHARES and UTILIZATION_LIMIT directives are unset, then Resource Manager uses the CPU_COUNT and CPU_MIN_COUNT settings for the PDB to manage CPU utilization.



Setting this parameter to a very low value can result in insufficient CPU allocation and poor performance.

See Also:

"CPU COUNT"

2.63 CREATE_BITMAP_AREA_SIZE

CREATE_BITMAP_AREA_SIZE specifies the amount of memory (in bytes) allocated for bitmap creation.

Property	Description
Parameter type	Integer
Default value	0



Property	Description
Modifiable	No
Modifiable in a PDB	Yes
Range of values	Operating system-dependent
Basic	No

This parameter is relevant only for systems containing bitmap indexes.



Oracle does not recommend using the CREATE_BITMAP_AREA_SIZE parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting PGA_AGGREGATE_TARGET instead. CREATE BITMAP AREA SIZE is retained for backward compatibility.

A larger value may speed up index creation.

Cardinality is the number of unique values in a column in relation to the number of rows in the table. If cardinality is very small, you can set a small value for this parameter. For example, if cardinality is only 2, then the value can be in kilobytes rather than megabytes. As a general rule, the higher the cardinality, the more memory is needed for optimal performance.



Oracle Database SQL Tuning Guide for more information on using bitmap indexes

2.64 CREATE STORED OUTLINES

CREATE_STORED_OUTLINES determines whether Oracle automatically creates and stores an outline for each query submitted during the session.

Property	Description
Parameter type	String
Syntax	<pre>CREATE_STORED_OUTLINES = { true false category_name } [NOOVERRIDE]</pre>
Default value	There is no default value.
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Values

true



Enables automatic outline creation for subsequent queries in the same session. These outlines receive a unique system-generated name and are stored in the DEFAULT category. If a particular query already has an outline defined for it in the DEFAULT category, then that outline will remain and a new outline will not be created.

false

Disables automatic outline creation during the session. This is the default.

category name

Enables the same behavior as true except that any outline created during the session is stored in the *category* name category.

NOOVERRIDE

NOOVERRIDE specifies that this system setting will not override the setting for any session in which this parameter was explicitly set. If you do not specify NOOVERRIDE, then this setting takes effect in all sessions.

See Also:

Oracle Database SQL Tuning Guide for more information on setting this parameter

2.65 CURSOR_BIND_CAPTURE_DESTINATION

CURSOR_BIND_CAPTURE_DESTINATION determines the location at which bind variables that are captured from SQL cursors are available.

Property	Description
Parameter type	String
Syntax	<pre>CURSOR_BIND_CAPTURE_DESTINATION = { off memory memory+disk }</pre>
Default value	memory+disk
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Values

off

Bind variables are not captured from SQL cursors.

memory

Bind variables are captured from SQL cursors, and are available only in memory (V\$ views).

memory+disk

Bind variables are captured from SQL cursors, and are available in memory (V\$ views) and disk (Automatic Workload Repository tables, SQL Tuning Set tables, and so on).

When you specify this value with a SQL ALTER SESSION or ALTER SYSTEM statement, enclose the value in single quotes or double quotes. For example:

```
ALTER SESSION SET CURSOR_BIND_CAPTURE_DESTINATION = 'memory+disk';
ALTER SYSTEM SET CURSOR BIND CAPTURE DESTINATION = "memory+disk";
```

2.66 CURSOR_INVALIDATION

CURSOR_INVALIDATION controls whether deferred cursor invalidation or immediate cursor invalidation is used for DDL statements by default.

Property	Description
Parameter type	String
Syntax	CURSOR_INVALIDATION = { DEFERRED IMMEDIATE }
Default value	IMMEDIATE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different instances can have different values.

Deferred invalidation reduces the number of cursor invalidations and spreads the recompilation workload over time. Note that when the recompilation workload is spread over time, a cursor may run with a sub-optimal plan until it is recompiled, and may incur small execution-time overhead.

Prior to Oracle Database 12c Release 2 (12.2.0.1), immediate cursor invalidation was used.

This parameter provides system or session level default for the DEFERRED or IMMEDIATE option for the INVALIDATION clause in DDL statements.

When this parameter is set to <code>DEFERRED</code>, an application can take advantage of reduced cursor invalidation without making any other application changes.

When this parameter is set to IMMEDIATE, the application will experience the same cursor invalidation behavior as in Oracle Database 12c Release 1 (12.1).

2.67 CURSOR_SHARING

CURSOR SHARING determines what kind of SQL statements can share the same cursors.

Property	Description
Parameter type	String
Syntax	CURSOR_SHARING = { EXACT FORCE }
Default value	EXACT
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No



Values

FORCE

Allows the creation of a new cursor if sharing an existing cursor, or if the cursor plan is not optimal.

EXACT

Only allows statements with identical text to share the same cursor.



Oracle Database SQL Tuning Guide before setting the <code>CURSOR_SHARING</code> parameter to learn about the performance implications

2.68 CURSOR_SPACE_FOR_TIME

CURSOR SPACE FOR TIME lets you use more space for cursors to save time.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No



The ${\tt CURSOR_SPACE_FOR_TIME}$ parameter is deprecated. It is retained for backward compatibility only.

This parameter affects both the shared SQL area and the client's private SQL area.

Most users will not need to set this parameter because of the significantly enhanced concurrency modifications introduced in Oracle Database 10g Release 2 (10.2.0.2) and later.

Values

true

Shared SQL areas are kept pinned in the shared pool. As a result, shared SQL areas are not aged out of the pool as long as an open cursor references them. Because each active cursor's SQL area is present in memory, execution is faster. However, the shared SQL areas never leave memory while they are in use. Therefore, you should set this parameter to true only when the shared pool is large enough to hold all open cursors simultaneously.



In addition, a setting of true retains the private SQL area allocated for each cursor between executions instead of discarding it after cursor execution, saving cursor allocation and initialization time.

false

Shared SQL areas can be deallocated from the library cache to make room for new SQL statements.



If this parameter is set to true, then the <code>SERIAL</code> REUSE parameter is disabled.

See Also:

"SERIAL_REUSE"

2.69 DATA_GUARD_MAX_IO_TIME

DATA_GUARD_MAX_IO_TIME sets the maximum number of seconds that can elapse before a process is considered hung while performing a regular I/O operation in an Oracle Data Guard environment. Regular I/O operations include read, write, and status operations.

See Also:

"DATA_GUARD_MAX_LONGIO_TIME," which enables you to set the timeout for long I/O operations, such as open and close operations, in an Oracle Data Guard environment

Property	Description
Parameter type	Integer
Default value	240
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	10 to 7200
Basic	No
Oracle RAC	Different instances can use different values.



2.70 DATA GUARD MAX LONGIO TIME

DATA GUARD MAX LONGIO TIME sets the maximum number of seconds that can elapse before a process is considered hung while performing a long I/O operation in an Oracle Data Guard environment. Long I/O operations include open and close operations.

Property	Description
Parameter type	Integer
Default value	240
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	10 to 7200
Basic	No
Oracle RAC	Different instances can use different values.

See Also:
"DATA_GUARD_MAX_IO_TIME"

2.71 DATA_GUARD_SYNC_LATENCY

DATA GUARD SYNC LATENCY controls how many seconds the Log Writer (LGWR) process waits beyond the response of the first in a series of Oracle Data Guard SYNC redo transport mode connections.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 to the number of seconds specified by the ${\tt NET_TIMEOUT}$ attribute for the ${\tt LOG_ARCHIVE_DEST_n}$ parameter
Basic	No
Oracle RAC	The same value must be used on all instances.

The default value is 0, which means that the LGWR will wait up to the number of seconds specified by the NET TIMEOUT attribute of the LOG ARCHIVE DEST n parameter for each SYNC standby destination.

For example, in an Oracle Data Guard configuration that has two standbys which receive redo in SYNC mode, you would define LOG ARCHIVE DEST 2 and LOG ARCHIVE DEST 3 to use SYNC transport with data guard sync latency set to a value of 2 seconds. When redo needs to be sent to the standbys, it is sent to both the LOG ARCHIVE DEST 2 and LOG ARCHIVE DEST 3 connections in parallel and the LGWR will wait for acknowledgement. If LOG ARCHIVE DEST 2

responds with its message acknowledging that the redo has been received first, the LGWR will not wait for LOG_ARCHIVE_DEST_3 to respond for more than 2 extra seconds (DATA_GUARD_SYNC_LATENCY). If LOG_ARCHIVE_DEST_3 does not respond within those 2 seconds the LGWR disconnects from LOG_ARCHIVE_DEST_3, putting it into the error state. Redo continues to be sent to LOG_ARCHIVE_DEST_2 as usual. After the duration of the REOPEN attribute for LOG_ARCHIVE_DEST_3 has elapsed, the LGWR reconnects to LOG_ARCHIVE_DEST_3 and redo transfer continues. In this manner, the maximum impact of having the second SYNC standby would be restricted to 2 seconds more than having only one SYNC standby, while allowing the Oracle Data Guard configuration to maintain the desired protection level.

LGWR will never wait longer than the value of the NET_TIMEOUT attribute of the LOG_ARCHIVE_DEST_n parameter, regardless of the value of DATA_GUARD_SYNC_LATENCY.

See Also:

"LOG_ARCHIVE_DEST_n"

2.72 DATA_TRANSFER_CACHE_SIZE

DATA_TRANSFER_CACHE_SIZE sets the size of the data transfer cache (in bytes) used to receive data blocks (typically from a primary database in an Oracle Data Guard environment) for consumption by an instance during execution of an RMAN RECOVER ... NONLOGGED BLOCK command.

Droporty	Description
Property	Description
Parameter type	Big integer
Syntax	DATA_TRANSFER_CACHE_SIZE = integer [K M G]
Default value	If SGA_TARGET is set, then if DATA_TRANSFER_CACHE_SIZE is not specified, the default size of the data transfer cache is 0 (internally determined by Oracle Database). If SGA_TARGET is set and DATA_TRANSFER_CACHE_SIZE is specified, then the user-specified value indicates the minimum value for the data transfer cache. If SGA_TARGET is not set, the data transfer cache will not be available.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 – 512M, rounded up to the next granule size
Basic	No
Oracle RAC	Multiple instances can have different values.

This initialization parameter needs to be set only for databases that do not use Automatic Memory Management and that use the RMAN RECOVER ... NONLOGGED BLOCK command.

2.73 DB_nK_CACHE_SIZE

DB nK CACHE SIZE (where n = 2, 4, 8, 16, 32) specifies the size of the cache for the nK buffers.

Property	Description
Parameter type	Big integer
Syntax	DB_[2 4 8 16 32]K_CACHE_SIZE = integer [K M G]
Default value	0 (additional block size caches are not configured by default)
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Minimum: 0 (values greater than zero are automatically modified to be either the granule size * number of processor groups, or 4 MB * number of CPUs, whichever is greater)
	Maximum: operating system-dependent
Basic	No

You can set this parameter only when DB_BLOCK_SIZE has a value other than nk. For example, if DB_BLOCK_SIZE=4096, then you cannot specify the parameter DB_4K_CACHE_SIZE (because the size for the 4 KB block cache is already specified by DB_CACHE_SIZE).

Do not set this parameter to zero if there are any online tablespaces with an nK block size.

Operating system-specific block size restrictions apply. For example, you cannot set DB_32K_CACHE_SIZE if the operating system's maximum block size is less than 32 KB. Also, you cannot set DB_2K_CACHE_SIZE if the minimum block size is greater than 2 KB.



Your operating system-specific Oracle documentation for more information on block size restrictions

2.74 DB BIG TABLE CACHE PERCENT TARGET

DB_BIG_TABLE_CACHE_PERCENT_TARGET specifies the cache section target size for automatic big table caching, as a percentage of the buffer cache.

Property	Description
Parameter type	String
Syntax	DB_BIG_TABLE_CACHE_PERCENT_TARGET = string
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	Multiple instances can have different values, but it is recommended to keep the big table cache section size uniform.

Automatic big table caching enables parallel queries and serial queries to use the buffer cache, which enhances the in-memory query capabilities of Oracle Database. Automatic big table

caching is designed primarily to enhance performance for data warehouse workloads, but it also improves performance in mixed workloads.

Starting in Oracle Database 12c Release 1 (12.1.0.2), table scans can use a different algorithm in the following scenarios:

Parallel queries:

In single-instance and Oracle Real Application Clusters (Oracle RAC) databases, parallel queries can use the automatic big table cache when the DB_BIG_TABLE_CACHE_PERCENT_TARGET initialization parameter is set to a nonzero value, and the PARALLEL_DEGREE_POLICY initialization parameter is set to AUTO or ADAPTIVE.

Serial gueries:

In a single-instance configuration only, serial queries can use the automatic big table cache when the <code>DB_BIG_TABLE_CACHE_PERCENT_TARGET</code> initialization parameter is set to a nonzero value.

When a nonzero value is specified for the <code>DB_BIG_TABLE_CACHE_PERCENT_TARGET</code> parameter, the value indicates the percentage of the buffer cache to reserve for the big table cache. The largest value that can be specified with the <code>DB_BIG_TABLE_CACHE_PERCENT_TARGET</code> parameter is 90, which reserves 10% of the buffer cache for usage besides table scans.

The default value of this parameter is 0. Therefore, automatic big table caching is not enabled by default. When automatic big table caching is not enabled, queries might run using the least recently used (LRU) mechanism for cached reads, or might decide to use direct reads for the table scan.

If a large table is about the size of the combined size of the big table cache of all instances, the table will be partitioned and cached or mostly cached on all instances. With in-memory parallel query, this could eliminate most disk reads for queries on the table, or the database could intelligently read from disk only for the portion of the table that does not fit in the big table cache. If the big table cache cannot cache all the tables to be scanned, only the most frequently accessed tables will be cached, and the rest will be read via direct read automatically.

Use these guidelines when setting the parameter:

- If you do not enable automatic degree of parallelism (DOP) in your Oracle RAC environment, do not set this parameter because the big table cache is not used in that situation.
- When setting this parameter, consider the workload mix: how much of the workload is for OLTP; insert, update, and random access; and how much of the workload involves table scans. Because data warehouse workloads often perform large table scans, you may consider giving the big table cache section a higher percentage of buffer cache space for data warehouses.
- This parameter can be dynamically changed if the workload changes. The change could take some time to reach the target (depending on the current workload) because buffer cache memory might be actively used at that time.

Note:

Automatic big table caching uses temperature and object-based algorithms to track medium and big tables. Oracle will cache very small tables, but they will not be tracked by automatic big table caching.



See Also:

- "PARALLEL_DEGREE_POLICY"
- See "V\$BT_SCAN_CACHE" and "V\$BT_SCAN_OBJ_TEMPS" for more information about the big table cache
- Oracle Database VLDB and Partitioning Guide for more information about this parameter and about automatic big table caching

2.75 DB BLOCK BUFFERS

DB BLOCK BUFFERS specifies the number of database buffers in the buffer cache.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Modifiable in a PDB	No
Range of values	50 to an operating system-specific maximum
Basic	No
Oracle RAC	Multiple instances can have different values, and you can change the values as needed.

Note:

DB_BLOCK_BUFFERS cannot be combined with the dynamic DB_CACHE_SIZE parameter; combining these parameters in the same parameter file will produce an error.

DB_BLOCK_BUFFERS is one of several parameters that contribute to the total memory requirements of the SGA of an instance.

This parameter, together with <code>DB_BLOCK_SIZE</code>, determines the total size of the buffer cache. Effective use of the buffer cache can greatly reduce the I/O load on the database. <code>DB_BLOCK_SIZE</code> can be specified only when the database is first created, so you use <code>DB_BLOCK_BUFFERS</code> to control the size of the buffer cache.

Note:

The $\tt DB_BLOCK_BUFFERS$ initialization parameter is deprecated. It is maintained for backward compatibility.

Oracle recommends that you use DB CACHE SIZE instead.

See Also:

"DB_CACHE_SIZE"

2.76 DB BLOCK CHECKING

DB_BLOCK_CHECKING specifies whether Oracle Database performs block checking for database blocks.

Property	Description
Parameter type	String
Syntax	<pre>DB_BLOCK_CHECKING = { FALSE OFF LOW MEDIUM TRUE FULL }</pre>
Default value	FALSE
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes, with the following restriction: If block checking is enabled for a CDB, then you cannot subsequently disable block checking in any of its PDBs. That is, if the value of DB_BLOCK_CHECKING in a CDB is LOW, MEDIUM, TRUE, or FULL, and you then attempt to set the value of DB_BLOCK_CHECKING in one of its PDBs to FALSE or OFF, an error will occur.
Basic	No

Values

OFF or FALSE

No block checking is performed for blocks in user tablespaces. However, semantic block checking for SYSTEM tablespace blocks is always turned on.

• LOW

Basic block header checks are performed after block contents change in memory (for example, after <code>UPDATE</code>, <code>INSERT</code> or <code>DELETE</code> statements, or after inter-instance block transfers in Oracle RAC).

MEDIUM

All LOW checks and full semantic checks are performed for all objects except indexes (whose contents can be reconstructed by a drop+rebuild on encountering a corruption).

FULL or TRUE

All LOW and MEDIUM checks and full semantic checks are performed for all objects.

Oracle checks a block by going through the data in the block, making sure it is logically self-consistent. Block checking can often prevent memory and data corruption. Block checking typically causes 1% to 10% overhead in most applications, depending on workload and the parameter value. Specific DML overhead may be higher. The more updates or inserts in a workload, the more expensive it is to turn on block checking. You should set DB BLOCK CHECKING to FULL if the performance overhead is acceptable.



For backward compatibility, the use of FALSE (implying OFF) and TRUE (implying FULL) is preserved.



Caution:

Before enabling block checking with this parameter, Oracle recommends that you detect and repair any logical corruptions in the database. Otherwise, a block that contains logical corruption will be marked as "soft corrupt" after block checking is enabled and the block is modified by a DML statement. This will result in ORA-1578 errors and the block will be unreadable. For more information about detecting and repairing logical corruptions, see *Oracle Database Backup and Recovery User's Guide*.



See Also:

Oracle Database Administrator's Guide for more information about this parameter

2.77 DB BLOCK CHECKSUM

DB_BLOCK_CHECKSUM determines whether DBWn and the direct loader will calculate a checksum (a number calculated from all the bytes stored in the block) and store it in the cache header of every data block when writing it to disk.

Property	Description
Parameter type	String
Syntax	DB_BLOCK_CHECKSUM = { OFF FALSE TYPICAL TRUE FULL }
Default value	TYPICAL
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

Checksums are verified when a block is read - only if this parameter is TYPICAL or FULL and the last write of the block stored a checksum. In FULL mode, Oracle also verifies the checksum before a change application from update/delete statements and recomputes it after the change is applied. In addition, Oracle gives every log block a checksum before writing it to the current log.

Most of the log block checksum is done by the generating foreground processes, while the LGWR or the LGWR worker processes (LG*nn* processes) perform the rest of the work, for better CPU and cache efficiency.

If this parameter is set to OFF, DBWn calculates checksums only for the SYSTEM tablespace, but not for user tablespaces. In addition, no log checksum is performed when this parameter is set to OFF.

Checksums allow Oracle to detect corruption caused by underlying disks, storage systems, or I/O systems. If set to FULL, DB BLOCK CHECKSUM also catches in-memory corruptions and stops

them from making it to the disk. Turning on this feature in TYPICAL mode causes only an additional 1% to 2% overhead. In the FULL mode it causes 4% to 5% overhead. Oracle recommends that you set DB BLOCK CHECKSUM to TYPICAL.

For backward compatibility the use of TRUE (implying TYPICAL) and FALSE (implying OFF) values is preserved.



Oracle Database Backup and Recovery Reference for more information about this parameter

2.78 DB_BLOCK_SIZE

DB BLOCK SIZE specifies (in bytes) the size of Oracle database blocks.

Property	Description
Parameter type	Integer
Default value	8192
Modifiable	No
Modifiable in a PDB	No
Range of values	2048 to 32768, but your operating system may have a narrower range
Basic	Yes
Oracle RAC	You must set this parameter for every instance, and multiple instances must have the same value.



Set this parameter at the time of database creation. Do not alter it afterward.

Typical values for DB_BLOCK_SIZE are 4096 and 8192. The value of this parameter must be a multiple of the physical block size at the device level.

The value for DB_BLOCK_SIZE in effect at the time you create the database determines the size of the blocks. The value must remain set to its initial value.

For Oracle Real Application Clusters (Oracle RAC), this parameter affects the maximum value of the FREELISTS storage parameter for tables and indexes. Oracle uses one database block for each freelist group. Decision support system (DSS) and data warehouse database environments tend to benefit from larger block size values.



64-bit operating systems support a maximum DB BLOCK SIZE value of 32768



See Also:

- Oracle Database Administrator's Guide for information on setting this parameter
- Oracle Database SQL Language Reference for information on freelist groups

2.79 DB_CACHE_ADVICE

 $\label{eq:decomposition} $$ $$ DB_CACHE_ADVICE $ enables or disables statistics gathering used for predicting behavior with different cache sizes through the $$ V$DB_CACHE_ADVICE $$ performance view.$

Property	Description
Parameter type	String
Syntax	DB_CACHE_ADVICE = { ON READY OFF }
Default value	If STATISTICS_LEVEL is set to TYPICAL or ALL, then ON
	If STATISTICS_LEVEL is set to BASIC, then OFF
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

Values

OFF

Advisory is turned off and the memory for the advisory is not allocated.

READY

Advisory is turned off but the memory for the advisory remains allocated. Allocating the memory before the advisory is actually turned on avoids the risk of an error when you switch the parameter to onledge N.

If the parameter is switched to this state from on, the contents of the view are preserved and the memory for the advisory is retained.

If the parameter is switched to this state from OFF, you may get an error.

• ON

Advisory is turned on. CPU and memory overheads are incurred. Attempting to set the parameter to this state when it is already in the OFF state may result in an error. Otherwise, the view (VDB_CACHE_ADVICE$) is reset and statistics are gathered to the newly refreshed view

If the parameter is in the READY state, you can set it to ON without any errors because the memory is already allocated. The view is reset and statistics are displayed in the newly refreshed view.



"V\$DB_CACHE_ADVICE"

2.80 DB_CACHE_SIZE

DB_CACHE_SIZE specifies the size of the DEFAULT buffer pool for buffers with the primary block size (the block size defined by the DB_BLOCK_SIZE initialization parameter).



Oracle recommends against setting this parameter in a PDB. The only shared memory sizing parameter that should be set in a PDB is SGA_TARGET, which specifies the maximum SGA that the PDB can use at any time.

Property	Description
Parameter type	Big integer
Syntax	DB_CACHE_SIZE = integer [K M G]
Default value	If SGA_TARGET is set: If the parameter is not specified, then the default is 0 (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool. If SGA_TARGET is not set, then the default is either 48 MB or 4 MB * number of CPUs, whichever is greater
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

The value must be at least 4M * number of cpus (smaller values are automatically rounded up to this value). A user-specified value larger than this is rounded up to the nearest granule size. A value of zero is invalid because it is needed for the DEFAULT memory pool of the primary block size, which is the block size for the SYSTEM tablespace.

See Also:

- "SGA_TARGET"
- Oracle Database Performance Tuning Guide and Oracle Database Administrator's Guide for more information on setting this parameter
- Oracle Multitenant Administrator's Guide for more information about the initialization parameters that control the memory usage of PDBs

2.81 DB_CREATE_FILE_DEST

DB CREATE FILE DEST specifies the default location for Oracle-managed datafiles.

Property	Description
Parameter type	String



Property	Description
Syntax	DB_CREATE_FILE_DEST = directory disk group
Default value	There is no default value.
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	Yes

This location is also used as the default location for Oracle-managed control files and online redo logs if none of the $DB_CREATE_ONLINE_LOG_DEST_n$ initialization parameters are specified.

If a file system directory is specified as the default location, then the directory must already exist; Oracle does not create it. The directory must have appropriate permissions that allow Oracle to create files in it. Oracle generates unique names for the files, and a file thus created is an Oracle Managed File.



Oracle Database Administrator's Guide for more information on setting this parameter and on Oracle Managed Files

2.82 DB CREATE ONLINE LOG DEST n

DB_CREATE_ONLINE_LOG_DEST_n (where n = 1, 2, 3, ... 5) specifies the default location for Oracle-managed control files and online redo logs.

Property	Description
Parameter type	String
Syntax	<pre>DB_CREATE_ONLINE_LOG_DEST_[1 2 3 4 5] = directory disk group</pre>
Default value	There is no default value.
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	Yes

If more than one $\begin{tabular}{l} DB_CREATE_ONLINE_LOG_DEST_n parameter is specified, then the control file or online redo log is multiplexed across the locations of the other <math>\begin{tabular}{l} DB_CREATE_ONLINE_LOG_DEST_n parameters. One member of each online redo log is created in each location, and one control file is created in each location.$

Specifying at least two parameters provides greater fault tolerance for the control files and online redo logs if one of the locations should fail.

If a file system directory is specified as the default location, then the directory must already exist; Oracle does not create it. The directory must have appropriate permissions that allow Oracle to create files in it. Oracle generates unique names for the files, and a file thus created is an Oracle Managed File.



Oracle Database Administrator's Guide for more information on setting this parameter and on Oracle Managed Files

2.83 DB DOMAIN

In a distributed database system, \DB_DOMAIN specifies the logical location of the database within the network structure.

Property	Description
Parameter type	String
Syntax	DB_DOMAIN = domain_name
Default value	There is no default value.
Modifiable	No
Modifiable in a PDB	Yes
Range of values	Any legal string of name components, separated by periods and up to 128 characters long (including the periods).
Basic	Yes
Oracle RAC	You must set this parameter for every instance, and multiple instances must have the same value.

You should set this parameter if this database is or ever will be part of a distributed system. The value consists of the extension components of a global database name, consisting of valid identifiers (any alphanumeric ASCII characters), separated by periods.



Oracle recommends that you specify <code>DB_DOMAIN</code> as a unique string for all databases in a domain.

This parameter allows one department to create a database without worrying that it might have the same name as a database created by another department. If one sales department's <code>DB_DOMAIN</code> is <code>JAPAN.EXAMPLE.COM</code>, then their <code>SALES</code> database (<code>SALES.JAPAN.EXAMPLE.COM</code>) is uniquely distinguished from another database with <code>DB_NAME = SALES</code> but with <code>DB_DOMAIN = US.EXAMPLE.COM</code>.

If you omit the domains from the name of a database link, Oracle expands the name by qualifying the database with the domain of your local database as it currently exists in the data dictionary, and then stores the link name in the data dictionary. DB_DOMAIN should start with an alphabetic character and exclude these characters:

"!@%^&*()+=\\|`~[{]};:'\",<>/?"



See Also:

- Oracle Database Administrator's Guide for more information on setting this parameter
- The data dictionary view "GLOBAL_NAME"

2.84 DB_FILE_MULTIBLOCK_READ_COUNT

DB_FILE_MULTIBLOCK_READ_COUNT specifies the maximum number of blocks read in one I/O operation during a sequential scan.

Property	Description
Parameter type	Integer
Default value	The default value corresponds to the maximum I/O size that can be efficiently performed and is platform-dependent
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	Operating system-dependent
Basic	No

DB_FILE_MULTIBLOCK_READ_COUNT is one of the parameters you can use to minimize I/O during table scans. The total number of I/Os needed to perform a full table scan depends on such factors as the size of the table, the multiblock read count, and whether parallel execution is being utilized for the operation.

The default value of this parameter is a value that corresponds to the maximum I/O size that can be performed efficiently. This value is platform-dependent and is 1MB for most platforms.

Because the parameter is expressed in blocks, it will be set to a value that is equal to the maximum I/O size that can be performed efficiently divided by the standard block size. Note that if the number of sessions is extremely large the multiblock read count value is decreased to avoid the buffer cache getting flooded with too many table scan buffers.

Even though the default value may be a large value, the optimizer will not favor large I/Os if you do not set this parameter. It will favor large I/Os only if you explicitly set this parameter to a large value.

Online transaction processing (OLTP) and batch environments typically have values in the range of 4 to 16 for this parameter. DSS and data warehouse environments tend to benefit most from maximizing the value of this parameter. The optimizer is more likely to choose a full table scan over an index if the value of this parameter is high.

The maximum value is the operating system's maximum I/O size expressed as Oracle blocks ((max I/O size)/DB_BLOCK_SIZE). If you set this parameter to a value greater than the maximum, then Oracle uses the maximum.





Oracle Database Performance Tuning Guide for information about how setting this parameter incorrectly can cause excessive I/O waits for some execution plans

2.85 DB FILE NAME CONVERT

DB_FILE_NAME_CONVERT is useful for creating a duplicate database for recovery purposes. It converts the filename of a new datafile on the primary database to a filename on the standby database.

Property	Description
Parameter type	String
Syntax	DB_FILE_NAME_CONVERT = 'string1' , 'string2' , 'string3' , 'string4' ,
	Where:
	 string1 is the pattern of the primary database filename string2 is the pattern of the standby database filename string3 is the pattern of the primary database filename string4 is the pattern of the standby database filename You can enclose each string in single or double quotation marks. You can specify as many pairs of primary and standby replacement strings as required. However, starting with Oracle Database 12c Release 2 (12.2), Oracle recommends that you limit the number of pairs to 99.
	Example:
	DB_FILE_NAME_CONVERT = '/dbs/t1/','/dbs/t1/s','dbs/t2/s'
Default value	There is no default value.
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

If you add a datafile to the primary database, you must add a corresponding file to the standby database. When the standby database is updated, this parameter converts the datafile name on the primary database to the datafile name on the standby database. The file on the standby database must exist and be writable, or the recovery process will halt with an error.

Set the value of this parameter to one or more pairs of strings. The first string in a pair is the pattern found in the datafile names on the primary database. The second string in a pair is the pattern found in the datafile names on the standby database.

If you specify an odd number of strings (the last string has no corresponding replacement string), an error is returned during startup. If the filename being converted matches more than one pattern in the pattern/replace string list, the first matched pattern takes effect. There is no limit on the number of pairs that you can specify in this parameter (other than the hard limit of the maximum length of multivalue parameters). However, starting with Oracle Database 12c Release 2 (12.2), Oracle recommends that you limit the number of pairs to 99.

You can also use <code>DB_FILE_NAME_CONVERT</code> to rename the datafiles in the clone control file when setting up a clone database during tablespace point-in-time recovery.

See Also:

- Oracle Database Backup and Recovery User's Guide for more information about database backup and recovery
- Oracle Data Guard Concepts and Administration for more information about Oracle Data Guard

2.86 DB_FILES

 $\mathtt{DB_FILES}$ specifies the maximum number of database files that can be opened for this database.

Property	Description
Parameter type	Integer
Default value	200
Modifiable	No
Modifiable in a PDB	Yes
Range of values	Minimum: the largest among the absolute file numbers of the datafiles in the database
	Maximum: operating system-dependent
Basic	No
Oracle RAC	Multiple instances must have the same value.

The maximum valid value is the maximum number of files, subject to operating system constraint, that will ever be specified for the database, including files to be added by ADD DATAFILE statements.

If you increase the value of $\mathtt{DB_FILES}$, then you must shut down and restart all instances accessing the database before the new value can take effect. If you have a primary and standby database, then they should have the same value for this parameter.

See Also:

- Oracle Real Application Clusters Administration and Deployment Guide for information on setting this parameter in an Oracle RAC environment
- Your operating system-specific Oracle documentation for the default value of this parameter

2.87 DB FLASH CACHE FILE

DB_FLASH_CACHE_FILE specifies file name(s) for the flash memory or disk group representing a collection of flash memory, for use with Database Smart Flash Cache.



Property	Description
Parameter type	String
Syntax	DB_FLASH_CACHE_FILE = filename [,filename] disk group
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

You can specify up to 16 file names for flash memory devices. For example, if there are three flash raw devices:

```
db_flash_cache_file = /dev/raw/sda, /dev/raw/sdb, /dev/raw/sdc
```

Specifying this parameter without also specifying the DB_FLASH_CACHE_SIZE initialization parameter is not allowed.

See Also:
"DB_FLASH_CACHE_SIZE"

2.88 DB_FLASH_CACHE_SIZE

DB_FLASH_CACHE_SIZE specifies the size of the Database Smart Flash Cache (flash cache). This parameter may only be specified at instance startup.

Property	Description
Parameter type	Big integer
Syntax	<pre>DB_FLASH_CACHE_SIZE = integer [K M G T] [, integer [K M G T]]</pre>
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 to (DB_BLOCK_SIZE * 256 MB)
	If DB_BLOCK_SIZE = 2 KB, then 0 to 512 GB
	 If DB_BLOCK_SIZE = 4 KB, then 0 to 1 TB
	 If DB_BLOCK_SIZE = 8 KB, then 0 to 2 TB
	 If DB_BLOCK_SIZE = 16 KB, then 0 to 4 TB
	 If DB_BLOCK_SIZE = 32 KB, then 0 to 8 TB
Basic	No

You can specify up to 16 file sizes, for each of the flash memory devices specified with <code>DB_FLASH_CACHE_FILE</code>. For example, if there are three flash raw devices, you can specify the sizes of each device as follows:

```
db_flash_cache_file = /dev/raw/sda, /dev/raw/sdb, /dev/raw/sdc
db_flash_cache_size = 32G, 32G, 64G
```

If your flash cache consists of one flash cache device, you can dynamically change this parameter to 0 for that flash cache device (disabling the flash cache) after the database is started. You can then reenable the flash cache by setting this parameter for the device back to the original value when the database was started. Dynamic resizing of DB_FLASH_CACHE_SIZE or reenabling flash cache to a different size is not supported.

If your flash cache includes multiple flash cache devices, you can dynamically change the parameter to 0 for a particular flash cache device (turning it off) after the database is started. You can then reenable that flash cache device by setting this parameter for the device back to the original value it had when the database was started (turning it back on).

For example, to turn off the /dev/raw/sdb flash cache device:

```
db_flash_cache_file = /dev/raw/sda, /dev/raw/sdb, /dev/raw/sdc
db_flash_cache_size = 32G, 0, 64G
```

And, to turn the /dev/raw/sdb flash cache device back on again:

```
db_flash_cache_file = /dev/raw/sda, /dev/raw/sdb, /dev/raw/sdc
db_flash_cache_size = 32G, 32G, 64G
```

```
✓ See Also:
"DB_FLASH_CACHE_FILE"
```

2.89 DB FLASHBACK LOG DEST

DB FLASHBACK LOG DEST specifies the location for flashback database log storage.

Property	Description
Parameter type	String
Syntax	DB_FLASHBACK_LOG_DEST = directory disk group
Default value	There is no default value.
Modifiable	ALTER SYSTEM SID='*'
	See "Modifying DB_FLASHBACK_LOG_DEST with ALTER SYSTEM"
Modifiable in a PDB	No
Basic	No
Oracle RAC	You must set this parameter for every instance, and multiple instances must have the same value.

This parameter allows you to specify a location for flashback database logs that is separate from the fast recovery area used by backups. For example, you could place flashback database logs in high-performance storage.

Before assigning a value to this parameter, you must set the <code>DB_FLASHBACK_LOG_DEST_SIZE</code> parameter to a nonzero value.

Modifying DB_FLASHBACK_LOG_DEST with ALTER SYSTEM

When modifying the DB_FLASHBACK_LOG_DEST parameter with the ALTER SYSTEM statement, you must disable and reenable flashback logging. For example:

```
SQL> alter database flashback off;
Database altered.

SQL> alter system set db_flashback_log_dest='/u01/oracle/dbs/ofba2';
System altered.

SQL> alter database flashback on;
Database altered.
```

Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

- "DB_FLASHBACK_LOG_DEST_SIZE"
- "V\$FLASHBACK_LOG_DEST"
- Oracle Database Backup and Recovery User's Guide for more information about this parameter

2.90 DB_FLASHBACK_LOG_DEST_SIZE

DB_FLASHBACK_LOG_DEST_SIZE specifies the hard limit (in bytes) on the total space to be used by flashback database log files stored in DB_FLASHBACK_LOG_DEST.

Property	Description
Parameter type	Big integer
Syntax	DB_FLASHBACK_LOG_DEST_SIZE = integer [K M G]
Default value	0
Modifiable	ALTER SYSTEM SID='*'
Modifiable in a PDB	No
Range of values	0 to operating system dependent
Basic	No
Oracle RAC	You must set this parameter for every instance, and multiple instances must have the same value.

Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

- "DB FLASHBACK LOG DEST"
- "V\$FLASHBACK_LOG_DEST"
- Oracle Database Backup and Recovery User's Guide for more information about this parameter

2.91 DB_FLASHBACK_RETENTION_TARGET

DB_FLASHBACK_RETENTION_TARGET specifies the upper limit (in minutes) on how far back in time the database may be flashed back.

Description
Integer
1440 (minutes)
ALTER SYSTEM SID='*'
No
0 to 2 ³¹ - 1
No

How far back one can flashback a database depends on how much flashback data Oracle has kept in the fast recovery area.

See Also:

Oracle Data Guard Broker for enabling Flashback Database on Oracle Data Guard primary and target standby databases

2.92 DB_INDEX_COMPRESSION_INHERITANCE

DB_INDEX_COMPRESSION_INHERITANCE dictates how index creation inherits compression attributes.

Property	Description
Parameter type	String



Property	Description
Syntax	DB_INDEX_COMPRESSION_INHERITANCE = { TABLESPACE TABLE ALL NONE }
Default value	NONE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	You must set this parameter for every instance, and multiple instances must have the same value.

Values:

TABLESPACE

Index inheritance is based on tablespace attributes.

TABLE

Index inheritance is based on table attributes.

• ALL

Index inheritance is based on table settings. However, if the table does not have default compression attributes, then index inheritance is based on tablespace attributes.

NONE

There is no index inheritance from the table or tablespace.

If the CREATE INDEX statement specifies compression attributes, then the value of DB_INDEX_COMPRESSION_INHERITANCE is ignored, and there is no inheritance from the table or tablespace. When there is table inheritance or tablespace inheritance (and the tablespace does not have an explicit index compression clause), then the following matrix is used:

Table/Tablespace Level Compression	Inherited Compression for Index
OLTP	ADVANCED LOW
QUERY LOW, QUERY HIGH	ADVANCED HIGH
ARCHIVE LOW, ARCHIVE HIGH	ADVANCED HIGH



The value of <code>DB_INDEX_COMPRESSION_INHERITANCE</code> is ignored when creating an index owned by <code>SYS</code>. Such indexes do not inherit index compression attributes from tables or tablespaces. If you want to compress an index owned by <code>SYS</code>, you must explicitly specify the index compression attributes in the <code>CREATE INDEX</code> or <code>ALTER INDEX</code> statement.



See Also:

Oracle Database Administrator's Guide for more information about index compression

2.93 DB_KEEP_CACHE_SIZE

DB_KEEP_CACHE_SIZE specifies the size of the KEEP buffer pool.

Property	Description
Parameter type	Big integer
Syntax	DB_KEEP_CACHE_SIZE = integer [K M G]
Default value	0 (DB_KEEP_CACHE_SIZE is not configured by default)
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Minimum: 0 (values greater than zero are automatically modified to be either the granule size * number of processor groups, or 4 MB * number of CPUs, whichever is greater)
	Maximum: operating system-dependent
Basic	No

The size of the buffers in the KEEP buffer pool is the primary block size (the block size defined by the $\texttt{DB_BLOCK_SIZE}$ initialization parameter).

See Also:

- "DB_RECYCLE_CACHE_SIZE"
- Oracle Database Performance Tuning Guide for information on setting these parameters and on using multiple buffer pools

2.94 DB_LOST_WRITE_PROTECT

DB LOST WRITE PROTECT enables or disables lost write detection.

Property	Description
Parameter type	String
Syntax	DB_LOST_WRITE_PROTECT = { AUTO TYPICAL FULL NONE }
Default value	AUTO
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes This parameter is modifiable in a PDB in Oracle Database 23ai, starting with Release Update 23.7.



Property	Description
Basic	No
Oracle RAC	In Oracle RAC instances, this parameter value is system-wide.

A data block lost write occurs when an I/O subsystem acknowledges the completion of the block write, while in fact the write did not occur in the persistent storage.

Values:

AUTO

When this parameter is set to AUTO on a primary database, the instance automatically decides whether it logs buffer cache reads in the redo log or not, depending on the status of the standby databases. Specifically, the primary database only logs buffer cache reads if physical standby databases with real time redo apply exist.

When this parameter is set to AUTO on a standby database, the instance will automatically decide whether it incurs additional performance overhead to perform lost write detection or not, depending on whether apply is keeping up. If apply lag is beyond the reasonable threshold, the standby database will skip lost write protection temporarily until redo apply catches up with primary again, to ensure the lowest Data Guard role transition timings.

The reasonable threshold is determined as follows:

- If Fast Start Failover (FSFO) is configured through Data Guard Broker, based on apply lag threshold, then the threshold is the lesser of 60 seconds and two-thirds of the FSFO apply lag threshold.
- Otherwise, the threshold is 60 seconds.

TYPICAL

When this parameter is set to TYPICAL on a primary database, the instance logs buffer cache reads for read/write tablespaces in the redo log, which is necessary for detection of lost writes.

When this parameter is set to TYPICAL on a standby database, or on a primary database during media recovery, the instance performs lost write detection, regardless of higher redo apply lags that may impact Data Guard role transition timings.

• FULL

When this parameter is set to FULL on a primary database, the instance logs reads for read-only tablespaces and read/write tablespaces.

When this parameter is set to FULL on a standby database, or on a primary database during media recovery, the instance performs lost write detection.

NONE

When this parameter is set to NONE on either a primary database or a standby database, no lost write detection functionality is enabled.

2.95 DB_NAME

DB_NAME specifies a database identifier of up to 8 characters.



Property	Description
Parameter type	String
Syntax	DB_NAME = database_name
Default value	There is no default value.
Modifiable	No
Modifiable in a PDB	No
Basic	Yes
Oracle RAC	You must set this parameter for every instance. Multiple instances must have the same value.

This parameter must be specified and must correspond to the name specified in the CREATE DATABASE statement.

If you have multiple databases, the value of this parameter should match the Oracle instance identifier of each one to avoid confusion with other databases running on the system. The value of $\mbox{DB}_{\mbox{NAME}}$ must be the same in both the standby and production initialization parameter files.

The database name specified in either the STARTUP command or the ALTER DATABASE ... MOUNT statement for each instance of the cluster database must correspond to the DB_NAME initialization parameter setting.

The following characters are valid in a database name: alphanumeric characters, underscore (_), number sign (#), and dollar sign (\$). No other characters are valid. The database name must start with an alphabetic character. Oracle removes double quotation marks before processing the database name. Therefore you cannot use double quotation marks to embed other characters in the name. The database name is case insensitive.



Oracle Database Administrator's Guide and Oracle Real Application Clusters
Administration and Deployment Guide for more information on setting this parameter

2.96 DB PERFORMANCE PROFILE

DB_PERFORMANCE_PROFILE specifies the performance profile for a database or a pluggable database (PDB).

Property	Description
Parameter type	String
Syntax	DB_PERFORMANCE_PROFILE = string
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No



Property	Description
Oracle RAC	The same value should be set on all instances.

A performance profile supports easier management for hundreds of databases or PDBs.

On Oracle Exadata, the performance profile for the regular database or CDB is pushed down to the Exadata storage cells. The performance profile is used for the management of Exadata I/O Resource Manager.

A CDB resource plan can specify different resource controls for a performance profile using the <code>DBMS_RESOURCE_MANAGER.CREATE_CDB_PROFILE_DIRECTIVE</code> procedure. A PDB with a matching performance profile will use the specified controls.

The value of $\mathtt{DB}_\mathtt{PERFORMANCE}_\mathtt{PROFILE}$ can be up to 30 characters and is not case sensitive.



Oracle Database PL/SQL Packages and Types Reference for information about the DBMS RESOURCE MANAGER.CREATE CDB PROFILE DIRECTIVE procedure.

2.97 DB RECOVERY AUTO REKEY

Use <code>DB_RECOVERY_AUTO_REKEY</code> to enable or disable automatic tablespace rekey recovery on Oracle Data Guard standby instances.

Property	Description
Parameter type	String
Syntax	DB_RECOVERY_AUTO_REKEY = { ON OFF }
Default value	ON
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	Multiple instances should use the same value.

This parameter is effective when Oracle Data Guard standby recovery encounters a redo that indicates that the primary database performed a tablespace rekey. If the value of this parameter is ON, then standby recovery automatically performs the corresponding tablespace rekey. If the value of this parameter is OFF, then standby recovery does not perform a tablespace rekey.

Note:

This parameter is available starting with Oracle Database 23ai.



2.98 DB_RECOVERY_FILE_DEST

DB_RECOVERY_FILE_DEST specifies the default location for the fast recovery area.

Property	Description
Parameter type	String
Syntax	DB_RECOVERY_FILE_DEST = directory disk group
Default value	There is no default value.
Modifiable	ALTER SYSTEM SID='*'
Modifiable in a PDB	No
Basic	Yes
Oracle RAC	You must set this parameter for every instance, and multiple instances must have the same value.

The fast recovery area contains multiplexed copies of current control files and online redo logs, as well as archived redo logs, flashback logs, and RMAN backups.

Specifying this parameter without also specifying the <code>DB_RECOVERY_FILE_DEST_SIZE</code> initialization parameter is not allowed.

See Also:

- Oracle Database Backup and Recovery User's Guide for information on setting up and configuring the fast recovery area
- "DB_RECOVERY_FILE_DEST_SIZE"

2.99 DB RECOVERY FILE DEST SIZE

DB_RECOVERY_FILE_DEST_SIZE specifies (in bytes) the hard limit on the total space to be used by target database recovery files created in the fast recovery area.

Property	Description
Parameter type	Big integer
Syntax	DB_RECOVERY_FILE_DEST_SIZE = integer [K M G]
Default value	0
Modifiable	ALTER SYSTEM SID='*'
Modifiable in a PDB	No
Basic	Yes
Oracle RAC	You must set this parameter for every instance, and multiple instances must have the same value.

Note that neither block 0 nor the OS block header of each Oracle file is included in this size. Allow an extra 10% for this data when computing the actual disk usage required for the fast recovery area.

See Also:

- Oracle Database Backup and Recovery User's Guide for information on setting up and configuring the fast recovery area
- "DB_RECOVERY_FILE_DEST"

2.100 DB_RECYCLE_CACHE_SIZE

DB RECYCLE CACHE SIZE specifies the size of the RECYCLE buffer pool.

Property	Description
Parameter type	Big integer
Syntax	DB_RECYCLE_CACHE_SIZE = integer [K M G]
Default value	0 (DB_RECYCLE_CACHE_SIZE is not configured by default)
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Minimum: 0 (values greater than zero are automatically modified to be either the granule size * number of processor groups, or 4 MB * number of CPUs, whichever is greater) Maximum: operating system-dependent
Basic	No

The size of the buffers in the RECYCLE pool is the primary block size (the block size defined by the DB_BLOCK_SIZE initialization parameter).

See Also:

- "DB_KEEP_CACHE_SIZE"
- Oracle Database Performance Tuning Guide for information on setting these parameters and on using multiple buffer pools

2.101 DB_SECUREFILE

DB SECUREFILE specifies whether to treat LOB files as SecureFiles.

Property	Description
Parameter type	String
Syntax	<pre>DB_SECUREFILE = { NEVER PERMITTED PREFERRED ALWAYS IGNORE }</pre>



Property	Description
Default value	PERMITTED if the COMPATIBLE initialization parameter is set to 11.2.0.1, 11.2.0.2, or 11.2.0.3, or PREFERRED if the COMPATIBLE initialization parameter is set to 12.0.0.0 or higher
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Values

NEVER

Any LOBs that are specified as SecureFiles are created as BasicFiles LOBs. All SecureFiles-specific storage options and features (for example, compress, encrypt, deduplicate) will cause an exception. The BasicFiles LOB defaults will be used for storage options not specified.

PERMITTED

LOBs are allowed to be created as SecureFiles.

PREFERRED

All LOBs are created as SecureFiles unless BASICFILE is explicitly specified in the LOB storage clause or the tablespace is a Manual Segment Space Management tablespace. When PREFERRED is set, cases where BASICFILE would otherwise be inherited from the partition or column level LOB storage are ignored; the LOBs will be created as SecureFiles instead.

ALWAYS

Attempts to create all LOBs as SecureFiles LOBs but creates any LOBs not in an Automatic Segment Space Managed (ASSM) tablespace as BasicFiles LOBs, unless SECUREFILE is explicitly specified. Any BasicFiles LOB storage options that are specified will be ignored and the SecureFiles LOB defaults will be used for all storage options not specified.

• IGNORE

The **SECUREFILE** keyword and all SecureFiles options are ignored.

If the COMPATIBLE initialization parameter is not set to 11.1.0 or higher, then LOBs are not treated as SecureFiles.

If there is a LOB column with two partitions (one that has a tablespace for which ASSM is enabled and one that has a tablespace for which ASSM is not enabled), then LOBs in the partition with the ASSM-enabled tablespace will be treated as SecureFiles and LOBs in the other partition will be treated as BasicFiles.

If an application is not certified with SecureFiles, the value of the DB_SECUREFILE initialization parameter can be set to PERMITTED to get the default behavior of Oracle Database 11g.



See Also:

Oracle Database SecureFiles and Large Objects Developer's Guide for an example of setting this parameter using the ALTER SYSTEM statement

2.102 DB_ULTRA_SAFE

DB ULTRA SAFE sets the default values for other parameters that control protection levels.

Property	Description
Parameter type	String
Syntax	DB_ULTRA_SAFE = { OFF DATA_ONLY DATA_AND_INDEX }
Default value	OFF
Modifiable	No
Modifiable in a PDB	No
Basic	No

Values

OFF

When any of DB_BLOCK_CHECKING, DB_BLOCK_CHECKSUM, or DB_LOST_WRITE_PROTECT are explicitly set, no changes are made.

- DATA_ONLY
 - DB BLOCK CHECKING will be set to MEDIUM.
 - DB_LOST_WRITE_PROTECT will be set to TYPICAL.
 - DB BLOCK CHECKSUM will be set to FULL.
- DATA_AND_INDEX
 - DB_BLOCK_CHECKING will be set to FULL.
 - DB LOST WRITE PROTECT will be set to TYPICAL.
 - DB BLOCK CHECKSUM will be set to FULL.

See Also:

- "DB_BLOCK_CHECKING"
- "DB_BLOCK_CHECKSUM"
- "DB_LOST_WRITE_PROTECT"



2.103 DB_UNIQUE_NAME

DB UNIQUE NAME specifies a globally unique name for the database.

Property	Description
Parameter type	String
Syntax	DB_UNIQUE_NAME = database_unique_name
Default value	Database instances: the value of DB_NAME
	Oracle Automatic Storage Management instances: +ASM
Modifiable	No
Modifiable in a PDB	No
Basic	Yes
Oracle RAC	Multiple instances must have the same value.

Databases with the same <code>DB_NAME</code> within the same <code>DB_DOMAIN</code> (for example, copies of a database created for reporting or a physical standby) must have a unique <code>DB_UNIQUE_NAME</code>. Every database's <code>DB_UNIQUE_NAME</code> must be unique within the enterprise.

The value of <code>DB_UNIQUE_NAME</code> can be up to 30 characters and is case insensitive. The following characters are valid in a database name: alphanumeric characters, underscore (_), number sign (#), and dollar sign (\$).

Note:

DB_UNIQUE_NAME is used by several components within an Oracle instance to default file names or file paths, such as the default for the DG_BROKER_CONFIG_FILE*n* initialization parameter. In these cases, on UNIX platforms, the dollar sign (\$) character will be removed from the path or file name, because the character is used by Oracle to define the start of an environment variable substitution within a path or file name.

Note:

As part of their operations, some database tools or utilities create a string that uniquely identifies a database. The string may include the DB_UNIQUE_NAME for a database, and other identifying information for the database, such as the database SID. Oracle Database restricts some identifiers to 30 characters, so using a short DB_UNIQUE_NAME can help prevent ORA-00972 "identifier is too long" messages from database tools and utilities that create a string that includes the DB_UNIQUE_NAME.



See Also:

Oracle Data Guard Concepts and Administration and Oracle Database Administrator's Guide for more information on setting this parameter

2.104 DB_UNRECOVERABLE_SCN_TRACKING

DB_UNRECOVERABLE_SCN_TRACKING enables or disables the tracking of unrecoverable (NOLOGGING) direct-path INSERT and direct-path load operations.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Multiple instances must have the same value

When the value is set to true, updates are made to the controlfile that maintains the V\$DATAFILE.UNRECOVERABLE_CHANGE# and V\$DATAFILE.UNRECOVERABLE_TIME columns. When the value is set to false, updates are not made to the controlfile. Setting this parameter to false may improve performance of direct-path NOLOGGING operations.

2.105 DB_WRITER_PROCESSES

DB_WRITER_PROCESSES specifies the initial number of Database Writer Processes for an instance. This parameter is useful for systems that modify data heavily.

Property	Description
Parameter type	Integer
Default value	1 or CPU_COUNT / 8, whichever is greater. If the number of processor groups is less than 100 but greater than the number of Database Writer Processes, then the number of Database Writer Processes is adjusted to be a multiple of the number of processor groups. If the number of Database Writer Processes is greater than or equal to the number of processor groups, then there is no adjustment.
	For Oracle Solaris on SPARC systems, the default value of this parameter is calculated using an internal algorithm.
Modifiable	No
Modifiable in a PDB	No
Range of values	1 to 100
Basic	No



There can be 1 to 100 Database Writer Processes. The names of the first 36 Database Writer Processes are DBW0-DBW9 and DBWa-DBWz. The names of the 37th through 100th Database Writer Processes are BW36-BW99.

See Also:

- Background Processes for additional information about the Database Writer Process
- Oracle Database Concepts for information on when the Database Writer Process writes dirty buffers to disk

2.106 DBFIPS_140

DBFIPS_140 enables Transparent Data Encryption (TDE) and DBMS_CRYPTO PL/SQL package program units to run in a mode compliant to the Federal Information Processing Standard (subsequently known as "FIPS mode").

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	All instances must use the same value.

Set this parameter to true to use TDE and <code>DBMS_CRYPTO</code> in FIPS mode. This means that only FIPS-compliant algorithms may be used. By default, this parameter is set to <code>false</code>. When this parameter is set to <code>false</code>, all algorithms (FIPS-compliant or not) may be used.

See Also:

- Oracle Database Security Guide for information about configuring Oracle Database FIPS 140 settings
- Oracle Database Security Guide for a table that describes the effect of setting the value of DBFIPS_140 to true or false on different platforms



2.107 DBNEST_ENABLE

DBNEST ENABLE allows you to enable or disable dbNest.

Note:

This parameter is applicable only on Linux systems.

Property	Description
Parameter type	String
Syntax	<pre>DBNEST_ENABLE = { NONE CDB_RESOURCE_PDB_ALL }</pre>
Default value	NONE
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	All instances should use the same value.

DbNest provides operating system resource isolation and management, file system isolation, and secure computing for PDBs. The infrastructure is implemented as a Linux-specific package that provides hierarchical containers, called nests. A CDB resides within a single parent nest, while PDBs reside within the individual child nests created within the parent. The isolation and resource management properties of a nest are separate. A full nest includes both isolation and resource management. Alternatively, you can configure a partial nest that includes only resource management.

You can set the following values for this parameter:

- NONE: Disables dbNest. This is the default value.
- CDB_RESOURCE_PDB_ALL: Enables a resource management-only nest for the CDB and a full nest for each PDB.

See Also:

Oracle Multitenant Administrator's Guide for information on securing and isolating resources using dbNest

2.108 DBNEST PDB FS CONF

DBNEST PDB FS CONF specifies the dbNest file system configuration file for a PDB.

Note:

This parameter is applicable only on Linux systems.

Property	Description
Parameter type	String
Syntax	DBNEST_PDB_FS_CONF = 'pathname'
Default value	None
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	Multiple instances can have different values.

A dbNest file system configuration file specifies directories to be mounted inside the nest for a PDB. These directories are mounted in addition to the default directories.

For pathname, specify the full operating system path for the file.

If you do not set this parameter, then only the default directories are mounted.



Oracle Multitenant Administrator's Guide for information on dbNest file system configuration files

2.109 DBWR IO SLAVES

DBWR IO SLAVES specifies the number of I/O server processes used by the DBWO process.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Modifiable in a PDB	No
Range of values	0 to operating system-dependent
Basic	No

DBWR_IO_SLAVES is relevant only on systems with only one database writer process (DBW0). The DBW0 process and its server processes always write to disk. By default, the value is 0 and I/O server processes are not used.

If you set <code>DBWR_IO_SLAVES</code> to a nonzero value, the number of I/O server processes used by the <code>ARCH</code> and <code>LGWR</code> processes is set to 4. However, the number of I/O server processes used by Recovery Manager is set to 4 only if asynchronous I/O is disabled (either your platform does not support asynchronous I/O or <code>disk</code> asynch io is set to <code>false</code>).

Typically, I/O server processes are used to simulate asynchronous I/O on platforms that do not support asynchronous I/O or that implement it inefficiently. However, you can use I/O server processes even when asynchronous I/O is being used. In that case the I/O server processes will use asynchronous I/O.



I/O server processes are also useful in database environments with very large I/O throughput, even if asynchronous I/O is enabled.

See Also:

- "BACKUP_TAPE_IO_SLAVES"
- Oracle Database Performance Tuning Guide for more information about this parameter

2.110 DDL_LOCK_TIMEOUT

DDL_LOCK_TIMEOUT specifies a time limit for how long DDL statements will wait in a DML lock queue.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 1,000,000 (in seconds)
Basic	No

A value of zero indicates a status of NOWAIT. The maximum value of 1,000,000 seconds will result in the DDL statement waiting forever to acquire a DML lock.

If a lock is not acquired before the timeout period expires, then an error is returned.

See Also:

- Oracle Database Administrator's Guide for more information about the DDL_LOCK_TIMEOUT parameter
- Oracle Database Development Guide for information about nonblocking and blocking DDL

2.111 DEFAULT_SHARING

DEFAULT_SHARING sets the value of the sharing clause in statements creating objects in an application root.

Property	Description
Parameter type	String
Syntax	DEFAULT SHARING = { NONE METADATA DATA EXTENDED DATA }



Property	Description
Default value	For types of objects that support sharing, METADATA is the default.
	For types of objects that do not support sharing, ${\tt NONE}$ is the default.
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	All instances must have the same value

Specifying SHARING= in the create DDL overrides the value of the DEFAULT SHARING parameter.

Example

Issuing the following ALTER SYSTEM statement in an application root sets the default value of the sharing clause to NONE in the server parameter file for the application root:

ALTER SYSTEM SET DEFAULT_SHARING = NONE SCOPE = SPFILE;

2.112 DEFERRED_SEGMENT_CREATION

DEFERRED SEGMENT CREATION specifies the semantics of deferred segment creation.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

If set to true, then segments for tables and their dependent objects (LOBs, indexes) will not be created until the first row is inserted into the table.

Before creating a set of tables, if it is known that a significant number of them will not be populated, then consider setting this parameter to true. This saves disk space and minimizes install time.

2.113 DG_BROKER_CONFIG_FILEn

 $DG_BROKER_CONFIG_FILE n$ (where n = 1, 2) specifies the names for the Data Guard broker configuration files.

Property	Description
Parameter type	String
Syntax	<pre>DG_BROKER_CONFIG_FILE[1 2] = filename</pre>
Default value	Operating system-dependent
Modifiable	ALTER SYSTEM



Property	Description
Modifiable in a PDB	No
Range of values	One filename
Basic	No

Every database that is part of a Data Guard broker configuration has two broker configuration files, which contain entries that describe the state and properties of the configuration (such as the sites and databases that are part of the configuration, the roles and properties of each of the databases, and the state of each of the elements of the configuration). Two files are provided to always maintain the last known good state of the configuration.

If $DG_BROKER_CONFIG_FILEn$ is not explicitly defined, then it is set to an operating system-specific default value at instance startup. This parameter can be altered only when the Data Guard broker is not running. See "DG_BROKER_START" for information on how to stop and start the broker.



Oracle Data Guard Broker for more information about setting this parameter

2.114 DG BROKER START

 ${\tt DG_BROKER_START}$ enables Oracle to determine whether the Oracle Data Guard broker should be started.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEMSID='*'
Modifiable in a PDB	No
Range of values	true false
Basic	No

The Oracle Data Guard broker consists of background processes, foreground processes, and the system global area (SGA). The background and foreground processes exist as long as the instance exists, whenever this parameter is set to true.

If the database is never going to be configured in an Oracle Data Guard broker configuration, then you can leave this parameter unspecified and accept the default value of false. If the database is part of an Oracle Data Guard broker configuration, then you can simplify automatic startup of the broker by setting this parameter to true in the initialization parameter file.



Oracle Data Guard Broker for examples of using this parameter to start the Oracle Data Guard broker

2.115 DIAGNOSTIC DEST

As of Oracle Database 11g Release 1 (11.1), the diagnostics for each database instance are located in a dedicated directory that is specified by the <code>DIAGNOSTIC_DEST</code> initialization parameter.

Property	Description
Parameter type	String
Syntax	<pre>DIAGNOSTIC_DEST = { pathname directory }</pre>
Default value	Derived from the value of the <code>\$ORACLE_BASE</code> environment variable. If <code>\$ORACLE_BASE</code> is not set, then derived from <code>ORACLE_BASE</code> as set by the Oracle Universal Installer. If <code>ORACLE_BASE</code> is not set, then <code>\$ORACLE_HOME/rdbms/log</code> is used.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	This parameter can be set on each instance. Oracle recommends that each instance in a cluster specify a <code>DIAGNOSTIC_DEST</code> directory location that is located on shared disk and that the same value for <code>DIAGNOSTIC_DEST</code> be specified for each instance.

The structure of the directory specified by <code>DIAGNOSTIC</code> <code>DEST</code> is as follows:

<diagnostic_dest>/diag/rdbms/<dbname>/<instname>

This location is known as the Automatic Diagnostic Repository (ADR) Home. For example, if the database name is proddb and the instance name is proddb1, the ADR home directory would be <diagnostic dest>/diag/rdbms/proddb/proddb1.

The following files are located under the ADR home directory:

- Trace files located in subdirectory <adr-home>/trace
- Alert logs located in subdirectory <adr-home>/alert. In addition, the alert.log file is now in XML format, which conforms to the Oracle ARB logging standard.
- Core files located in the subdirectory <adr-home>/cdump
- Incident files the occurrence of each serious error (for example, ORA-600, ORA-1578, ORA-7445) causes an incident to be created. Each incident is assigned an ID and dumping for each incident (error stack, call stack, block dumps, and so on) is stored in its own file, separated from process trace files. Incident dump files are located in <adr-home>/incident/<incdir#>. You can find the incident dump file location inside the process trace file.





Oracle Automatic Storage Management Administrator's Guide for an example of the diagnostic directory for an Oracle ASM instance

2.116 DIAGNOSTICS_CONTROL

DIAGNOSTICS_CONTROL enables you to control and monitor the users who perform potentially unsafe database diagnostic operations.

Property	Description
Parameter type	String
Syntax	<pre>DIAGNOSTICS_CONTROL = { ERROR WARNING IGNORE }</pre>
Default value	IGNORE
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	The same value must be used on all instances.

When diagnosing Oracle database issues, an Oracle Support Service staff member might instruct you to perform diagnostic operations. Some diagnostic operations are harmless. However, others could have adverse effects if not performed properly with the guidance of an Oracle Support Service staff member. In order to control and monitor who performs such diagnostic operations, Oracle Database 20 introduces the following security controls:

Potentially unsafe diagnostic operations are internally identified in the database. These
operations include setting certain debug events and debug actions with the ALTER SESSION
or ALTER SYSTEM statement. For example:

```
ALTER SESSION SET EVENTS '1357 trace name context forever, level 2'; ALTER SYSTEM SET EVENTS 'trace[vos]';
```

- A user can be authorized to perform potentially unsafe diagnostic operations, either by being granted the ENABLE DIAGNOSTICS system privilege, or by being granted the DBA role, or by authenticating as SYSDBA.
- The DIAGNOSTICS_CONTROL initialization parameter enables you to specify how the database responds when an unauthorized user attempts to perform these diagnostic operations. The possible values are:
 - ERROR: When an unauthorized user attempts to perform a potentially unsafe diagnostic operation, the attempt fails with error ORA-01031: insufficient privileges.
 - WARNING: When an unauthorized user attempts to perform a potentially unsafe diagnostic operation, the attempt succeeds, but a warning is written to the alert log.
 - IGNORE: When an unauthorized user attempts to perform a potentially unsafe diagnostic operation, the attempt succeeds and no error messages or warnings occur. This is the default.

Note:

When altering the <code>DIAGNOSTICS_CONTROL</code> initialization parameter, be aware of the following guidelines:

- To modify this parameter with the ALTER SYSTEM command, you must be authenticated AS SYSDBA.
- In Oracle RAC environments, it is currently not possible to modify this parameter on remote instances by specifying the SID clause in the ALTER SYSTEM SET statement.

2.117 DISK_ASYNCH IO

DISK_ASYNCH_IO controls whether I/O to datafiles, control files, and logfiles is asynchronous (that is, whether parallel server processes can overlap I/O requests with CPU processing during table scans).

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No

If your platform supports asynchronous I/O to disk, Oracle recommends that you leave this parameter set to its default value. However, if the asynchronous I/O implementation is not stable, you can set this parameter to false to disable asynchronous I/O. If your platform does not support asynchronous I/O to disk, this parameter has no effect.

If you set <code>DISK_ASYNCH_IO</code> to false, then you can increase <code>DB_WRITER_PROCESSES</code> or use <code>DBWR_IO_SLAVES</code> to simulate asynchronous I/O.

See Also:

- "DB WRITER PROCESSES"
- "DBWR IO SLAVES"
- Oracle Database Performance Tuning Guide for more information

2.118 DISPATCHERS

DISPATCHERS configures dispatcher processes in the shared server architecture.

Property	Description
Parameter type	String
Syntax	DISPATCHERS = 'dispatch_clause'
Syntax	dispatch_clause::=
	(PROTOCOL = protocol)
	(ADDRESS = address)
	(DESCRIPTION = description)
	[options_clause]
Syntax	options_clause::=
	(DISPATCHERS = integer
	SESSIONS = integer
	CONNECTIONS = integer
	MULTIPLEX = {1 ON YES TRUE
	0 OFF NO FALSE BOTH IN OUT}
	LISTENER = tnsname
	SERVICE = service
	<pre>INDEX = integer)</pre>
Default value	If SHARED_SERVERS is greater than 0, then DISPATCHERS defaults to '(PROTOCOL=tcp)', causing 1 TCP/IP dispatcher to be created.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

The parsing software supports a name-value syntax to enable the specification of attributes in a position-independent, case-insensitive manner. For example:

```
DISPATCHERS = '(PROTOCOL=TCP) (DISPATCHERS=3)'
```

Attributes may be specified using the full attribute name or any substring beginning with the first 3 characters. For example, <code>SESSIONS</code> can be specified as <code>SES</code>, <code>SESSI</code>, and so on.

Specify only one of the following attributes: PROTOCOL, ADDRESS, or DESCRIPTION. If you specify either ADDRESS or DESCRIPTION, then you can specify additional network attributes. Doing so supports multi-homed hosts.

dispatch_clause

PROTOCOL

The network protocol for which the dispatcher generates a listening endpoint.

ADDRESS

The network protocol address of the endpoint on which the dispatchers listen.

DESCRIPTION

The network description of the endpoint on which the dispatchers listen, including the protocol address.

options_clause

DISPATCHERS

The initial number of dispatchers to start. The default is 1.

SESSIONS

The maximum number of network sessions to allow for each dispatcher. The default is operating system-specific. Most operating systems have a default of 16 KB.

CONNECTIONS

The maximum number of network connections to allow for each dispatcher. The default is operating system-specific.

MULTIPLEX

Enables the Oracle Connection Manager session multiplexing feature.

- The values 1, ON, YES, TRUE, and BOTH indicate that Network Session Multiplex is enabled for both incoming and outgoing network connections.
- The value IN indicates that Network Session Multiplex is enabled for incoming network connections.
- The value OUT indicates that Network Session Multiplexing is enabled for outgoing network connections.
- The values 0, NO, OFF, and FALSE indicate that Network Session Multiplexing is disabled for both incoming and outgoing network connections. This is the default.
- LISTENER

Specifies the network name of an address or address list of the Oracle Net listeners with which the dispatchers will register.

The LISTENER attribute facilitates administration of multi-homed hosts. This attribute specifies the appropriate listeners with which the dispatchers will register. The LISTENER attribute takes precedence over the LOCAL_LISTENER and REMOTE_LISTENER parameters. See "LOCAL_LISTENER" and "REMOTE_LISTENER".

SERVICE

Specifies one or more names by which clients can connect to the dispatchers. The SERVICE attribute takes precedence over any settings configured by using the SRVCTL command-line utility, the GDSCTL command-line utility, or the DBMS_SERVICE PL/SQL package.

INDEX

Use this attribute in an ALTER SYSTEM SET DISPATCHERS statement to indicate which dispatcher configuration you want to modify. (If you specify INDEX in the initialization parameter file, the Oracle Database ignores it.) In an ALTER SYSTEM statement, INDEX specifies the order in which the parameter's values were initialized. The value ranges from 0 (for the first dispatcher configuration) to one less than the total number of dispatcher configurations you define.

For example, if you specify 3 dispatcher configurations in the initialization parameter file, you would modify the third dispatcher configuration by specifying INDEX=2 in the ALTER SYSTEM statement. You could also add another dispatcher configuration in the ALTER SYSTEM statement by specifying INDEX=3.



If INDEX is not specified in the ALTER SYSTEM statement, then the PROTOCOL, ADDRESS, or DESCRIPTION attributes must be specified, and if a dispatcher configuration matching this PROTOCOL, ADDRESS, or DESCRIPTION exists, then that configuration will be modified. Otherwise, a new configuration will be added.

See Also:

- "SHARED_SERVERS"
- Oracle Database Net Services Administrator's Guide and Oracle Database Administrator's Guide for more information on setting this parameter

2.119 DISTRIBUTED_LOCK_TIMEOUT

DISTRIBUTED_LOCK_TIMEOUT specifies the amount of time (in seconds) for distributed transactions to wait for locked resources.

Property	Description
Parameter type	Integer
Default value	60
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	1 to 2 ³¹ - 1
Basic	No



Oracle Database Concepts for more information on data concurrency

2.120 DML LOCKS

DML_LOCKS specifies the maximum number of DML locks—one for each table modified in a transaction.

Property	Description
Parameter type	Integer
Default value	Derived: 4 * TRANSACTIONS
Modifiable	No
Modifiable in a PDB	No
Range of values	20 to unlimited; a setting of 0 disables enqueues
Basic	No



Property	Description
Oracle RAC	You must set this parameter for every instance, and all instances must have positive values or all must be 0.

A **DML lock** is a lock obtained on a table that is undergoing a DML operation (insert, update, delete). The <code>DML_LOCKS</code> value should equal the grand total of locks on tables currently referenced by all users. For example, if three users are modifying data in one table, then three entries would be required. If three users are modifying data in two tables, then six entries would be required.

The default value assumes an average of four tables referenced for each transaction. For some systems, this value may not be enough.

Enqueues are shared memory structures that serialize access to database resources. If you set the value of <code>DML_LOCKS</code> to 0, enqueues are disabled and performance is slightly increased. However, you should be aware of the following restrictions when you set you <code>DML_LOCKS</code> to 0:

- You cannot use DROP TABLE, CREATE INDEX statements
- You cannot use explicit lock statements such as LOCK TABLE IN EXCLUSIVE MODE
- Enterprise Manager cannot run on any instances for which DML LOCKS is set to 0

Oracle holds more locks during parallel DML than during serial execution. Therefore, if your database supports a lot of parallel DML, you may need to increase the value of this parameter.

See Also:

- Oracle Database Concepts for a discussion of lock and enqueue resources needed for parallel DML
- Oracle Database Concepts for more information on data concurrency

2.121 DNFS_BATCH_SIZE

DNFS_BATCH_SIZE controls the number of asynchronous I/O's that can be queued by an Oracle process when Direct NFS Client is enabled.

Property	Description
Parameter type	Integer
Default value	4096
Modifiable	No
Modifiable in a PDB	No
Range of values	0 - 4096
Basic	No

In environments where the NFS server cannot handle a large number of outstanding asynchronous I/O requests, use this parameter to limit the number of I/O's issued by an Oracle foreground process. The recommended setting for this parameter is to start at 128 and increase or decrease it based on NFS server performance.





Oracle Database Performance Tuning Guide for additional information about the ${\tt DNFS}\ {\tt BATCH}\ {\tt SIZE}$ initialization parameter

2.122 DRCP_CONNECTION_LIMIT

DRCP_CONNECTION_LIMIT specifies the maximum number of Database Resident Connection Pooling (DRCP) connections for a PDB.

Property	Description
Parameter type	Integer
Default value	In CDB\$ROOT: 0 (unlimited)
	In a PDB:
	 If a persistent value for SESSIONS was explicitly set for the PDB (ALTER SYSTEM SET SESSIONS = n SCOPE={SPFILE BOTH}), and the PDB was subsequently restarted, then the default value is (10 * SESSIONS). Otherwise, the PDB inherits the value for this parameter from CDB\$ROOT.
Modifiable	ALTER SYSTEM
Woulliable	ALIER SISIEM
Modifiable in a PDB	Yes
Range of values	0 to operating system-dependent
	A value of $\boldsymbol{0}$ specifies that there is no limit on the number of DRCP connections.
Basic	No
Oracle RAC	Different instances can use different values.

DRCP_CONNECTION_LIMIT is a PDB-inherited parameter. The value of this parameter in CDB\$ROOT is not a CDB-wide limit. It is the default value for each PDB.

2.123 DRCP_DEDICATED_OPT

Use $\protect\operatorname{DRCP_DEDICATED_OPT}$ to enable or disable the use of dedicated optimization with Database Resident Connection Pooling (DRCP).

Dedicated optimization causes DRCP to behave like a dedicated server when the number of connections to the DRCP broker is less than the DRCP maximum size.

Property	Description
Parameter type	String
Syntax	DRCP_DEDICATED_OPT = { YES NO }
Default value	NO
Modifiable	ALTER SYSTEM



Property	Description
Modifiable in a PDB	Yes, according to the following guidelines:
	 When the ENABLE_PER_PDB_DRCP initialization parameter is set to true, DRCP_DEDICATED_OPT can be modified only at the PDB level; it cannot be modified for the CDB root. When the ENABLE_PER_PDB_DRCP initialization parameter is set to false, DRCP_DEDICATED_OPT can be modified only for the CDB root; it cannot be modified at the PDB level.
Basic	No
Oracle RAC	Different instances can use different values.

When the value of this parameter is set to YES, the use of dedicated optimization with DRCP is enabled. When it is set to NO, the use of dedicated optimization with DRCP is disabled.

See Also:

- Oracle Database Administrator's Guide for more information about setting this parameter
- "ENABLE PER PDB DRCP"

2.124 DST_UPGRADE_INSERT_CONV

DST_UPGRADE_INSERT_CONV specifies whether internal operators will be allocated on top of TIMESTAMP WITH TIME ZONE (TSTZ) columns of tables which have not been upgraded during the upgrade window of daylight saving time patching for TIMESTAMP WITH TIME ZONE data.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

Values

true

Internal operators will be allocated on top of TSTZ columns of tables which have not been upgraded. This is the default.

false

Internal operators will not be allocated on top of TSTZ columns of tables which have not been upgraded.

When DST_UPGRADE_INSERT_CONV is set to true during the upgrade window of the daylight saving time patching process:

- SELECT queries on tables with TSTZ data which have not been upgraded will use internal
 operators on top of TSTZ columns to present TSTZ data as if they were recorded using the
 new time zone translation rules.
- DML on tables with TSTZ data which have not been upgraded will use internal operators
 on top of TSTZ columns to ensure that the TSTZ data is recorded using the old time zone
 translation rules in order to be consistent with the existing TSTZ data in the same tables.

Note:

Oracle strongly recommends that this parameter is set to true throughout the upgrade window of the daylight saving time patching process. This parameter reduces the performance impact since indexes on TSTZ columns will be disabled whenever internal operators are allocated. If the parameter is set to false, then indexes will be used and this may affect performance of queries against TSTZ data during the DST upgrade window. Turning off this parameter during the upgrade window may corrupt data on disk when DMLs occur for tables with TSTZ data which have not yet been upgraded.

See Also:

Oracle Database Globalization Support Guide for more information about this parameter

2.125 ENABLE AUTOMATIC MAINTENANCE PDB

ENABLE_AUTOMATIC_MAINTENANCE_PDB can be used to enable or disable the running of automated maintenance tasks for all the PDBs in a CDB or for individual PDBs in a CDB.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	The same value should be used for all instances.

Note:

The value of this parameter in CDB\$ROOT (the root of a CDB) has no effect in the root. Automated maintenance tasks are always run in the root, regardless of the setting of this parameter.



By default, the value of <code>ENABLE_AUTOMATIC_MAINTENANCE_PDB</code> is true in CDB\$ROOT (the root container in the CDB) and in the individual PDBs in a CDB. This means that by default, automated maintenance tasks are run for the CDB root and all the PDBs in the CDB.

When you change the value of <code>ENABLE_AUTOMATIC_MAINTENANCE_PDB</code> in the CDB root, the new value takes effect in the root and in all the PDBs in the CDB.

Therefore, if you change the value of <code>ENABLE_AUTOMATIC_MAINTENANCE_PDB</code> in the CDB root to false, the value of <code>ENABLE_AUTOMATIC_MAINTENANCE_PDB</code> is also changed to false in all of the PDBs in the CDB.

You can also change the value of <code>ENABLE_AUTOMATIC_MAINTENANCE_PDB</code> in any of the individual PDBs in a CDB, and the value that is set for each individual PDB will be honored. This enables you to enable or disable automated maintenance tasks for individual PDBs.

See Also:

- "AUTOTASK_MAX_ACTIVE_PDBS" for information about specifying the maximum number of PDBs that can schedule automated maintenance tasks at the same time
- Oracle Database Administrator's Guide for more information about managing automated database maintenance tasks

2.126 ENABLE DDL LOGGING

ENABLE_DDL_LOGGING enables or disables the writing of a subset of data definition language (DDL) statements to a DDL log.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

The DDL log is a file that has the same format and basic behavior as the alert log, but it only contains the DDL statements issued by the database. The DDL log is created only for the RDBMS component and only if the <code>ENABLE_DDL_LOGGING</code> initialization parameter is set to <code>true</code>. When this parameter is set to <code>false</code>, DDL statements are not included in the DDL log, but they are included in the alert log.

The DDL log contains one log record for each DDL statement issued by the database. The DDL log is included in IPS incident packages.

There are two DDL logs that contain the same information. One is an XML file, and the other is a text file. The DDL log is stored in the log/ddl subdirectory of the ADR home.

When ENABLE DDL LOGGING is set to true, the following DDL statements are written to the log:

- ALTER/CREATE/DROP/TRUNCATE CLUSTER
- ALTER/CREATE/DROP FUNCTION
- GRANT
- ALTER/CREATE/DROP INDEX
- ALTER/CREATE/DROP OUTLINE
- ALTER/CREATE/DROP PACKAGE
- ALTER/CREATE/DROP PACKAGE BODY
- ALTER/CREATE/DROP PROCEDURE
- ALTER/CREATE/DROP PROFILE
- REVOKE
- ALTER/CREATE/DROP SEQUENCE
- CREATE/DROP SYNONYM
- ALTER/CREATE/DROP/RENAME/TRUNCATE TABLE
- ALTER/CREATE/DROP TRIGGER
- ALTER/CREATE/DROP TYPE
- ALTER/CREATE/DROP TYPE BODY
- DROP USER
- ALTER/CREATE/DROP VIEW

Note:

The DDL statement written to the log may be truncated. You can use DDL triggers to view the entire DDL statement. See *Oracle Database PL/SQL Language Reference* for more information about DDL triggers.

See Also:

Oracle Database Licensing Information User Manual for licensing information for the ENABLE_DDL_LOGGING initialization parameter.

2.127 ENABLE_DNFS_DISPATCHER

ENABLE DNFS DISPATCHER enables dispatcher support for the Oracle Direct NFS client.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No



Property	Description
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	All instances should use the same value.

When this parameter is set to true in the initialization parameter file, dispatcher processes will be spawned for the Oracle Direct NFS client.

The number of dispatcher processes spawned is calculated as one-eighth of the value of the CPU COUNT parameter.

See Also:
"CPU_COUNT"

2.128 ENABLE_GOLDENGATE_REPLICATION

ENABLE_GOLDENGATE_REPLICATION controls services provided by the RDBMS for Oracle GoldenGate (both capture and apply services).

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	All instances must have the same setting

Set this parameter to true to enable RDBMS services used by Oracle GoldenGate.

This parameter primarily controls supplemental logging required to support logical replication of new data types and operations. The redo log file is designed to be applied physically to a database, therefore the default contents of the redo log file often do not contain sufficient information to allow logged changes to be converted into SQL statements. Supplemental logging adds extra information into the redo log files so that replication can convert logged changes into SQL statements without having to access the database for each change. Previously these extra changes were controlled by the supplemental logging DDL. Now the ENABLE_GOLDENGATE_REPLICATION parameter must also be set to enable the required supplemental logging for any new data types or operations.

All enhancements to supplemental logging required to support logical replication are also controlled by this parameter.

The RDBMS services controlled by this parameter also include (but are not limited to):

- Transparent Data Encryption (including Tablespace Encryption) utilities used by GoldenGate Extract
- Service to read redo logs used by GoldenGate Extract
- Service to suppress triggers used by GoldenGate Replicat
- Service to handle transient duplicate handling used by GoldenGate Replicat
- Service to bypass referential integrity checking used by GoldenGate Replicat
- · Services required to run Oracle GoldenGate in Integrated Extract and Integrated Replicat

Note:

When this database parameter is set to true, you must have a valid Oracle GoldenGate license or an authorized managed cloud client to use the enabled features. GoldenGate licenses or authorized managed cloud clients are required for all 3rd party or end-user tools that are used with this feature. Refer to *Oracle GoldenGate Licensing Information* and ensure that you have obtained the appropriate licensing before setting this parameter to true.

2.129 ENABLE IMC WITH MIRA

ENABLE_IMC_WITH_MIRA enables or disables the In-Memory Column Store and Oracle Data Guard Multi-Instance Redo Apply, at the same time, on an Active Data Guard standby database.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	Different instances can use different values.

The value of this parameter is relevant only on the instance where MRP0 is spawned at the start of Redo Apply on a standby database.

2.130 ENABLE_PER_PDB_DRCP

ENABLE_PER_PDB_DRCP controls whether Database Resident Connection Pooling (DRCP) configures one connection pool for the entire CDB or one isolated connection pool for each PDB.

Property	Description
Parameter type	Boolean
Default value	false



Property	Description
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	Multiple instances must have the same value.

Possible values:

true

DRCP creates one isolated connection pool for each PDB. You can connect to a particular PDB and configure, manage, and monitor its connection pool according to the specific requirements for that PDB. For example, you can set values for the MIN_AUTH_SERVERS, MAX_AUTH_SERVERS, and DRCP_DEDICATED_OPT initialization parameters for a PDB, and you can use the DBMS CONNECTION POOL package to manage the connection pool for a PDB.

When <code>ENABLE_PER_PDB_DRCP</code> is set to true, there is no connection pool for the entire CDB. The values of the <code>MIN_AUTH_SERVERS</code>, <code>MAX_AUTH_SERVERS</code>, and <code>DRCP_DEDICATED_OPT</code> initialization parameters have no meaning in the CDB root and you cannot alter them, nor can you use the <code>DBMS_CONNECTION_POOL_package</code> while connected to the CDB root.

false

DRCP creates one connection pool for the entire CDB. This connection pool is shared by all PDBs. You can set the MIN_AUTH_SERVERS, MAX_AUTH_SERVERS, and DRCP_DEDICATED_OPT initialization parameters only in the CDB root, and you can use the DBMS CONNECTION POOL only while connected to the CDB root.

See Also:

- "MIN_AUTH_SERVERS"
- "MAX_AUTH_SERVERS"
- "DRCP_DEDICATED_OPT"
- Oracle Database PL/SQL Packages and Types Reference for more information on the DBMS CONNECTION POOL package

2.131 ENABLE PLUGGABLE DATABASE

ENABLE PLUGGABLE DATABASE is a bootstrap initialization parameter to create a CDB.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No



Property	Description
Range of values	true false
Basic	No

This parameter enables a database in NOMOUNT startup mode to know that it is a CDB.



See *Oracle Multitenant Administrator's Guide* for more information about CDBs and PDBs.

2.132 ENABLED_PDBS_ON_STANDBY

ENABLED_PDBS_ON_STANDBY specifies which pluggable databases (PDBs) to replicate on an Oracle Data Guard standby database.

Property	Description
Parameter type	String
Syntax	<pre>ENABLED_PDBS_ON_STANDBY = PDB-list</pre>
Default value	* if no value is specified for this parameter in the init.ora file
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	All instances should use the same value.



This parameter is meaningful only on standby databases. Its settings are ignored on a primary database.

PDB-list accepts a list of PDB names represented by glob patterns such as "PDB?" or "PDB*a", "PDB2". The glob pattern rules are like those of UNIX shells. The asterisk (*) and question mark (?) wildcard characters are supported. The ? wildcard character matches exactly one unknown character, and the * wildcard character matches any number of unknown characters. Also, the minus sign (-) character can be used as the first character in a PDB name to indicate that the PDB should be excluded on the standby database.

The following characters are valid in a PDB name: alphanumeric characters, underscore (_), number sign (#), and dollar sign (\$). No other characters are valid. Oracle removes double quotation before processing the PDB name. Therefore you cannot use double quotation marks to embed other characters in the name. The PDB name is case insensitive. These are the same naming conventions as for a database name (as described in the DB_NAME initialization parameter description).

These rules apply for the <code>ENABLED_PDBS_ON_STANDBY</code> parameter:



- Any PDB names that match glob patterns starting with minus sign (-) will not be part of the standby database.
- If a PDB name matches several patterns in the *PDB-list* at the same time, the rightmost matched pattern has the highest precedence.
- If this parameter is not specified in the init.ora file, it is assumed that "*" is specified by default.
- If one or more patterns are specified in this parameter, it is assumed that "-*" is implicitly specified in the leftmost position. For example, ENABLED_PDBS_ON_STANDBY="PDB1", "PDB2" is equivalent to ENABLED_PDBS_ON_STANDBY="-*", "PDB1", "PDB2". It means that a PDB name that does not match any patterns is not allowed to be part of the standby.

Examples

These examples assumes that nine new PDBs named PDB1A, PDB1B, PDB1C, PDB2A, PDB2B, PDB2C, PDB3A, PDB3B, and PDB3C are being added to the primary database.

- 1. If STANDBY1 sets ENABLED_PDBS_ON_STANDBY="*" on STANDBY1, then all nine PDBs will be created on STANDBY1.
- 2. If STANDBY1 sets ENABLED_PDBS_ON_STANDBY="PDB1*" on STANDBY1, then PDB1A, PDB1B, and PDB1C will be created on STANDBY1.
- 3. If STANDBY1 sets ENABLED_PDBS_ON_STANDBY="PDB*A" on STANDBY1, then PDB1A, PDB2A, and PDB3A will be created on STANDBY1.
- 4. If STANDBY1 sets ENABLED_PDBS_ON_STANDBY="PDB1*", "-PDB*A" on STANDBY1, then PDB1B and PDB1C will be created on STANDBY1.
- 5. If STANDBY1 sets ENABLED_PDBS_ON_STANDBY="*", "-PDB*A", "-PDB*B" on STANDBY1, then PDB1C, PDB2C and PDB3C will be created on STANDBY1. All other PDBs match "*", and the rightmost pattern has the higher precedence, so they are excluded on the standby.
- 6. If STANDBY1 sets ENABLED_PDBS_ON_STANDBY="*", "-PDB*A", "PDB2A" on STANDBY1, then PDB1A and PDB3A are excluded, but all other PDBs including PDB2A will be created

2.133 ENCRYPT NEW TABLESPACES

ENCRYPT NEW TABLESPACES specifies whether to encrypt newly created user tablespaces.

Property	Description
Parameter type	String
Syntax	<pre>ENCRYPT_NEW_TABLESPACES = { CLOUD_ONLY ALWAYS DDL }</pre>
Default value	CLOUD_ONLY
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value should be specified for all instances.

The values that can be specified for the <code>ENCRYPT_NEW_TABLESPACES</code> parameter have the following meanings:

CLOUD ONLY:



When a user tablespace is created in the Oracle Cloud, it will be transparently encrypted if the <code>ENCRYPTION</code> ... <code>ENCRYPT</code> clause for the SQL <code>CREATE</code> <code>TABLESPACE</code> statement is not specified. The encryption algorithm is determined by the value of the <code>TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM</code> initialization parameter. When a user tablespace is created in an on-premises database, the <code>ENCRYPTION</code> clause of the <code>CREATE</code> <code>TABLESPACE</code> statement determines if the tablespace is encrypted.

CLOUD ONLY is the default value.

ALWAYS:

Whether the user tablespace is created in the Oracle Cloud or in an on-premises database, the tablespace will be transparently encrypted if the ENCRYPTION ... ENCRYPT clause is not specified in the CREATE TABLESPACE statement. The encryption algorithm is determined by the value of the TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM initialization parameter.

• DDL:

Whether the user tablespace is created in the Oracle Cloud or in an on-premises database, the CREATE TABLESPACE statement follows the specified DDL. If no ENCRYPTION clause is specified, then the tablespace will not be encrypted. If the ENCRYPTION USING ... ENCRYPT clause is specified, then the specified algorithm will be used to encrypt the tablespace. If ENCRYPTION ENCRYPT is specified, but no algorithm is specified, then the encryption algorithm is determined by the value of the TABLESPACE ENCRYPTION DEFAULT ALGORITHM initialization parameter.

Note:

ENCRYPT_NEW_TABLESPACES is deprecated in Oracle Database 23ai and may be removed in a future release. Oracle recommends that you use the TABLESPACE ENCRYPTION initialization parameter instead.

See Also:

- TABLESPACE ENCRYPTION
- TABLESPACE ENCRYPTION DEFAULT ALGORITHM
- Oracle Database Transparent Data Encryption Guide for more information about tablespace encryption
- Oracle Database SQL Language Reference for more information about the CREATE TABLESPACE statement



2.134 ERROR_MESSAGE_DETAILS

Use <code>ERROR_MESSAGE_DETAILS</code> to control whether the database raises an additional stacked error message, with further details about the data values encountering the error, when displaying error messages.

Property	Description
Parameter type	String
Syntax	<pre>ERROR_MESSAGE_DETAILS = { ON OFF DISALLOWED }</pre>
Default value	ON
Modifiable	ALTER SESSION can be used to set this parameter to ON or OFF.
	ALTER SYSTEM can be used to set this parameter to ON, OFF, or DISALLOWED.
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different instances can use different values.

This parameter allows you to instruct the database to show or omit explanatory details about data values causing an error. Providing this information usually makes it easier for the user to understand why the error occurred. If this is not desired, this parameter can be used to omit these additional details.

The detailed error message is being raised as a stacked error, meaning that applications will see the root cause error as the main error and will have to further traverse the stack to gain insights into the details. SQL command line tools will automatically traverse the entire stack and print the entire error message stack.

You can set this parameter for the database instance or PDB with the ALTER SYSTEM statement, or for the current session with the ALTER SESSION statement, using these values:

- ON: The database will provide additional explanatory details when displaying error messages. This is the default.
- OFF: The database will not provide additional explanatory details. This is the behavior prior to Oracle Database 23ai.

Additionally, you can set the following value only at the database instance or PDB level with the ALTER SYSTEM statement:

• DISALLOWED: The database will not provide additional explanatory details, and users cannot override this behavior for the current session with the ALTER SESSION statement.



This parameter is available starting with Oracle Database 23ai.

2.135 EVENT

EVENT is a parameter used only to debug the system.

Property	Description
Parameter type	String
Default value	There is no default value.
Modifiable	No
Modifiable in a PDB	No
Basic	No

Do not alter the value of this parameter except under the supervision of Oracle Support Services staff.

2.136 EXTERNAL_KEYSTORE_CREDENTIAL_LOCATION

EXTERNAL_KEYSTORE_CREDENTIAL_LOCATION specifies the location of the secure external keystore.

Property	Description
Parameter type	String
Syntax	EXTERNAL_KEYSTORE_CREDENTIAL_LOCATION = file-location
Default value	No default value
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	This parameter can be set on each instance. Oracle recommends that if the instances have a shared location, then use a directory on the shared location for this parameter and set the same value on all Oracle RAC instances. If the database does not have a shared location, then each instance of the database will have its own directory and the value should be set per-instance.

TDE keystore credentials can be stored in a secure external keystore to automate Transparent Data Encryption (TDE) operations.

To automate the setting of the Master Key in a newly-provisioned PDB, set <code>EXTERNAL_KEYSTORE_CREDENTIAL_LOCATION</code> so that the CDB knows the location of the secure external keystore.

When a file specification is set with this parameter, the secure external keystore is looked for in that location. If the secure external keystore is not found in that location, then this parameter is ignored.



Oracle Database Transparent Data Encryption Guide for more information about using TDE



2.137 FAL_CLIENT

FAL_CLIENT specifies the FAL (fetch archive log) client name that is used by the FAL service, configured through the FAL_SERVER initialization parameter, to refer to the FAL client.

Property	Description
Parameter type	String
Syntax	FAL_CLIENT = string
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

The value is an Oracle Net service name, which is assumed to be configured properly on the FAL server system to point to the FAL client (standby database).



Oracle Data Guard Concepts and Administration for more information about FAL server

2.138 FAL_SERVER

FAL SERVER specifies the FAL (fetch archive log) server for a standby database.

Property	Description
Parameter type	String
Syntax	FAL_SERVER = string
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

The value is an Oracle Net service name, which is assumed to be configured properly on the standby database system to point to the desired FAL server.



Oracle Data Guard Concepts and Administration for more information about FAL server



2.139 FAST_START_MTTR_TARGET

FAST_START_MTTR_TARGET enables you to specify the number of seconds the database takes to perform crash recovery of a single instance.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 to 3600 seconds
Basic	No
Oracle RAC	Multiple instances can have different values, and you can change the values at run time.

When specified, FAST START MTTR TARGET is overridden by LOG CHECKPOINT INTERVAL.

See Also:

- Oracle Database Performance Tuning Guide for an example of setting the lower bound for this parameter
- Oracle Database Performance Tuning Guide for an example of setting the upper bound for this parameter
- Oracle Database Performance Tuning Guide for an example of selecting a preliminary value for this parameter

2.140 FAST_START_PARALLEL_ROLLBACK

FAST_START_PARALLEL_ROLLBACK specifies the degree of parallelism used when recovering terminated transactions.

Property	Description
Parameter type	String
Syntax	FAST_START_PARALLEL_ROLLBACK = { HIGH LOW FALSE }
Default value	LOW
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Terminated transactions are transactions that are active before a system failure. If a system fails when there are uncommitted parallel DML or DDL transactions, then you can speed up transaction recovery during startup by using this parameter.

Values

FALSE

Parallel rollback is disabled

• LOW

Limits the maximum degree of parallelism to 2 * CPU COUNT

HIGH

Limits the maximum degree of parallelism to 4 * CPU COUNT

If you change the value of this parameter, then transaction recovery will be stopped and restarted with the new implied degree of parallelism.



Oracle Database VLDB and Partitioning Guide for more information about this parameter

2.141 FILE_MAPPING

FILE MAPPING specifies whether file mapping is enabled.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	true false
Basic	No



The ${\tt FILE_MAPPING}$ initialization parameter is deprecated. It is still supported for backward compatibility.

The FMON background process is started to manage the mapping information when file mapping is enabled.

See Also:

Oracle Database Administrator's Guide for more information about the file mapping feature



2.142 FILEIO_NETWORK_ADAPTERS

FILEIO_NETWORK_ADAPTERS specifies a list of network adapters that can be used to access the disk storage.

Property	Description
Parameter type	String
Syntax	<pre>FILEIO_NETWORK_ADAPTERS = adapter_name [, adapter_name]</pre>
Default value	There is no default value.
Modifiable	No
Modifiable in a PDB	No
Range of values	One or more network adapter names, separated by commas
Basic	No

On platforms where the database files reside in network attached storage, this parameter provides the storage access library the list of network adapters that can be used to access the storage.

The network adapter name is a fully qualified address name of the network card that can be accessed through the host name database or using the Network Information Service. The components of the adapter name are separated by periods. For example, the following is a fully qualified adapter name:

ib1.oracle.com

2.143 FILESYSTEMIO_OPTIONS

FILESYSTEMIO_OPTIONS specifies I/O operations for file system files. This parameter is platform-specific and has a default value that is suitable for a particular platform.

Property	Description
Parameter type	String
Syntax	FILESYSTEMIO_OPTIONS = { ASYNCH DIRECTIO SETALL NONE }
Default value	Varies by database version and operating system.
Modifiable	No
Modifiable in a PDB	No
Basic	No

Values:

- ASYNCH Enable asynchronous I/O on file system files, which has no timing requirement for transmission.
- DIRECTIO Enable direct I/O on file system files, which bypasses the buffer cache.
- SETALL Enable both asynchronous and direct I/O on file system files.
- NONE Disable both asynchronous and direct I/O on file system files.



See Also:

- Your platform-specific documentation for more details
- Oracle Database Performance Tuning Guide for information about setting this parameter before running I/O calibration

2.144 FIXED DATE

FIXED_DATE enables you to set a constant date that SYSDATE will always return instead of the current date.

Property	Description
Parameter type	String
Syntax	FIXED_DATE = [YYYY-MM-DD-HH24:MI:SS (or the default Oracle date format) NONE]
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

To undo a fixed date setting, specify FIXED_DATE=NONE. This parameter is useful primarily for testing. The value can be in the format shown above or in the default Oracle date format, without a time.

2.145 FORWARD LISTENER

FORWARD_LISTENER specifies the name of a listener to which a connection must be forwarded by an existing set of remote listeners.

Property	Description
Parameter type	String
Syntax	FORWARD_LISTENER = listener-name
Default value	NULL
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	A different value can be set on different instances.

The parameter is useful when it is difficult to change an existing client connect string, for example, after an offsite database has been moved into the Oracle Cloud. With the FORWARD_LISTENER parameter, clients can continue to connect to their offsite listener, and the offsite listener forwards the connection to the Oracle Cloud listener.

Once a forward listener has been configured through the FORWARD_LISTENER parameter, the LOCAL LISTENER parameter can be cleared by setting its value to -oracle-none- so that all the

connections coming to an existing set of remote listeners configured through the REMOTE_LISTENER parameter are forwarded only to listeners configured through FORWARD LISTENER.

The FORWARD LISTENER parameter can also be specified in the LISTENER NETWORKS parameter.

Examples

The following setup can be used to forward all the database connections coming to an existing customer SCAN listener to the Oracle Cloud SCAN listener.

REMOTE_LISTENER=customer-scan FORWARD_LISTENER=oracle-cloud-scan LOCAL_LISTENER=oracle-none-

See Also:

- "LOCAL_LISTENER"
- "REMOTE_LISTENER"
- "LISTENER NETWORKS"
- Oracle Clusterware Administration and Deployment Guide for information about SCAN listeners

2.146 GCS_SERVER_PROCESSES

GCS_SERVER_PROCESSES specifies the number of background GCS server processes (LMSn and LMnn) to serve the inter-instance traffic among Oracle RAC instances.

If there is high Cache Fusion congestion on the system, the number of GCS server processes could be increased to reduce Cluster wait time. Additionally, the number of GCS server processes could be increased to reduce the application brownout time associated with Cluster reconfiguration and Dynamic Re-mastering (DRM).

Property	Description
Parameter type	Integer



Property	Description
Default value	Oracle calculates the default value as follows (in order of precedence):
	1. If CLUSTER_DATABASE is set to false, then 0
	2. If Oracle ASM, then 1
	3. If 1 - 3 CPUS, then 1
	4. If 4 - 15 CPUs , then 2
	If 128 or more CPUs and SGA is 100 GB or more, then (CPUs / 6). If the result includes a fraction, then the fraction is disregarded.
	 Otherwise, the value is 2 + (CPUs / 32). If the result includes a fraction, then the fraction is disregarded. For example, if you have 20 CPUs, then 2 + (20 / 32) would equal 2 GCS processes.
	7. On NUMA-enabled systems with 32 or more CPUs, the value is rounded up to a multiple of the number of NUMA processor groups, with a limit of (CPUs / 4) rounded down to a multiple of the number of NUMA processor groups.
Modifiable	No
Modifiable in a PDB	No
Range of values	0 if Oracle RAC is disabled (CLUSTER_DATABASE is set to false)
	1 to 100 if Oracle RAC is enabled (CLUSTER_DATABASE is set to true)
Basic	No
Oracle RAC	Multiple instances can have different values.

GCS server processes are only seen in an Oracle RAC environment.

2.147 GLOBAL_NAMES

 ${\tt GLOBAL_NAMES}$ specifies whether a database link is required to have the same name as the database to which it connects.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

If the value of <code>GLOBAL_NAMES</code> is <code>false</code>, then no check is performed. If you use or plan to use distributed processing, then Oracle recommends that you set this parameter to <code>true</code> to ensure the use of consistent naming conventions for databases and links in a networked environment.





Oracle Database Administrator's Guide for more information on setting this parameter

2.148 GLOBAL_TXN_PROCESSES

GLOBAL_TXN_PROCESSES specifies the initial number of GTXn background processes (GTX0, ... GTX9 and GTXa, ... GTXj) per instance to support global (XA) transactions in an Oracle RAC environment.

Property	Description
Parameter type	Integer
Default value	1
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 to 20
Basic	No
Oracle RAC	Multiple instances can have different values.

If you want to disable the GTXn background processes, then you must set <code>GLOBAL_TXN_PROCESSES</code> to 0 in your parameter file. Setting this parameter to 0 will disable the XA support on an Oracle RAC database. Error <code>ORA-55712</code> will be returned if you try to run XA transactions on an Oracle RAC database with this parameter set to 0. You can change the setting to a nonzero value at run time to turn on the support for XA.

GLOBAL_TXN_PROCESSES is useful for systems that process global (XA) transactions heavily. You do not need to specify a value for this parameter since Oracle Database automatically determines the number of processes and autotunes them, as necessary. GTXn background processes are only seen in an Oracle RAC environment.



Oracle Database Development Guide for more information about this parameter

2.149 GROUP_BY_POSITION_ENABLED

GROUP_BY_POSITION_ENABLED controls whether a positive integer in a GROUP BY clause is treated as the ordinal position of an expression in the select list.

Property	Description
Parameter type	Boolean
Default value	FALSE
Modifiable	ALTER SESSION, ALTER SYSTEM



Property	Description
Modifiable in a PDB	Yes
Range of values	TRUE FALSE
Basic	No
Oracle RAC	The same value must be used on all instances.

In SQL SELECT statements, you can specify the GROUP BY clause when you want to group the selected rows that share a common value for an expression and return a single row of summary information for each group.

In previous releases, if you wanted to group rows according to the value of an expression in the select list, you specified the expression as it appeared in the select list. For example:

```
SELECT department_id, COUNT(*)
  FROM employees
  GROUP BY department id;
```

Starting with Oracle Database 23ai, when <code>GROUP_BY_POSITION_ENABLED</code> is set to <code>TRUE</code>, you can specify <code>GROUP BY position</code>, where <code>position</code> is a positive integer that represents the ordinal position of an expression in the select list. For example, the following <code>SELECT</code> statement groups the selected rows by the value of <code>department id</code>:

```
SELECT department_id, COUNT(*)
  FROM employees
  GROUP BY 1;
```

The following SELECT statement groups the selected rows, first by the value of manager_id, and then by the value of hired year:

```
SELECT manager_id, EXTRACT(YEAR FROM hire_date) AS hired_year, COUNT(*)
FROM employees
GROUP BY 1, 2;
```

Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

Oracle Database SQL Language Reference for more information about the $\tt GROUP\ BY$ clause of the $\tt SELECT\ statement$

2.150 HASH AREA SIZE

HASH AREA SIZE specifies the maximum amount of memory, in bytes, to be used for hash joins.

Property	Description
Parameter type	Integer
Default value	Derived: 2 * SORT_AREA_SIZE
Modifiable	ALTER SESSION
Modifiable in a PDB	No
Range of values	0 to operating system-dependent
Basic	No

Note:

Oracle does not recommend using the <code>HASH_AREA_SIZE</code> parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting <code>PGA_AGGREGATE_TARGET</code> instead. <code>HASH_AREA_SIZE</code> is retained for backward compatibility.

HASH_AREA_SIZE is relevant to parallel execution operations and to the query portion of DML or DDL statements.

See Also:

Oracle Database Concepts for information on hash joins in general

2.151 HEARTBEAT_BATCH_SIZE

Use <code>HEARTBEAT_BATCH_SIZE</code> to configure the heartbeat batch size, which is the number of heartbeats sent during one heartbeat period.

Property	Description
Parameter type	Integer
Default value	5
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	2 to 100
Basic	No
Oracle RAC	Multiple instances must have the same value.

2.152 HEAT_MAP

Use <code>HEAT_MAP</code> to enable or disable both the Heat Map and Automatic Data Optimization (ADO) features.

Property	Description
Parameter type	String
Syntax	HEAT_MAP = { ON OFF }
Default value	OFF
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	If specified, use the same value on all instances

Setting the ${\tt HEAT_MAP}$ initialization parameter to ${\tt ON}$ causes the database to track read and write access of all segments, as well as modification of database blocks, due to DMLs and DDLs. These activities are not tracked for objects in the ${\tt SYSTEM}$ and ${\tt SYSAUX}$ tablespaces.

See Also:

- Oracle Database VLDB and Partitioning Guide for more information about enabling and disabling the Heat Map feature
- Oracle Database VLDB and Partitioning Guide for more information about Heat Map tracking

2.153 HI_SHARED_MEMORY_ADDRESS

HI_SHARED_MEMORY_ADDRESS specifies the starting address at run time of the system global area (SGA).

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Modifiable in a PDB	No
Basic	No

This parameter is ignored on platforms that specify the SGA's starting address at linktime.

On 64-bit platforms, use <code>HI_SHARED_MEMORY_ADDRESS</code> to specify the high-order 32 bits of a 64-bit address. Use <code>SHARED_MEMORY_ADDRESS</code> to specify the low-order 32 bits of the address (see "SHARED_MEMORY_ADDRESS"). If both parameters are 0 or unspecified, the SGA address defaults to a platform-specific location.

2.154 HS_AUTOREGISTER

HS_AUTOREGISTER enables or disables automatic self-registration of Heterogeneous Services (HS) agents.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	true false
Basic	No

When enabled, information is uploaded into the server's data dictionary to describe a previously unknown agent class or a new agent version.

Oracle recommends that you set this parameter to true. Oracle incurs less overhead when establishing subsequent connections through the same agent if self-registered information is available in the server's data dictionary.



Oracle Database Heterogeneous Connectivity User's Guide for more information about using HS agents

2.155 HYBRID_READ_ONLY

HYBRID READ ONLY enables or disables hybrid read-only mode for a pluggable database (PDB).

Property	Description
Parameter type	Boolean
Default value	FALSE
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	TRUE FALSE
Basic	No
Oracle RAC	Different instances can use different values.

Hybrid read-only mode allows a PDB to operate in read/write mode or read-only mode, depending on the type of user that is connected to the PDB. The PDB operates in read/write mode for CDB common users, and read-only mode for application common users and local users. Hybrid read-only mode allows CDB common users to perform patching and maintenance operations on an open PDB, while application common users and local users can use the PDB as if it were open in read-only mode.

- To enable hybrid read-only mode for a PDB, connect to the PDB and set
 HYBRID_READ_ONLY to TRUE. The PDB must be in READ WRITE mode before you enable
 hybrid read-only mode.
- To disable hybrid read-only mode for a PDB, connect to the PDB and set HYBRID READ ONLY to FALSE.



You can determine whether a PDB is in hybrid read-only mode by querying the IS_HYBRID_READ_ONLY column of the V\$CONTAINER_TOPOLOGY view.

When a PDB is in hybrid read-only mode, the OPEN_MODE column in the V\$CONTAINERS and V\$PDBS views reflects the mode for the querying user, that is, the OPEN_MODE column returns READ WRITE when queried by a CDB common user, and READ ONLY when queried by an application common user or a local user.

Note:

When you set the value of this parameter in a PDB to TRUE, the TRUE value will not persist when the PDB is closed and reopened. You can remedy this in one of the following ways:

- Specify HYBRID READ ONLY when opening the PDB:
 - ALTER PLUGGABLE DATABASE pdb name OPEN HYBRID READ ONLY;
- After opening the PDB, connect to the PDB and set this parameter to TRUE again:

ALTER SYSTEM SET HYBRID READ ONLY = TRUE;

Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

"V\$CONTAINER_TOPOLOGY"

2.156 IDENTITY_PROVIDER_CONFIG

Use IDENTITY_PROVIDER_CONFIG to configure the database to use Microsoft Azure Active Directory (Azure AD) as an external identity provider.

Property	Description
Parameter type	String



Property	Description
Syntax	<pre>IDENTITY_PROVIDER_CONFIG = '{ "application_id_uri" : "string", "tenant_id" : "string", "app_id" : "string" }'</pre>
	Where:
	 The application_id_uri value is obtained from the registered application, to be mapped in the JWT "aud" (audience) claim. It must be domain qualified and start with "https://" to specify the OCI database to access. The tenant_id value is obtained from the tenant configuration and is also available on the database application registration overview page. The app_id value is obtained from the registered resource application and is also available on the database application registration overview page.
Default value	None
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value must be used on all instances.

This parameter is effective only when the database is enabled to use Azure AD as an external identity provider, that is, when the <code>IDENTITY_PROVIDER_TYPE</code> initialization parameter is set to <code>AZURE AD</code>. You must use this parameter to specify Azure AD configuration settings.

Example:

```
ALTER SYSTEM SET IDENTITY_PROVIDER_CONFIG =

'{ "application_id_uri" : "https://example.com/pdb1",
    "tenant_id" : "111a1111-a11a-111a-111111111111a",
    "app_id" : "11aa1a1-aaaa-1111-1111-1111aa111111" }'
SCOPE=BOTH;
```

Note:

Setting the value of this parameter is only one step in a series of steps for configuring the database to use Azure AD. Refer to *Oracle Database Security Guide* for the complete steps before you set this parameter.

Note:

For additional information about using this parameter with Oracle Autonomous Database, refer to "Use Azure Active Directory (Azure AD) with Autonomous Database".



This parameter is not available to update directly for Oracle Autonomous Database Serverless.

Note:

This parameter is available in Oracle Database 19c, starting with version 19.16, and in Oracle Database 23ai. This parameter is not available in Oracle Database 21c.

See Also:

"IDENTITY_PROVIDER_TYPE"

2.157 IDENTITY_PROVIDER_TYPE

IDENTITY_PROVIDER_TYPE specifies an external identity provider for the database.

Property	Description
Parameter type	String
Syntax	<pre>IDENTITY_PROVIDER_TYPE = { AZURE_AD OCI_IAM NONE }</pre>
Default value	NONE
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value must be used on all instances.

Values:

- AZURE_AD Enables the database to use Microsoft Azure Active Directory (Azure AD) as an external identity provider.
 - When you specify this value, you must also specify configuration settings for Azure AD by setting the <code>IDENTITY_PROVIDER_CONFIG</code> initialization parameter.
- OCI_IAM Enables the database to use Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) as an external identity provider.
 - When you specify this value, there is no need to specify configuration settings with the <code>IDENTITY_PROVIDER_CONFIG</code> initialization parameter.
- NONE The database will not use Azure AD or OCI IAM as an external identity provider.



Note:

Setting the value of this parameter is only one step in a series of steps for configuring an external identity provider for the database. Refer to *Oracle Database Security Guide* for the complete set of steps for configuring Azure AD or OCI IAM before you set this parameter value.

Note:

The following resources offer additional information about using this parameter with Oracle Autonomous Database:

- "Use Azure Active Directory (Azure AD) with Autonomous Database"
- "Use Identity and Access Management (IAM) Authentication with Autonomous Database"

Note:

This parameter is not available to update directly for Oracle Autonomous Database Serverless.

Note:

This parameter is available in Oracle Database 19c, starting with version 19.16, and in Oracle Database 23ai. This parameter is not available in Oracle Database 21c.

See Also:

"IDENTITY PROVIDER CONFIG"

2.158 IFILE

Use IFILE to embed another parameter file within the current parameter file.

Property	Description
Parameter type	Parameter file
Syntax	<pre>IFILE = parameter_file_name</pre>
Default value	There is no default value.
Modifiable	No
Modifiable in a PDB	No



Property	Description
Range of values	Valid parameter filenames
Basic	No
Oracle RAC	Multiple instances can have different values.

For example:

IFILE = COMMON.ORA

You can have up to three levels of nesting. In this example, the file COMMON.ORA could contain a second IFILE parameter for the file COMMON2.ORA, which could contain a third IFILE parameter for the file GCPARMS.ORA. You can also include multiple parameter files in one parameter file by listing IFILE several times with different values:

```
IFILE = DBPARMS.ORA
IFILE = GCPARMS.ORA
IFILE = LOGPARMS.ORA
```



You must list multiple entries on contiguous lines of the parameter file.

2.159 IGNORE_SESSION_SET_PARAM_ERRORS

IGNORE_SESSION_SET_PARAM_ERRORS controls whether the database ignores errors when clients attempt to modify session parameter values.

Property	Description
Parameter type	String
Syntax	<pre>IGNORE_SESSION_SET_PARAM_ERRORS = parameter_name [, parameter_name]</pre>
Default value	None
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Multiple instances should use the same value.

The <code>IGNORE_SESSION_SET_PARAM_ERRORS</code> parameter allows you to change the outcome of the following scenario:

- 1. An Oracle database uses PDB lockdown profiles to prevent certain initialization parameters from being modified with the ALTER SESSION statement.
- 2. A database client attempts to modify one of those parameters with the ALTER SESSION statement. The client has no knowledge that the parameter is locked down.
- The database returns an error to the client.



To prevent the database from returning an error in the preceding scenario, set the value of <code>IGNORE_SESSION_SET_PARAM_ERRORS</code> to a comma-separated list of locked down initialization parameters. Thereafter, if a client attempts to modify one of those parameters with the <code>ALTER SESSION</code> statement, the database will ignore the modification attempt and indicate to the client that the operation was successful.

You can set the IGNORE_SESSION_SET_PARAM_ERRORS parameter while connected to a specific PDB, or you can apply it to all PDBs by setting it in the CDB root.

Example

The CURSOR_INVALIDATION and CURSOR_SHARING parameters are locked down in one of the PDBs in your database. If a client application attempts to modify either of these parameters, you want the database to ignore the request and allow the application to continue running without incurring an error. To achieve this, you can connect to the PDB and set the IGNORE SESSION SET PARAM ERRORS parameter. For example:

ALTER SYSTEM SET IGNORE_SESSION_SET_PARAM_ERRORS = CURSOR_INVALIDATION, CURSOR SHARING;

2.160 INMEMORY ADG ENABLED

INMEMORY_ADG_ENABLED indicates whether in-memory for Active Data Guard is enabled in addition to the in-memory cache size.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	The same value should be used on all instances. If the value for this parameter is changed on one instance, the parameter's value should be changed to the same value on the other instances.

For Active Data Guard, media recovery needs to retrieve in-memory objects when applying redo and to invalidate the related objects after the query advance. This parameter controls whether media recovery does the retrieving and invalidating.

This parameter should be set on standby databases. The value of this parameter is meaningless on a primary database.

If the standby is an Oracle RAC instance, then all the different standby instances must have this parameter set to the same value.

This parameter is a dynamic system-modifiable parameter whose value can be changed only when Active Data Guard recovery is not running.

If you start multi-instance redo apply (MIRA) Active Data Guard recovery, all instances involved in MIRA must have the parameter set to the same value, otherwise MIRA will signal an error.

Example

If the standby is an Oracle RAC instance, this parameter should be set to the same value on all the Oracle RAC instances:

SQL> alter system set inmemory_adg_enabled=true sid='*';

See Also:

- "INMEMORY_SIZE"
- Oracle Database In-Memory Guide for more information about deploying an In-Memory Column Store (IM column store) in an Active Data Guard environment

2.161 INMEMORY_AUTOMATIC_LEVEL

INMEMORY_AUTOMATIC_LEVEL is used to enable the Automatic In-Memory feature, which automates the management of the In-Memory Column Store (IM column store) to help ensure that the working data set is in the IM column store at all times.

Property	Description
Parameter type	String
Syntax	<pre>INMEMORY_AUTOMATIC_LEVEL = { LOW MEDIUM HIGH OFF }</pre>
Default value	OFF
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	All instances should use the same value.

Typically, among all the IM enabled segments, only a subset is actively queried at any time. This subset is known as the working data set. The working data set is expected to change over time for many applications.

INMEMORY_AUTOMATIC_LEVEL attempts to keep the working data set in the IM column store at all times by moving segments in and out of the IM column store based on access patterns.

Oracle recommends that you provision enough memory for the working data set to fit in the IM column store.

The following values can be set:

- Low: The database evicts cold segments from the IM column store when it is under memory pressure.
- MEDIUM: The database evicts cold segments from the IM column store when it is under memory pressure. This level includes an additional optimization that ensures that any hot segment that was not populated because of memory pressure is populated first.
- HIGH: All segments in the database are enabled for the IM column store, except segments belonging to Oracle-maintained users and segments from external tables. Automatic In-

Memory manages all segments enabled for the IM column store, keeping hot segments in the IM column store and evicting cold segments. This level includes an additional optimization that ensures that any hot segment that was not populated because of memory pressure is populated first.

• OFF: Automatic In-Memory is disabled. This value returns the IM column store to the behavior that existed prior to Oracle Database 18c, when the Automatic In-Memory features was introduced. If you do not expect a stable working data set, set the parameter to OFF. This is the default value.

See Also:

- Oracle Database Licensing Information User Manual to learn which database offerings support the Automatic In-Memory feature
- Oracle Database In-Memory Guide for more information about configuring the Automatic In-Memory feature

2.162 INMEMORY CLAUSE DEFAULT

INMEMORY_CLAUSE_DEFAULT enables you to specify a default In-Memory Column Store (IM column store) clause for new tables and materialized views.

Note:

The setting of his parameter has no effect on In-Memory external tables or hybrid partitioned tables.

Property	Description
Parameter type	String
Syntax	<pre>INMEMORY_CLAUSE_DEFAULT = '[INMEMORY] [NO INMEMORY] [other- clauses]'</pre>
Syntax	<pre>other-clauses::= [compression-clause] [priority-clause] [rac-clause]</pre>
Syntax	<pre>compression-clause::= NO MEMCOMPRESS MEMCOMPRESS FOR { DML QUERY [LOW HIGH] CAPACITY [LOW HIGH] }¹</pre>
Syntax	<pre>priority-clause::= PRIORITY { LOW MEDIUM HIGH CRITICAL NONE }¹</pre>
Syntax	<pre>rac-clause::= [distribute-clause] [duplicate-clause]¹</pre>
Syntax	distribute-clause::= DISTRIBUTE [AUTO BY ROWID RANGE] 1
Syntax	<pre>duplicate-clause::= NO DUPLICATE DUPLICATE [ALL] 1</pre>
Default value	An empty string



Property	Description
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	All instances should use the same value

¹ See Table 2-2 for more information about this clause.

If the INMEMORY_CLAUSE_DEFAULT parameter is unset or set to an empty string (the default), only tables and materialized views explicitly specified as INMEMORY will be populated into the IM column store. Setting the value of the INMEMORY_CLAUSE_DEFAULT parameter to NO INMEMORY has the same effect as setting it to the default value.

If the INMEMORY_CLAUSE_DEFAULT parameter is set, then any *newly created* table or materialized view specified as INMEMORY will inherit unspecified attributes from this parameter. This can force certain in-memory options by default that are not explicitly specified in the syntax. For example, if the INMEMORY_CLAUSE_DEFAULT parameter is set to MEMCOMPRESS FOR CAPACITY LOW and a table is created as INMEMORY PRIORITY HIGH, then the table is treated as if it was declared as INMEMORY MEMCOMPRESS FOR CAPACITY LOW PRIORITY HIGH.

If INMEMORY is specified as part of this parameter, then all newly created tables and materialized views will be populated into the IM column store, except tables and materialized views explicitly specified as NO INMEMORY. For example, if this parameter is set to INMEMORY MEMCOMPRESS FOR CAPACITY HIGH, then all new tables will be created as if this clause were present in the SQL CREATE TABLE statement. If there is a default INMEMORY value for the tablespace for a given segment, then it will override the value for this parameter.

Table 2-2 Meaning of INMEMORY_CLAUSE_DEFAULT Parameter Values

Syntax	Description
INMEMORY	Specifies that all newly-created tables and materialized views populate the IM column store unless they are specified as NO INMEMORY in the SQL CREATE TABLE or CREATE MATERIALIZED VIEW statement
NO INMEMORY	Specifies that only tables and materialized views explicitly specified as INMEMORY in the SQL CREATE TABLE or CREATE MATERIALIZED VIEW statements populate the IM column store
compression-clause	Specifies that in-memory compression should be used for the instance. Use the MEMCOMPRESS FOR values to specify the in-memory compression level.
NO MEMCOMPRESS	When ${\tt NO}\ {\tt MEMCOMPRESS}$ is specified, no in-memory compression is done in the IM column store.
MEMCOMPRESS FOR	MEMCOMPRESS FOR is used to indicate the in-memory compression level for the IM column store.
DML	When DML is specified, the IM column store is optimized for DML operations, and some lightweight in-memory compression may be done.
QUERY	When QUERY is specified, the in-memory compression level is for high performance. If QUERY is specified without LOW or HIGH, it defaults to QUERY LOW.



Syntax	Description
QUERY LOW	When QUERY LOW is specified, the in-memory compression level provides the highest performance.
QUERY HIGH	When QUERY HIGH is specified, the in-memory compression level provides a balance between compression and performance, weighted toward performance.
CAPACITY	When CAPACITY is specified without LOW or HIGH, it defaults to CAPACITY LOW.
CAPACITY LOW	When CAPACITY LOW is specified, the in-memory compression level a balance between compression and performance, weighted toward capacity.
CAPACITY HIGH	When CAPACITY HIGH is specified, the in-memory compression leve is for highest capacity.
priority-clause	Specifies the priority to use when populating tables in the IM column store. Use the PRIORITY values to specify the priority.
	By default, the population of a table in the IM column store can be delayed until the database determines it is useful.
	On database instance startup, tables are populated in priority order.
PRIORITY NONE	When PRIORITY NONE is specified, the population of a table in the IM column store can be delayed until the database determines it is useful
	This is the default value when no priority is specified.
PRIORITY LOW	When PRIORITY LOW is specified for a table or tables, the population of those tables in the IM column store is done before tables that have no priority specified.
PRIORITY MEDIUM	When PRIORITY MEDIUM is specified for a table or tables, the population of those tables in the IM column store is done before table that have no priority and PRIORITY LOW specified.
PRIORITY HIGH	When PRIORITY HIGH is specified for a table or tables, the populatio of those tables in the IM column store is done before tables that have no priority, PRIORITY LOW, and PRIORITY MEDIUM specified.
PRIORITY CRITICAL	When PRIORITY CRITICAL is specified for a table or tables, the population of those tables in the IM column store is done before table that have no priority, PRIORITY LOW, PRIORITY MEDIUM, and PRIORITY HIGH specified.
rac-clause	Specifies how tables in the IM column store will be managed among Oracle Real Application Clusters (Oracle RAC) instances. Use the distribute-clause and duplicate-clause to specify how tables in the IM store will be managed in Oracle RAC instances.
	For a non-Oracle RAC database, these settings have no effect, because the whole table or partition has to be on the single instance.
distribute-clause	Specifies how a table is distributed among Oracle RAC instances.
DISTRIBUTE AUTO	Specifies that the database will automatically decide how to distribute tables in the IM column store across the Oracle RAC instances based on the type of partitioning and the value of the duplicate-clause.
	DISTRIBUTE AUTO is the default, and it is also used when DISTRIBUTE is specified by itself.
DISTRIBUTE BY ROWID RANGE	Specifies that the tables in the IM column store will be distributed by rowid range to different Oracle RAC instances.



Table 2-2 (Cont.) Meaning of INMEMORY_CLAUSE_DEFAULT Parameter Values

Syntax	Description
duplicate-clause	Specifies how many copies of each In-Memory Compression Unit (IMCU) of the tables in the IM column store will be spread across all the Oracle RAC instances.
	Note: The duplicate-clause is only applicable if you are using Oracle Real Application Clusters (Oracle RAC) on an engineered system. Otherwise, the duplicate-clause is ignored and there is only one copy of each IMCU in memory.
NO DUPLICATE	Data is not duplicated across Oracle RAC instances. This is the default.
DUPLICATE	Data is duplicated on another Oracle RAC instance, resulting in data existing on a total of two Oracle RAC instances.
DUPLICATE ALL	Data is duplicated across all Oracle RAC instances. If you specify DUPLICATE ALL, then the database uses the DISTRIBUTE AUTO setting, regardless of whether or how you specify the distribute-clause.

Examples

The following statement causes no tables to populate the IM column store:

```
alter system set INMEMORY_CLAUSE_DEFAULT='NO INMEMORY'
scope=both;
```

The following statement causes new tables and materialized views (except those specified as NO INMEMORY) to populate the IM column store at the high capacity compression level:

```
alter system set
INMEMORY_CLAUSE_DEFAULT='INMEMORY MEMCOMPRESS FOR CAPACITY HIGH'
scope=both;
```

The following statement causes new tables (even those specified as NO INMEMORY) to populate the IM column store at the highest performance compression level at LOW priority:

```
alter system set
INMEMORY_CLAUSE_DEFAULT='INMEMORY MEMCOMPRESS FOR QUERY LOW PRIORITY LOW'
scope=both;
```

The following statement causes new tables (even those specified as NO INMEMORY) to populate the IM column store without any in-memory compression:

```
alter system set
INMEMORY_CLAUSE_DEFAULT='INMEMORY NO MEMCOMPRESS'
scope=both;
```

The following statement causes tables in the IM column store to be duplicated on every Oracle RAC instance, unless on a non-engineered system. For a non-engineered system, the duplicate-clause (DUPLICATE ALL) will be ignored and tables in the column store will be automatically distributed across the Oracle RAC instance, with only one copy of each IMCU in the tables in the IM column store:

```
alter system set
INMEMORY_CLAUSE_DEFAULT='INMEMORY MEMCOMPRESS FOR QUERY
DISTRIBUTE AUTO DUPLICATE ALL'
scope=both;
```



The following statement sets the value of the <code>INMEMORY_CLAUSE_DEFAULT</code> parameter back to its default value, the empty string:

alter system set
INMEMORY_CLAUSE_DEFAULT=''
scope=both;

See Also:

- "INMEMORY FORCE"
- "INMEMORY_QUERY"
- "INMEMORY_SIZE"
- "V\$IM SEGMENTS"
- "V\$IM USER SEGMENTS"
- "QUERY_REWRITE_INTEGRITY"
- Oracle Database In-Memory Guide for an introduction to the IM column store
- Oracle Database In-Memory Guide for more information about the IM column store
- Oracle Database In-Memory Guide for more information on IM column store compression methods
- Oracle Database In-Memory Guide for more information on priority levels for populating a database object in the IM column store
- Oracle Database SQL Language Reference for more information on the CREATE TABLE statement
- Oracle Database SQL Language Reference for more information on the CREATE MATERIALIZED VIEW statement

2.163 INMEMORY_DEEP_VECTORIZATION

INMEMORY DEEP VECTORIZATION enables or disables the deep vectorization framework.

Property	Description
Parameter type	Boolean
Default value	TRUE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	TRUE FALSE
Basic	No

Deep vectorization can optimize complex SQL operations (such as joins) by pipelining physical operators inside each SQL operator and vectorizing them.



See Also:

Oracle Database In-Memory Guide for information about deep vectorization

2.164 INMEMORY EXPRESSIONS USAGE

INMEMORY_EXPRESSIONS_USAGE controls which In-Memory Expressions (IM expressions) are populated into the In-Memory Column Store (IM column store) and are available for queries.

Property	Description
Parameter type	String
Syntax	<pre>INMEMORY_EXPRESSIONS_USAGE = { STATIC_ONLY DYNAMIC_ONLY ENABLE DISABLE }</pre>
Default value	ENABLE
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value must be used on all instances.

The four values for this parameter are:

- STATIC_ONLY: Tables enabled for in-memory and containing certain data types such as
 Oracle numbers or JSON will have these columns populated in the IM column store using
 a more efficient representation. Note that this setting will increase the in-memory footprint
 for some tables. A static configuration enables the IM column store to cache OSON (binary
 JSON) columns, which are marked with an IS_JSON check constraint. Internally, an
 OSON column is a hidden virtual column named SYS_IME_OSON. In contrast, a dynamic
 configuration automatically creates and populates frequently used expressions.
- DYNAMIC_ONLY: IM expressions will be automatically created and populated into the IM column store, if used in conjunction with PL/SQL procedure
 DBMS_INMEMORY.IME_CAPTURE_EXPRESSIONS. Note that setting this value will increase the inmemory footprint for some tables.
- ENABLE: Both static and dynamic IM expressions will be populated into the IM column store and available to be used by queries. Note that setting this value will increase the inmemory footprint for some tables. This is the default value.
- DISABLE: No IM expressions of any kind will be populated into the IM column store.

Changing the mode of a parameter does not have an immediate effect on existing in-memory data. For example, if the <code>inmemory_expressions_usage</code> clause is changed from <code>DYNAMIC_ONLY</code> to <code>DISABLE</code>, the stored IM expressions are not immediately removed. The next repopulation will not bring back the expressions in memory, thereby effectively removing the expressions.



See Also:

- "INMEMORY_VIRTUAL_COLUMNS"
- Oracle Database In-Memory Guide for more information about IM expressions
- Oracle Database PL/SQL Packages and Types Reference for more information about the DBMS INMEMORY PL/SQL package

2.165 INMEMORY_FORCE

INMEMORY_FORCE allows you to specify whether objects, such as tables, materialized views, and partitions, that are specified as INMEMORY are populated into the In-Memory Column Store (IM column store) or not.

Property	Description
Parameter type	String
Syntax	<pre>INMEMORY_FORCE = { DEFAULT OFF BASE_LEVEL CELLMEMORY_LEVEL }</pre>
Default value	DEFAULT
Modifiable	You can use ALTER SYSTEM to dynamically modify this parameter setting as follows: From DEFAULT to OFF From OFF to DEFAULT From BASE_LEVEL to DEFAULT From CELLMEMORY_LEVEL to DEFAULT From CELLMEMORY_LEVEL to OFF
	All other modifications of this parameter require you to update the initialization file and restart the instance.
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	All instances should use the same value.

The following values can be set:

DEFAULT

The IM column store is populated with objects that are specified as INMEMORY.

OFF

Even if the IM column store is configured on this instance, no objects are populated in memory.

BASE LEVEL

Enables Database In-Memory Base Level. The Base Level lets you to experiment with Oracle Database In-Memory features without purchasing the Oracle Database In-Memory option.

When the Base Level is enabled, the value of the INMEMORY_SIZE initialization parameter is limited to 16 GB for a CDB. In an Oracle RAC environment, the value of INMEMORY SIZE is



limited to 16 GB for each instance. The compression level for all objects and columns is automatically and transparently set to QUERY LOW. Automatic In-Memory is disabled, and In-Memory Column Store feature tracking is tracked for "In-Memory Base Level" rather than "In-Memory Column Store." The CellMemory feature is disabled for Oracle Exadata.

• CELLMEMORY LEVEL

CellMemory is an Exadata feature that allows you to use Exadata Smart Flash Cache to populate data in in-memory columnar format. In older releases of Oracle Database, in order to use the CellMemory feature, you were required to enable the IM column store (by setting INMEMORY_SIZE to a value greater than 0), even if you had no intention of using the IM column store. This incurred the overhead of enabling the IM column store without any benefit. Starting with Oracle Database release 19c, version 19.8, you can use the CellMemory feature without enabling the IM column store by setting INMEMORY_FORCE=CELLMEMORY_LEVEL and INMEMORY_SIZE=0. With these settings, the IM column store is not enabled and queries can use CellMemory to scan objects.

Note that if the value of <code>INMEMORY_SIZE</code> is greater than <code>0</code>, then setting <code>INMEMORY_FORCE=CELLMEMORY_LEVEL</code> is equivalent to setting <code>INMEMORY_FORCE=DEFAULT</code>. The IM column store is enabled, even if you only use CellMemory.

See Also:

- "INMEMORY CLAUSE DEFAULT"
- "INMEMORY_MAX_POPULATE_SERVERS"
- "INMEMORY QUERY"
- "INMEMORY_SIZE"
- Oracle Database In-Memory Guide for an introduction to the IM column store
- Oracle Database In-Memory Guide for more information about the IM column store
- Oracle Database Licensing Information User Manual to learn which database offerings support the Database In-Memory Base Level feature

2.166 INMEMORY_MAX_POPULATE_SERVERS

INMEMORY_MAX_POPULATE_SERVERS specifies the maximum number of background populate servers to use for In-Memory Column Store (IM column store) population, so that these servers do not overload the rest of the system.

Property	Description
Parameter type	Integer
Default value	Half of the value of CPU_COUNT or the PGA_AGGREGATE_TARGET value divided by 512M, whichever is less.
	See the "CPU_COUNT" description for information about how CPU_COUNT is calculated.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No



Property	Description
Range of values	0 to a value based on the number of cores in the system.
Basic	No
Oracle RAC	All instances should use the same value.

This parameter has meaning only if the INMEMORY_SIZE parameter is also set to a positive value.

The value to use for this parameter depends on the number of cores in the system. A certain percentage of CPU should be allocated for in-memory background population, and this parameter should be set accordingly. You can also set this parameter to 0 to temporarily disable populate tasks on the system from executing.

Note:

Be careful not to set the value of this parameter too high. If it is set close to the number of cores or higher, no CPU could be left for the rest of the system to run.

Note:

The IM column store is not populated if this parameter is set to 0.

See Also:

- "INMEMORY_CLAUSE_DEFAULT"
- "INMEMORY_FORCE"
- "INMEMORY_QUERY"
- "INMEMORY_SIZE"
- Oracle Database In-Memory Guide for an introduction to the IM column store
- Oracle Database In-Memory Guide for more information about the IM column store

2.167 INMEMORY_OPTIMIZED_ARITHMETIC

INMEMORY_OPTIMIZED_ARITHMETIC encodes the NUMBER data type in in-memory tables compressed with QUERY LOW as a fixed-width native integer scaled by a common exponent.

Property	Description
Parameter type	String
Syntax	<pre>INMEMORY_OPTIMIZED_ARITHMETIC = { ENABLE DISABLE }</pre>



Property	Description
Default value	DISABLE
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different values can be specified on different instances.

The Oracle Database NUMBER format can incur a significant performance overhead when executing queries because arithmetic operations cannot be performed natively in hardware.

When INMEMORY_OPTIMIZED_ARITHMETIC is set to ENABLE, for tables compressed with QUERY LOW, NUMBER columns are encoded as a fixed-width native integer scaled by a common exponent. This In-Memory optimized number format enables fast calculations using SIMD hardware. By using SIMD vector processing, arithmetic operations, simple aggregations, and group-by aggregations can benefit significantly.

Not all row sources in the query processing engine have support for the In-Memory optimized number format. Therefore, the IM column store must store both the traditional Oracle Database NUMBER data type and the In-Memory optimized number type. This means that the acceleration in analytic query performance comes at a cost of increased space overhead.

When INMEMORY_OPTIMIZED_ARITHMETIC is set to DISABLE (the default), the database does not use the optimized encoding.

If INMEMORY_OPTIMIZED_ARITHMETIC is set to ENABLE and then to DISABLE, the optimized number format for existing IMCUs is not dropped immediately. Instead, as the IM column store repopulates IMCUs, the new IMCUs do not use the optimized encoding.



Oracle Database In-Memory Guide for details about the improved performance and increased space overhead when <code>INMEMORY_OPTIMIZED_ARITHMETIC</code> is enabled

2.168 INMEMORY OPTIMIZED_DATE

Use INMEMORY OPTIMIZED DATE to enable or disable the In-Memory Optimized Dates feature.

Property	Description
Parameter type	String
Syntax	<pre>INMEMORY_OPTIMIZED_DATE = { ENABLE DISABLE }</pre>
Default value	DISABLE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different instances can use different values.



When this parameter is set to ENABLE, In-Memory can run date-based analytic queries much faster by leveraging the In-Memory Expressions framework.



This parameter is available starting with Oracle Database 23ai.



Oracle Database In-Memory Guide for more information about the In-Memory Optimized Dates feature

2.169 INMEMORY_QUERY

INMEMORY_QUERY is used to enable or disable in-memory queries for the entire database at the session or system level.

Property	Description
Parameter type	String
Syntax	<pre>INMEMORY_QUERY = { ENABLE DISABLE }</pre>
Default value	ENABLE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	All instances should use the same value.

This parameter is helpful when you want to test workloads with and without the use of the In-Memory Column Store (IM column store).

This parameter enables in-memory queries for the entire database by default when the ${\tt INMEMORY}$ ${\tt SIZE}$ parameter is specified.

Set this parameter to DISABLE if you want to disable in-memory queries.



See Also:

- "INMEMORY_CLAUSE_DEFAULT"
- "INMEMORY_FORCE"
- "INMEMORY_MAX_POPULATE_SERVERS"
- "INMEMORY_SIZE"
- Oracle Database In-Memory Guide for an introduction to the IM column store
- Oracle Database In-Memory Guide for more information about the IM column store

2.170 INMEMORY SIZE

INMEMORY_SIZE sets the size of the In-Memory Area, which contains the IM Column Store (IM column store) on a database instance.

Property	Description
Parameter type	Big integer
Syntax	<pre>INMEMORY_SIZE = integer [K M G]</pre>
Default value	0
Modifiable	You can use ALTER SYSTEM to increase or decrease the value of this parameter. However, the value of this parameter must always be greater than or equal to its value at instance startup.
Modifiable in a PDB	Yes
Range of values	0 to the amount of memory left in the SGA after other allocations
	A value of 0 specifies that the IM column store is not allocated and no In- Memory features are enabled.
	A value greater than 0 and less than 100M specifies that the IM column store is enabled for future growth, but no memory is currently provisioned to the In-Memory Area.
Basic	No
Oracle RAC	All instances should use the same value.

The database must be restarted after setting this parameter to enable the IM column store.

Typically this parameter should be set to at least the size needed to accommodate all the tables that will use the IM column store. It can be set higher to allow for growth of those tables or to accommodate other tables that will use the IM column store in the future.

This parameter can also be set per PDB to limit the maximum size of the IM column store for that PDB. Note that the sum of the PDB values do not have to equal the CDB value, and the sum of the PDB values may even be greater than the CDB value.

Unless this parameter is specifically set on a PDB, each PDB inherits the CDB value, which means they can use all of the available IM column store.



The value specified for this parameter counts toward SGA_TARGET. For example, if you set SGA_TARGET to 10 GB and you set INMEMORY_SIZE to 2 GB, then 20% of the SGA_TARGET setting is allocated to the In-Memory Area.

Starting with Oracle Database 23ai, Automatic In-Memory Sizing can automatically grow or shrink the In-Memory Area based on the benefits of the column store if the following conditions are true:

- SGA TARGET > 0
- INMEMORY AUTOMATIC LEVEL = MEDIUM OF HIGH
- VECTOR MEMORY SIZE is set to 0 or not specified

In-Memory Dynamic Scans require the Resource Manager. Therefore, the Resource Manager is automatically enabled when you change the value of <code>INMEMORY_SIZE</code> from 0 to a non-zero value. No specific resource plan is required.

Database In-Memory Base Level allows a maximum of 16 GB for the size of the In-Memory Area. Therefore, if Database In-Memory Base Level is enabled (the value of the INMEMORY_FORCE initialization parameter is set to BASE_LEVEL), then the value of INMEMORY_SIZE cannot exceed 16 GB for a CDB. In an Oracle RAC environment, the value of INMEMORY_SIZE is limited to 16 GB for each instance.

See Also:

- "INMEMORY_CLAUSE_DEFAULT"
- "INMEMORY_FORCE"
- "INMEMORY_MAX_POPULATE_SERVERS"
- "INMEMORY QUERY"
- Oracle Database In-Memory Guide for an introduction to the IM column store
- Oracle Database In-Memory Guide for more information about the IM column store
- Oracle Database In-Memory Guide for an example of using the INMEMORY_SIZE parameter

2.171

INMEMORY_TRICKLE_REPOPULATE_SERVERS_PERCENT

INMEMORY_TRICKLE_REPOPULATE_SERVERS_PERCENT limits the maximum number of background populate servers used for In-Memory Column Store (IM column store) repopulation, as trickle repopulation is designed to use only a small percentage of the populate servers.

Property	Description
Parameter type	Integer
Default value	1
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No



Property	Description
Range of values	0 to 50
Basic	No
Oracle RAC	All instances should use the same value.

The value for this parameter is a percentage of the <code>INMEMORY_MAX_POPULATE_SERVERS</code> initialization parameter value.

For example, if this parameter is set to 5 and <code>INMEMORY_MAX_POPULATE_SERVERS</code> is set to 10, then on average half of a core is used for trickle repopulation.

The default value of 1 is good in most cases. In some cases, if you want to disable trickle repopulate, this parameter can be set to 0. If you want to keep the system more aggressively up to date (at the expense of more background CPU), you can set the parameter to higher values such as 5 or 10.

A value of greater than 50 is not allowed, so that at least half of the populate servers are available for other (re)populate tasks. On some systems, a value of less than 50 can be problematic, depending on other workload.

This parameter has meaning only if the INMEMORY_SIZE parameter is also set to a positive value.

2.172 INMEMORY_VIRTUAL_COLUMNS

INMEMORY_VIRTUAL_COLUMNS controls which user-defined virtual columns are stored as In-Memory virtual columns (IM columns).

	Book at the control of the control o
Property	Description
Parameter type	String
Syntax	<pre>INMEMORY_VIRTUAL_COLUMNS = { ENABLE MANUAL DISABLE }</pre>
Default value	MANUAL
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value must be used on all instances.

IM virtual columns improve query performance by avoiding the necessity of repeated calculations. Also, the database can scan and filter IM virtual columns using techniques such as SIMD vector processing.

These values can be set for the parameter:

- ENABLE: For a table or partition that has been enabled for in-memory storage, all virtual columns will be stored in-memory at the default table or partition memcompress level unless:
 - They have been explicitly excluded using the no inmemory syntax.
 - They have been altered to have a different memcompress level than the base table or partition, in which case they will be stored at the specified memcompress level.



- MANUAL: This is the default value for the parameter. For a table or partition that has been enabled for in-memory storage, no virtual columns will be stored in-memory unless:
 - They have been explicitly marked for inmemory, in which case they will be stored inmemory at the table or partition memcompress level.
 - They have been marked for inmemory with a different memcompress level than the base table or partition, in which case they will be stored at the specified memcompress level.
- DISABLE: For a table or partition that has been enabled for in-memory storage, no virtual
 columns will ever be stored in-memory. Any changes to the inmemory_column_clause for
 a virtual column including changes in memcompress level will be recorded, but not acted
 upon with regards to population of virtual columns.

See Also:

- "INMEMORY EXPRESSIONS USAGE"
- Oracle Database In-Memory Guide for more information about IM virtual columns

2.173 INSTANCE_ABORT_DELAY_TIME

INSTANCE_ABORT_DELAY_TIME specifies how much time to delay an internal initiated instance termination (in seconds), such as when an irrecoverable process dies or an irrecoverable instance error occurs.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 and higher
Basic	No
Oracle RAC	Different instances should use the same value.

This parameter does not apply to a SHUTDOWN ABORT operation or any instance termination necessary to implement a user command.

Setting this parameter to a value greater than 0 enables a DBA to take some actions before a termination occurs due to an irrecoverable error. Note that since the instance is in an irrecoverable state, the DBA should not be too ambitious with the actions taken because some processes and/ or resources may be corrupted or unavailable, making complex actions impossible. Oracle does not guarantee what it is possible when an instance is in this state. A message is written to the alert log when the delayed termination is initiated. The value will not apply in the case of PMON death.

The larger the specified value, the longer the instance stays up, and the potential increases for other problems to occur. If you set this parameter, Oracle recommends setting it to a value between 0 and 60.

2.174 INSTANCE_GROUPS

Used with the PARALLEL_INSTANCE_GROUP parameter, INSTANCE_GROUPS lets you restrict parallel query operations to a limited number of instances.

Property	Description
Parameter type	String
Syntax	<pre>INSTANCE_GROUPS = group_name [, group_name]</pre>
Default value	There is no default value.
Modifiable	No
Modifiable in a PDB	No
Range of values	One or more instance group names, separated by commas
Basic	No
Oracle RAC	Multiple instances can have different values.



The INSTANCE_GROUPS parameter is deprecated. It is retained for backward compatibility only.

INSTANCE GROUPS is an Oracle RAC parameter that you can specify only in parallel mode.

This parameter specifies one or more instance groups and assigns the current instance to those groups. If one of the specified groups is also specified in the PARALLEL_INSTANCE_GROUP parameter, then Oracle allocates query processes for a parallel operation from this instance.

See Also:

- Oracle Real Application Clusters Administration and Deployment Guide for more information on parallel query execution in a Real Application Clusters environment
- "PARALLEL INSTANCE GROUP"

2.175 INSTANCE_MODE

INSTANCE MODE indicates whether the instance is read-write, read-only, or read-mostly.

Property	Description
Parameter type	String
Syntax	<pre>INSTANCE_MODE = { READ-WRITE READ-ONLY READ-MOSTLY }</pre>
Default value	READ-WRITE
Modifiable	No



Property	Description
Basic	No
Oracle RAC	See the Oracle RAC restrictions in the initialization parameter description below.

A READ-WRITE instance is a regular Oracle instance.

A READ-ONLY instance is an Oracle instance that can only be opened in read-only mode.

A READ-MOSTLY instance is an Oracle instance that performs very few database writes.

These restrictions apply when setting the parameter on different Oracle Real Application Clusters (Oracle RAC) instances:

- 1. There must be at least one running instance with <code>INSTANCE_MODE</code> set to <code>READ-WRITE</code> at any given time. If the only instance with <code>INSTANCE_MODE</code> set to <code>READ-WRITE</code> goes down, all other instances will be brought down by Oracle automatically.
- An instance with INSTANCE_MODE set to READ-ONLY cannot be the first instance to open a
 database or any pluggable database (PDB) among all instances in an Oracle RAC cluster.
- 3. Oracle does not support co-existence of an instance with INSTANCE_MODE set to READ-ONLY and an instance with INSTANCE MODE set to READ-MOSTLY in the same Oracle RAC cluster.

2.176 INSTANCE_NAME

In an Oracle Real Application Clusters environment, multiple instances can be associated with a single database service. Clients can override Oracle's connection load balancing by specifying a particular instance by which to connect to the database. Instance_name specifies the unique name of this instance.

Property	Description
Parameter type	String
Syntax	<pre>INSTANCE_NAME = instance_id</pre>
Default value	The instance's SID
	Note: The SID identifies the instance's shared memory on a host, but may not uniquely distinguish this instance from other instances.
Modifiable	No
Modifiable in a PDB	No
Range of values	Alphanumeric ASCII characters and the underscore (_) character, up to a maximum of 255 characters
Basic	No

In a single-instance database system, the instance name is usually the same as the database name.



See Also:

Oracle Real Application Clusters Administration and Deployment Guide and Oracle Database Net Services Administrator's Guide for more information

2.177 INSTANCE NUMBER

INSTANCE_NUMBER specifies a unique number that maps the instance to one free list group for each database object created with storage parameter FREELIST GROUPS.

Property	Description
Parameter type	Integer
Default value	0 (zero)
Modifiable	No
Modifiable in a PDB	No
Range of values	1 to maximum number of instances specified when the database was created
Basic	Yes
Oracle RAC	You must set this parameter for every instance, and all instances must have different values.

INSTANCE_NUMBER is an Oracle RAC parameter that can be specified in parallel mode or exclusive mode.

The INSTANCE parameter of the ALTER TABLE ... ALLOCATE EXTENT statement assigns an extent to a particular free list group. If you set INSTANCE_NUMBER to the value specified for the INSTANCE parameter, the instance uses that extent for inserts and for updates that expand rows.

The practical maximum value of this parameter is the maximum number of instances specified in the CREATE DATABASE statement. The absolute maximum is operating system-dependent.



Oracle Real Application Clusters Administration and Deployment Guide for more information

2.178 INSTANCE_TYPE

INSTANCE_TYPE specifies whether the instance is a database instance, an Oracle Automatic Storage Management (Oracle ASM) instance, or an Oracle ASM Proxy instance.

Property	Description
Parameter type	String
Syntax	<pre>INSTANCE_TYPE = { RDBMS ASM ASMPROXY }</pre>



Property	Description
Default value	RDBMS
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	Multiple instances must have the same value.

Values

RDBMS

The instance is a database instance.

ASM

The instance is an Oracle ASM instance.

ASMPROXY

The instance is an Oracle ASM proxy instance.



Oracle Automatic Storage Management Administrator's Guide for information about managing Oracle Flex ASM

2.179 IORM_LIMIT_POLICY

IORM_LIMIT_POLICY specifies the policy for computing the Exadata I/O Resource Management (IORM) share and limit values for pluggable databases.

Property	Description
Parameter type	String
Syntax	<pre>IORM_LIMIT_POLICY = { RM_PLAN CPU_COUNT_RATIO }</pre>
Default value	RM_PLAN
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	Multiple instances should use the same value.

Values:

- RM PLAN
 - If a CDB resource plan is not specified by the user, that is, the CDB resource plan is set to DEFAULT_CDB_PLAN, then this setting gives equal IORM shares to all PDBs and specifies no utilization limits. If the CPU_MIN_COUNT initialization parameter is set, the IORM share values are determined by using the RAC-wide PDB value of



CPU_MIN_COUNT. Otherwise, the IORM share values for each PDB are determined using the RAC-wide PDB value of CPU COUNT.

- If a user-defined CDB resource plan has been specified by setting the
 RESOURCE_MANAGER_PLAN initialization parameter, then the IORM share and utilization
 limit values for the PDBs are obtained from that resource plan.
- CPU_COUNT_RATIO

If the CPU_MIN_COUNT initialization parameter is set, the IORM share values are determined by using the RAC-wide PDB value of CPU_MIN_COUNT. Otherwise, the IORM share values for each PDB are determined using the RAC-wide PDB value of CPU_COUNT. In either case, IORM utilization limit values for each PDB are computed as a ratio of the RAC-wide PDB CPU_COUNT to the RAC-wide CDB CPU_COUNT.

The IORM utilization limit applies only to flash devices. It is not applicable for hard disks or XRMEM devices. The utilization limit is used for managing flash IOPS and throughput, and is not used to manage PDB flash cache space usage.



This parameter allows a third setting, CPU_COUNT_BASED . However, this setting is not supported.

Note:

This parameter is available starting with Oracle Database 23ai.

2.180 JAVA JIT ENABLED

JAVA_JIT_ENABLED enables or disables the Just-in-Time (JIT) compiler for the Oracle Java Virtual Machine (OracleJVM) environment.

Property	Description
Parameter type	Boolean
Default value	Operating system-dependent
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

For platforms that support the JIT compiler, the default value of this parameter is true; otherwise the default value is false. Attempting to set this parameter to true on unsupported platforms will result in an error.



See Also:

Oracle Database Java Developer's Guide for more information about setting this parameter

2.181 JAVA_MAX_SESSIONSPACE_SIZE

JAVA_MAX_SESSIONSPACE_SIZE specifies (in bytes) the maximum amount of session space made available to a Java program executing in the server.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Modifiable in a PDB	No
Range of values	0 to 2 GB - 1
Basic	No

Java session space is the memory that holds Java state from one database call to another. When a user's session-duration Java state attempts to exceed the amount specified by <code>JAVA_MAX_SESSIONSPACE_SIZE</code>, the Java virtual machine terminates the session with an out-of-memory failure.

See Also:

- Oracle Database Java Developer's Guide for more information about this parameter
- "JAVA_SOFT_SESSIONSPACE_LIMIT"

2.182 JAVA POOL SIZE

JAVA_POOL_SIZE specifies (in bytes) the size of the Java pool, from which the Java memory manager allocates most Java state during run-time execution. This memory includes the shared in-memory representation of Java method and class definitions, as well as the Java objects that are migrated to the Java session space at end-of-call.

Property	Description
Parameter type	Big integer
Syntax	JAVA_POOL_SIZE = integer [K M G]
Default value	If SGA_TARGET is set: If the parameter is not specified, then the default is 0 (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool. If SGA_TARGET is not set: 24 MB, rounded up to the nearest granule size

Property	Description
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Minimum: 0 (values greater than zero are rounded up to the nearest granule size)
	Maximum: operating system-dependent
Basic	No



Oracle Database Java Developer's Guide for more information about this parameter

2.183 JAVA_SOFT_SESSIONSPACE_LIMIT

JAVA_SOFT_SESSIONSPACE_LIMIT specifies (in bytes) a **soft limit** on Java memory usage in a session, as a means to warn you if a user's session-duration Java state is using too much memory.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Modifiable in a PDB	No
Range of values	0 to 2 GB - 1
Basic	No

Java session space is the memory that holds Java state from one database call to another. When a user's session-duration Java state exceeds the size specified by JAVA SOFT SESSIONSPACE LIMIT, Oracle generates a warning that goes into the trace files.

✓ See Also:

- Oracle Database Java Developer's Guide for more information on this parameter
- "JAVA_MAX_SESSIONSPACE_SIZE"

2.184 JOB_QUEUE_PROCESSES

JOB_QUEUE_PROCESSES specifies the maximum number of job workers per CDB or PDB that can be created for the execution of DBMS JOB jobs and Oracle Scheduler (DBMS SCHEDULER) jobs.

Property	Description
Parameter type	Integer
Default value	Derived. The lesser value of:
	• CPU_COUNT * 20
	• SESSIONS / 4
	If the result of the above derivation is less than twice the number of open containers in the CDB, then the value of this parameter is adjusted to equal twice the number of open containers in the CDB. Containers include CDB\$ROOT, PDB\$SEED, PDBs, application roots, application seeds, and application PDBs. You can obtain the number of open containers in a CDB with the following query:
	<pre>SELECT COUNT(*) FROM V\$CONTAINERS WHERE open_mode != 'MOUNTED';</pre>
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 4000
Basic	No
Oracle RAC	Multiple instances can have different values.

DBMS_JOB and Oracle Scheduler share the same job coordinator and job workers, and they are both controlled by the JOB_QUEUE_PROCESSES parameter. The actual number of job workers created for Oracle Scheduler jobs is auto-tuned by the Scheduler depending on several factors, including available resources, Resource Manager settings, and currently running jobs. However, the combined total number of job workers running DBMS_JOB jobs and Oracle Scheduler jobs in a CDB or PDB can never exceed the value of JOB_QUEUE_PROCESSES for that CDB or PDB.

The default value for <code>JOB_QUEUE_PROCESSES</code> provides a compromise between quality of service for applications and reasonable use of system resources. However, it is possible that the default value does not suit every environment. In such cases, you can use the following quidelines to fine tune this parameter:

In a CDB root:

Set JOB_QUEUE_PROCESSES to the maximum number of job workers that can be used simultaneously in the entire CDB. Oracle recommends that you set the value of this parameter to at least twice the number of open containers in the CDB, otherwise, there might be severe starvation between PDBs trying to run multiple jobs. If JOB_QUEUE_PROCESSES is set to 0 in a CDB root, then DBMS_JOB and Oracle Scheduler jobs cannot run in the CDB root or in any PDB, regardless of the JOB_QUEUE_PROCESSES setting at the PDB level.

In a PDB:

Set JOB_QUEUE_PROCESSES to the maximum number of job workers that can be used simultaneously in the PDB. The actual number depends on the resources assigned by Resource Manager and the demand in other containers. When multiple PDBs request jobs, Oracle Scheduler attempts to give all PDBs a fair share of the processes. Oracle recommends that you set the value of this parameter to at least 2 in a PDB. However, if you do not want to run DBMS_JOB and Oracle Scheduler jobs in a PDB, then set JOB QUEUE PROCESSES to 0 in the PDB.



Materialized views and AutoTask use Oracle Scheduler for automatic refreshes. Setting $JOB_QUEUE_PROCESS$ to 0 will disable these features and any other features that use Oracle Scheduler or DBMS JOB.

Note:

DBMS_JOB is deprecated in Oracle Database 12c Release 2 (12.2.0.1) and may be removed in a future release. Oracle recommends that you use DBMS_SCHEDULER instead.

See Also:

- Oracle Database PL/SQL Packages and Types Reference for more information on the DBMS SCHEDULER package
- Oracle Database Data Warehousing Guide for more information on managing materialized views
- Oracle Database Advanced Queuing User's Guide for more information about job queue processes
- Oracle Database Administrator's Guide for more information about the maximum number of scheduler job processes

2.185 JSON_BEHAVIOR

 $\tt JSON_BEHAVIOR$ allows you to specify various default behaviors, at the session level, for certain SQL/JSON functions and conditions.

Property	Description
Parameter type	String
Syntax	<pre>JSON_BEHAVIOR = "flag:value[;flag:value]"</pre>
	There must be no spaces inside of the double quotation marks.
Default value	None
Modifiable	ALTER SESSION
Modifiable in a PDB	No
Basic	No

This parameter allows you to change the default return data type, default type-compatibility, and default error behavior during a session for certain SQL/JSON functions and conditions. This lets you enforce consistent session-level JSON processing behavior, reducing the need for explicit overriding of default behaviors within individual SQL statements.

For example, by default, the <code>JSON_QUERY</code> function uses lax type-compatibility when comparing data. You can use this parameter to specify that all invocations of the <code>JSON_QUERY</code> function in a session use strict type-compatibility, by default.

You can specify one or more of the following flags, in any order:

JSON QUERY RET VARCHAR

Value: TRUE or FALSE (default)

Specifies the default return data type for the <code>JSON_QUERY</code> function when the <code>RETURNING</code> data type clause is omitted.

If you set this flag to TRUE, then the return data type for the JSON_QUERY function is VARCHAR2 (4000), regardless of its input data type.

The default value of FALSE causes the $\tt JSON_QUERY$ function to use the data type of its input for the return data type.

LAX JSON EXISTS

Value: TRUE (default) or FALSE

Specifies the default type-compatibility for the $\tt JSON_EXISTS$ condition when the $\tt TYPE$ (STRICT | LAX) clause is omitted.

If you set this flag to <code>FALSE</code>, then the <code>JSON_EXISTS</code> condition will behave as if were run with <code>TYPE(STRICT)</code> syntax. That is, when searching for a value in an array, if an array value does not match the data type of the search value, then the condition will behave as specified by the <code>ON ERROR</code> clause.

The default value of TRUE causes the $\tt JSON_EXISTS$ condition to behave as if it were run with the default $\tt TYPE(LAX)$ syntax, and it will attempt to convert array values to the search value data type, when necessary.

• LAX JSON QUERY

Value: TRUE (default) or FALSE

Specifies the default type-compatibility for the <code>JSON_QUERY</code> function when the <code>TYPE(STRICT|LAX)</code> clause is omitted.

If you set this flag to <code>FALSE</code>, then the <code>JSON_QUERY</code> function behaves as if it were run with <code>TYPE(STRICT)</code> syntax. That is, when searching for a value in an array, if an array value does not match the data type of the search value, then the function will behave as specified by the <code>ON ERROR</code> clause.

The default value of TRUE causes the $\tt JSON_QUERY$ function to behave as if it were run with the default $\tt TYPE(LAX)$ syntax, and it will attempt to convert array values to the search value data type, when necessary.

LAX JSON TABLE

Value: TRUE (default) or FALSE

Specifies the default type-compatibility for the JSON_TABLE function when the TYPE (STRICT | LAX) clause is omitted.

If you set this flag to <code>FALSE</code>, then the <code>JSON_TABLE</code> function behaves as if were run with <code>TYPE(STRICT)</code> syntax. That is, if the value of the function's chosen key is not of the specified return data type, then the function will behave as specified by the <code>ON ERROR</code> clause.

The default value of TRUE causes the $\tt JSON_TABLE$ function to behave as if it were run with the default $\tt TYPE(LAX)$ syntax, and it will attempt to convert the chosen key to the specified return data type, when necessary.

• LAX JSON VALUE

Value: TRUE (default) or FALSE



Specifies the default type-compatibility for the <code>JSON_VALUE</code> function when the <code>TYPE(STRICT|LAX)</code> clause is omitted.

If you set this flag to <code>FALSE</code>, then the <code>JSON_VALUE</code> function behaves as if were run with <code>TYPE(STRICT)</code> syntax. That is, if the function's return value is not of the specified return data type, then the function will behave as specified by the <code>ON ERROR</code> clause.

The default value of TRUE causes the JSON_VALUE function to behave as if it were run with the default TYPE (LAX) syntax, and it will attempt to convert the return value to the specified return data type, when necessary.

ON ERROR

Value: ERROR or NULL (default)

Specifies the default behavior for the functions <code>JSON_QUERY</code>, <code>JSON_TABLE</code> (without the <code>EXISTS</code> clause), and <code>JSON_VALUE</code> when the <code>ON ERROR</code> clause is omitted. By default, these functions return a null value in the event of an error, which can make it difficult to identify invalid queries.

If you set this flag to ERROR, then these functions will return an error for invalid queries.

The default value of \mathtt{NULL} causes these functions to return a null value in the event of an error.

Examples

```
ALTER SESSION SET JSON_BEHAVIOR = "ON_ERROR:ERROR";

ALTER SESSION SET JSON_BEHAVIOR =
"LAX JSON QUERY:FALSE;LAX JSON TABLE:FALSE;LAX JSON VALUE:FALSE";
```

Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

Oracle Database SQL Language Reference for more information about the $\tt JSON_QUERY, JSON_TABLE, and JSON_VALUE functions, and the <math>\tt JSON_EXISTS$ condition

2.186 JSON_EXPRESSION_CHECK

Use JSON EXPRESSION CHECK to enable or disable JSON query statement checking.

Property	Description
Parameter type	String
Syntax	<pre>JSON_EXPRESSION_CHECK = { ON OFF }</pre>
Default value	OFF
Modifiable	ALTER SESSION, ALTER SYSTEM



Property	Description
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different instances can use different values.

Set this parameter to ON to enable JSON query statement checking; set it to OFF to disable JSON query statement checking.



This parameter is available starting with Oracle Database 23ai.

See Also:

 ${\it JSON-Relational\ Duality\ Developer's\ Guide\ for\ more\ information\ about\ } \\ {\it JSON_EXPRESSION_CHECK}$

2.187 LARGE POOL SIZE

LARGE POOL SIZE specifies (in bytes) the size of the large pool allocation heap.

Property	Description
Parameter type	Big integer
Syntax	LARGE_POOL_SIZE = integer [K M G]
Default value	If SGA_TARGET is set, but a value is not specified for LARGE_POOL_SIZE, then the default is 0 (internally determined by the Oracle database). If LARGE_POOL_SIZE is specified, then the user-specified value indicates a minimum value for the memory pool. If SGA_TARGET is not set, then the default is 0.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Minimum: the granule size
	Maximum: operating system-dependent
Basic	No

The large pool allocation heap is used in shared server systems for session memory, by parallel execution for message buffers, and by backup processes for disk I/O buffers. Parallel execution allocates buffers out of the large pool only when SGA TARGET is set.

You can specify the value of this parameter using a number, optionally followed by K or M to specify kilobytes or megabytes, respectively. If you do not specify K or M, then the number is taken as bytes.

Oracle Database Performance Tuning Guide for more information on setting this parameter

2.188 LDAP DIRECTORY ACCESS

LDAP_DIRECTORY_ACCESS specifies whether Oracle refers to Oracle Internet Directory for user authentication information.

Note:

Enterprise User Security (EUS) is deprecated with Oracle Database 23ai.

Oracle recommends that you migrate to using Centrally Managed Users (CMU). This feature enables you to directly connect with Microsoft Active Directory without an intervening directory service for enterprise user authentication and authorization to the database. If your Oracle Database is in the cloud, you can also choose to move to one of the newer integrations with a cloud identity provider.

Property	Description
Parameter type	String
Syntax	LDAP_DIRECTORY_ACCESS = { NONE PASSWORD SSL }
Default value	NONE
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

If directory access is turned on, then this parameter also specifies how users are authenticated.

Values

NONE

Oracle does not refer to Oracle Internet Directory for Enterprise User Security information.

PASSWORD

Oracle tries to connect to the enterprise directory service using the database password stored in the database wallet. If that fails, then the Oracle Internet Directory connection fails and the database will not be able to retrieve enterprise roles and schema mappings upon enterprise user login.

• SSL

Oracle tries to connect to Oracle Internet Directory using SSL.



Oracle Database Enterprise User Security Administrator's Guide for more information on Enterprise User Security

Using LDAP_DIRECTORY_ACCESS with PDBs

PDBs can use password or SSL authentication with Oracle Internet Directory for EUS, or Microsoft Active Directory for CMU, when the parameter value is set accordingly.

The LDAP DIRECTORY ACCESS initialization parameter is PDB-specific and can be set as follows:

- When you use the ALTER SYSTEM command to set the value of LDAP_DIRECTORY_ACCESS
 while connected to the CDB root:
 - If you specify the CONTAINER=ALL clause, then the setting applies to the CDB root and all PDBs.
 - If you omit the CONTAINER=ALL clause, or specify the CONTAINER=CURRENT clause, then
 the setting applies only to the CDB root.
- When you use the ALTER SYSTEM command to set the value of LDAP_DIRECTORY_ACCESS while connected to a PDB, the setting applies only to that PDB.
- When you set the value of LDAP_DIRECTORY_ACCESS in an initialization parameter file, the setting applies only to the CDB root; it does not apply to the PDBs.

For a CDB, if the wallet location is not specified in sqlnet.ora, then the default database wallet path is:

ORACLE BASE/admin/db-unique-name/pdb-GUID/wallet (if ORACLE BASE is set)

or:

ORACLE HOME/admin/db-unique-name/pdb-GUID/wallet (if ORACLE BASE is not set)

The exception is for the root database, which has a default wallet path of:

ORACLE BASE/admin/db-unique-name/wallet (if ORACLE BASE is set)

or:

ORACLE_HOME/admin/db-unique-name/wallet (if ORACLE_BASE is not set)

All PDBs in a CDB have the same database unique name. By placing wallets in the default location, each PDB can have its own identity.

Note that because there is only one sqlnet.ora file for a CDB, the wallet location in sqlnet.ora is only used by the CDB root container. Because each PDB must have its own wallet, a PDB wallet will be specified by the pdb-GUID under the wallet location in sqlnet.ora.

For the root container of a CDB, the wallet location is:

WALLET_LOCATION_specified_in_sqlnet.ora

For each PDB of the CDB, the wallet location is:

WALLET_LOCATION_specified_in_sqlnet.ora/pdb-GUID/



Note:

Oracle databases are registered with Oracle Internet Directory using Database Configuration Assistant (DBCA). For registration with Oracle Internet Directory to work, all the PDBs for a CDB must be registered using DBCA.

Using LDAP_DIRECTORY_ACCESS with Non-CDBs

For non-CDBs, if the wallet location is not specified in sqlnet.ora, then the default database wallet path is:

ORACLE BASE/admin/db-unique-name/wallet (if ORACLE BASE is set)

or:

ORACLE HOME/admin/db-unique-name/wallet (if ORACLE BASE is not set)



Oracle Database Enterprise User Security Administrator's Guide for an example of setting the value of this parameter to SSL in the server parameter file using ALTER SYSTEM

2.189 LDAP DIRECTORY SYSAUTH

LDAP_DIRECTORY_SYSAUTH allows or disallows directory-based authorization for users granted administrative privileges, such as SYSDBA, SYSOPER, SYSBACKUP, SYSDG, and SYSKM.

Property	Description
Parameter type	String
Syntax	LDAP_DIRECTORY_SYSAUTH = { yes no }
Default value	no
Modifiable	No
Modifiable in a PDB	Yes
Basic	Yes

When LDAP_DIRECTORY_SYSAUTH is set to yes, directory users are allowed to connect to the database as SYSDBA, SYSOPER, SYSBACKUP, SYSDG, or SYSKM, if they have mapped database global users that are granted corresponding administrative privileges such as SYSDBA, SYSOPER, SYSBACKUP, SYSDG, and SYSKM.

When LDAP_DIRECTORY_SYSAUTH is set to no, directory users are not allowed to connect to the database as SYSDBA, SYSOPER, SYSBACKUP, SYSDG, or SYSKM, even if they have mapped database global users that are granted corresponding administrative privileges such as SYSDBA, SYSOPER, SYSBACKUP, SYSDG, and SYSKM.

When modifying this parameter in a PDB, use the ALTER SYSTEM command with SCOPE=SPFILE.

2.190 LICENSE_MAX_SESSIONS

LICENSE MAX SESSIONS specifies the maximum number of concurrent user sessions allowed.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 to number of session licenses
Basic	No
Oracle RAC	Multiple instances can have different values, but the total for all instances mounting a database should be less than or equal to the total number of sessions licensed for that database.



Oracle no longer offers licensing by the number of concurrent sessions. Therefore the LICENSE_MAX_SESSIONS and LICENSE_SESSIONS_WARNING initialization parameters have been deprecated.

When the limit specified by LICENSE_MAX_SESSIONS is reached, only users with the RESTRICTED SESSION privilege can connect to the database. Users who are not able to connect receive a warning message indicating that the system has reached maximum capacity.

A zero value indicates that concurrent usage (session) licensing is not enforced. If you set this parameter to a nonzero number, you might also want to set <code>LICENSE_SESSIONS_WARNING</code> (see "LICENSE_SESSIONS_WARNING").

Do not enable both concurrent usage licensing and user licensing. Set either LICENSE_MAX_SESSIONS or LICENSE_MAX_USERS to zero.



Oracle Database Administrator's Guide for more information about this parameter

2.191 LICENSE_MAX_USERS

LICENSE MAX USERS specifies the maximum number of users you can create in the database.

Property	Description
Parameter type	Integer
Default value	0



Modifiable 2	ALTER SYSTEM
	ALIEN SISIEM
Modifiable in a PDB	No
Range of values	0 to number of user licenses
Basic	No
Oracle RAC (Oracle recommends that multiple instances have the same value.

When you reach the limit specified by LICENSE_MAX_USERS, you cannot create more users. You can, however, increase the limit.



Oracle no longer offers licensing by the number of concurrent sessions. Therefore the ${\tt LICENSE_MAX_SESSIONS}$ and ${\tt LICENSE_SESSIONS_WARNING}$ initialization parameters have been deprecated.

See Also:

Oracle Database Administrator's Guide for more information about this parameter

2.192 LICENSE_SESSIONS_WARNING

LICENSE_SESSIONS_WARNING specifies a warning limit on the number of concurrent user sessions.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 to value of LICENSE_MAX_SESSIONS parameter
Basic	No
Oracle RAC	Multiple instances can have different values.



Oracle no longer offers licensing by the number of concurrent sessions. Therefore the $\mbox{LICENSE_MAX_SESSIONS}$ and $\mbox{LICENSE_SESSIONS_WARNING}$ initialization parameters have been deprecated.

When this limit specified by LICENSE_SESSIONS_WARNING is reached, additional users can connect, but Oracle writes a message in the alert log for each new connection. Users with RESTRICTED SESSION privilege who connect after the limit is reached receive a warning message stating that the system is nearing its maximum capacity.

If this parameter is set to zero, no warning is given as you approach the concurrent usage (session) limit. If you set this parameter to a nonzero number, you should also set LICENSE_MAX_SESSIONS (see "LICENSE_MAX_SESSIONS").

See Also:

Oracle Database Administrator's Guide for more information on setting this parameter

2.193 LISTENER_NETWORKS

LISTENER_NETWORKS specifies one or more sets of local, forward, and remote listeners for cross-registration. All listeners within the same *network name* will cross-register.

Property	Description
Parameter type	String
Syntax	LISTENER_NETWORKS =
	'((NAME=network_name) (LOCAL_LISTENER=["]listener_address[,]["]) (FORWARD_LISTENER=["]listener_address[,]["]) [(REMOTE_LISTENER=["]listener_address[,]["])])' [,]
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

If a *network_name* is specified multiple times, then the resulting listener set is the union of each specification. This can be used to specify sets that require more than 255 characters, which is the per element limit.

The *listener_address* specifies a string that is an address, address list, or an alias that resolves to an address or address list of Oracle Net listeners. If an alias, the address or address list is specified in the TNSNAMES.ORA file or another address repository as configured for your system.

If a comma is used to specify a *listener_address* list, then the set of addresses must be surrounded by double quotes.

- "LOCAL_LISTENER"
- "FORWARD LISTENER"
- "REMOTE LISTENER"
- Oracle Database Net Services Administrator's Guide for more information on setting this parameter

2.194 LOB_SIGNATURE_ENABLE

LOB SIGNATURE ENABLE is used to enable or disable the LOB locator signature feature.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Multiple instances must have the same value.

You can secure your LOBs by enabling the LOB locator signature feature. A LOB locator is a pointer to the location of a large object (LOB) value. If the LOB_SIGNATURE_ENABLED initialization parameter is set to true, then when you create a LOB, Oracle Database automatically assigns a signature to the LOB locator. When Oracle Database receives a request from a client, it uses the signature to determine if any tampering with the LOB locator has occurred.

When this parameter is set to true, you also have the option of further securing your LOBs by encrypting your LOB locator signature keys.



Oracle Database Security Guide for more information on LOB locator signatures and encrypting LOB locator signature keys

2.195 LOCAL LISTENER

LOCAL_LISTENER specifies a network name that resolves to an address or address list of Oracle Net local listeners (that is, listeners that run on the same system as this instance). The address or address list is specified in the TNSNAMES.ORA file or other address repository as configured for your system.

Property	Description
Parameter type	String
Syntax	LOCAL_LISTENER = network_name
Default value	(ADDRESS = (PROTOCOL=TCP) (HOST= $hostname$) (PORT=1521)) where $hostname$ is the network name of the local host.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

- Oracle Database Concepts for more information about listener processes and dispatcher processes
- Oracle Database Net Services Administrator's Guide and your operating systemspecific Oracle documentation for more information about specifying network addresses for the protocols on your system

2.196 LOCK_NAME_SPACE

LOCK_NAME_SPACE specifies the namespace that the distributed lock manager (DLM) uses to generate lock names.

Property	Description
Parameter type	String
Syntax	LOCK_NAME_SPACE = namespace
Default value	There is no default value.
Modifiable	No
Modifiable in a PDB	No
Range of values	Up to 8 alphanumeric characters. No special characters allowed.
Basic	No

Note:

The LOCK_NAME_SPACE parameter is deprecated. It is retained for backward compatibility only.

Consider setting this parameter if a standby or clone database has the same database name on the same cluster as the primary database.

If the standby database resides on the same file system as the primary database, then set LOCK NAME SPACE in the standby parameter file to a distinct value such as the following:

LOCK_NAME_SPACE = standby

2.197 LOCK_SGA

LOCK SGA locks the entire SGA into physical memory.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No

It is usually advisable to lock the SGA into real (physical) memory, especially if the use of virtual memory would include storing some of the SGA using disk space. This parameter is ignored on platforms that do not support it.



Oracle Database Performance Tuning Guide for more information about using this parameter

2.198 LOG_ARCHIVE_CONFIG

 $LOG_ARCHIVE_CONFIG$ enables or disables the sending of redo logs to remote destinations and the receipt of remote redo logs, and specifies the unique database names (DB_UNIQUE_NAME) for each database in the Data Guard configuration.

Property	Description
Parameter type	String
Syntax	LOG_ARCHIVE_CONFIG =
	<pre>{ [SEND NOSEND] [RECEIVE NORECEIVE] [DG_CONFIG=(remote_db_unique_name1 [, remote_db_unique_name30) NODG_CONFIG] }</pre>
Default value	NULL
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

Values

SEND

Enables the sending of redo logs to remote destinations

NOSEND

Disables the sending of redo logs to remote destinations

RECEIVE

Enables the receipt of remotely archived redo logs

NORECEIVE

Disables the receipt of remotely archived redo logs

DG CONFIG

Specifies a list of up to 30 unique database names (defined with the DB_UNIQUE_NAME initialization parameter) for all of the databases in the Data Guard configuration.

NODG_CONFIG

Eliminates the list of service provider names previously specified with the DG_CONFIG option.

When this parameter has not been set, its default value is <code>NULL</code>. However, the <code>NULL</code> value is treated as if the parameter has been set to <code>\SEND</code>, <code>RECEIVE</code>, <code>NODG</code> <code>CONFIG'</code>.

2.199 LOG_ARCHIVE_DEST

Use LOG ARCHIVE DEST to specify the destination to which redo log files will be archived.

Property	Description
Parameter type	String
Syntax	LOG_ARCHIVE_DEST = filespec
Default value	Null
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Any valid path or device name, except raw partitions
Basic	No
Oracle RAC	Multiple instances can have different values.



For Enterprise Edition users, this parameter has been deprecated in favor of the $LOG_ARCHIVE_DEST_n$ parameters. If Oracle Enterprise Edition is not installed or it is installed, but you have not specified any $LOG_ARCHIVE_DEST_n$ parameters, this parameter is valid.

LOG_ARCHIVE_DEST is applicable only if you are running the database in ARCHIVELOG mode or are recovering a database from archived redo logs. LOG_ARCHIVE_DEST is incompatible with the LOG_ARCHIVE_DEST_n parameters, and must be defined as the null string ("") or (' ') when any LOG_ARCHIVE_DEST_n parameter has a value other than a null string. Use a text string to specify the default location and root of the disk file or tape device when archiving redo log files.

(Archiving to tape is not supported on all operating systems.) The value cannot be a raw partition.

If LOG_ARCHIVE_DEST is not explicitly defined and all the LOG_ARCHIVE_DEST_n parameters have null string values, LOG_ARCHIVE_DEST is set to an operating system-specific default value on instance startup.

To override the destination that this parameter specifies, either specify a different destination for manual archiving or use the SQL*Plus statement ARCHIVE LOG START filespec for automatic archiving, where filespec is the new archive destination. To permanently change the destination, use the statement ALTER SYSTEM SET LOG_ARCHIVE_DEST = filespec, where filespec is the new archive destination.

Neither LOG_ARCHIVE_DEST nor LOG_ARCHIVE_FORMAT have to be complete file or directory specifiers themselves; they only need to form a valid file path after the variables are substituted into LOG_ARCHIVE_FORMAT and the two parameters are concatenated together.

See Also:

- Oracle Database Backup and Recovery User's Guide
- "LOG_ARCHIVE_DUPLEX_DEST", "LOG_ARCHIVE_MIN_SUCCEED_DEST", and "V\$ARCHIVE DEST" for more information on setting this parameter
- Your Oracle operating system-specific documentation for the default value and for an example of how to specify the destination path or file name using LOG ARCHIVE DEST

2.200 LOG_ARCHIVE_DEST_n

The LOG_ARCHIVE_DEST_n initialization parameter defines up to 31 (where n = 1, 2, 3, ... 31) destinations, each of which *must* specify either the LOCATION or the SERVICE attribute to specify where to archive the redo data.

Property	Description
Parameter type	String



Property	Description
Syntax	LOG_ARCHIVE_DEST_[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31] =
	<pre>{ null_string { LOCATION=path_name SERVICE=service_name } [MANDATORY] [REOPEN[=seconds]] [DELAY[=minutes]] [ENCRYPTION=ENABLED DISABLED] [GROUP=group] [NOREGISTER] [PRIORITY=priority] [TEMPLATE=template] [ALTERNATE=destination] [MAX_FAILURE=count] [SYNC ASYNC] [AFFIRM NOAFFIRM] [NET_TIMEOUT=seconds] [VALID_FOR=(redo_log_type,database_role)] [DB_UNIQUE_NAME] [COMPRESSION={ENABLE DISABLE ZLIB LZO}] }</pre>
Default value	There is no default value.
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	No
Basic	Yes

All other attributes except the LOCATION or SERVICE attributes are optional. Note that whether you are specifying the LOCATION attribute or the SERVICE attribute, it must be the first attribute supplied in the list of attributes.

If you choose not to enter any attributes, then you can specify a NULL string by entering the following:

```
LOG ARCHIVE DEST n=' ';
```

You set the attributes for the $LOG_ARCHIVE_DEST_n$ initialization parameter to control different aspects of how redo transport services transfer redo data from a production or primary database destination to another (standby) database destination. You can query the $VARCHIVE_DEST$ view to see the current attribute settings for each destination (n).

Note:

Some of the attributes for this parameter are deprecated, but are being retained for backward compatibility. See "Deprecated Attributes for LOG_ARCHIVE_DEST_n".

For every $LOG_ARCHIVE_DEST_n$ initialization parameter that you define, you must specify a corresponding $LOG_ARCHIVE_DEST_STATE_n$ parameter. The $LOG_ARCHIVE_DEST_STATE_n$ (where n is an integer from 1 to 31) initialization parameter specifies whether the corresponding destination is currently enabled or disabled.

"LOG_ARCHIVE_DEST_STATE_n"

Destinations LOG_ARCHIVE_DEST_11 through LOG_ARCHIVE_DEST_31 do not support the SYNC, ARCH, LOCATION, MANDATORY, or ALTERNATE attributes, and cannot be specified as the target of the ALTERNATE attribute.

LOG_ARCHIVE_DEST_11 through LOG_ARCHIVE_DEST_31 can only be used when the COMPATIBLE initialization parameter is set to 11.2.0 or higher.

Values

See Also:

Oracle Data Guard Concepts and Administration for detailed descriptions of all the values listed in this section

AFFIRM and NOAFFIRM

Control whether a redo transport destination acknowledges received redo data before or after writing it to the standby redo log. The default is NOAFFIRM.

ALTERNATE=LOG_ARCHIVE_DEST_n

Specifies an alternate archiving destination to be used when the original destination fails. There is no default value; if an alternate destination is not specified, then archiving does not automatically change to another destination if the original destination fails.

ASYNC

The redo data generated by a transaction need not have been received at a destination which has this attribute before that transaction can commit. This is the default behavior if neither SYNC nor ASYNC is specified.

COMPRESSION

Indicates whether network compression is enabled or disabled, or disabled, or whether the ZLIB or LZO algorithm is used. The possible values include:

- DISABLE: Compression is disabled.
- ENABLE: Compression is enabled. The ZLIB compression algorithm is used.
- ZLIB: ZLIB compression is used.
- LZO: LZO compression is used.

The Advanced Compression option is required in order to use the COMPRESSION attribute.

• DB UNIQUE NAME=name

Specifies a unique name for the database at this destination. You must specify a name; there is no default value.

DELAY[=minutes]

Specifies a minimum time lag between when redo data is archived on a standby site retrieving redo from a primary and when the archived redo log file is applied to the standby database or any standbys cascading from it. If you specify the DELAY attribute without a time interval, the default is 30 minutes.

• ENCRYPTION=DISABLE | ENABLE

Controls whether encryption of the redo stream sent to Zero Data Loss Recovery Appliance (Recovery Appliance) is enabled or disabled. The default value for the attribute is DISABLE.

This attribute cannot be used with the COMPRESSION, LOCATION, or SYNC attributes.

To use the ENCRYPTION attribute, you must set the COMPATIBLE initialization parameter to 11.2.0.4 or higher on the protected database.

See Also:

Zero Data Loss Recovery Appliance Administrator's Guide for introductory information about Recovery Appliance

GROUP=group

The GROUP attribute is used to specify membership in a specific collection of log archive destinations. Groups are numbered 1 through 8. The default group (GROUP=0) is special in that it cannot be assigned. The default group is populated with all destinations that are not explicitly assigned to a group. All groups other than the default group must consist of a set of remote destinations. No group (other than the default group) can contain local destinations.

LOCATION=local_disk_directory or USE DB RECOVERY FILE DEST

Specifies either a local file system destination or the directory, file system, or Automatic Storage Management disk group that will serve as the fast recovery area. You must specify this attribute for at least one destination. You can specify either a local disk directory or fast recovery area with the LOCATION attribute. You *must* include either the LOCATION or the SERVICE attribute for each destination to specify where to archive the redo data.

MANDATORY

Specifies that the transmission of redo data to the destination must succeed before the local online redo log file can be made available for reuse. If the MANDATORY attribute is not specified, then the destination is optional.

• MAX_FAILURE

Controls the consecutive number of times redo transport services attempt to reestablish communication and transmit redo data to a failed destination before the primary database gives up on the destination. See the MAX_FAILURE description in *Oracle Data Guard Concepts and Administration* for usage notes about how this attribute is handled differently in Oracle Database 12c Release 2 (12.2.0.1) and in Oracle Database 12c Release 1 (12.1.0.2).

• NET TIMEOUT=seconds

Specifies the number of seconds the log writer process on the primary system waits for status from the SYNC (NSSn) process before terminating the network connection. The default is 30 seconds.

NOREGISTER

Indicates that the location of the archived redo log file is not to be recorded at the corresponding destination.

PRIORITY=priority

The PRIORITY attribute is used to specify preference within a group of log archive destinations. Priorities are numbered 1 through 8. A lower value represents a higher priority. The lowest priority (PRIORITY=8) is special in the sense that if that priority is active then all destinations at that priority will be made active. If any higher priority destination returns to service then that destination will be made active and all low priority destinations will be made inactive because they will be receiving redo from one of the other redo destinations, either through cascading or a Far Sync.

REOPEN[=seconds]

Specifies the minimum number of seconds before the archiver processes (ARCn) or the log writer process (LGWR) should try again to access a previously failed destination. The default is 300 seconds.

SERVICE=net_service_name

Specifies a valid Oracle Net service name (SERVICE=net_service_name) that identifies the remote Oracle database instance to which redo data will be sent. Each destination *must* specify either the LOCATION or the SERVICE attribute. There is no default net service name.

SYNC

The redo data generated by a transaction must have been received by every enabled destination which has this attribute before that transaction can commit.

TEMPLATE=filename_template_%t_%s_%r

Specifies a path name and a file name template for archived redo log files created at a redo transport destination that contain redo data from the database where this attribute is specified. This attribute overrides the value of the LOG_ARCHIVE_FORMAT initialization parameter at a redo transport destination. This attribute does not have a default value.

• VALID FOR=(redo log type, database role)

Identifies when redo transport services can transmit redo data to destinations based on the following factors:

- redo_log_type—whether online redo log files, standby redo log files, or both are currently being archived on the database at this destination
- database_role—whether the database is currently running in the primary or the standby role

Deprecated Attributes for LOG ARCHIVE DEST n

The following attributes are deprecated for the $LOG_ARCHIVE_DEST_n$ parameter. They are retained for backward compatibility only.

Table 2-3 Deprecated Attributes on the LOG_ARCHIVE_DEST_n Initialization Parameter

Deprecated Attribute	Alternative
ARCH	Specify SYNC or ASYNC. ASYNC is the default if neither attribute is specified.
LGWR	Specify SYNC or ASYNC. ASYNC is the default if neither attribute is specified.



Table 2-3 (Cont.) Deprecated Attributes on the LOG_ARCHIVE_DEST_n Initialization Parameter

Deprecated Attribute	Alternative
OPTIONAL	Destinations are optional by default.
VERIFY	None. This attribute is only used with the deprecated ARCH attribute.

In addition, note the following changes to the ASYNC and SYNC attributes:

- The BLOCKS keyword on the ASYNC attribute is no longer needed.
 - It is no longer necessary to set this keyword because Data Guard dynamically adjusts the block count up or down to an appropriate number of blocks, as necessary.
- The PARALLEL and NOPARALLEL keywords on the SYNC attribute are no longer needed.

2.201 LOG_ARCHIVE_DEST_STATE_n

The LOG_ARCHIVE_DEST_STATE_n parameters (where n = 1, 2, 3, ... 31) specify the availability state of the corresponding destination.

Property	Description
Parameter type	String
Syntax	LOG_ARCHIVE_DEST_STATE_[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31] = { enable defer alternate reset }
Default value	enable
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	No
Basic	Yes

The parameter suffix (1 through 31) specifies one of the corresponding $LOG_ARCHIVE_DEST_n$ destination parameters.

Values

enable

Specifies that a valid log archive destination can be used for a subsequent archiving operation (automatic or manual). This is the default.

defer

Specifies that valid destination information and attributes are preserved, but the destination is excluded from archiving operations until reenabled.

alternate

Specifies that a log archive destination is not enabled but will become enabled if communications to another destination fail.

reset

Functions the same as defer, but clears any error messages for the destination if it had previously failed.

The LOG_ARCHIVE_DEST_STATE_n parameters have no effect on the ENABLE state for the LOG ARCHIVE DEST or LOG ARCHIVE DUPLEX DEST parameters.

The V\$ARCHIVE_DEST dynamic performance view shows values in use for the current session. The DEST_ID column of that view corresponds to the archive destination suffix n.

See Also:

- Oracle Data Guard Concepts and Administration for more information about this parameter
- "V\$ARCHIVE DEST"

2.202 LOG_ARCHIVE_DUPLEX_DEST

LOG_ARCHIVE_DUPLEX_DEST is similar to the initialization parameter LOG_ARCHIVE_DEST. This parameter specifies a second archive destination: the **duplex** archive destination. This duplex archive destination can be either a must-succeed or a best-effort archive destination, depending on how many archive destinations must succeed (as specified in the LOG ARCHIVE MIN SUCCEED DEST parameter).

Property	Description
Parameter type	String
Syntax	LOG_ARCHIVE_DUPLEX_DEST = filespec
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Either a null string or any valid path or device name, except raw partitions
Basic	No

Note:

If you are using Oracle Enterprise Edition, this parameter is deprecated in favor of the $LOG_ARCHIVE_DEST_n$ parameters. If Oracle Enterprise Edition is not installed or it is installed but you have not specified any $LOG_ARCHIVE_DEST_n$ parameters, this parameter is valid.

The default setting of a null string ("") or (' ') indicates that a duplex archive destination does not exist.

- "LOG_ARCHIVE_DEST_n"
- "LOG_ARCHIVE_MIN_SUCCEED_DEST"
- "V\$ARCHIVE_DEST"
- Oracle Database Administrator's Guide for an example of using this parameter to specify an optional secondary archive destination

2.203 LOG_ARCHIVE_FORMAT

Use LOG ARCHIVE FORMAT to specify the default filename format when archiving redo log files.

Property	Description
Parameter type	String
Syntax	LOG_ARCHIVE_FORMAT = filename
Default value	Operating system-dependent
Modifiable	No
Modifiable in a PDB	No
Range of values	Any string that resolves to a valid filename
Basic	No
Oracle RAC	Multiple instances can have different values, but identical values are recommended.

LOG_ARCHIVE_FORMAT is applicable only if you are using the redo log in ARCHIVELOG mode. Use a text string and variables to specify the default filename format when archiving redo log files. The string generated from this format is appended to the string specified in the LOG ARCHIVE DEST parameter.

The following variables can be used in the format:

- %s log sequence number
- %s log sequence number, zero filled
- %t thread number
- %T thread number, zero filled
- %a activation ID
- %d database ID

%r resetlogs ID that ensures unique names are constructed for the archived log files across multiple incarnations of the database

Using uppercase letters for the variables (for example, %S) causes the value to be fixed length and padded to the left with zeros. An example of specifying the archive redo log filename format follows:

LOG_ARCHIVE_FORMAT = 'log%t_%s_%r.arc'

Archive log file names must contain each of the elements \$s (sequence), \$t (thread), and \$r (resetlogs ID) to ensure that all archive log file names are unique. If the LOG_ARCHIVE_FORMAT initialization parameter is set in the parameter file, then make sure the parameter value contains the \$s, \$t, and \$r elements. Otherwise, the following error is displayed at the time of instance startup:

ORA-19905: log archive format must contain %s, %t and %r

Neither LOG_ARCHIVE_DEST nor LOG_ARCHIVE_FORMAT have to be complete file or directory specifiers themselves; they only need to form a valid file path after the variables are substituted into LOG_ARCHIVE_FORMAT and the two parameters are concatenated together.

LOG ARCHIVE FORMAT is ignored in these cases:

- For archived log files that go to the fast recovery area
- When LOG_ARCHIVE_DEST[_n] points to the root of an Oracle ASM disk group (for example, +DATA). The directory of a disk group (for example, +DATA/logs) must be specified for the parameter to be honored.

In these cases where <code>LOG_ARCHIVE_FORMAT</code> is ignored, an Oracle ASM file name is used. See Oracle Automatic Storage Management Administrator's Guide for more information on Oracle ASM file names.

See Also:

- Oracle Database Backup and Recovery User's Guide, Oracle Data Guard Concepts and Administration, and Oracle Real Application Clusters Administration and Deployment Guide for more information about this parameter
- Your operating system- specific Oracle documentation for the default value and range of values for LOG_ARCHIVE_FORMAT

2.204 LOG_ARCHIVE_MAX_PROCESSES

LOG_ARCHIVE_MAX_PROCESSES specifies the maximum number of ARCn processes that can be created.

Property	Description
Parameter type	Integer
Default value	4
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	1 to 30
Basic	No



- "Background Processes" for more information about ARCn processes
- Oracle Database Administrator's Guide for more information about LOG_ARCHIVE_MAX_PROCESSES

2.205 LOG_ARCHIVE_MIN_SUCCEED_DEST

LOG_ARCHIVE_MIN_SUCCEED_DEST defines the minimum number of destinations that must succeed in order for the online logfile to be available for reuse.

Property	Description	
Parameter type	Integer	
Default value	1	
Modifiable	ALTER SESSION, ALTER SYSTEM	
Modifiable in a PDB	Yes	
Range of values	1 to 10 if you are using LOG_ARCHIVE_DEST_n	
	1 or 2 if you are using LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST	
Basic	No	

If you are using the $LOG_ARCHIVE_DEST_n$ parameters and automatic archiving is enabled, then the value of this parameter cannot exceed the number of enabled, valid destinations specified as MANDATORY plus the number of enabled, valid destinations that are configured with the OPTIONAL and LOCATION attributes.

If you are using $LOG_ARCHIVE_DEST$ and $LOG_ARCHIVE_DUPLEX_DEST$ and automatic archiving is enabled, a value of 1 specifies that the destination specified in $LOG_ARCHIVE_DEST$ must succeed. A value of 2 specifies that the destinations specified in both parameters must succeed.

If the value of this parameter is less than the number of enabled, valid MANDATORY destinations, this parameter is ignored in favor of the MANDATORY destination count. If the value is more than the number of enabled, valid MANDATORY destinations, then some of the enabled, valid destinations configured with the OPTIONAL and LOCATION attributes are treated as MANDATORY.

You can switch dynamically from using the older parameters to the LOG_ARCHIVE_DEST_n parameter using ALTER SYSTEM, as follows:

- 1. Set LOG ARCHIVE MIN SUCCEED DEST to 1.
- 2. Set the value of LOG ARCHIVE DEST and LOG ARCHIVE DUPLEX DEST to the null string.
- 3. Set the desired number of destinations for the LOG ARCHIVE DEST n parameters.
- 4. Reset LOG ARCHIVE MIN SUCCEED DEST to the desired value.



- Oracle Database Administrator's Guide for more information on setting this parameter
- "LOG_ARCHIVE_DEST_n", "LOG_ARCHIVE_DUPLEX_DEST", and
 "V\$ARCHIVE_DEST" for information on related parameters

2.206 LOG_ARCHIVE_TRACE

LOG_ARCHIVE_TRACE enables and controls the generation of comprehensive trace information for log archiving and redo transport activity.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768
Basic	No
Oracle RAC	Multiple instances can have different values.

The additional tracing that is output when setting <code>LOG_ARCHIVE_TRACE</code> to a non-zero value can appear in trace files for an archive process, RFS process, LGWR process, SYNC process, ASYNC process, foreground process, MRP process, recovery process, log apply process, startup process, shutdown process, and other processes that use redo transport services.

Table 2-4 describes the valid LOG ARCHIVE TRACE values.

Table 2-4 LOG_ARCHIVE_TRACE Values

Value	Description	More Information
0	Disables archivelog tracing (this is the default)	more information
1	High-level tracing	High-level tracing provides a small amount of tracing from most redo transport processes. When a value is specified that is not high-level (not 1), the high-level tracing for that value will be output to the trace file even if the high-level tracing has not been specifically requested. For example, if you set LOG_ARCHIVE_TRACE to 128, you will get all the trace messages pertaining to the FAL engine, <i>including</i> the high-level tracing from the FAL engine.



Table 2-4 (Cont.) LOG_ARCHIVE_TRACE Values

Value	Description	More Information
2	Tracks Data Guard interfaces	This tracing level is for the various Data Guard configuration parameters, the various Data Guard related SQL commands, the various Data Guard table accesses, and some Data Guard internal interfaces.
4	Tracks common redo transport services	This tracing level is for a number of common services provided by redo transport to various internal users (such as the RFS process, the LGWR process, and the ASYNC process). These include locking primitives, use of common VOS (Virtual Operating System) interfaces, task scheduling, process monitoring, control file access, and other miscellaneous common services.
8	Tracks database protection mode	
16	Tracks disk and network I/O requests	
32	Tracks redo transport destinations	
64	Tracks redo transport archive engine	
128	Tracks redo transport FAL (fetch archive log) engine	
256	Tracks Physical, Logical, and Availability Machine clients of RFS process	
512	Tracks redo transport of LGWR, SYNC, and ASYNC processes	
1024	Tracks redo transport RFS process	
2048	Tracks redo transport gap resolution	
4096	Tracks real-time apply	
8192	Tracks recovery process	
16384	Tracks redo transport buffer management	
32768	Tracks LogMiner dictionary	

You can combine tracing levels by adding together the values of the desired tracing levels. For example, a setting of 3 will generate level 1 and level 2 trace output. You can set different values for the primary and standby database.

When this parameter is set to the default value of 0, Oracle will still generate appropriate alert and trace entries in response to error conditions. If you change the value of this parameter dynamically in an ALTER SYSTEM statement, the change takes effect immediately.

Many of the trace messages from redo transport processes include a routine name at the beginning of the trace message. The first part of the routine name specifies the module for that routine. Table 2-5 shows the tracing value used for each redo transport module. Note that the tracing value for transport modules can change from release to release.

Table 2-5 Tracing Values Used for Redo Transport Modules

Module Name	Purpose	Tracing Value
krsa	Locking primitives	4
krsb	Buffer management	16384
krsc	Crash recovery	64
krsd	Destination	32
krse	Archive engine	64
krsf	FAL engine	128
krsg	Gap resolution	2048
krsh	Helper	4
krsi	Input/output	16
krsj	Protection mode	8
krsk	Control file access	4
krsl	Log writer	512
krsm	MRP process	8192
krsn	Configuration	2
krso	Process monitor	4
krsp	Physical standby and Availability Machine RFS client	256
krsq	SQL commands	2
krsr	RFS process	1024
krst	Tables	2
krsu	Network I/O	16
krsv	VOS front end	4
krsw	ASYNC process	512
krsx	Internal interfaces	2

Oracle Database Administrator's Guide

2.207 LOG_BUFFER

 ${\tt LOG_BUFFER}$ specifies the amount of memory (in bytes) that Oracle uses when buffering redo entries to a redo log file.

Property	Description
Parameter type	Big integer
Default value	2 MB to 32 MB, depending on the SGA size and CPU count
Modifiable	No



Property	Description
Modifiable in a PDB	No
Range of values	2 MB to operating system-dependent
Basic	No

Redo log entries contain a record of the changes that have been made to the database block buffers. The LGWR process writes redo log entries from the log buffer to a redo log file.

The log buffer size depends on the number of redo strands in the system. One redo strand is allocated for every 16 CPUs and has a default size of 2 MB. Oracle allocates a minimum of 2 redo strands per instance. When the log buffer size is not specified, any remaining memory in the redo granules is given to the log buffer.

See Also:

- Your operating system-specific Oracle documentation for the default value and range of values
- Oracle Database Performance Tuning Guide for information on resizing the redo log buffer using this parameter

2.208 LOG_CHECKPOINT_INTERVAL

LOG_CHECKPOINT_INTERVAL specifies the frequency of checkpoints in terms of the number of redo log file blocks that can exist between an incremental checkpoint and the last block written to the redo log. This number refers to physical operating system blocks, not database blocks.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 to 2 ³¹ - 1
Basic	No
Oracle RAC	Multiple instances can have different values.

Regardless of this value, a checkpoint always occurs when switching from one online redo log file to another. Therefore, if the value exceeds the actual redo log file size, checkpoints occur only when switching logs. Checkpoint frequency is one of the factors that influence the time required for the database to recover from an unexpected failure.



Note:

- Specifying a value of 0 (zero) for LOG_CHECKPOINT_INTERVAL has the same effect as setting the parameter to infinity and causes the parameter to be ignored. Only nonzero values of this parameter are considered meaningful.
- Recovery I/O can also be limited by setting the LOG_CHECKPOINT_TIMEOUT
 parameter or by the size specified for the smallest redo log. For information on
 which mechanism is controlling checkpointing behavior, query the
 V\$INSTANCE RECOVERY view.

See Also:

- "LOG_CHECKPOINT_TIMEOUT"
- "V\$INSTANCE_RECOVERY"
- Oracle Database Performance Tuning Guide for information on disabling or removing this parameter when the FAST START MTTR TARGET parameter is set

2.209 LOG CHECKPOINT TIMEOUT

LOG_CHECKPOINT_TIMEOUT specifies (in seconds) the amount of time that has passed since the incremental checkpoint at the position where the last write to the redo log (sometimes called the **tail of the log**) occurred. This parameter also signifies that no buffer will remain dirty (in the cache) for more than *integer* seconds.

Property	Description
Parameter type	Integer
Default value	1800
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	0 to 2 ³¹ - 1
Basic	No
Oracle RAC	Multiple instances can have different values.

Specifying a value of 0 for the timeout disables time-based checkpoints. Hence, setting the value to 0 is not recommended unless FAST START MTTR TARGET is set.

Note:

- A checkpoint scheduled to occur because of this parameter is delayed until the completion of the previous checkpoint if the previous checkpoint has not yet completed.
- Recovery I/O can also be limited by setting the LOG_CHECKPOINT_INTERVAL
 parameter or by the size specified for the smallest redo log. For information on
 which mechanism is controlling checkpointing behavior, query the
 V\$INSTANCE RECOVERY view.

See Also:

- "LOG_CHECKPOINT_TIMEOUT"
- "V\$INSTANCE RECOVERY"
- Oracle Database Performance Tuning Guide for information on disabling or removing this parameter when the FAST START MTTR TARGET parameter is set

2.210 LOG CHECKPOINTS TO ALERT

LOG_CHECKPOINTS_TO_ALERT lets you log your checkpoints to the alert log. Doing so is useful for determining whether checkpoints are occurring at the desired frequency.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	true false
Basic	No

2.211 LOG_FILE_NAME_CONVERT

LOG_FILE_NAME_CONVERT converts the filename of a new log file on the primary database to the filename of a log file on the standby database.

Property	Description
Parameter type	String



Property	Description
Syntax	<pre>LOG_FILE_NAME_CONVERT = 'string1' , 'string2' , 'string3' , 'string4' ,</pre>
	Where:
	 string1 is the pattern of the primary database filename
	 string2 is the pattern of the standby database filename
	 string3 is the pattern of the primary database filename
	 string4 is the pattern of the standby database filename
	You can enclose each string in single or double quotation marks.
	You can specify as many pairs of primary and standby replacement strings as required.
	Example:
	LOG_FILE_NAME_CONVERT = '/dbs/t1/','/dbs/t1/s','dbs/t2/','dbs/t2/s'
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

If you add a log file to the primary database, you must add a corresponding file to the standby database.

Set the value of this parameter to one or more pairs of strings. The first string is the pattern found in the log file names on the primary database. The second string is the pattern found in the log file names on the standby database.

When the standby database is updated, this parameter converts the log file name on the primary database to the log file name on the standby database. The file must exist on the standby database and must be writable or the recovery process will halt with an error.

If you specify an odd number of strings (the last string has no corresponding replacement string), an error is signalled during startup. If the filename being converted matches more than one pattern in the pattern/replace string list, the first matched pattern takes effect. There is no limit on the number of pairs that you can specify in this parameter (other than the hard limit of the maximum length of multivalue parameters).

You should also use <code>LOG_FILE_NAME_CONVERT</code> to rename the logfiles in the clone control file when setting up the clone database during tablespace point-in-time recovery.

Note:

The LOG_FILE_NAME_CONVERT parameter applies only to online logs (not to archived logs).

See Also:

Oracle Data Guard Concepts and Administration

2.212 LONG_MODULE_ACTION

 $\verb|LONG_MODULE_ACTION| enables the use of longer lengths for modules and actions.$

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	The same value must be used on all instances.

Module length was 48 bytes and action length was 32 bytes in Oracle Database releases prior to Oracle Database 12c Release 2 (12.2.0.1).

Starting with Oracle Database 12c Release 2 (12.2.0.1):

- If LONG_MODULE_ACTION is set to TRUE (the default value), then the length of modules and actions will be 64 bytes each.
- If LONG_MODULE_ACTION is set to FALSE, then the length of modules will be 48 bytes, and the length of actions will be 32 bytes.

2.213 MANDATORY USER PROFILE

MANDATORY_USER_PROFILE specifies the mandatory user profile for a CDB or PDB.

Property	Description
Parameter type	String
Syntax	MANDATORY_USER_PROFILE = profile_name
Default value	There is no default value
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Multiple instances must use the same value.

For profile_name, specify the name of a mandatory user profile. Mandatory user profiles are common profiles created in the CDB root with the following SQL statement:

```
CREATE MANDATORY PROFILE ... CONTAINER = ALL;
```

You can set the value of MANDATORY USER PROFILE in the CDB root or in a PDB.

 When you set this parameter in the CDB root, the specified mandatory user profile applies to the CDB root and all PDBs.



• When you set this parameter in a PDB, the specified mandatory user profile applies only to that PDB and overrides the mandatory user profile for the CDB root, if any.

Only common users with the commonly granted ALTER SYSTEM privilege or the commonly granted SYSDBA privilege can set the value of MANDATORY_USER_PROFILE in the CDB root or in a PDB. This prevents local PDB administrators from circumventing any restrictions that are enforced at the CDB root.



See *Oracle Database Security Guide* for more information about mandatory user profiles

2.214 MAX_AUTH_SERVERS

MAX_AUTH_SERVERS specifies the maximum number of authentication servers in the authentication pool. This pool authenticates user connections when client applications connect to Database Resident Connection Pooling (DRCP).

Property	Description
Parameter type	Integer
Default value	25
Modifiable	ALTER SYSTEM
Modifiable in a PDB	 Yes, according to the following guidelines: When the ENABLE_PER_PDB_DRCP initialization parameter is set to true, MAX_AUTH_SERVERS can be modified only at the PDB level; it cannot be modified for the CDB root. When the ENABLE_PER_PDB_DRCP initialization parameter is set to false, MAX_AUTH_SERVERS can be modified only for the CDB root; it cannot be modified at the PDB level.
Range of values	Minimum: MIN_AUTH_SERVERS or 1, whichever is the greater value Maximum: operating system-dependent
Basic	No
Oracle RAC	Different instances can use different values.

The value of this parameter must be greater than or equal to the value of the MIN_AUTH_SERVERS initialization parameter, which specifies the minimum number of authentication servers in the authentication pool. The exception is when MIN_AUTH_SERVERS = 0. In this case, the value of MAX_AUTH_SERVERS must be greater than or equal to 1.

✓ See Also:

- Oracle Database Administrator's Guide for more information about setting this parameter
- "ENABLE_PER_PDB_DRCP"
- "MIN_AUTH_SERVERS"



2.215 MAX_COLUMNS

Use MAX COLUMNS to enable or disable Wide Tables for the database.

Property	Description
Parameter type	String
Syntax	MAX_COLUMNS = { STANDARD EXTENDED }
Default value	STANDARD
Modifiable	No
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Multiple instances must use the same value.

Values:

• To enable Wide Tables, set this parameter to EXTENDED. With this setting, the maximum number of columns allowed in a database table or view is 4096.

The COMPATIBLE initialization parameter must be set to 23.0.0.0 or higher in order to set MAX_COLUMNS = EXTENDED.

 To disable Wide Tables, set this parameter to STANDARD. With this setting, the maximum number of columns allowed in a database table or view is 1000. This matches the behavior in releases prior to Oracle Database 23ai.

You can change the value of MAX_COLUMNS from STANDARD to EXTENDED at any time. However, you can change the value of MAX_COLUMNS from EXTENDED to STANDARD only when all tables and views in the database contain 1000 or fewer columns.



The increased column limit of 4096 is supported by Oracle Database 23ai clients, such as SQL*Plus, OCI, JDBC-OCI, unmanaged ODP.NET, and open source drivers. Older client versions (pre-Oracle Database 23ai) do not support the increased column limit and cannot access more than 1000 columns in a table or view.

Note:

This parameter is available starting with Oracle Database 23ai.



2.216 MAX_DATAPUMP_JOBS_PER_PDB

MAX_DATAPUMP_JOBS_PER_PDB determines the maximum number of concurrent Oracle Data Pump jobs per PDB.

Property	Description
Parameter type	String
Syntax	MAX_DATAPUMP_JOBS_PER_PDB = { AUTO integer }
Default value	AUTO
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 250 or AUTO
Basic	No
Oracle RAC	The same value must be used on all instances.

When this parameter has a value of AUTO, Oracle Data Pump will derive its actual value to be 50% of the SESSIONS initialization parameter.

A value that is too large could cause Oracle Data Pump to consume too many system resources, while a value that is too small could prevent users from performing their Oracle Data Pump tasks.

The main resource Oracle Data Pump uses is shared pool in the System Global Area (SGA) for the database. Parallel jobs increase the number of sessions and, depending on the job, the number of PX servers used.

See Also:

- Oracle Database Utilities for more information about using Oracle Data Pump with CDBs
- Oracle Database PL/SQL Packages and Types Reference for more information about the DBMS_DATAPUMP PL/SQL package

2.217 MAX_DATAPUMP_PARALLEL_PER_JOB

MAX_DATAPUMP_PARALLEL_PER_JOB specifies the maximum number of parallel processes allowed per Oracle Data Pump job.

Property	Description
Parameter type	String
Syntax	MAX_DATAPUMP_PARALLEL_PER_JOB = { integer AUTO }
Default value	AUTO
Range of values	1 to 1024, or AUTO
Modifiable	ALTER SYSTEM



Property	Description
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different instances can use different values.

When this parameter has a value of AUTO, Oracle Data Pump derives its value to be 50 percent of the value of the SESSIONS initialization parameter.

2.218 MAX_DISPATCHERS

 ${\tt MAX_DISPATCHERS}$ specifies the maximum number of dispatcher processes allowed to be running simultaneously.

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	If MAX_DISPATCHERS is specified, then it should be greater than or equal to the number of dispatchers specified by the DISPATCHERS parameter and less than the number of processes specified by the PROCESSES parameter.
Basic	No

This parameter can be overridden by the DISPATCHERS parameter and is maintained for backward compatibility with older releases.

See Also:

- Oracle Database Administrator's Guide for more information about this parameter
- Your operating system-specific Oracle documentation for the default value and range of values

2.219 MAX_DUMP_FILE_SIZE

MAX DUMP FILE SIZE specifies the maximum size of trace files (excluding the alert log).

Property	Description
Parameter type	String
Syntax	MAX_DUMP_FILE_SIZE = { integer [K M G] UNLIMITED }



Property	Description
Default value	Oracle Database Free: 32M (4 segments of 8 MB each)
	For all other Oracle Database offerings, the default value is 1G, which means:
	 The maximum size for trace files generated by foreground processes is 1 GB (5 segments of 200 MB each). The maximum size for other trace files, such as incident trace files and trace files generated by background processes, is 10 GB (5 segments of 2 GB each).
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to unlimited, or UNLIMITED
Basic	No

You can change this limit if you are concerned that trace files may use too much space.

- A numeric value for MAX_DUMP_FILE_SIZE specifies the maximum size in operating system blocks.
- A numeric value followed by a K or M or G suffix specifies the file size in kilobytes, megabytes, or gigabytes.
- The special value string UNLIMITED means that there is no upper limit on trace file size. Thus, dump files can be as large as the operating system permits.

When the trace file is limited in size, it may be automatically split into multiple files, called segments, if needed. This process is call trace file segmentation. The segments will have the same file name as the active trace file, but with an extra segment number appended.

The trace file can be split into a maximum of 5 segments, and the size of each segment will typically be 1/5th of the trace file limit.

When the combined size of all the trace file segments exceeds the specified limit, the oldest segment is deleted, and a new, empty segment is created. Thus, the trace file always contains the most recent trace information. The first segment is never deleted, because it may contain relevant information about the initial state of the process, unless it ages out and gets purged. Rotation will happen between the remaining segments only.

Note that trace file segmentation will not occur when this parameter is set to a value less than 25 MB.

See Also:

Oracle Database Administrator's Guide and Oracle Database SQL Tuning Guide for more information on setting this parameter



2.220 MAX_IDLE_BLOCKER_TIME

MAX_IDLE_BLOCKER_TIME specifies the maximum number of minutes that a blocking session can be idle. After that point, the session is automatically terminated.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to the maximum integer. The value of 0 indicates that there is no limit.
Basic	No
Oracle RAC	Different instances can use different values.

A session is considered to be a blocking session when it is holding resources required by other sessions. For example:

- The session is holding a lock required by another session.
- The session is a parallel operation and its consumer group, PDB, or database has either reached its maximum parallel server limit or has queued parallel operations.
- The session's PDB or database instance is about to reach its SESSIONS or PROCESSES limit.

This parameter differs from the MAX_IDLE_TIME parameter in that MAX_IDLE_TIME applies to all sessions (blocking and non-blocking), whereas MAX_IDLE_BLOCKING_TIME applies only to blocking sessions. Therefore, in order for MAX_IDLE_BLOCKING_TIME to be effective, its limit must be less than the MAX_IDLE_TIME limit.



This parameter does not have an effect on parallel query processes, nor on SYS user sessions.

See Also:

"MAX IDLE TIME"

2.221 MAX_IDLE_TIME

MAX_IDLE_TIME specifies the maximum number of minutes that a session can be idle. After that point, the session is automatically terminated.

Property	Description
Parameter type	Integer

Property	Description
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to the maximum integer. The value of 0 indicates that there is no limit.
Basic	No
Oracle RAC	Different instances can use different values.

"MAX_IDLE_BLOCKER_TIME" to learn how to set a lower idle time limit for sessions that are holding resources required by other sessions

2.222 MAX_IOPS

MAX_IOPS enables you to set the maximum number of I/Os that can be issued per second on a per pluggable database (PDB) basis. This parameter is used to throttle PDB I/Os.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to the maximum Integer value. A very low value (for example, under 100 l/Os per second) is not recommended.
Basic	No
Oracle RAC	Different values can be set on different instances.

DBWR I/Os, control file I/Os, password file I/Os and other critical I/Os are exempted from the rate limit set by this parameter, but their I/Os are accounted for while throttling. Because of these exemptions, the PDB's actual I/O rate may sometimes exceed the limit.

This feature is enabled for multitenant container database (CDB) only. The feature is not supported on Oracle Exadata.

This parameter can be set from inside a PDB. If the parameter is set in CDB\$ROOT, all the PDBs in that CDB will inherit the parameter value from CDB\$ROOT. This parameter cannot be set in a non-CDB environment.

The default value of 0 means that no limits are set on the maximum number of I/Os that can be issued per second in a PDB.

If Oracle processes need to wait because of this IO rate limit, the wait event is resmgr: I/O rate limit.



- "MAX_MBPS"
- "resmgr: I/O rate limit"

Examples

This example shows how to use SQL statements to set a maximum of 3000 I/Os per second on a PDB named CDB1 PDB1:

```
alter session set container = cdb1_pdb1;
alter system set max iops = 3000;
```

2.223 MAX_MBPS

MAX_MBPS enables you to set the maximum number of megabytes (MB) of I/Os issued per second on a per pluggable database (PDB) basis. This parameter is used to throttle PDB I/Os.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to the maximum Integer value. A very low value (for example, under 25 MB per second) is not recommended.
Basic	No
Oracle RAC	Different values can be set on different instances.

DBWR I/Os, control file I/Os, password file I/Os and other critical I/Os are exempted from the throughput limit set by this parameter, but their I/Os are accounted for while throttling. Because of these exemptions, the PDB's actual I/O rate may sometimes exceed the limit.

This feature is enabled for multitenant container database (CDB) only. The feature is not supported on Oracle Exadata.

This parameter can be set from inside a PDB. If the parameter is set in CDB\$ROOT, all the PDBs in that CDB will inherit the parameter value from CDB\$ROOT. This parameter cannot be set in a non-CDB environment.

The default value of 0 means that no limits are set on the maximum number of megabytes (MB) of I/Os that can be issued per second in a PDB.

If Oracle processes need to wait because of this IO rate limit, the wait event is resmgr: I/O rate limit.



- "MAX_IOPS"
- "resmgr: I/O rate limit"

Examples

This example shows how to use SQL statements to set a maximum of 200 megabytes per second on a PDB named CDB1 PDB1:

```
alter session set container = cdb1_pdb1;
alter system set max mbps = 200;
```

2.224 MAX_PDBS

 ${\tt MAX_PDBS}$ allows you to limit the number of pluggable databases (PDBs) that can be created in a CDB or an application container.

Property	Description
Parameter type	Integer
Default value	Depends on the Oracle Database edition Possible values: 5, 254, or 4098
Modifiable	ALTER SYSTEM SID='*'
Modifiable in a PDB	No. However, this parameter can be modified in the application root of an application container.
Range of values	Minimum: 0
	Maximum: The default value for the Oracle Database edition (5, 254, or 4098)
Basic	No
Oracle RAC	The same value must be used on all instances.

This parameter can only be set while connected to the CDB root or an application root.

This parameter applies to PDBs, application containers, and application PDBs. An application PDB is a PDB that resides in an application container.

This parameter does not apply to the PDB seed (PDB\$SEED), application seeds, or application root clones.



The maximum number of PDBs that you are allowed to create in a CDB depends on the licensing policy for your Oracle Database edition. To learn more, refer to the Oracle Multitenant information in the "Consolidation" table in *Oracle Database Licensing Information User Manual*.



2.225 MAX_SAGA_DURATION

MAX_SAGA_DURATION specifies the maximum duration for a saga (in seconds).

If a saga exists for longer than the amount of time specified by this parameter, it is considered incomplete and is eligible for termination.

Property	Description
Parameter type	Integer
Default value	86400 seconds (equivalent to 24 hours)
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	1 to 2 ³¹ - 1
Basic	No
Oracle RAC	Multiple instances must have the same value.



This parameter is available starting with Oracle Database 23ai.

2.226 MAX_SHARED_SERVERS

MAX_SHARED_SERVERS specifies the maximum number of shared server processes allowed to be running simultaneously. Setting this parameter enables you to reserve process slots for other processes, such as dedicated servers.

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	If MAX_SHARED_SERVERS is specified, then it should be greater than or equal to SHARED_SERVERS and less than PROCESSES.
Basic	No

When you want to reduce the range of shared servers, you can reduce MAX_SHARED_SERVERS before reducing SHARED_SERVERS. If MAX_SHARED_SERVERS is lower than SHARED_SERVERS, then the number of shared servers will not vary but will remain at the constant level specified by SHARED_SERVERS. If MAX_SHARED_SERVERS is not specified, then a shared server process may be spawned as long as the number of free process slots is greater than 1 / 8 the maximum number of processes, or 2 if PROCESSES is less than 24.



- "SHARED_SERVERS"
- "PROCESSES"
- Oracle Database Administrator's Guide for more information on setting this parameter
- Oracle Database Concepts for information on processes
- Your operating system-specific Oracle documentation for the default value and range of values

2.227 MAX_STRING_SIZE

MAX_STRING_SIZE controls the maximum size of VARCHAR2, NVARCHAR2, and RAW data types in SQL.

Property	Description
Parameter type	String
Syntax	MAX_STRING_SIZE = { STANDARD EXTENDED }
Default value	STANDARD
Modifiable	ALTER SYSTEM SID='*'1
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Multiple instances must use the same value.

¹ Use ALTER SYSTEM only when the database is in UPGRADE mode, and run the utl32k.sql script afterward, as explained in this section.

STANDARD means that the length limits for Oracle Database releases prior to Oracle Database 12c apply (for example, 4000 bytes for VARCHAR2 and NVARCHAR2, and 2000 bytes for RAW).

EXTENDED means that the 32767 byte limit introduced in Oracle Database 12c applies.

The COMPATIBLE initialization parameter must be set to 12.0.0.0 or higher to set MAX STRING SIZE = EXTENDED.

You can change the value of MAX_STRING_SIZE from STANDARD to EXTENDED. However, you cannot change the value of MAX STRING SIZE from EXTENDED to STANDARD.

By setting MAX_STRING_SIZE = EXTENDED, users are taking an explicit action that could introduce application incompatibility in their database. Applications that do not want to use the expanded data types can be rewritten for compatibility with either setting; for example, these applications could use explicit CASTs to fix the length of VARCHAR2 expressions during CREATE TABLE AS SELECT.

Altering $\texttt{MAX_STRING_SIZE}$ will update database objects and possibly invalidate them, as follows:



- Tables with virtual columns will be updated with new data type metadata for virtual columns of VARCHAR2 (4000), 4000-byte NVARCHAR2, or RAW (2000) type.
 - Functional indexes will become unusable if a change to their associated virtual columns causes the index key to exceed index key length limits. Attempts to rebuild such indexes will fail with ORA-01450: maximum key length exceeded.
- Views will be invalidated if they contain VARCHAR2 (4000), 4000-byte NVARCHAR2, or RAW (2000) typed expression columns.
- Materialized views will be updated with new metadata VARCHAR2 (4000), 4000-byte NVARCHAR2, and RAW (2000) typed expression columns

Increasing the Maximum Size of VARCHAR2, NVARCHAR2, and RAW Columns in a CDB

To increase the maximum size of VARCHAR2, NVARCHAR2, and RAW columns in a CDB and in all the PDBs in the CDB:

- Connect to the CDB AS SYSDBA.
- 2. In the root, change the setting of MAX STRING SIZE to EXTENDED:

```
ALTER SESSION SET CONTAINER=CDB$ROOT;
ALTER SYSTEM SET max string size=extended SCOPE=SPFILE;
```



The root continues to use STANDARD semantics even after MAX_STRING_SIZE is set to EXTENDED. The reason for setting MAX_STRING_SIZE to EXTENDED in the root is so all the PDBs in the CDB can inherit the EXTENDED setting from the root.

- 3. Shut down the CDB.
- Restart the CDB in UPGRADE mode.

```
startup upgrade;
```

5. Use the catcon.pl script to run the rdbms/admin/utl32k.sql script in the root and in all the PDBs in the CDB to increase the maximum size of the VARCHAR2, NVARCHAR2, and RAW columns. The --force_pdb_mode 'UPGRADE' option is used to ensure that all PDBs, including application root clones, are opened in migrate mode. Enter the SYS password when prompted:

```
$ cd $ORACLE_HOME/rdbms/admin
$ mkdir /scratch/mydir/utl32k_cdb_pdbs_output
$ $ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl -u SYS --
force_pdb_mode 'UPGRADE' -d $ORACLE_HOME/rdbms/admin -l '/scratch/mydir/
utl32k_cdb_pdbs_output' -b
utl32k_cdb_pdbs_output utl32k.sql
catcon: ALL catcon-related output will be written to [/scratch/mydir/
utl32k_cdb_pdbs_output/utl32k_cdb_pdbs_output_catcon_23172.lst]
catcon: See [/scratch/mydir/utl32k_cdb_pdbs_output/
utl32k_cdb_pdbs_output*.log] files for output generated by scripts
catcon: See [/scratch/mydir/utl32k_cdb_pdbs_output/
utl32k_cdb_pdbs_output_*.lst] files for spool files, if any
Enter Password:
```



```
catcon.pl: completed successfully
s
```



The utl32k.sql script increases the maximum size of the VARCHAR2, NVARCHAR2, and RAW columns for the views where this is required. The script does not increase the maximum size of the VARCHAR2, NVARCHAR2, and RAW columns in some views because of the way the SQL for those views is written.

- 6. Connect to the CDB AS SYSDBA and shut down the database.
- 7. Restart the CDB in NORMAL mode.

startup;

8. Use the catcon.pl script to run the rdbms/admin/utlrp.sql script to recompile invalid objects in the root and in all the PDBs in the CDB. The --force_pdb_mode 'READ WRITE' option is used to ensure that all the PDBs (including application root clones) are opened in read write mode. Enter the SYS password when prompted:

```
$ cd $ORACLE_HOME/rdbms/admin
$ mkdir /scratch/mydir/utlrp_cdb_pdbs_output
$ $ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl -u SYS --
force_pdb_mode 'READ WRITE' -d $ORACLE_HOME/rdbms/admin -l '/scratch/mydir/
utlrp_cdb_pdbs_output' -b utlrp_cdb_pdbs_output utlrp.sql
catcon: ALL catcon-related output will be written to [/scratch/mydir/
utlrp_cdb_pdbs_output/utlrp_cdb_pdbs_output_catcon_24271.lst]
catcon: See [/scratch/mydir/utlrp_cdb_pdbs_output/
utlrp_cdb_pdbs_output*.log] files for output generated by scripts
catcon: See [/scratch/mydir/utlrp_cdb_pdbs_output/
utlrp_cdb_pdbs_output_*.lst] files for spool files, if any
Enter Password:
catcon.pl: completed successfully
$
```

See Also:

Oracle Multitenant Administrator's Guide for information about using the catcon.pl script to run Oracle-supplied scripts in a CDB and PDBs.

Increasing the Maximum Size of VARCHAR2, NVARCHAR2, and RAW Columns in a PDB

To increase the maximum size of VARCHAR2, NVARCHAR2, and RAW columns in a PDB:

- 1. Shut down the PDB.
- Reopen the PDB in migrate mode.



Note:

The following SQL statement can be used to reopen a PDB in migrate mode when the current container is the PDB:

ALTER PLUGGABLE DATABASE pdb-name OPEN UPGRADE;

- 3. Change the setting of MAX_STRING_SIZE in the PDB to EXTENDED.
- 4. Run the rdbms/admin/utl32k.sql script in the PDB. You must be connected AS SYSDBA to run the utl32k.sql script.
- 5. Reopen the PDB in NORMAL mode.

Note:

The utl32k.sql script increases the maximum size of the VARCHAR2, NVARCHAR2, and RAW columns for the views where this is required. The script does not increase the maximum size of the VARCHAR2, NVARCHAR2, and RAW columns in some views because of the way the SQL for those views is written.

6. Run the rdbms/admin/utlrp.sql script in the PDB to recompile invalid objects. You must be connected AS SYSDBA to run the script.

See Also:

Oracle Multitenant Administrator's Guide for more information about modifying the open mode of PDBs.

Increasing the Maximum Size of VARCHAR2, NVARCHAR2, and RAW Columns in an Oracle RAC Database

To increase the maximum size of VARCHAR2, NVARCHAR2, and RAW columns in an Oracle RAC database:

- 1. Shut down all of the Oracle RAC database instances, except one.
- 2. Restart the Oracle RAC database instance in UPGRADE mode.
- 3. Change the setting of MAX STRING SIZE to EXTENDED.
- 4. Run the rdbms/admin/utl32k.sql script in the Oracle RAC database instance. You must be connected AS SYSDBA to run the script.
- 5. Restart all Oracle RAC database instances in NORMAL mode.



Note:

The utl32k.sql script increases the maximum size of the VARCHAR2, NVARCHAR2, and RAW columns for the views where this is required. The script does not increase the maximum size of the VARCHAR2, NVARCHAR2, and RAW columns in some views because of the way the SQL for those views is written.

6. Run the rdbms/admin/utlrp.sql script to recompile invalid objects. You must be connected AS SYSDBA to run the script.

Increasing the Maximum Size of VARCHAR2, NVARCHAR2, and RAW Columns in an Oracle Data Guard Logical Standby Database

To increase the maximum size of VARCHAR2, NVARCHAR2, and RAW columns in an Oracle Data Guard logical standby database:

- 1. Shut down the Oracle Data Guard primary database and logical standby database.
- 2. Restart the primary database and logical standby database in UPGRADE mode.
- 3. Change the setting of MAX_STRING_SIZE to EXTENDED on the primary database and logical standby database.
- 4. Run the rdbms/admin/utl32k.sql script on both the primary database and the logical standby database. You must be connected AS SYSDBA to run the script.
- 5. Restart the primary database and logical standby database in NORMAL mode.

Note:

The ut132k.sql script increases the maximum size of the VARCHAR2, NVARCHAR2, and RAW columns for the views where this is required. The script does not increase the maximum size of the VARCHAR2, NVARCHAR2, and RAW columns in some views because of the way the SQL for those views is written.

- 6. Run the rdbms/admin/utlrp.sql script on the primary database and logical standby database to recompile invalid objects. You must be connected AS SYSDBA to run the script.
- Restart SQL Apply.

See Also:

Oracle Database Globalization Support Guide for more information about the ${\tt MAX_STRING_SIZE}$ parameter



2.228 MEMOPTIMIZE_POOL_SIZE

MEMOPTIMIZE_POOL_SIZE sets the size of the memoptimize pool, a memory area in the system global area (SGA) used by the Memoptimized Rowstore.

Property	Description
Parameter type	Big integer
Syntax	MEMOPTIMIZE_POOL_SIZE = integer [K M G]
Default value	0
Modifiable	No
Modifiable in a PDB	No
Range of values	0 to no maximum
Basic	No
Oracle RAC	Different values can be used on different instances.

The Memoptimized Rowstore improves the data query performance of applications, such as Internet of Things (IoT), that frequently query tables based on primary key values.

The Memoptimized Rowstore provides the capability of fast lookup of data for the tables that are mainly queried based on primary key columns.

This parameter specifies an integer value to indicate the amount of SGA to use for allocating the following structures for the memoptimize pool:

- The size of the buffer cache region: This is the total number of blocks for all MEMOPTIMIZE FOR READ tables.
- The size of the hash index segmented data structure pointing to the special blocks of MEMOPTIMIZE FOR READ tables.

Calculate the buffer cache requirement for the table being considered for MEMOPTIMIZE FOR READ, and include an additional 25% memory requirement for the hash index segmented data structure.

These structures are allocated from SGA at instance startup.

The value specified for this parameter counts toward SGA_TARGET. For example, if you set SGA_TARGET to 10 GB and you set MEMOPTIMIZE_POOL_SIZE to 2 GB, then 20% of the SGA_TARGET setting is allocated to the memoptimize pool.

Unlike other SGA components such as the buffer cache and shared pool, the memoptimize pool size is not controlled by automatic memory management. The database does not automatically shrink the memoptimize pool when the buffer cache or shared pool requires more memory, or increase the memoptimize pool when it runs out of space. You can only increase the size of the memoptimize pool by manually adjusting the MEMOPTIMIZE_POOL_SIZE initialization parameter.

See Also:

Oracle Database Performance Tuning Guide for more information about the Memoptimized Rowstore and the memoptimize pool.

2.229 MEMOPTIMIZE_WRITE_AREA_SIZE

MEMOPTIMIZE_WRITE_AREA_SIZE specifies the size of the ingest buffer in the large pool of the SGA. The ingest buffer is where Memoptimized Rowstore fast ingest inserts are buffered.

Property	Description
Parameter type	Big integer
Syntax	MEMOPTIMIZE_WRITE_AREA_SIZE = integer [K M G]
Default value	0
Modifiable	ALTER SYSTEM SID='*'
Modifiable in a PDB	No
Range of values	0 to operating system-dependent
Basic	No
Oracle RAC	You must set this parameter for every instance, and multiple instances must have the same value.



This parameter is available starting with Oracle Database 23ai.

2.230 MEMOPTIMIZE WRITES

MEMOPTIMIZE_WRITES allows you to instruct the database to always use fast ingest when inserting data into tables enabled for Memoptimized Rowstore fast ingest.

Property	Description
Parameter type	String
Syntax	MEMOPTIMIZE_WRITES = { HINT ON }
Default value	HINT
Modifiable	ALTER SESSION
Modifiable in a PDB	No
Basic	No

The fast ingest feature of Memoptimized Rowstore was introduced in Oracle Database 19c. At that time, the only way to direct the database to use fast ingest was to specify the MEMOPTIMIZE_WRITE hint in the INSERT statement.

Starting with Oracle Database 23ai, you can also direct the database to use fast ingest by setting the MEMOPTIMIZE WRITES initialization parameter as follows:

• ON - When inserting data into tables enabled for fast ingest, the fast ingest feature is always used, regardless of whether the MEMOPTIMIZE_WRITE hint is specified in the INSERT statement.

• HINT - When inserting data into tables enabled for fast ingest, the fast ingest feature is used only when the MEMOPTIMIZE WRITE hint is specified in the INSERT statement.

Note:

This parameter is available starting with Oracle Database 23ai.

2.231 MEMORY_MAX_SIZE

 ${\tt MEMORY_MAX_SIZE} \ \ specifies \ the \ maximum \ value \ to \ which \ a \ DBA \ can \ set \ the \ {\tt MEMORY_SIZE} \ initialization \ parameter.$

Property	Description
Parameter type	Big integer
Syntax	<pre>MEMORY_MAX_SIZE = integer [K M G]</pre>
Default value	The value of the MEMORY_SIZE initialization parameter
Modifiable	No
Modifiable in a PDB	No
Range of values	MEMORY_SIZE to the physical memory size available to the Oracle database
Basic	No
Oracle RAC	Different instances can use different values.

Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

"MEMORY_SIZE" for more information about how the settings of MEMORY_MAX_SIZE and MEMORY SIZE affect each other

2.232 MEMORY MAX TARGET

MEMORY_MAX_TARGET specifies the maximum value to which a DBA can set the MEMORY_TARGET initialization parameter.

Property	Description
Parameter type	Big integer
Syntax	<pre>MEMORY_MAX_TARGET = integer [K M G]</pre>
Default value	0



Property	Description
Modifiable	No
Modifiable in a PDB	No
Range of values	0 to the physical memory size available to the Oracle Database
Basic	No

Note:

On Linux platforms only, Oracle recommends against explicitly setting values for MEMORY_TARGET and MEMORY_MAX_TARGET in the initialization parameter file at the time of instance startup. Doing so may result in suboptimal small page allocation in the SGA. Furthermore, if the USE_LARGE_PAGES parameter is set to ONLY and you attempt to set values for MEMORY_TARGET and MEMORY_MAX_TARGET, the instance will fail to start.

See Also:

- Oracle Database Administrator's Guide for more information about managing memory
- "MEMORY_TARGET"
- "USE_LARGE_PAGES"

2.233 MEMORY SIZE

MEMORY SIZE specifies the size of instance-wide usable memory.

Property	Description
Parameter type	Big integer
Syntax	MEMORY_SIZE = integer [K M G]
Default value	0
Modifiable	You can use ALTER SYSTEM to increase or decrease the value of this parameter. However, the value of this parameter must always be greater than or equal to its value at instance startup.
Modifiable in a PDB	No
Range of values	1536 MB to MEMORY_MAX_SIZE
Basic	No
Oracle RAC	Different instances can use different values.

In a text-based initialization parameter file, if you omit MEMORY_MAX_SIZE and include a value for MEMORY_SIZE, then the database automatically sets MEMORY_MAX_SIZE to the value of MEMORY_SIZE. If you omit the line for MEMORY_SIZE and include a value for MEMORY_MAX_SIZE, the MEMORY SIZE parameter defaults to 0. After instance startup, you can dynamically change



 ${\tt MEMORY_SIZE} \ to \ a \ nonzero \ value, \ provided \ that \ it \ does \ not \ exceed \ the \ value \ of \ MEMORY \ MAX \ SIZE.$

Based on the value of MEMORY_SIZE, the values for the SGA and PGA parameters SGA_TARGET and PGA_AGGREGATE_LIMIT will be calculated internally for the instance. The sizing will also depend on the large page settings.

When MEMORY_SIZE is set to a nonzero value, the CDB-level values for the following SGA and PGA parameters will be ignored by the database: SGA_MAX_SIZE, PGA_AGGREGATE_LIMIT, and SGA_TARGET. If you assign a value to any of these parameters at the time of instance startup, the database will not report an error, but the parameter value will be ignored. If you dynamically modify any of these parameters, an error will be reported.

When MEMORY_SIZE is set to a nonzero value, the MEMORY_MAX_TARGET and MEMORY_TARGET parameters should be set to 0. If you attempt to set either of these parameters to a nonzero value, the request will be ignored, and a warning will be written to the alert log at startup.



This parameter is available starting with Oracle Database 23ai.

See Also:

"MEMORY_MAX_SIZE"

2.234 MEMORY_TARGET

MEMORY TARGET specifies the Oracle system-wide usable memory.

The database tunes memory to the MEMORY_TARGET value, reducing or enlarging the SGA and PGA as needed.

Property	Description
Parameter type	Big integer
Syntax	<pre>MEMORY_TARGET = integer[K M G]</pre>
Default value	0 (SGA autotuning is disabled for DEFERRED mode autotuning requests, but allowed for IMMEDIATE mode autotuning requests)
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	152 MB to MEMORY_MAX_TARGET
Basic	No

MEMORY_TARGET should be set higher than or equal to the sum of the current sizes of the SGA and PGA.

In a text-based initialization parameter file, if you omit MEMORY_MAX_TARGET and include a value for MEMORY_TARGET, then the database automatically sets MEMORY_MAX_TARGET to the value of MEMORY_TARGET. If you omit the line for MEMORY_TARGET and include a value for



MEMORY_MAX_TARGET, the MEMORY_TARGET parameter defaults to zero. After startup, you can then dynamically change MEMORY_TARGET to a nonzero value, provided that it does not exceed the value of MEMORY MAX TARGET.

Total memory usage can grow beyond the value of MEMORY_TARGET. For example, memory is allocated to PL/SQL tables and varrays regardless of the value of MEMORY_TARGET as long as memory is available at the operating system level.

In the **Default value** field, IMMEDIATE mode autotuning requests are necessary to avoid ORA-04031 errors. The DEFERRED and IMMEDIATE modes are reflected in the OPER_MODE column of the V\$MEMORY RESIZE OPS view.

Note:

On Linux platforms only, Oracle recommends against explicitly setting values for MEMORY_TARGET and MEMORY_MAX_TARGET in the initialization parameter file at the time of instance startup. Doing so may result in suboptimal small page allocation in the SGA. Furthermore, if the USE_LARGE_PAGES parameter is set to ONLY and you attempt to set values for MEMORY_TARGET and MEMORY_MAX_TARGET, the instance will fail to start.

Note:

The default value of SGA_MAX_SIZE depends on the values of MEMORY_TARGET and MEMORY_MAX_TARGET.

See Also:

- "MEMORY MAX TARGET"
- "USE LARGE PAGES"
- Oracle Database Administrator's Guide for more information about managing memory
- Oracle Multitenant Administrator's Guide for information about the initialization parameters that control the memory usage of PDBs

2.235 MIN AUTH SERVERS

MIN_AUTH_SERVERS specifies the minimum number of authentication servers in the authentication pool. This pool authenticates user connections when client applications connect to Database Resident Connection Pooling (DRCP).

Property	Description
Parameter type	Integer



Property	Description
Default value	If the ENABLE_PER_PDB_DRCP initialization parameter is set to true, then the default value for MIN_AUTH_SERVERS is 0 in each PDB.
	If the ENABLE_PER_PDB_DRCP initialization parameter is set to false, then the default value for MIN_AUTH_SERVERS is 1 in the CDB root.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	 Yes, according to the following guidelines: When the ENABLE_PER_PDB_DRCP initialization parameter is set to true, MIN_AUTH_SERVERS can be modified only at the PDB level; it cannot be modified for the CDB root. When the ENABLE_PER_PDB_DRCP initialization parameter is set to false, MIN_AUTH_SERVERS can be modified only for the CDB root; it cannot be modified at the PDB level.
Range of values	Minimum: 0
	Maximum: MAX_AUTH_SERVERS or PROCESSES, whichever is the lesser value
Basic	No
Oracle RAC	Different instances can use different values.

The value of this parameter must be less than or equal to the values of both the MAX_AUTH_SERVERS initialization parameter and the PROCESSES initialization parameter. The MAX_AUTH_SERVERS parameter specifies the maximum number of authentication servers in the authentication pool. The PROCESSES parameter specifies the maximum number of operating system user processes that can simultaneously connect to Oracle.

See Also:

- Oracle Database Administrator's Guide for more information about setting this parameter
- "ENABLE_PER_PDB_DRCP"
- "MAX_AUTH_SERVERS"
- "PROCESSES"

2.236 MLE_PROG_LANGUAGES

Use MLE_PROG_LANGUAGES to enable or disable Oracle Database Multilingual Engine (MLE) completely, or selectively enable certain languages.

Property	Description
Parameter type	String
Syntax	MLE_PROG_LANGUAGES = { ALL JAVASCRIPT OFF }
Default value	ALL
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes



Property	Description
Basic	No
Oracle RAC	Multiple instances must have the same value.

Values:

- ALL Enable MLE completely.
- JAVASCRIPT Enable MLE for JavaScript only. Note that MLE currently supports JavaScript as its sole language. Therefore, the JAVASCRIPT setting has the same effect as the ALL setting.
- OFF Disable MLE completely.

If MLE is disabled completely at the CDB level, it cannot be enabled at the PDB or session level. Similarly, if MLE is disabled completely at the PDB level, it cannot be enabled at the session level.



This parameter is available starting with Oracle Database 23ai.

See Also:

Oracle Database Multilingual Engine JavaScript Developer's Guide for more information about this parameter

2.237 MULTISHARD_QUERY_DATA_CONSISTENCY

MULTISHARD_QUERY_DATA_CONSISTENCY enables you to specify a data consistency setting for multi-shard queries.

Property	Description
Parameter type	String
Syntax	MULTISHARD_QUERY_DATA_CONSISTENCY = { STRONG SHARD_LOCAL DELAYED_STANDBY_ALLOWED }
Default value	STRONG
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The value of this parameter can be different on different Oracle RAC instances.

You can use MULTISHARD_QUERY_DATA_CONSISTENCY to avoid the cost of SCN synchronization when executing multi-shard queries across shards, which can be globally distributed.

MULTISHARD QUERY DATA CONSISTENCY can be set at the system level or the session level.

The values that can be set for MULTISHARD QUERY DATA CONSISTENCY are:

- STRONG: With this setting, SCN synchronization is performed across all shards, and data is consistent across all shards. This setting provides global consistent read capability. This is the default value.
- SHARD_LOCAL: With this setting, SCN synchronization is not performed across all shards. Data is consistent within each shard. This setting provides the most current data.
- DELAYED_STANDBY_ALLOWED: With this setting, SCN synchronization is not performed across
 all shards. Data is consistent within each shard. This setting allows data to be fetched from
 Data Guard standby databases when possible (for example, depending on load balancing),
 and may return stale data from standby databases.



Oracle Globally Distributed Database Guide for more information about specifying consistency levels in a multi-shard query

2.238 NLS CALENDAR

NLS CALENDAR specifies which calendar system Oracle uses.

Property	Description
Parameter type	String
Syntax	NLS_CALENDAR = "calendar_system"
Default value	None, implies GREGORIAN
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Range of values	Any valid calendar format name
Basic	No

NLS CALENDAR can have one of the following values:

- Arabic Hijrah
- English Hijrah
- Ethiopian
- GREGORIAN
- Japanese Imperial
- Persian
- ROC Official (Republic of China)
- Thai Buddha

For example, suppose NLS_CALENDAR is set to English Hijrah. The date format is DD Month YYYY. If the date is March 7, 2025, then the SYSDATE is displayed as follows:



SELECT SYSDATE FROM DUAL;
SYSDATE
----07 Ramadan 1446



The value of the initialization parameter NLS_CALENDER is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. If the initialization parameter is not specified, the initial session value becomes <code>GREGORIAN</code>. This initial value is overridden by a client-side value if the client is OCI-based and the <code>NLS_LANG</code> client setting (environment variable) is defined.

See Also:

Oracle Database Globalization Support Guide for a listing of available calendar systems

2.239 NLS_COMP

NLS COMP specifies the collation behavior of the database session.

Property	Description
Parameter type	String
Syntax	NLS_COMP = { BINARY LINGUISTIC ANSI }
Default value	BINARY
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Basic	No

Values

BINARY

Normally, comparisons in the \mbox{WHERE} clause and in PL/SQL blocks is binary unless you specify the $\mbox{NLSSORT}$ function.

LINGUISTIC

Comparisons for all SQL operations in the WHERE clause and in PL/SQL blocks should use the linguistic sort specified in the NLS_SORT parameter. To improve the performance, you can also define a linguistic index on the column for which you want linguistic comparisons.

ANSI

A setting of ANSI is for backward compatibility; in general, you should set ${\tt NLS_COMP}$ to ${\tt LINGUISTIC}$



Note:

Unless you explicitly set the value for NLS_COMP in your initialization parameter file, a default value of NULL is shown in the following views: V\$PARAMETER, V\$PARAMETER, V\$PARAMETER, V\$PARAMETER2, V\$SYSTEM_PARAMETER2, and NLS_INSTANCE_PARAMETERS. However, the actual default value, and behavior, is BINARY. Note that you cannot change the default to NULL, because NULL is not among the valid values.

Examples

See Oracle Database Globalization Support Guide for examples of using this parameter.



The value of this initialization parameter NLS_COMP is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overridden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the NLS_LANG client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.

2.240 NLS_CURRENCY

NLS_CURRENCY specifies the string to use as the local currency symbol for the L number format element. The default value of this parameter is determined by NLS_TERRITORY.

Property	Description
Parameter type	String
Syntax	NLS_CURRENCY = currency_symbol
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Range of values	Any valid character string, with a maximum of 10 bytes (not including null)
Basic	No

Note:

The value of this initialization parameter NLS_CURRENCY is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overridden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the NLS_LANG client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.



- Oracle Database Globalization Support Guide for more information about this parameter
- Oracle Database SQL Language Reference for information on number format elements

2.241 NLS_DATE_FORMAT

NLS_DATE_FORMAT specifies the default datetime format model to use with the TO_CHAR and TO DATE functions. The default value of this parameter is determined by NLS TERRITORY.

Property	Description
Parameter type	String
Syntax	NLS_DATE_FORMAT = "format"
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Range of values	Any valid datetime format model
Basic	No

The value of this parameter can be any valid datetime format model, and the value must be surrounded by double quotation marks. For example:

NLS_DATE_FORMAT = "MM/DD/YYYY"

Note:

The value of this initialization parameter NLS_DATE_FORMAT is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overridden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the NLS_LANG client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.

See Also:

- Oracle Database SQL Language Reference for more information on datetime format models
- Oracle Database Globalization Support Guide for more information about this parameter



2.242 NLS DATE LANGUAGE

NLS_DATE_LANGUAGE specifies the language to use for the spelling of day and month names and date abbreviations (a.m., p.m., AD, BC) returned by the TO_DATE and TO_CHAR functions.

Property	Description
Parameter type	String
Syntax	NLS_DATE_LANGUAGE = language
Default value	Derived from NLS_LANGUAGE
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Range of values	Any valid NLS_LANGUAGE value
Basic	No

Note:

The value of this initialization parameter NLS_DATE_LANGUAGE is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overridden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the NLS_LANG client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.

See Also:

- Oracle Database Globalization Support Guide for more information about this parameter
- Oracle Database SQL Language Reference for information on the TO_DATE function.
- Oracle Database SQL Language Reference for information on the TO_CHAR function.

2.243 NLS_DUAL_CURRENCY

NLS_DUAL_CURRENCY specifies the dual currency symbol (such as "Euro") for the territory. The default is the dual currency symbol defined in the territory of your current language environment.

Property	Description
Parameter type	String
Syntax	NLS_DUAL_CURRENCY = currency_symbol



Property	Description
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Range of values	Any valid format name up to 10 characters
Basic	No

Note:

The value of this initialization parameter <code>NLS_DUAL_CURRENCY</code> is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overridden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the <code>NLS_LANG</code> client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.

See Also:

Oracle Database Globalization Support Guide for more information about this parameter

2.244 NLS ISO CURRENCY

NLS_ISO_CURRENCY determines the string to use as the international currency symbol corresponding to the C number format element in a call to the TO_CHAR function.

Property	Description
Parameter type	String
Syntax	NLS_ISO_CURRENCY = territory
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Range of values	Any valid NLS_TERRITORY value
Basic	No

Local currency symbols can be ambiguous. For example, a dollar sign (\$) can refer to U.S. dollars or Australian dollars. ISO Specification 4217 defines unique "international" currency symbols for the currencies of specific territories or countries. The value of the NLS_ISO_CURRENCY parameter is the Oracle name of the territory whose ISO currency symbol is returned in place of the C number format element. For example, if NLS_ISO_CURRENCY is set to AMERICA, the string 'USD' is returned by TO_CHAR where the C element is specified in the format.



Note:

The value of this initialization parameter NLS_ISO_CURRENCY is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overridden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the NLS_LANG client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.

See Also:

- Oracle Database Globalization Support Guide for more information about this parameter
- Oracle Database SQL Language Reference for information on number format elements

2.245 NLS_LANGUAGE

NLS LANGUAGE specifies the default language of the database.

Property	Description
Parameter type	String
Syntax	NLS_LANGUAGE = language
Default value	Operating system-dependent, derived from the ${\tt NLS_LANG}$ environment variable
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Range of values	Any valid language name
Basic	Yes

This language specified by NLS_LANGUAGE is used for messages, day and month names, symbols for AD, BC, a.m., and p.m., and the default sorting mechanism. This parameter also determines the default values of the parameters NLS_DATE_LANGUAGE and NLS_SORT.

Note:

The value of this initialization parameter NLS_LANGUAGE is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overridden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the NLS_LANG client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.



Examples

See these examples of using the NLS LANGUAGE parameter:

- For an example of setting NLS_LANGUAGE to Italian, see *Oracle Database Globalization* Support Guide.
- For an example of overriding default values for NLS_LANGUAGE and NLS_TERRITORY during a session, see Oracle Database Globalization Support Guide.

See Also:

- Oracle Database Globalization Support Guide for more information about this parameter.
- Oracle Database Globalization Support Guide for a complete list of languages that can be specified using this parameter
- Oracle Database Globalization Support Guide for information on overriding the default values for this parameter
- Your operating system-specific Oracle documentation and the release notes for your country

2.246 NLS_LENGTH_SEMANTICS

NLS LENGTH SEMANTICS is used to specify length semantics.

Property	Description
Parameter type	String
Syntax	NLS_LENGTH_SEMANTICS = string
	Example: NLS_LENGTH_SEMANTICS = 'CHAR'
Default value	BYTE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	BYTE CHAR
Basic	No

The session-level value of NLS_LENGTH_SEMANTICS specifies the default length semantics to use for VARCHAR2 and CHAR table columns, user-defined object attributes, and PL/SQL variables in database objects created in the session. This default may be overridden by the explicit length semantics qualifiers BYTE and CHAR in column, attribute, and variable definitions.

The instance-level value of NLS_LENGTH_SEMANTICS provides a default for the session-level value if NLS_LENGTH_SEMANTICS is not set explicitly by the database client through the NLS_LENGTH_SEMANTICS client environment variable (does not apply to JDBC Thin clients), or the ALTER SESSION SET NLS_LENGTH_SEMANTICS statement.

NCHAR, NVARCHAR2, CLOB, and NCLOB columns are always character-based.

Sessions logged in as SYS do not use the NLS_LENGTH_SEMANTICS parameter. They use BYTE length semantics for all created objects unless overridden by the explicit BYTE and CHAR qualifiers in object definitions (SQL DDL statements).



Oracle strongly recommends that you do NOT set the <code>NLS_LENGTH_SEMANTICS</code> parameter to <code>CHAR</code> in the instance or server parameter file. This may cause many existing installation scripts to unexpectedly create columns with character length semantics, resulting in run-time errors, including buffer overflows.

See Also:

Oracle Database Globalization Support Guide for more information about this parameter

2.247 NLS_NCHAR_CONV_EXCP

NLS_NCHAR_CONV_EXCP determines whether an error is reported when there is data loss during an implicit or explicit character type conversion between NCHAR/NVARCHAR2 and CHAR/VARCHAR2.

Description
String
NLS_NCHAR_CONV_EXCP = { TRUE FALSE }
FALSE
ALTER SESSION, ALTER SYSTEM
Yes
No

The default value results in no error being reported.



Oracle Database Globalization Support Guide for more information about this parameter

2.248 NLS_NUMERIC_CHARACTERS

NLS_NUMERIC_CHARACTERS specifies the characters to use as the group separator and decimal character.

Property	Description
Parameter type	String
Syntax	NLS_NUMERIC_CHARACTERS =
	"decimal_character group_separator"
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Basic	No

NLS_NUMERIC_CHARACTERS overrides those characters defined implicitly by NLS_TERRITORY. The group separator separates integer groups (that is, thousands, millions, billions, and so on). The decimal separates the integer portion of a number from the decimal portion.

You can specify any character as the decimal or group separator. The two characters specified must be single-byte and must be different from each other. The characters cannot be any numeric character or any of the following characters: plus (+), minus sign (-), less than sign (<), greater than sign (>). Either character can be a space.

For example, if you want to specify a comma as the decimal character and a space as the group separator, you would set this parameter as follows:

NLS NUMERIC CHARACTERS = ", "

Note:

The value of this initialization parameter <code>NLS_NUMERIC_CHARACTERS</code> is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overridden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the <code>NLS_LANG</code> client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.

See Also:

Oracle Database Globalization Support Guide for more information about this parameter

2.249 NLS_SORT

NLS_SORT specifies the collating sequence for character value comparison in various SQL operators and clauses.

Property	Description
Parameter type	String
Syntax	<pre>NLS_SORT = { BINARY linguistic_definition }</pre>



Property	Description
Default value	Derived from NLS_LANGUAGE
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Range of values	BINARY or any valid linguistic definition name
Basic	No

For example, NLS_SORT specifies the collating sequence for character value comparison in these SQL operators and clauses: ORDER BY, GROUP BY, comparison conditions (=, <>, <=, >=), IN, BETWEEN, LIKE, MIN/MAX, GREATEST/LEAST, and INSTR.

- If the value is BINARY, then comparison is based directly on byte values in the binary encoding of the character values being compared. The ordering depends on the character set of the compared values, which is either the database character set (for VARCHAR2, CHAR, LONG, and CLOB) or the national character set (for NVARCHAR2, NCHAR, and NCLOB).
- If the value is a named linguistic sort, then comparison is defined by this sort. A linguistic
 sort uses various rules to achieve ordering expected by speakers of one or more natural
 languages. This is usually the same ordering that is used in dictionaries and telephone
 directories in those languages.

The exact operators and query clauses that obey the NLS_SORT parameter depend on the value of the NLS_COMP parameter. If an operator or clause does not obey the NLS_SORT value, as determined by NLS_COMP, the collation used is BINARY.

The BINARY comparison is faster and uses less resources than any linguistic comparison but for text in a natural language, it does not provide ordering expected by users.

The value of NLS_SORT affects execution plans of queries. Because a standard index cannot be used as a source of values sorted in a linguistic order, an explicit sort operation must usually be performed instead of an index range scan. A functional index on the NLSSORT function may be defined to provide values sorted in a linguistic order and reintroduce the index range scan to the execution plan.

Note:

The value of the initialization parameter NLS_SORT is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overriden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the NLS_LANG client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.

See Also:

 Oracle Database Globalization Support Guide for more information about this parameter and a current listing of values you can specify



2.250 NLS_TERRITORY

 ${\tt NLS_TERRITORY}$ specifies the name of the territory whose conventions are to be followed for day and week numbering.

Property	Description
Parameter type	String
Syntax	NLS_TERRITORY = territory
Default value	Operating system-dependent
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Range of values	Any valid territory name
Basic	Yes

This parameter also establishes the default date format, the default decimal character and group separator, and the default ISO and local currency symbols.

For information on these settings, see "NLS_DATE_FORMAT", "NLS_NUMERIC_CHARACTERS", "NLS_CURRENCY", and "NLS_ISO_CURRENCY".



The value of this initialization parameter <code>NLS_TERRITORY</code> is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overridden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the <code>NLS_LANG</code> client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.

Examples

For an example of overriding the default value for the NLS_TERRITORY parameter, see *Oracle Database Globalization Support Guide*.

See Also:

- Oracle Database Globalization Support Guide for a complete list of territories
- Your operating system-specific Oracle documentation for the territory-dependent default values for these parameters

2.251 NLS_TIMESTAMP_FORMAT

 ${\tt NLS_TIMESTAMP_FORMAT} \ \ {\tt defines} \ \ {\tt the} \ \ {\tt datetime} \ \ {\tt format} \ \ {\tt model} \ \ {\tt to} \ \ {\tt use} \ \ {\tt with} \ \ {\tt the} \ \ {\tt TO_CHAR} \ \ {\tt and} \ \ {\tt TO} \ \ {\tt TIMESTAMP} \ \ {\tt functions}.$

Property	Description
Parameter type	String
Syntax	NLS_TIMESTAMP_FORMAT = "format"
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Range of values	Any valid datetime format model
Basic	No

The value must be surrounded by quotation marks as follows:

```
NLS TIMESTAMP FORMAT = 'YYYY-MM-DD HH:MI:SS.FF'
```

You can specify the value of NLS_TIMESTAMP_FORMAT by setting it in the initialization parameter file. You can specify its value for a client as a client environment variable.

You can also alter the value of NLS_TIMESTAMP_FORMAT by changing its value in the initialization parameter and then restarting the instance. To alter the value during a session use the ALTER SESSION SET statement.

Note:

The value of this initialization parameter NLS_TIMESTAMP_FORMAT is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overridden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the NLS_LANG client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.

See Also:

- Oracle Database SQL Language Reference for more information on datetime format models
- Oracle Database Globalization Support Guide for more information about this parameter

2.252 NLS_TIMESTAMP_TZ_FORMAT

NLS_TIMESTAMP_TZ_FORMAT defines the default datetime format model to use with the TO_CHAR and $TO_TIMESTAMP_TZ$ functions.

Property	Description
Parameter type	String
Syntax	NLS_TIMESTAMP_TZ_FORMAT = "format"



Property	Description
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Modifiable in a PDB	Yes
Range of values	Any valid datetime format model
Basic	No

The value must be surrounded by quotation marks as follows:

NLS_TIMESTAMP_TZ_FORMAT = 'YYYY-MM-DD HH:MI:SS.FF TZH:TZM'

You can specify the value of NLS_TIMESTAMP_TZ_FORMAT by setting it in the initialization parameter file. You can specify its value for a client as a client environment variable.

You can also alter the value of <code>NLS_TIMESTAMP_TZ_FORMAT</code> by changing its value in the initialization parameter and then restarting the instance. To alter the value during a session use the <code>ALTER_SESSION_SET</code> statement.

Note:

The value of this initialization parameter <code>NLS_TIMESTAMP_TZ_FORMAT</code> is used to initialize the session value of this parameter, which is the actual value referenced by the SQL query processing. This initial value is overridden by a client-side value if the client uses the Oracle JDBC driver or if the client is OCI-based and the <code>NLS_LANG</code> client setting (environment variable) is defined. The initialization parameter value is, therefore, usually ignored.

See Also:

- Oracle Database SQL Language Reference for more information on datetime format models
- Oracle Database Globalization Support Guide for more information about this parameter

2.253 NONCDB_COMPATIBLE

NONCOB_COMPATIBLE enables you to get behavior similar to a non-CDB when issuing SQL commands inside a PDB in a CDB.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No



Property	Description
Range of values	true false
Basic	No

Set this parameter if you are using a single PDB in your CDB configuration and you have legacy code that causes <code>ORA-65040</code> when you run it in the PDB.

Values

TRUE

Indicates the behavior for SQL statements will be like a non-CDB although the statements are issued in a PDB in a CDB.

FALSE

Indicates the behavior will for SQL statements will be like a CDB. This is the default value of the parameter.

There are some statements (such as ALTER DB BACKUP CONTROLFILE) which can be issued in a non-CDB, but in a CDB they must be issued in the root (because they affect the whole CDB) and would result in an error if issued in a PDB.

Some ALTER DATABASE or ALTER SYSTEM statements are not permitted inside a PDB, and they will fail if NONCDB_COMPATIBLE=FALSE is set in init.ora. However, these statements will succeed if NONCDB_COMPATIBLE=TRUE is set.

See Also:

- Oracle Multitenant Administrator's Guide for an introduction to PDBs and CDBs
- Oracle Multitenant Administrator's Guide for information about the initialization parameters that control the memory usage of PDBs
- Oracle Multitenant Administrator's Guide for information about the ALTER SYSTEM statements that can be run in a PDB.

2.254 OBJECT_CACHE_MAX_SIZE_PERCENT

The **object cache** is a memory block on the client that allows applications to store entire objects and to navigate among them without round trips to the server.

OBJECT_CACHE_MAX_SIZE_PERCENT specifies the percentage of the optimal cache size that the session object cache can grow past the optimal size.

Property	Description
Parameter type	Integer
Default value	10
Modifiable	ALTER SESSION, ALTER SYSTEM DEFERRED
Modifiable in a PDB	Yes
Range of values	0 to operating system-dependent maximum



Property	Description
Basic	No

The maximum size is equal to the optimal size plus the product of this percentage and the optimal size. When the cache size exceeds this maximum size, the system will attempt to shrink the cache to the optimal size.

See Also:

- "OBJECT_CACHE_OPTIMAL_SIZE" for a description of the object cache
- Pro*C/C++ Developer's Guide and Oracle Call Interface Developer's Guide for information on precompiler use of the object cache

2.255 OBJECT CACHE OPTIMAL SIZE

OBJECT_CACHE_OPTIMAL_SIZE specifies (in bytes) the size to which the session object cache is reduced when the size of the cache exceeds the maximum size.

Property	Description
Parameter type	Integer
Default value	51200000 (50M)
Modifiable	ALTER SESSION, ALTER SYSTEM DEFERRED
Modifiable in a PDB	Yes
Range of values	10 KB to operating system-dependent maximum
Basic	No

The **object cache** is a memory block on the client that allows applications to store entire objects and to navigate among them without round trips to the server. On the server, it is used to cache frequently used objects requested as a result of client requests to help in performance.



*Pro*C/C++ Developer's Guide* and *Oracle Call Interface Developer's Guide* for information on precompiler use of the object cache

2.256 OFS_THREADS

 ${\tt OFS_THREADS}$ sets the maximum number of Oracle file system (OFS) threads that can be started to service Oracle file system requests.

Property	Description
Parameter type	Integer
Default value	4
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	2 to 128
Basic	No
Oracle RAC	The same value should be specified on all instances.



This initialization parameter is supported only on the Linux operating system.

See Also:

Oracle Database Administrator's Guide for more information about the Oracle Database NFS server feature

2.257 OLAP_PAGE_POOL_SIZE

OLAP_PAGE_POOL_SIZE specifies (in bytes) the size of the OLAP page pool.

Property	Description
Parameter type	Big integer
Syntax	OLAP_PAGE_POOL_SIZE = integer [K M G]
Default value	0
Modifiable	ALTER SESSION, ALTER SYSTEM DEFERRED
Modifiable in a PDB	Yes
Range of values	0, 2097152 (2 MB) to 2 ³¹ - 1
Basic	No



\mathbf{A}

Caution:

Oracle strongly recommends that you leave this parameter set to its default value of 0. This instructs the database to autonomously adjust the OLAP page pool size to serve the requirements of the analytic workspaces. Setting this parameter to a value other than 0 can lead to errors in OLAP processing or wasted memory.

See Also:

Oracle OLAP User's Guide for more information about the OLAP option for Oracle Database

2.258 ONE_STEP_PLUGIN_FOR_PDB_WITH_TDE

If a pluggable database (PDB) has Transparent Data Encryption-encrypted (TDE-encrypted) tables or tablespaces, you can enable <code>ONE_STEP_PLUGIN_FOR_PDB_WITH_TDE</code> on the target CDB to simplify the move of TDE keys in a single step PDB move operation.

ONE_STEP_PLUGIN_FOR_PDB_WITH_TDE eliminates the need of having to manually provide a keystore password when you import the TDE keys into the PDB after it has moved to the target CDB.

Property	Description
Parameter type	Boolean
Default value	FALSE
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	A different value can be set for this parameter on different Oracle RAC instances.

The default for ONE_STEP_PLUGIN_FOR_PDB_WITH_TDE is FALSE.

When <code>ONE_STEP_PLUGIN_FOR_PDB_WITH_TDE</code> is set to <code>TRUE</code> on the target CDB, the plug in of the PDB does not require a keystore password.



Note:

The ONE_STEP_PLUGIN_FOR_PDB_WITH_TDE parameter is deprecated in Oracle Database 23ai.

ONE_STEP_PLUGIN_FOR_PDB_WITH_TDE enables you to clone remotely or to relocate encrypted PDBs without providing the Transparent Data Encryption (TDE) keystore password. However, EXTERNAL STORE provides the same functionality, and is universally applicable to all ADMINISTER KEY MANAGEMENT statements that do not change the TDE configuration. Instead of using

ONE_STEP_PLUGIN_FOR_PDB_WITH_TDE, Oracle recommends that you use the IDENTIFIED BY EXTERNAL STORE clause for the ADMINISTER KEY MANAGEMENT statement.

2.259 OPEN_CURSORS

OPEN_CURSORS specifies the maximum number of open cursors (handles to private SQL areas) a session can have at once. You can use this parameter to prevent a session from opening an excessive number of cursors.

Property	Description
Parameter type	Integer
Default value	50
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 65535
Basic	Yes

It is important to set the value of <code>OPEN_CURSORS</code> high enough to prevent your application from running out of open cursors. The number will vary from one application to another. Assuming that a session does not open the number of cursors specified by <code>OPEN_CURSORS</code>, there is no added overhead to setting this value higher than actually needed.

See Also:

- Oracle Database Performance Tuning Guide for more information on setting this parameter
- Your operating system-specific Oracle documentation for the range of values

2.260 OPEN LINKS

OPEN_LINKS specifies the maximum number of concurrent open connections to remote databases in one session. These connections include database links, as well as external procedures and cartridges, each of which uses a separate process.

Property	Description
Parameter type	Integer
Default value	4
Modifiable	No
Modifiable in a PDB	Yes
Range of values	0 to 32768
Basic	No

Oracle counts one open link for the following:

- For each user that references a public or private database link
- For each external procedure or cartridge connection when it is executed for the first time

Both types of connections close when the session ends. You can also close a database link connection explicitly by issuing an ALTER SESSION CLOSE DATABASE LINK statement.

You should set this parameter to allow for the external procedure and cartridge connections expected during the session plus the number of databases referred to in typical distributed transactions (that is, a single SQL statement that references multiple databases), so that all the databases can be open to execute the statement. For example, if queries alternately access databases A, B, and C, and <code>OPEN_LINKS</code> is set to 2, time will be lost waiting while one connection is broken and another made. Increase the value if many different databases are accessed over time.

This parameter refers only to connections used for distributed transactions. Direct connections to a remote database specified as an application connects are not counted.

If you set OPEN LINKS to 0, then no distributed transactions are allowed.

In a multitenant container database (CDB), the <code>OPEN_LINKS</code> parameter can be set at both the root and at the PDB level using either an initialization parameter file (PFILE) or server parameter file (SPFILE). You need to restart the instance or reopen the PDB for a new <code>OPEN_LINKS</code> value to become effective.



"OPEN_LINKS_PER_INSTANCE" for information on setting open connections globally for a database instance

2.261 OPEN_LINKS_PER_INSTANCE

OPEN_LINKS_PER_INSTANCE specifies the maximum number of migratable open connections globally for each database instance.

Property	Description
Parameter type	Integer
Default value	4
Modifiable	No



Property	Description
Modifiable in a PDB	No
Range of values	0 to 2 ³¹ - 1
Basic	No
Oracle RAC	Multiple instances can have different values.

XA transactions use migratable open connections so that the connections are cached after a transaction is committed. Another transaction can use the connection, provided the user who created the connection is the same as the user who owns the transaction.

OPEN_LINKS_PER_INSTANCE is different from OPEN_LINKS, which indicates the number of connections from a session. The OPEN_LINKS parameter is not applicable to XA applications.

See Also:

- "OPEN_LINKS"
- Oracle Database Development Guide for more information about using this parameter in Oracle XA applications

2.262 OPTIMIZER ADAPTIVE PLANS

OPTIMIZER_ADAPTIVE_PLANS controls adaptive plans. Adaptive plans are execution plans built with alternative choices that are decided at run time based on statistics collected as the query executes.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	The same value must be set on all instances

Setting this parameter to false disables the following adaptive features:

- Nested loop join/hash join selection
- Star transformation bitmap pruning
- Adaptive parallel distribution method

See Also:

Oracle Database SQL Tuning Guide for information about adaptive plans

2.263 OPTIMIZER_ADAPTIVE_REPORTING_ONLY

OPTIMIZER ADAPTIVE REPORTING ONLY controls reporting-only mode for adaptive optimizations.

Property	Description
Parameter type	Boolean
Default value	FALSE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	TRUE FALSE
Basic	No

When OPTIMIZER_ADAPTIVE_REPORTING_ONLY is set to FALSE, reporting-only mode is off, and the adaptive optimizations are enabled as usual.

When <code>OPTIMIZER_ADAPTIVE_REPORTING_ONLY</code> is set to <code>TRUE</code>, adaptive optimizations run in reporting-only mode. With this setting, the information required for an adaptive optimization is gathered, but no action is taken to change the plan. For instance, an adaptive plan will always choose the default (optimizer-chosen) plan, but information is collected on what plan to adapt to in non-reporting mode. This information can be viewed in the adaptive plan report.

This parameter affects only adaptive optimizations that are enabled.

See Also:

- Oracle Database SQL Tuning Guide for more information about adaptive plans and automatic reoptimization
- Oracle Database SQL Tuning Guide for information on controlling adaptive optimization

2.264 OPTIMIZER_ADAPTIVE_STATISTICS

OPTIMIZER_ADAPTIVE_STATISTICS controls adaptive statistics. Some query shapes are too complex to rely on base table statistics alone, so the optimizer augments these statistics with adaptive statistics.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	The same value must be set on all instances
Oracle RAC	The same value must be set on all instances



Setting this parameter to false disables the following adaptive features:

- SQL plan directives
- Statistics feedback for joins
- Adaptive dynamic sampling for parallel execution



Setting $OPTIMIZER_ADAPTIVE_STATISTICS$ to false preserves the statistics feedback functionality that was introduced in Oracle Database 11g.

OPTIMIZER_ADAPTIVE_STATISTICS does not control the creation of SQL plan directives. SQL plan directives will be created even if this parameter is false, but they will not be used to refine SQL execution plans with dynamic sampling.

See Also:

Oracle Database SQL Tuning Guide for information about adaptive plans

2.265 OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES

OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES enables or disables the automatic recognition of repeatable SQL statements, as well as the generation of SQL plan baselines for such statements.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

See Also:

Oracle Database SQL Tuning Guide for more information about the optimizer



2.266 OPTIMIZER_CAPTURE_SQL_QUARANTINE

 ${\tt OPTIMIZER_CAPTURE_SQL_QUARANTINE} \ \ \textbf{enables} \ \ \textbf{or disables} \ \ \textbf{the automatic creation of SQL} \\ \textbf{Quarantine configurations}.$

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Different instances can use different values.

Values:

true

Enables the automatic creation of SQL Quarantine configurations. If the Resource Manager terminates a SQL statement because the statement has exceeded resource limits, then the database automatically creates a SQL Quarantine configuration for the execution plan used by the terminated SQL statement. Note that the *plan* for a terminated SQL statement is quarantined, not the statement itself.

false

Disables the automatic creation of SQL Quarantine configurations.

See Also:

- "DBA_SQL_QUARANTINE"
- "OPTIMIZER_USE_SQL_QUARANTINE"
- Oracle Database SQL Tuning Guide for more information about SQL Quarantine

2.267 OPTIMIZER_CROSS_SHARD_RESILIENCY

OPTIMIZER CROSS SHARD RESILIENCY enables resilient execution of cross shard queries.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No



Property	Description
Oracle RAC	The same value must be used on all instances.

When this parameter is set to true and a cross-shard query fails on one or more shards, the query execution continues on the Oracle Data Guard standbys of the failed shards. This parameter is not set to true by default, because there may be performance overhead for the resilient query execution.

2.268 OPTIMIZER_DYNAMIC_SAMPLING

OPTIMIZER_DYNAMIC_SAMPLING controls both when the database gathers dynamic statistics, and the size of the sample that the optimizer uses to gather the statistics.

Property	Description
Parameter type	Integer
Default value	If OPTIMIZER_FEATURES_ENABLE is set to 10.0.0 or higher, then 2
	If OPTIMIZER_FEATURES_ENABLE is set to 9.2.0, then 1
	If OPTIMIZER_FEATURES_ENABLE is set to 9.0.1 or lower, then 0
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 11
Basic	No



Dynamic statistics were called dynamic sampling in releases earlier than Oracle Database 12c Release 1 (12.1).

If the value of OPTIMIZER_DYNAMIC_SAMPLING is set to 11, the OPTIMIZER_FEATURES_ENABLE setting has no effect on the OPTIMIZER_DYNAMIC_SAMPLING setting.



Oracle Database SQL Tuning Guide for detailed information about the values (0-11) that can be set for the OPTIMIZER DYNAMIC SAMPLING parameter.

2.269 OPTIMIZER_FEATURES_ENABLE

OPTIMIZER_FEATURES_ENABLE acts as an umbrella parameter for enabling a series of optimizer features based on an Oracle release number.

Property	Description
Parameter type	String
Syntax	OPTIMIZER_FEATURES_ENABLE = { 8.0.0 8.0.3 8.0.4 8.0.5 8.0.6 8.0.7 8.1.0 8.1.3 8.1.4 8.1.5 8.1.6 8.1.7 9.0.0 9.0.1 9.2.0 9.2.0.8 10.1.0 10.1.0.3 10.1.0.4 10.1.0.5 10.2.0.1 10.2.0.2 10.2.0.3 10.2.0.4 10.2.0.5 11.1.0.6 11.1.0.7 11.2.0.1 11.2.0.2 11.2.0.3 11.2.0.4 12.1.0.1 12.1.0.2 12.2.0.1 18.1.0 19.1.0 21.1.0 23.1.0 }
Default value	23.1.0
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

For example, if you upgrade your database from release 19c to release 23ai, but you want to keep the release 19c optimizer behavior, you can do so by setting this parameter to 19.1.0. At a later time, you can try the enhancements introduced in releases up to and including release 23ai by setting the parameter to 23.1.0.



When setting this parameter to a value representing Oracle Database 18c or later, you must specify three numeric values separated by periods, such as 21.1.0 or 23.1.0.

Table 2-6 describes some of the optimizer features that are enabled when you set the OPTIMIZER FEATURES ENABLE parameter to 12.1.0.2 or a later release.

Table 2-6 Optimizer Features for Oracle Database 12c and Later Releases

Features	12.1.0.2	12.2.0.1	18c	19c	21c	23ai
Adaptive Query Optimization	Х	Х	Х	Χ	Х	Х
Online statistics gathering for bulk loads	Χ	Χ	Χ	Χ	Χ	Χ
Session level statistics for Global Temporary Tables	X	Χ	Χ	X	Χ	Χ
Multi-table left outer joins	Χ	Χ	Χ	Χ	Χ	Χ
Lateral views	Χ	Χ	Χ	Χ	Χ	Χ
Batch table access by rowid	Χ	Χ	Χ	Χ	Χ	Χ
Null accepting semi joins	Χ	Χ	Χ	Χ	Χ	Χ
Scalar subquery unnesting	Χ	Χ	Χ	Χ	Χ	Χ
Conversion of joins that produce unnecessary duplicates to semi-joins	X	Χ	Χ	Χ	Χ	Χ
Parallel Union and Parallel Union All operations	Χ	Χ	Χ	Χ	Χ	Χ
Enhance Auto DOP	Χ	Χ	Χ	Χ	Χ	Χ
Approximate count distinct	X	Χ	Χ	X	X	X



Table 2-6 (Cont.) Optimizer Features for Oracle Database 12c and Later Releases

Features	12.1.0.2	12.2.0.1	18c	19c	21c	23ai
Support for Oracle Database In-Memory	Х	Х	Х	Х	X	Х
Group-by and aggregation elimination	Χ	Χ	Χ	Χ	Χ	Χ
Query rewrite for approximate query processing		Χ	Χ	Χ	Χ	Χ
Statistics advisor		Χ	Χ	Χ	Χ	Χ
Support for sharded databases		Χ	Χ	Χ	Χ	Χ
Expression tracking		Χ	Χ	Χ	Χ	Χ
Space-saving algorithm for partition synopses		Χ	Χ	X	Χ	Х
Oracle In-Memory Database statistics		Χ	Χ	X	Χ	X
Support for sharding		Χ	Χ	Χ	Χ	Χ
Cost-based OR expansion		Χ	Χ	Χ	Χ	Χ
Sub-query elimination		Χ	Χ	Χ	Χ	Χ
Multi-column key join elimination		Χ	Χ	Χ	Χ	Χ
SQL Quarantine				Χ	Χ	Χ
Gathering and use of real-time statistics				Χ	Χ	Χ
Use of automatic indexes				Χ	Χ	X
Exists-to-any rewrite of subqueries					Χ	Х
WITH clause filter predicate pushdown					Χ	X
Number of distinct value (NDV) modeling for real time statistics					Х	Х
Pushing down group-by into union-all branches						Х
Subsumption of views or subqueries						Х

See Also:

Oracle Database SQL Tuning Guide for more information about the optimizer

2.270 OPTIMIZER_IGNORE_HINTS

OPTIMIZER IGNORE HINTS enables embedded hints to be ignored.

Property	Description
Parameter type	Boolean
Default value	FALSE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Different values can be set on different instances.

When this parameter is set to TRUE, the optimizer ignores embedded hints.

The default value is FALSE. When this parameter's value is FALSE, the optimizer does not ignore embedded hints.

2.271 OPTIMIZER_IGNORE_PARALLEL_HINTS

OPTIMIZER IGNORE PARALLEL HINTS enables embedded parallel hints to be ignored.

Property	Description
Parameter type	Boolean
Default value	FALSE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Different values can be set on different instances.

When this parameter is set to TRUE, the optimizer ignores embedded parallel hints.

The default value is FALSE. When this parameter's value is FALSE, the optimizer does not ignore parallel embedded hints.

2.272 OPTIMIZER_INDEX_CACHING

OPTIMIZER_INDEX_CACHING lets you adjust the behavior of cost-based optimization to favor nested loops joins and IN-list iterators.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 100
Basic	No

The cost of executing an index using an IN-list iterator or of executing a nested loops join when an index is used to access the inner table depends on the caching of that index in the buffer cache. The amount of caching depends on factors that the optimizer cannot predict, such as the load on the system and the block access patterns of different users.

You can modify the optimizer's assumptions about index caching for nested loops joins and IN-list iterators by setting this parameter to a value between 0 and 100 to indicate the percentage of the index blocks the optimizer should assume are in the cache. Setting this parameter to a higher value makes nested loops joins and IN-list iterators look less expensive to the optimizer. As a result, it will be more likely to pick nested loops joins over hash or sort-merge joins and to pick indexes using IN-list iterators over other indexes or full table scans. The default for this parameter is 0, which results in default optimizer behavior.

See Also:

Oracle Database SQL Language Reference for additional information about this initialization parameter

2.273 OPTIMIZER INDEX COST ADJ

OPTIMIZER_INDEX_COST_ADJ lets you tune optimizer behavior for access path selection to be more or less index friendly—that is, to make the optimizer more or less prone to selecting an index access path over a full table scan.

Property	Description
Parameter type	Integer
Default value	100
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	1 to 10000
Basic	No

The default for this parameter is 100 percent, at which the optimizer evaluates index access paths at the regular cost. Any other value makes the optimizer evaluate the access path at that percentage of the regular cost. For example, a setting of 50 makes the index access path look half as expensive as normal.

Note:

The adjustment does not apply to user-defined cost functions for domain indexes.

See Also:

Oracle Database SQL Language Reference for additional information about this initialization parameter

2.274 OPTIMIZER_INMEMORY_AWARE

OPTIMIZER_INMEMORY_AWARE enables or disables all of the optimizer cost model enhancements for in-memory.

Property	Description	
Parameter type	Boolean	
Default value	true	
Modifiable	ALTER SESSION, ALTER SYSTEM	

Property	Description
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	All instances should use the same value

Setting the parameter to false causes the optimizer to ignore the in-memory property of tables during the optimization of SQL statements. This behavior can also be achieved by setting the OPTIMIZER FEATURES ENABLE initialization parameter to values lower than 12.1.0.2.

See Also:

- "OPTIMIZER FEATURES ENABLE"
- Oracle Database Concepts and Oracle Database SQL Tuning Guide for more information about the optimizer

2.275 OPTIMIZER_MODE

OPTIMIZER_MODE establishes the default behavior for choosing an optimization approach for the instance.

Property	Description
Parameter type	String
Syntax	OPTIMIZER_MODE =
	{ FIRST_ROWS_[1 10 100 1000] FIRST_ROWS ALL_ROWS }
Default value	ALL_ROWS
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Values

• FIRST_ROWS_n

The optimizer uses a cost-based approach and optimizes with a goal of best response time to return the first n rows (where n = 1, 10, 100, 1000).

FIRST ROWS

The optimizer uses a mix of costs and heuristics to find a best plan for fast delivery of the first few rows.

FIRST_ROWS is available for backward compatibility and plan stability; use FIRST_ROWS_n instead.

ALL ROWS

The optimizer uses a cost-based approach for all SQL statements in the session and optimizes with a goal of best throughput (minimum resource use to complete the entire statement).

See Also:

- Oracle Database SQL Tuning Guide for more information on setting this parameter
- Oracle Database Concepts and Oracle Database SQL Tuning Guide for more information about the optimizer

2.276 OPTIMIZER_REAL_TIME_STATISTICS

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Different values can be set on different instances.

When this parameter is set to true, the database automatically gathers real-time statistics during conventional DML operations. The default setting is false, which means real-time statistics are disabled.

2.277 OPTIMIZER_SESSION_TYPE

 ${\tt OPTIMIZER_SESSION_TYPE} \ controls \ how \ the \ database \ performs \ automatic \ indexing \ for \ SQL \ statements \ in \ the \ session.$

Property	Description
Parameter type	String
Syntax	OPTIMIZER_SESSION_TYPE = { NORMAL CRITICAL ADHOC }
Default value	NORMAL
Modifiable	ALTER SESSION
Modifiable in a PDB	No
Basic	No

Values:

NORMAL

Automatic indexing is performed normally for SQL statements in the session. The automatic indexing process identifies auto index candidates, creates auto indexes, and verifies them against SQL statements.

• CRITICAL

Automatic indexing is performed for SQL statements in the session, with a higher priority given to long-running statements. The automatic indexing process identifies auto index candidates and creates auto indexes. However, auto indexes are verified against SQL statements in descending order of elapsed execution time for each statement.

ADHOC

Automatic indexing is suspended for SQL statements in the session. The automatic indexing process does not identify auto index candidates, create auto indexes, or verify auto indexes against SQL statements. This setting is useful when running ad hoc queries or testing new functionality.



Oracle Database Administrator's Guide for more information about how automatic indexing works

2.278 OPTIMIZER_USE_INVISIBLE_INDEXES

OPTIMIZER USE INVISIBLE INDEXES enables or disables the use of invisible indexes.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

Values

true

Invisible indexes are treated as visible (normal) indexes.

false

Invisible indexes will not be considered by the optimizer but will still be maintained by DML operations.

2.279 OPTIMIZER_USE_PENDING_STATISTICS

OPTIMIZER_USE_PENDING_STATISTICS specifies whether the optimizer uses pending statistics when compiling SQL statements.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

See Also

Oracle Database SQL Tuning Guide for more information on setting this parameter

2.280 OPTIMIZER_USE_SQL_PLAN_BASELINES

OPTIMIZER_USE_SQL_PLAN_BASELINES enables or disables the use of SQL plan baselines stored in SQL Management Base.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

When enabled, the optimizer looks for a SQL plan baseline for the SQL statement being compiled. If one is found in SQL Management Base, then the optimizer will cost each of the baseline plans and pick one with the lowest cost.

See Also:

- Oracle Database SQL Tuning Guide for information on enabling automatic initial plan capture
- Oracle Database SQL Tuning Guide for information about configuring the capture and use of SQL plan baselines

2.281 OPTIMIZER_USE_SQL_QUARANTINE

OPTIMIZER_USE_SQL_QUARANTINE determines whether the optimizer considers SQL Quarantine configurations when choosing an execution plan for a SQL statement.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Different instances can use different values.

Values:

true

The optimizer considers SQL Quarantine configurations when choosing an execution plan for a SQL statement. The statement will be terminated if it would otherwise use a quarantined execution plan.

• false

The optimizer ignores SQL Quarantine configurations when choosing an execution plan for a SQL statement. It may use any execution plan, regardless of its quarantine status.

See Also:

- "DBA_SQL_QUARANTINE"
- "OPTIMIZER_CAPTURE_SQL_QUARANTINE"
- Oracle Database SQL Tuning Guide for more information about SQL Quarantine

2.282 OS AUTHENT PREFIX

OS_AUTHENT_PREFIX specifies a prefix that Oracle Database uses to authenticate users attempting to connect to the server.

Property	Description
Parameter type	String
Syntax	OS_AUTHENT_PREFIX = authentication_prefix
Default value	OPS\$
Modifiable	No
Modifiable in a PDB	No
Basic	No



Oracle Database concatenates the value of this parameter to the beginning of the user's operating system account name. When a connection request is attempted, Oracle Database compares the prefixed username with Oracle user names in the database.

The default value of this parameter is OPS\$ for backward compatibility with previous versions. However, you might prefer to set the prefix value to "" (a null string), thereby eliminating the addition of any prefix to operating system account names.



The text of the OS_AUTHENT_PREFIX parameter is case sensitive on some operating systems.

See Also:

- Oracle Database Security Guide for more information on setting this parameter
- Your operating system-specific Oracle documentation for the default value

2.283 OS ROLES

 ${\tt OS_ROLES}$ determines whether Oracle or the operating system identifies and manages the roles of each username.

Note:

Enterprise User Security (EUS) is deprecated with Oracle Database 23ai.

Oracle recommends that you migrate to using Centrally Managed Users (CMU). This feature enables you to directly connect with Microsoft Active Directory without an intervening directory service for enterprise user authentication and authorization to the database. If your Oracle Database is in the cloud, you can also choose to move to one of the newer integrations with a cloud identity provider.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No

Values

TRUE



The operating system completely manages the role grants for all database usernames. When a user attempts to create a session, the username's security domain is initialized using the roles identified by the operating system.

Revocation by Oracle of roles granted by the operating system is ignored, as are any roles previously granted by Oracle.

FALSE

Oracle identifies and manages the roles.



Oracle Database Administrator's Guide and Oracle Database Enterprise User Security Administrator's Guide for more information on roles and on setting this parameter

2.284 OUTBOUND_DBLINK_PROTOCOLS

OUTBOUND_DBLINK_PROTOCOLS specifies the network protocols allowed for communicating for outbound database links in the database.

Property	Description
Parameter type	String
Syntax	OUTBOUND_DBLINK_PROTOCOLS = { ALL NONE [TCP [,] TCPS [,] IPC] }
Default value	ALL
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	The same value must be used on all instances.

Specify a value of ALL to allow all network protocols, and a value of NONE to disallow all network communication protocols for database link communication. Specify a single value or a list of comma separated network communication protocols to allow only certain network protocols for outbound database link communication.

2.285 PARALLEL_ADAPTIVE_MULTI_USER

PARALLEL_ADAPTIVE_MULTI_USER, when set to true, enables an adaptive algorithm designed to improve performance in multiuser environments that use parallel execution.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No

Property	Description
Range of values	true false
Basic	No

The algorithm automatically reduces the requested degree of parallelism based on the system load at query startup time. The effective degree of parallelism is based on the default degree of parallelism, or the degree from the table or hints, divided by a reduction factor.

The algorithm assumes that the system has been tuned for optimal performance in a singleuser environment.

Tables and hints use the default degree of parallelism.

Note:

The PARALLEL_ADAPTIVE_MULTI_USER initialization parameter is deprecated in Oracle Database 12c Release 2 (12.2.0.1) and may be removed in a future release. Oracle recommends that you use the parallel statement queuing feature instead.

See Also:

- Oracle Database SQL Language Reference for more information about optimizer hints
- Oracle Database VLDB and Partitioning Guide for more information about parallel statement queuing

2.286 PARALLEL DEGREE LIMIT

PARALLEL_DEGREE_LIMIT limits the degree of parallelism used by the optimizer to ensure that parallel server processes do not flood the system.

Property	Description
Parameter type	String
Syntax	PARALLEL_DEGREE_LIMIT = { CPU AUTO IO integer }
Default value	CPU
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

With automatic degree of parallelism, Oracle automatically decides whether a statement should execute in parallel and what degree of parallelism the statement should use. The optimizer automatically determines the degree of parallelism for a statement based on the resource requirements of the statement. However, PARALLEL_DEGREE_LIMIT enforces the limit for the degree of parallelism used by the optimizer.

Values

CPU

The maximum degree of parallelism is limited by the number of CPUs in the system. The formula used to calculate the limit is PARALLEL_THREADS_PER_CPU * CPU_COUNT * the number of instances available (by default, all the opened instances on the cluster but can be constrained using PARALLEL INSTANCE GROUP or service specification). This is the default.

AUTC

This value is equivalent to the CPU value.

• TC

The maximum degree of parallelism the optimizer can use is limited by the I/O capacity of the system. The value is calculated by dividing the total system throughput by the maximum I/O bandwidth per process. You must run the <code>DBMS_RESOURCE_MANAGER.CALIBRATE_IO</code> procedure on the system to use the IO setting. This procedure will calculate the total system throughput and the maximum I/O bandwidth per process.

integer

A numeric value for this parameter specifies the maximum degree of parallelism the optimizer can choose for a SQL statement when automatic degree of parallelism is active. Automatic degree of parallelism is only enabled if PARALLEL_DEGREE_POLICY is set to ADAPTIVE, AUTO, or LIMITED.

See Also:

- Oracle Database VLDB and Partitioning Guide for information about automatic degree of parallelism
- Oracle Database PL/SQL Packages and Types Reference for information on the DBMS RESOURCE MANAGER.CALIBRATE 10 procedure

2.287 PARALLEL DEGREE POLICY

PARALLEL_DEGREE_POLICY specifies whether automatic degree of parallelism, statement queuing, and in-memory parallel execution will be enabled.

Property	Description
Parameter type	String
Syntax	PARALLEL_DEGREE_POLICY = { MANUAL LIMITED AUTO ADAPTIVE }
Default value	MANUAL
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No



Values



Automatic degree of parallelism will be enabled regardless of the value of PARALLEL_DEGREE_POLICY if a PARALLEL hint is used at the SQL statement level.

MANUAL

Disables automatic degree of parallelism, statement queuing, and in-memory parallel execution. This reverts the behavior of parallel execution to what it was prior to Oracle Database 11*g* Release 2 (11.2). This is the default.

LIMITED

Enables automatic degree of parallelism for some statements but statement queuing and in-memory Parallel Execution are disabled. Automatic degree of parallelism is only applied to those statements that access tables or indexes decorated explicitly with the DEFAULT degree of parallelism using the PARALLEL clause. Statements that do not access any tables or indexes decorated with the DEFAULT degree of parallelism will retain the MANUAL behavior.

AUTO

Enables automatic degree of parallelism, statement queuing, and in-memory parallel execution.

ADAPTIVE

This value enables automatic degree of parallelism, statement queuing and in-memory parallel execution, similar to the AUTO value. In addition, performance feedback is enabled. Performance feedback helps to improve the degree of parallelism automatically chosen for repeated SQL statements. After the initial execution of a statement, the degree of parallelism chosen by the optimizer is compared to the degree of parallelism computed based on the actual execution performance. If they vary significantly, then the statement is marked for re-parse and the initial execution performance statistics (for example, CPU-time) are provided as feedback for subsequent executions. The optimizer uses the initial execution performance statistics to better determine a degree of parallelism for subsequent executions.

See Also:

- Oracle Database SQL Language Reference for information about PARALLEL hints
- Oracle Database VLDB and Partitioning Guide for information about automatic degree of parallelism

2.288 PARALLEL_EXECUTION_MESSAGE_SIZE

PARALLEL_EXECUTION_MESSAGE_SIZE specifies the size of messages used for parallel execution (formerly referred to as parallel query, PDML, Parallel Recovery, replication).

Property	Description
Parameter type	Integer
Default value	Operating system-dependent
Modifiable	No
Modifiable in a PDB	No
Range of values	Minimum: 2148
	Maximum: 65536, but some operating systems may have a smaller value
Basic	No
Oracle RAC	Multiple instances must have the same value.

On most platforms, the default value is as follows:

- 16384 bytes if COMPATIBLE is set to 11.2.0 or higher
- 2148 bytes if COMPATIBLE is less than 11.2.0

The default value is adequate for most applications. Larger values require a larger shared pool. Larger values result in better performance at the cost of higher memory use. For this reason, replication gets no benefit from increasing the size.



Oracle Database VLDB and Partitioning Guide to learn how this parameter affects memory consumption for parallel operations, including parallel execution

2.289 PARALLEL_FORCE_LOCAL

PARALLEL FORCE LOCAL controls parallel execution in an Oracle RAC environment.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

By default, the parallel server processes selected to execute a SQL statement can operate on any or all Oracle RAC nodes in the cluster. By setting PARALLEL_FORCE_LOCAL to true, the parallel server processes are restricted so that they can only operate on the same Oracle RAC node where the query coordinator resides (the node on which the SQL statement was executed).



See Also:

Oracle Database VLDB and Partitioning Guide for more information about this parameter

2.290 PARALLEL INSTANCE GROUP

Used in conjunction with services or with the <code>INSTANCE_GROUPS</code> parameter, <code>PARALLEL_INSTANCE_GROUP</code> lets you restrict parallel query operations to a limited number of instances.

Property	Description
Parameter type	String
Syntax	PARALLEL_INSTANCE_GROUP = service_name group_name
Default value	There is no default value; parallel execution is enabled across all currently active instances.
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	Any service name or any group name specified in the <code>INSTANCE_GROUPS</code> parameter of any active instance
Basic	No
Oracle RAC	Multiple instances can have different values.

PARALLEL_INSTANCE_GROUP is an Oracle RAC parameter that you can specify in parallel mode only. Note that the INSTANCE GROUPS parameter has been deprecated.

This parameter identifies the parallel instance group Oracle will use for spawning parallel execution processes. If used in conjunction with services, then parallel operations will spawn parallel execution processes only on instances defined in the service. If used in conjunction with INSTANCE_GROUPS, then parallel operations will spawn parallel execution processes only on instances that specify a matching group in their INSTANCE GROUPS parameter.

If the value assigned to <code>PARALLEL_INSTANCE_GROUP</code> is the name of a service or group that does not exist, then the operation runs serially. No parallelism is used.

2.291 PARALLEL_MAX_SERVERS

PARALLEL_MAX_SERVERS specifies the maximum number of parallel execution processes and parallel recovery processes for an instance. As demand increases, Oracle Database increases the number of processes from the number created at instance startup up to this value.

Property	Description
Parameter type	Integer
Default value	PARALLEL_THREADS_PER_CPU * CPU_COUNT * concurrent_parallel_users * 5
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes

Property	Description
Range of values	0 to 32767
Basic	No
Oracle RAC	Multiple instances can have different values.



This parameter applies to parallel execution in exclusive mode as well as in a Real Application Clusters environment.

The number of concurrent parallel users running at default degree of parallelism on an instance depends on the memory initialization parameter settings for the instance. For example, if the MEMORY_TARGET or SGA_TARGET initialization parameter is set, then the number of concurrent_parallel_users = 4. If neither MEMORY_TARGET or SGA_TARGET is set, then PGA_AGGREGATE_TARGET is examined. If a value is set for PGA_AGGREGATE_TARGET, then concurrent_parallel_users = 2. If a value is not set for PGA_AGGREGATE_TARGET, then concurrent parallel users = 1.

The database system always reserves a certain number of reserved processes. Therefore, the following two values are taken into account when setting the PARALLEL_MAX_SERVERS initialization parameter:

- The default value of PARALLEL_MAX_SERVERS determined using the calculation in the table above
- The value of the PROCESSES initialization parameter minus the number of reserved processes

The lower of the two values is used as the default value of PARALLEL_MAX_SERVERS, and if you attempt to explicitly set PARALLEL_MAX_SERVERS to a value that is higher than either of the values, then the setting is adjusted to the lower of the two values.

When the PARALLEL_MAX_SERVERS parameter is set to a value lower than the value in the table above, the lower default value enables the database to start service processes and allows user processes to connect to the database.

If the FAST_START_PARALLEL_ROLLBACK initialization parameter is set to LOW or HIGH, then its value should be taken into account when setting PARALLEL_MAX_SERVERS. If FAST_START_PARALLEL_ROLLBACK = LOW, then PARALLEL_MAX_SERVERS should not be set to a value lower than CPU_COUNT * 2. If FAST_START_PARALLEL_ROLLBACK = HIGH, then PARALLEL MAX_SERVERS should not be set to a value lower than CPU_COUNT * 4.

The default value for PARALLEL_MAX_SERVERS for a PDB is determined using the calculation in the table above with the PDB's CPU COUNT value.





If you set this parameter too low, then some queries may not have a parallel execution process available to them during query processing. If you set it too high, then memory resource shortages may occur during peak periods, which can degrade performance.

See Also:

Oracle Database SQL Tuning Guide for more information about parallel execution

2.292 PARALLEL MIN DEGREE

 ${\tt PARALLEL_MIN_DEGREE} \ controls \ the \ minimum \ degree \ of \ parallelism \ computed \ by \ automatic \ degree \ of \ parallelism.$

Property	Description
Parameter type	String
Syntax	PARALLEL_MIN_DEGREE = [n CPU]
Default value	1
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	A different value can be set on different instances.

The value of PARALLEL_MIN_DEGREE is either a number that corresponds to the lower bound on the degree of parallelism computed by automatic degree of parallelism, or the string value CPU, which is computed by the function CPU_COUNT * PARALLEL_THREADS_PER_CPU.

The default value of PARALLEL_MIN_DEGREE is 1.

Note:

PARALLEL MIN DEGREE has no impact in either of these cases:

- When the value of PARALLEL MIN DEGREE is greater than the value of CPU COUNT
- When the object is Oracle-owned, such as a dictionary table or view created on a dictionary table

See Also:

- "PARALLEL_DEGREE_LIMIT"
- Oracle Database VLDB and Partitioning Guide for information about how the optimizer automatically determines the degree of parallelism for a statement

2.293 PARALLEL_MIN_PERCENT

PARALLEL_MIN_PERCENT lets you specify the minimum percentage of the requested number of parallel execution processes required for parallel execution.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SESSION
Modifiable in a PDB	No
Range of values	0 to 100
Basic	No
Oracle RAC	Multiple instances can have different values.

This parameter controls the behavior for parallel operations when parallel statement queuing is not enabled (when PARALLEL_DEGREE_POLICY is set to manual or limited). It ensures that an operation always gets a minimum percentage of parallel execution servers or errors out. Setting this parameter ensures that parallel operations will not execute unless adequate resources are available. The default value of 0 means that no minimum percentage of processes has been set.

Consider the following settings:

```
PARALLEL_MIN_PERCENT = 50
PARALLEL_MIN_SERVERS = 5
PARALLEL MAX SERVERS = 10
```

If 8 of the 10 parallel execution processes are busy, only 2 processes are available. If you then request a query with a degree of parallelism of 8, the minimum 50% will not be met.

You can use this parameter with PARALLEL_ADAPTIVE_MULTI_USER. In a multi-user environment, an individual user or application can set PARALLEL_MIN_PERCENT to a minimum value until sufficient resources are available on the system and an acceptable degree of parallelism is returned.



See Also:

- Oracle Database SQL Tuning Guide for more information about parallel execution
- "PARALLEL_DEGREE_POLICY", "PARALLEL_MAX_SERVERS",
 "PARALLEL_MIN_SERVERS", and "PARALLEL_ADAPTIVE_MULTI_USER"

2.294 PARALLEL MIN SERVERS

PARALLEL_MIN_SERVERS is the number of parallel execution processes Oracle creates when the instance is started. These processes will be kept active to service parallel statements.

Property	Description
Parameter type	Integer
Default value	CPU_COUNT * PARALLEL_THREADS_PER_CPU * 2
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Default value to the value of PARALLEL_MAX_SERVERS
Basic	No
Oracle RAC	Multiple instances can have different values.

Note:

This parameter applies to parallel execution in exclusive mode as well as in an Oracle Real Application Clusters environment.

Note:

When the PROCESSES initialization parameter is set to a value that is lower than the documented default value for the PARALLEL_MIN_SERVERS parameter in the table above, the database sets the default value of PARALLEL_MIN_SERVERS to a value that is lower than the documented default value. The lower default value enables the database to start service processes and allows user processes to connect to the database.

See Also:

Oracle Database Administrator's Guide for more information about parallel execution servers

2.295 PARALLEL_MIN_TIME_THRESHOLD

PARALLEL_MIN_TIME_THRESHOLD specifies the minimum execution time a statement should have before the statement is considered for automatic degree of parallelism.

Property	Description
Parameter type	String
Syntax	PARALLEL_MIN_TIME_THRESHOLD = { AUTO integer }
Default value	AUTO
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

By default, this parameter is set to 10 seconds. Automatic degree of parallelism is only enabled if Parallel Degree Policy is set to ADAPTIVE, AUTO, or LIMITED.

If all tables referenced by a SQL statement use In-Memory Column Store (IM column store), then PARALLEL MIN TIME THRESHOLD defaults to 1.



Oracle Database VLDB and Partitioning Guide for information about automatic degree of parallelism

2.296 PARALLEL_SERVERS_TARGET

PARALLEL_SERVERS_TARGET specifies the number of parallel server processes allowed to run parallel statements before statement queuing will be used.

Property	Description
Parameter type	Integer
Default value	For a CDB:
	Equal to the PARALLEL_MAX_SERVERS value for the CDB.
	For a PDB or non-CDB:
	PARALLEL_THREADS_PER_CPU * CPU_COUNT * concurrent_parallel_users * 2
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to PARALLEL_MAX_SERVERS
Basic	No

When the parameter PARALLEL_DEGREE_POLICY is set to ADAPTIVE or AUTO, Oracle will queue SQL statements that require parallel execution, if the necessary parallel server processes are not available. In a single instance database, statement queuing will begin once the number of parallel server processes active on the system is equal to or greater than

PARALLEL_SERVERS_TARGET. In an Oracle RAC database, a statement running on one instance can allocate parallel server processes on another instance when necessary. For example, if a statement running on one instance requires parallel execution, but the number of parallel server processes active on that instance is equal to or greater than the value of PARALLEL_SERVERS_TARGET for that instance, then the statement can allocate parallel server processes on a different instance. Therefore, statement queuing in an Oracle RAC database will begin only when every instance has reached its PARALLEL_SERVER_TARGET threshold.

Note:

Consumer groups that have been marked with the PARALLEL_STMT_CRITICAL directive set to BYPASS_QUEUE are allowed to bypass the parallel statement queue, and therefore may drive the total number of active parallel server processes beyond PARALLEL_SERVERS_TARGET. Parallel statements issued with PARALLEL_DEGREE_POLICY not set to ADAPTIVE and AUTO can also drive the total number of active parallel server processes beyond PARALLEL_SERVERS TARGET.

By default, PARALLEL_SERVERS_TARGET is set lower than the maximum number of parallel server processes allowed on the system (PARALLEL_MAX_SERVERS) to ensure each parallel statement will get all of the parallel server resources required and to prevent overloading the system with parallel server processes.

The number of concurrent parallel users running at default degree of parallelism on an instance depends on the memory initialization parameter settings for the instance. For example, if the MEMORY_TARGET or SGA_TARGET initialization parameter is set, then the number of concurrent_parallel_users = 4. If neither MEMORY_TARGET or SGA_TARGET is set, then PGA_AGGREGATE_TARGET is examined. If a value is set for PGA_AGGREGATE_TARGET, then concurrent_parallel_users = 2. If a value is not set for PGA_AGGREGATE_TARGET, then concurrent parallel users = 1.

Note that all serial (non-parallel) statements will execute immediately even if statement queuing has been activated.

By default, all PDB queries are subjected to queuing at the PDB level first by the PDB's PARALLEL_SERVERS_TARGET value, and then at the CDB level by the CDB's PARALLEL_SERVERS_TARGET value. This default behavior prevents any SQL statement inside a PDB from getting downgraded if parallel servers are exhausted in the CDB.

Parallel statement queuing is enabled by default at the CDB level because the CDB has a default value for PARALLEL_SERVERS_TARGET. You can disable parallel statement queuing at the CDB level by using ALTER SYSTEM to set PARALLEL SERVERS TARGET to 0 for the CDB.

The default value for PARALLEL_SERVERS_TARGET for a PDB is determined using the calculation in the table above with the PDB's CPU_COUNT value.



Note:

A PDB can set a lower limit for parallel execution servers than the limit specified in the CDB resource plan. When the PARALLEL_SERVERS_TARGET initialization parameter is set in a PDB, and parallel execution server limit is specified for a PDB in the CDB resource plan, then the lower of the two limits is enforced. For example, assume that the PARALLEL_SERVERS_TARGET initialization parameter is set to 100 in the CDB root. Also assume that hrpdb has its PARALLEL_SERVERS_TARGET initialization parameter set to 50 and the CDB plan has a directive for hrpdb with parallel_server_limit set to 70%. In this case, the limit for parallel execution servers for hrpdb is 50, because 50 is lower than the CDB resource plan limit of 70 for hrpdb.

See Also:

- "PARALLEL DEGREE POLICY"
- "PARALLEL MAX SERVERS"
- "MEMORY_TARGET"
- "SGA_TARGET"
- "PGA AGGREGATE TARGET"
- Oracle Database VLDB and Partitioning Guide for more information about this parameter

2.297 PARALLEL_THREADS_PER_CPU

PARALLEL_THREADS_PER_CPU describes the number of parallel execution processes or threads that a CPU can handle during parallel execution.

Property	Description
Parameter type	Integer
Default value	1
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Any nonzero number
Basic	No

Note

This parameter applies to parallel execution in exclusive mode as well as in an Oracle Real Application Clusters environment.

This parameter is used in determining the default values of other parallel execution related parameters, for example, PARALLEL_MAX_SERVERS. It is also used in determining the default degree of parallelism for SQL statements, and determining the upper bound for the degree of parallelism in automatic degree of parallelism. The default value is adequate in most cases.

See Also:

- Oracle Database SQL Tuning Guide for more information about parallel execution
- Oracle Database VLDB and Partitioning Guide for information about how the optimizer automatically determines the degree of parallelism for a statement

2.298 PDB FILE NAME CONVERT

PDB_FILE_NAME_CONVERT maps names of existing files to new file names when processing a CREATE PLUGGABLE DATABASE statement, as well as when processing the ENABLE PLUGGABLE DATABASE clause of the CREATE DATABASE statement, if the <code>file_name_convert_clause</code> is not specified and Oracle Managed Files is not enabled.

Property	Description
Parameter type	String
Syntax	<pre>PDB_FILE_NAME_CONVERT = 'string1' , 'string2' , 'string3' , 'string4' ,</pre>
	Where:
	 string1 is the pattern of the existing filename
	 string2 is the pattern of the new filename
	 string3 is the pattern of the existing filename
	 string4 is the pattern of the new filename
	You can enclose each string in single or double quotation marks.
	You can specify as many pairs of existing and new replacement strings as required.
	Example:
	<pre>PDB_FILE_NAME_CONVERT = '/dbs/t1/','/dbs/t1/ s','dbs/t2/','dbs/t2/s'</pre>
Default value	There is no default value
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	You must set this parameter for every instance, and multiple instances must have the same value.

File name patterns specified in this initialization parameter cannot match files or directories managed by Oracle Managed Files.



Oracle Multitenant Administrator's Guide for more information about this parameter

2.299 PDB_LOCKDOWN

PDB LOCKDOWN determines the PDB lockdown profile that applies to a PDB.

Property	Description
Parameter type	String
Syntax	PDB_LOCKDOWN = pdb-lockdown-profile-name
Default value	NULL
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	All instances should have the same value

A PDB lockdown profile is a mechanism to restrict operations (such as setting values of certain parameters and using certain options) that can be performed by users connected to a given PDB. You can also restrict execution of any packages that allow network access, for example, UTL SMTP.

You create lockdown profiles using the SQL CREATE LOCKDOWN PROFILE statement. Then you can set a profile using the SQL ALTER SESSION or ALTER SYSTEM statement. See the Examples section.

This parameter can be set using the ALTER SYSTEM statement with scope set to MEMORY, SPFILE, or BOTH.

The lockdown profile for PDBs can be specified by a common user with common ALTER SYSTEM or common SYSDBA privilege.

If a PDB lockdown profile is dropped, any PDB to which the dropped profile was assigned (by means of storing the dropped profile name in the PDB_LOCKDOWN parameter) will continue to have its PDB_LOCKDOWN parameter set to the dropped lockdown profile name. However, the PDB will not have any restrictions imposed by the dropped lockdown profile.

Lockdown profiles can now be created in an application root and are referred to as application lockdown profiles.

A CDB common user with common SYSDBA or common ALTER SYSTEM privilege can only set PDB_LOCKDOWN to a CDB lockdown profile. Similarly, an application common user with application common SYSDBA or application common ALTER SYSTEM privilege can only set PDB_LOCKDOWN to an application lockdown profile.

An application common user cannot overwrite PDB_LOCKDOWN if PDB_LOCKDOWN is already set to a CDB lockdown profile in an application root or application PDB.

If the PDB_LOCKDOWN parameter in a PDB is set to the name of a lockdown profile different from that in its ancestor (for a CDB, the CDB root or, for application PDBs, the application root), the following will govern the interaction between restrictions imposed by these profiles:



- If the PDB_LOCKDOWN parameter in a PDB (including an application PDB) is set to a CDB lockdown profile, lockdown profiles specified by the PDB_LOCKDOWN parameter in CDB root (and for application PDBs, the application root) are ignored.
- If the PDB_LOCKDOWN parameter in an application PDB is set to an application lockdown
 profile while the PDB_LOCKDOWN parameter in the application root or CDB root is set to a
 CDB lockdown profile, in addition to the rules stipulated in the application lockdown profile,
 the DISABLE rules from the CDB lockdown profile set in its nearest ancestor (that is, an
 application root or CDB root) are inherited.
- If there are conflicts between rules comprising the CDB lockdown profile and the application lockdown profile, the rules in the CDB lockdown profile will take precedence (for example, the OPTION_VALUE clause of a CDB lockdown profile will take precedence over the OPTION_VALUE clause of an application lockdown profile).

Examples

This example shows how the SYS user can connect to the database AS SYSDBA and use the CREATE LOCKDOWN PROFILE statement in the root of a CDB to define a new lockdown profile. After defining the new lockdown profile, the SYS user can assign the new lockdown profile to a PDB using the PDB LOCKDOWN parameter:

See Also:

- "DBA LOCKDOWN PROFILES"
- Oracle Multitenant Administrator's Guide for an introduction to PDB lockdown profiles
- Oracle Database SQL Language Reference for more information about the CREATE LOCKDOWN PROFILE statement



2.300 PDB_OS_CREDENTIAL

 ${\tt PDB_OS_CREDENTIAL} \ determines \ the \ identity \ of \ the \ operating \ system \ user \ (OS \ user) \ employed \ when \ interacting \ with \ the \ operating \ system \ from \ a \ PDB.$

Property	Description
Parameter type	String
Syntax	PDB_OS_CREDENTIAL = credential
Default value	None
Modifiable	No
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value should be used for all instances



The Oracle OS user will continue to be used when interacting with the operating system from the root.

The Oracle OS user is usually a highly privileged user, and using the same user for operating system interactions for every PDB is not recommended. Also, using the same OS user for operating system interactions from different PDBs may compromise data belonging to a given PDB.

In contrast, using an OS user described by a credential whose name is specified as a value of the PDB_OS_CREDENTIAL parameter helps ensure that operating system interactions are performed as a less powerful user and provides the ability to protect data belonging to one PDB from being accessed by users connected to another PDB. A credential is an object that is created using the CREATE CREDENTIAL procedure for the DBMS CREDENTIAL package.

The operating system interactions that are done as the OS user name specified in the credential include:

- External jobs that do not already have an operating system credential specified
- External table pre-processors
- PL/SQL library executions

This parameter can be specified for all the PDBs in a CDB but the CDB-wide value can be overridden for a specific PDB and can be modified *only* by a common administrative user with the EXECUTE privilege for the DBMS_CREDENTIAL PL/SQL package and the ALTER SYSTEM system privilege.

If a value is not set for this parameter for a given PDB, the Oracle OS User will continue to be used when interacting with the operating system from that PDB.



- Oracle Database Security Guide for an example of setting an OS user for a PDB using this parameter
- Oracle Multitenant Administrator's Guide for conceptual information about CDBs and PDBs
- Oracle Multitenant Administrator's Guide for information about managing CDBs and PDBs
- Oracle Database PL/SQL Packages and Types Reference for information about creating a credential using the DBMS CREDENTIAL.CREATE CREDENTIAL procedure

2.301 PERMIT_92_WRAP_FORMAT

PERMIT_92_WRAP_FORMAT allows Oracle Database release 9.2 wrapped versions of PL/SQL source text to be used in Oracle Database releases 10.2 and later when this parameter is set to true.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	Multiple instances should use the same value

Oracle recommends that wrapped files be created using the PL/SQL Wrapper utility from Oracle Database release 10 or later.



Oracle Database PL/SQL Language Reference for more information about PL/SQL source text wrapping

2.302 PGA_AGGREGATE_LIMIT

PGA_AGGREGATE_LIMIT specifies a limit on the aggregate PGA memory consumed by the instance.

Property	Description
Parameter type	Big integer



Property	Description
Syntax	PGA_AGGREGATE_LIMIT = integer [K M G]
Default value	If MEMORY_TARGET is set, then PGA_AGGREGATE_LIMIT defaults to the MEMORY_MAX_TARGET value.
	If MEMORY_TARGET is not set, then PGA_AGGREGATE_LIMIT defaults to 200% of PGA_AGGREGATE_TARGET.
	If MEMORY_TARGET is not set, and PGA_AGGREGATE_TARGET is explicitly set to 0, then the value of PGA_AGGREGATE_LIMIT is set to 90% of the physical memory size minus the total SGA size.
	In all cases, the default PGA_AGGREGATE_LIMIT is at least 2GB and at least 3MB times the PROCESSES parameter (and at least 5MB times the PROCESSES parameter for an Oracle RAC instance).
	For a PDB, the default value is the same as the CDB's default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	Do not attempt to set PGA_AGGREGATE_LIMIT below its default value, even in a parameter file (pfile), or instance startup will fail. However, PGA_AGGREGATE_LIMIT can be set to 0 either in a parameter file or dynamically after startup. If a value of 0 is specified, it means there is no limit to the aggregate PGA memory consumed by the instance.
Basic	No

Actions Taken When PGA_AGGREGATE_LIMIT is Exceeded

Parallel queries will be treated as a unit. First, the sessions that are using the most untunable memory will have their calls terminated. Then, if the total PGA memory usage is still over the limit, the sessions that are using the most untunable memory will be terminated.

SYS processes and background processes other than job queue processes will not be subjected to any of the actions described in this section. Instead, if they are using the most untunable memory, they will periodically write a brief summary of their PGA usage to a trace file.



Note:

This parameter is optional for pluggable databases (PDBs). When this parameter is set for a PDB, it specifies the maximum PGA size for the PDB.

To be able to use Resource Manager in a CDB to control the amount of memory each PDB can use:

- The NONCDB_COMPATIBLE initialization parameter must be set to FALSE at the CDB level (in the root of the CDB).
- The MEMORY TARGET initialization parameter must not be set at the CDB level.
- You must set the PGA_AGGREGATE_LIMIT initialization parameter in a PDB to a
 value that meets these requirements:
 - Less than or equal to the PGA AGGREGATE LIMIT value set at the CDB level
 - Greater than or equal to twice the value of PGA_AGGREGATE_TARGET set in the PDB

When you set PGA_AGGREGATE_LIMIT in a PDB to a value that does not meet these requirements, you receive an error. If these requirements are violated after the PDB's parameter is set (for example, if the PGA_AGGREGATE_LIMIT value is changed at the CDB level), Oracle will adjust the PDB's value to meet these requirements.

See Also:

- "MEMORY_TARGET"
- "PGA_AGGREGATE_TARGET"
- Oracle Multitenant Administrator's Guide for more information about the initialization parameters that control the memory usage of PDBs
- Oracle Database Performance Tuning Guide for more information about this parameter

2.303 PGA_AGGREGATE_TARGET

PGA_AGGREGATE_TARGET specifies the target aggregate PGA memory available to all server processes attached to the instance.

Property	Description
Parameter type	Big integer
Syntax	PGA_AGGREGATE_TARGET = integer [K M G]
Default value	10 MB or 20% of the size of the SGA, whichever is greater
	For a PDB, the default value is the same as the CDB's default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes



Property	Description
Range of values	Minimum: 10 MB
	Maximum: 4096 GB - 1
Basic	Yes

To set a hard limit for aggregate PGA memory, use the PGA AGGREGATE LIMIT parameter.

Setting PGA_AGGREGATE_TARGET to a nonzero value has the effect of automatically setting the WORKAREA_SIZE_POLICY parameter to AUTO. With this setting, SQL working areas used by memory-intensive SQL operators (such as sort, group-by, hash-join, bitmap merge, and bitmap create) will be automatically sized. A nonzero value for this parameter is the default since, unless you specify otherwise, Oracle sets it to 20% of the SGA or 10 MB, whichever is greater.

Setting PGA_AGGREGATE_TARGET to 0 automatically sets the WORKAREA_SIZE_POLICY parameter to MANUAL. With this setting, SQL working areas are sized using the * AREA SIZE parameters.

Oracle attempts to keep the amount of private memory below the target specified by this parameter by adapting the size of the working areas to private memory. When increasing the value of this parameter, you indirectly increase the memory allotted to working areas. Consequently, more memory-intensive operations are able to run fully in memory and fewer will work their way over to disk.

If Automatic Memory Management is enabled (MEMORY_TARGET is set to a positive value) and PGA_AGGREGATE_TARGET is also set to a positive value, the PGA_AGGREGATE_TARGET value acts as the minimum value for the size of the instance PGA.

Note:

This parameter is optional for pluggable databases (PDBs). When this parameter is set for a PDB, it specifies the target aggregate PGA size for the PDB.

To be able to use Resource Manager in a CDB to control the amount of memory each PDB can use:

- The NONCDB_COMPATIBLE initialization parameter must be set to FALSE at the CDB level (in the root of the CDB).
- The MEMORY TARGET initialization parameter must not be set at the CDB level.
- You must set the PGA_AGGREGATE_TARGET initialization parameter in a PDB to a value that meets these requirements:
 - Less than or equal to the PGA AGGREGATE TARGET value set at the CDB level
 - Less than or equal to 50% of the PGA_AGGREGATE_LIMIT initialization parameter value set at the CDB level
 - Less than or equal to 50% of the PGA AGGREGATE LIMIT value set in the PDB

When you set PGA_AGGREGATE_TARGET in a PDB to a value that does not meet these requirements, you receive an error. If these requirements are violated after the PDB's parameter is set (for example, if the PGA_AGGREGATE_TARGET value is changed at the CDB level), Oracle will adjust the PDB's value to meet these requirements.



- "WORKAREA_SIZE_POLICY"
- Oracle Database Performance Tuning Guide for more information about configuring memory
- "PGA AGGREGATE LIMIT"

2.304 PKCS11 LIBRARY LOCATION

PKCS11_LIBRARY_LOCATION specifies the PKCS#11 library location for Transparent Data Encryption (TDE).

Property	Description
Parameter type	String
Syntax	PKCS11_LIBRARY_LOCATION = 'library_location'
Default value	None
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value must be used on all instances.

For <code>library_location</code>, specify the full operating system path to the PKCS#11 library. For example:

```
ALTER SYSTEM SET PKCS11_LIBRARY_LOCATION = '/opt/oracle/extapi/64/pkcs11/okv/21.8/liborapkcs.so';
```

```
See Also:
"V$PKCS11_PATH"
```

2.305 PLSCOPE_SETTINGS

PLSCOPE_SETTINGS controls the compile time collection, cross-reference, and storage of PL/SQL source code identifier data.

Property	Description
Parameter type	String



Property	Description
Syntax	PLSCOPE_SETTINGS = 'value_clause [, value_clause]'
	value_clause::=
	{ IDENTIFIERS STATEMENTS } : { ALL NONE PUBLIC (for IDENTIFIERS only) PLSQL (for IDENTIFIERS only) PLSQL (for IDENTIFIERS only) }
Default value	'IDENTIFIERS:NONE'
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Values

• IDENTIFIERS:ALL

Enables the collection of all source code identifier data.

IDENTIFIERS:NONE

Disables the collection of all source code identifier data.

IDENTIFIERS: PUBLIC

Enables the collection of all PUBLIC user identifier data (except for DEFINITION)

IDENTIFIERS:SOL

Enables the collection of all SQL identifier data.

IDENTIFIERS:PLSOL

Enables the collection of all PLSQL identifier data.

STATEMENTS:ALL

Enables the collection of all SQL statements used in PL/SQL.

STATEMENTS:NONE

Disables the collection of all statements.

PLSCOPE_SETTINGS can be set on a session, system, or per-library unit (ALTER COMPILE) basis. The current setting of PLSCOPE_SETTINGS for any library unit can be attained by querying the *_PLSQL_OBJECT_SETTINGS views. Any identifier data collected by setting this parameter can be accessed using the * IDENTIFIERS views.

When a STATEMENTS setting is not specified, and IDENTIFIERS is specified but set to a value other than NONE, STATEMENTS defaults to a setting of ALL, which is equal to:

IDENTIFIERS: [ALL|PLSQL|PLSQL|PUBLIC]

Examples

The following parameter setting causes PL/SQL and SQL identifiers and statements to be collected:

PLSCOPE SETTINGS = 'IDENTIFIERS:ALL, STATEMENTS:ALL'



The following parameter setting causes only SQL identifiers and statements to be collected:

```
PLSCOPE SETTINGS = 'IDENTIFIERS:SQL, STATEMENTS:ALL'
```

The following parameter setting causes only PL/SQL identifiers to be collected:

```
PLSCOPE SETTINGS = 'IDENTIFIERS:PLSQL, STATEMENTS:NONE'
```

The following parameter setting causes no identifier data to be collected, and drops any existing identifier data:

```
PLSCOPE_SETTINGS = 'IDENTIFIERS: NONE'
```

See Also:

- "ALL_PLSQL_OBJECT_SETTINGS," "DBA_PLSQL_OBJECT_SETTINGS," and "USER_PLSQL_OBJECT_SETTINGS"
- "ALL_IDENTIFIERS,""DBA_IDENTIFIERS," and "USER_IDENTIFIERS"
- Oracle Database PL/SQL Language Reference for more information about this parameter

2.306 PLSQL_CCFLAGS

 ${\tt PLSQL_CCFLAGS}\ provides\ a\ mechanism\ that\ allows\ PL/SQL\ programmers\ to\ control\ conditional\ compilation\ of\ each\ PL/SQL\ library\ unit\ independently.$

Property	Description
Parameter type	String
Syntax	PLSQL_CCFLAGS = ' <v1>:<c1>,<v2>:<c2>,,<vn>:<cn>'</cn></vn></c2></v2></c1></v1>
Default value	Empty string
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	Any string literal that satisfies the internal syntax
Basic	No
Examples	ALTER SESSION SET PLSQL_CCFLAGS = 'DeBug:TruE';
	ALTER SESSION SET PLSQL_CCFLAGS = 'debug:TRUE';

Values

<vi>> has the form of an unquoted PL/SQL identifier. It is unrestricted and can be a
reserved word or a keyword. The text is insensitive to case. Each one is known as a flag or
flag name. Each <vi>> can occur more than once in the string, each occurrence can have a
different flag value, and the flag values can be of different kinds.



 <ci> is one of the following: a PL/SQL boolean literal, a PLS_INTEGER literal, or the literal NULL. The text is insensitive to case. Each one is known as a flag value and corresponds to a flag name.

You can define any allowable value for PLSQL_CCFLAGS. However, Oracle recommends that this parameter be used for controlling the conditional compilation of debugging or tracing code. It is recommended that the following identifiers not be used as flag name values:

- Names of Oracle parameters (for example, NLS LENGTH SEMANTICS)
- Identifiers with any of the following prefixes: PLS_, PLSQL_, PLSCC_, ORA_, ORACLE_, DBMS_,
 SYS

See Also:

Oracle Database PL/SQL Language Reference for more information about this parameter

2.307 PLSQL CODE TYPE

PLSQL CODE TYPE specifies the compilation mode for PL/SQL library units.

Property	Description
Parameter type	String
Syntax	PLSQL_CODE_TYPE = { INTERPRETED NATIVE }
Default value	INTERPRETED
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Values

INTERPRETED

PL/SQL library units will be compiled to PL/SQL bytecode format. Such modules are executed by the PL/SQL interpreter engine.

NATIVE

PL/SQL library units (with the possible exception of top-level anonymous PL/SQL blocks) will be compiled to native (machine) code. Such modules will be executed natively without incurring any interpreter overhead.

When the value of this parameter is changed, it has no effect on PL/SQL library units that have already been compiled. The value of this parameter is stored persistently with each library unit.

If a PL/SQL library unit is compiled native, all subsequent automatic recompilations of that library unit will use native compilation.



Oracle Database PL/SQL Language Reference for more information about this parameter

2.308 PLSQL DEBUG

PLSQL DEBUG specifies whether or not PL/SQL library units will be compiled for debugging.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No



The $PLSQL_DEBUG$ parameter is deprecated. It is retained for backward compatibility only.

Values

true

PL/SQL library units will be compiled for debugging

false

PL/SQL library units will be compiled for normal execution

When PLSQL_DEBUG is set to true, PL/SQL library units are always compiled INTERPRETED in order to be debuggable.

When the value of this parameter is changed, it has no effect on PL/SQL library units that have already been compiled. The value of this parameter is stored persistently with each library unit.

2.309 PLSQL_FUNCTION_DYNAMIC_STATS

PLSQL_FUNCTION_DYNAMIC_STATS specifies, at the session level, whether dynamic statistics are allowed to be generated for PL/SQL functions.

Property	Description
Parameter type	String
Syntax	<pre>PLSQL_FUNCTION_DYNAMIC_STATS = { PREFERENCE ON OFF CHOOSE }</pre>



Property	Description
Default value	PREFERENCE
Modifiable	ALTER SESSION
Modifiable in a PDB	No
Basic	No

Dynamic statistics is an optimization technique in which the database uses recursive SQL to scan a small random sample of the blocks in a table. In earlier releases, the optimizer made the decision to allow dynamic statistics for some PL/SQL functions, and disallow dynamic statistics for others.

Starting with Oracle Database 23ai, Release Update 23.8, Oracle provides users with more control over dynamic statistics for PL/SQL functions. The DBMS_STATS package offers procedures that allow you to set and view global-level and function-level preferences for generating dynamic statistics for PL/SQL functions.

Additionally, the PLSQL_FUNCTION_DYNAMIC_STATS parameter allows you to override these preferences for a particular session. You can specify the following values:

- PREFERENCE Do not override. Use the global-level and function-level preferences that
 were set with the DBMS_STATS package for generating dynamic statistics for PL/SQL
 functions. This is the default.
- ON Enable dynamic statistics in the session for all PL/SQL functions.
- OFF Disable dynamic statistics in the session for all PL/SQL functions.
- CHOOSE Allow the optimizer to choose whether dynamic statistics are generated in the session for each PL/SQL function.

Note:

This parameter is available starting with Oracle Database 23ai, Release Update 23.8.

See Also:

- Oracle Database PL/SQL Packages and Types Reference for more information about the DBMS STATS package
- Oracle Database SQL Tuning Guide for more information about this parameter



2.310 PLSQL_IMPLICIT_CONVERSION_BOOL

PLSQL_IMPLICIT_CONVERSION_BOOL allows or disallows implicit conversions in PL/SQL between numeric data types and the BOOLEAN data type, and between character data types and the BOOLEAN data type.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

When this parameter is set to true, PL/SQL supports the following implicit data type conversions:

- Numeric data type to BOOLEAN data type
 - Non-zero numeric values are implicitly converted to the BOOLEAN value TRUE
 - The numeric value of 0 is implicitly converted to the BOOLEAN value FALSE
- Character data type to BOOLEAN data type
 - Character values such as 'true', 't', 'yes', 'y', '1', and 'on' (case-insensitive) are implicitly converted to the BOOLEAN value TRUE
 - Character values such as 'false', 'f', 'no', 'n', '0', and 'off' (case-insensitive)
 are implicitly converted to the BOOLEAN value FALSE
- BOOLEAN data type to numeric data type
 - The BOOLEAN value TRUE is implicitly converted to the numeric value 1
 - The BOOLEAN value FALSE is implicitly converted to the numeric value 0
- BOOLEAN data type to character data type
 - The BOOLEAN value TRUE is implicitly converted to the character value 'TRUE'
 - The BOOLEAN value FALSE is implicitly converted to the character value 'FALSE'

When this parameter is set to false, PL/SQL does not support such implicit conversions.



This parameter is available starting with Oracle Database 23ai.



2.311 PLSQL_OPTIMIZE_LEVEL

PLSQL_OPTIMIZE_LEVEL specifies the optimization level that will be used to compile PL/SQL library units. The higher the setting of this parameter, the more effort the compiler makes to optimize PL/SQL library units.

Property	Description
Parameter type	Integer
Default value	2
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 3
Basic	No

Values

• 0

Maintains the evaluation order and hence the pattern of side effects, exceptions, and package initializations of Oracle9*i* and earlier releases. Also removes the new semantic identity of BINARY_INTEGER and PLS_INTEGER and restores the earlier rules for the evaluation of integer expressions. Although code will run somewhat faster than it did in Oracle9*i*, use of level 0 will forfeit most of the performance gains of PL/SQL in Oracle Database 10*g*.

• 1

Applies a wide range of optimizations to PL/SQL programs including the elimination of unnecessary computations and exceptions, but generally does not move source code out of its original source order.

• 2

Applies a wide range of modern optimization techniques beyond those of level 1 including changes which may move source code relatively far from its original location.

• 3

Applies a wide range of optimization techniques beyond those of level 2, automatically including techniques not specifically requested.

Generally, setting this parameter to 2 pays off in better execution performance. If, however, the compiler runs slowly on a particular source module or if optimization does not make sense for some reason (for example, during rapid turnaround development), then setting this parameter to 1 will result in almost as good a compilation with less use of compile-time resources.

The value of this parameter is stored persistently with the library unit.

- Oracle Database PL/SQL Language Reference for more information about this parameter
- Oracle Database Development Guide for an example of using this parameter

2.312 PLSQL_V2_COMPATIBILITY

PLSQL_V2_COMPATIBILITY is used to specify whether nonstandard behavior that PL/SQL Version 2 allows will be allowed in PL/SQL Version 8.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No



The PLSQL_V2_COMPATIBILITY parameter is deprecated. It is retained for backward compatibility only.

PL/SQL Version 2 allows some nonstandard behavior that Version 8 disallows. If you want to retain that behavior for backward compatibility, set PLSQL_V2_COMPATIBILITY to true. If you set it to false, then PL/SQL Version 8 behavior is enforced and Version 2 behavior is not allowed.

See Also:

Oracle Database PL/SQL Language Reference for a description of the differences between PL/SQL Version 2 and Version 8, and for more information on setting this parameter

2.313 PLSQL_WARNINGS

PLSQL_WARNINGS enables or disables the reporting of warning messages by the PL/SQL compiler, and specifies which warning messages to show as errors.

Property	Description
Parameter type	String



Property	Description
Syntax	PLSQL_WARNINGS = 'value_clause' [, 'value_clause']
Syntax	value_clause::=
	{ ENABLE DISABLE ERROR }:
	{ ALL
	SEVERE
	INFORMATIONAL
	PERFORMANCE
	{ integer
	(integer [, integer])
	}
	}
Default value	'DISABLE:ALL'
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Examples	<pre>PLSQL_WARNINGS = 'ENABLE:SEVERE', 'DISABLE:INFORMATIONAL'; PLSQL_WARNINGS = 'DISABLE:ALL'; PLSQL_WARNINGS = 'DISABLE:5000', 'ENABLE:5001', 'ERROR:5002'; PLSQL_WARNINGS = 'ENABLE:(5000,5001,5002)', 'DISABLE: (6000,6001)';</pre>

value_clause

Multiple value clauses may be specified, enclosed in quotes and separated by commas. Each value clause is composed of a qualifier, a colon (:), and a modifier.

Qualifier values:

• ENABLE

Enable a specific warning or a set of warnings

DISABLE

Disable a specific warning or a set of warnings

• ERROR

Treat a specific warning or a set of warnings as errors

Modifier values:

• ALL

Apply the qualifier to all warning messages

SEVERE

Apply the qualifier to only those warning messages in the SEVERE category

INFORMATIONAL

Apply the qualifier to only those warning messages in the INFORMATIONAL category

PERFORMANCE

Apply the qualifier to only those warning messages in the PERFORMANCE category



Oracle Database PL/SQL Language Reference for more information about this parameter

2.314 PMEM FILESTORE

PMEM_FILESTORE specifies the Persistent Memory Filestores (PMEM Filestores) that the database automatically mounts during instance startup.

Property	Description
Parameter type	String
Syntax	<pre>PMEM_FILESTORE = ('mount_point', 'backing_file' [, 'mount_point', 'backing_file'])</pre>
Default value	None
Modifiable	No
Modifiable in a PDB	No
Basic	No

You can specify one or more PMEM Filestores to be mounted during instance startup.

Specify the following pair of values for each PMEM Filestore:

- mount_point: The operating system directory path for the mount point of the PMEM Filestore
- backing file: The operating system file path for the backing file of the PMEM Filestore

For example:

```
PMEM_FILESTORE=('/var/db/db_1', '/var/db_storage/db1.f')
PMEM_FILESTORE=('/var/db/db_1', '/var/db_storage/db1.f', '/var/db/db_2', '/var/db storage/db2.f')
```

2.315 PRE_PAGE_SGA

PRE PAGE SGA determines whether Oracle accesses the entire SGA at instance startup.

Operating system page table entries are then prebuilt for each page of the SGA.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	No
Modifiable in a PDB	No



Property	Description
Range of values	true false
Basic	No

When this parameter is set to true, the database server touches the entire SGA after the instance startup. These accesses are done by a background process and are initiated after the instance startup. The accesses cause the OS to allocate memory to the SGA and populate the page tables, making the SGA memory resident and increasing database performance.



This setting does not prevent your operating system from paging or swapping the SGA after it is initially read into memory.

Note:

The PRE_PAGE_SGA parameter is deprecated in Oracle Database 23ai because it is obsolete. Setting this parameter typically provides little or no potential performance benefits, and can create problems.

The Oracle Database design for SGA packaging has evolved over time, so that process start-up effects on the SGA are initiated after instance startup, and there is little to no benefit in changing the value for <code>PRE_PAGE_SGA</code>. The only use case for this parameter is on Oracle Exadata systems, which should have <code>PRE_PAGE_SGA</code> set to <code>TRUE</code>.

2.316 PRIORITY_TXNS_HIGH_WAIT_TARGET

PRIORITY_TXNS_HIGH_WAIT_TARGET specifies the maximum number of seconds that a HIGH priority transaction will wait for a row lock before the Priority Transactions feature rolls back a lower priority transaction holding the lock.

Property	Description
Parameter type	Integer
Default value	2147483647
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	1 to 2147483647
Basic	No
Oracle RAC	Different instances can use different values.

The Priority Transactions feature automatically rolls back low-priority transactions that are blocking higher priority transactions from obtaining row locks.

This parameter is one of four initialization parameters that enable you to configure Priority Transactions. The other parameters are:

- TXN_PRIORITY Specifies a priority (HIGH, MEDIUM, or LOW) for all transactions in a user session
- PRIORITY_TXNS_MEDIUM_WAIT_TARGET Specifies the maximum number of seconds that a
 MEDIUM priority transaction will wait for a row lock
- PRIORITY_TXNS_MODE Specifies whether to enable Priority Transactions or test Priority Transactions by running in tracking mode

Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

- "TXN_PRIORITY"
- "PRIORITY_TXNS_MEDIUM_WAIT_TARGET"
- "PRIORITY_TXNS_MODE"
- Oracle Database Administrator's Guide for more information about Priority Transactions

2.317 PRIORITY_TXNS_MEDIUM_WAIT_TARGET

PRIORITY_TXNS_MEDIUM_WAIT_TARGET specifies the maximum number of seconds that a MEDIUM priority transaction will wait for a row lock before the Priority Transactions feature rolls back a lower priority transaction holding the lock.

Property	Description
Parameter type	Integer
Default value	2147483647
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	1 to 2147483647
Basic	No
Oracle RAC	Different instances can use different values.

The Priority Transactions feature automatically rolls back low-priority transactions that are blocking higher priority transactions from obtaining row locks.

This parameter is one of four initialization parameters that enable you to configure Priority Transactions. The other parameters are:

 TXN_PRIORITY - Specifies a priority (HIGH, MEDIUM, or LOW) for all transactions in a user session

- PRIORITY_TXNS_HIGH_WAIT_TARGET Specifies the maximum number of seconds that a
 HIGH priority transaction will wait for a row lock
- PRIORITY_TXNS_MODE Specifies whether to enable Priority Transactions or test Priority Transactions by running in tracking mode



This parameter is available starting with Oracle Database 23ai.

See Also:

- "TXN_PRIORITY"
- "PRIORITY_TXNS_HIGH_WAIT_TARGET"
- "PRIORITY_TXNS_MODE"
- Oracle Database Administrator's Guide for more information about Priority Transactions

2.318 PRIORITY_TXNS_MODE

PRIORITY TXNS MODE specifies the mode for Priority Transactions.

Property	Description
Parameter type	String
Syntax	PRIORITY_TXNS_MODE = { ROLLBACK TRACK }
Default value	ROLLBACK
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Multiple instances must have the same value.

Values:

 ROLLBACK - This setting enables Priority Transactions. The database will automatically roll back low-priority transactions that are blocking higher priority transactions from obtaining row locks.

Before enabling Priority Transactions, ensure that the following initialization parameters are configured properly:

- PRIORITY_TXNS_HIGH_WAIT_TARGET Specifies the maximum number of seconds that a
 HIGH priority transaction will wait for a row lock
- PRIORITY_TXNS_MEDIUM_WAIT_TARGET Specifies the maximum number of seconds that a MEDIUM priority transaction will wait for a row lock

 TXN_PRIORITY - Specifies a priority (HIGH, MEDIUM, or LOW) for all transactions in a user session

When running in ROLLBACK mode, you can track the performance of Priority Transactions by monitoring the following statistics:

- txns rollback priority_txns_high_wait_target
- txns rollback priority_txns_medium_wait_target

TRACK - This setting allows you to track the potential behavior of Priority Transactions. You
can use this mode to tune the Priority Transactions initialization parameters before
enabling ROLLBACK mode.

The database determines when HIGH and MEDIUM priority transactions would be rolled back, given the current values of the Priority Transactions initialization parameters. However, instead of actually rolling back the transactions, the database only increments the following statistics:

- txns track mode priority_txns_high_wait_target
- txns track mode priority txns medium wait target

You can also monitor the following wait events to determine the amount of time transactions are waiting for row locks:

- eng: TX row lock (HIGH priority)
- enq: TX row lock (MEDIUM priority)
- enq: TX row lock (LOW priority)

Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

- "PRIORITY_TXNS_HIGH_WAIT_TARGET"
- "PRIORITY TXNS MEDIUM WAIT TARGET"
- "TXN PRIORITY"
- Oracle Database Administrator's Guide for more information about Priority Transactions

2.319 PRIVATE_TEMP_TABLE_PREFIX

PRIVATE_TEMP_TABLE_PREFIX specifies the prefix that the database uses for private temporary tables.

Property	Description
Parameter type	String



Property	Description
Syntax	PRIVATE_TEMP_TABLE_PREFIX = string
Default value	ORA\$PTT_
Modifiable	ALTER SYSTEM DEFERRED
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value must be specified on each instance.

The default value used for the PRIVATE_TEMP_TABLE_PREFIX parameter is ORA\$PTT_.

If you choose to specify a different prefix value, it must begin with the string <code>ORA\$</code> and must be unique across the database. Use the following SQL query to determine if the prefix value is unique (in the query, replace <code>prefix-value</code> with your actual prefix value):

```
select count(*) from obj$ where name like '<prefix-value>%';
```

If the above query returns 0, then the prefix you specified is an appropriate value to set.



After setting the prefix, regular table, view, and object names cannot use the same prefix.

See Also:

- "DBA_PRIVATE_TEMP_TABLES"
- "USER_PRIVATE_TEMP_TABLES"
- Oracle Database Administrator's Guide for an introduction to private temporary tables

2.320 PROCESSES

PROCESSES specifies the maximum number of operating system user processes that can simultaneously connect to Oracle.

Property	Description
Parameter type	Integer
Default value	The value is derived, and it typically depends on the number of cores reported in the alert log.
Modifiable	No
Modifiable in a PDB	No
Range of values	80 to operating system dependent



Property	Description
Basic	Yes
Oracle RAC	Multiple instances can have different values.

The value for this parameter should allow for all background processes such as locks, job queue processes, and parallel execution processes.

The default values of the SESSIONS and TRANSACTIONS parameters are derived from this parameter. Therefore, if you change the value of PROCESSES, you should evaluate whether to adjust the values of those derived parameters.

See Also:

- Your operating system-specific Oracle documentation for the range of values
- Oracle Database Concepts for an introduction to Oracle database instance processes

2.321 PROCESSOR_GROUP_NAME

PROCESSOR_GROUP_NAME specifies the name of the processor group that this instance should run in.

Property	Description
Parameter type	String
Syntax	PROCESSOR_GROUP_NAME = string
Default value	There is no default value
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	For Oracle RAC databases, it is recommended that the operating system processor groups for each database instance are all named the same, so that the same PROCESSOR_GROUP_NAME value can be used. However, you can choose to use different names. In this case, you would set the PROCESSOR_GROUP_NAME explicitly for each database instance.

This initialization parameter is supported on the Linux and Solaris operating systems.

PROCESSOR_GROUP_NAME instructs the database instance to run itself within the specified operating system processor group. All Oracle processes will be bound to the CPUs in this group and will only run on these CPUs.

Notes:

- Oracle recommends that this parameter be set only for databases on which the USE DEDICATED BROKER initialization parameter is also set to TRUE.
- For NUMA systems, all SGA and PGA memory allocated by the database instance will be allocated from NUMA nodes within the group.
- Starting with Oracle Database Appliance (ODA) release 19.9, you can use
 PROCESSOR_GROUP_NAME to specify the set of CPU cores on which the database
 on ODA will run. For more information, refer to Oracle Database Appliance X8-2
 Deployment and User's Guide.



"USE_DEDICATED_BROKER"

2.322 QUERY_REWRITE_ENABLED

Use QUERY REWRITE ENABLED to enable or disable query rewriting globally for the database.

Property	Description
Parameter type	String
Syntax	QUERY_REWRITE_ENABLED = { false true force }
Default value	If OPTIMIZER_FEATURES_ENABLE is set to 10.0.0 or higher, then true
	If OPTIMIZER_FEATURES_ENABLE is set to 9.2.0 or lower, then false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Multiple instances can have different values.

Values

false

Oracle does not use rewrite.

true

Oracle costs the query with rewrite and without rewrite and chooses the method with the lower cost.

• force

Oracle always uses rewrite and does not evaluate the cost before doing so. Use force when you know that the query will always benefit from rewrite and when reduction in compile time is important.

To take advantage of query rewrite for a particular materialized view, you must enable query rewrite for that materialized view, and you must enable cost-based optimization.

- Oracle Database Data Warehousing Guide for information on query rewrite of materialized views
- Oracle Database SQL Tuning Guide and "OPTIMIZER_MODE" for information on cost-based optimization

2.323 QUERY_REWRITE_INTEGRITY

QUERY_REWRITE_INTEGRITY determines the degree to which Oracle must enforce query rewriting. At the safest level, Oracle does not use query rewrite transformations that rely on unenforced relationships.

Property	Description
Parameter type	String
Syntax	<pre>QUERY_REWRITE_INTEGRITY = { enforced trusted stale_tolerated }</pre>
Default value	enforced
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Multiple instances can have different values.

QUERY_REWRITE_INTEGRITY is relevant for materialized views as well as for foreign key constraints in NOVALIDATE state.

Values

enforced

Oracle enforces and guarantees consistency and integrity.

trusted

Oracle allows rewrites using relationships that have been declared, but that are not enforced by Oracle.

stale tolerated

Oracle allows rewrites using unenforced relationships. Materialized views are eligible for rewrite even if they are known to be inconsistent with the underlying detail data. You must set the <code>QUERY_REWRITE_INTEGRITY</code> initialization parameter to <code>stale_tolerated</code> before querying an external table in the In-Memory Column Store (IM column store).

If a foreign key constraint is in <code>NOVALIDATE</code> state, join elimination is not done when <code>QUERY_REWRITE_INTEGRITY=enforced</code>. This means that queries with joins over a foreign key constraint that is in <code>RELY NOVALIDATE</code> state can potentially take longer to parse and execute as the optimizer does not trust the <code>RELY</code>.



- "INMEMORY_CLAUSE_DEFAULT"
- Oracle Database Data Warehousing Guide for more information about query rewrite for materialized views
- Oracle Database Data Warehousing Guide for more information about the QUERY REWRITE INTEGRITY parameter

2.324 RDBMS_SERVER_DN

RDBMS_SERVER_DN specifies the Distinguished Name (DN) of the Oracle server. It is used for retrieving Enterprise Roles from an enterprise directory service.



Enterprise User Security (EUS) is deprecated with Oracle Database 23ai.

Oracle recommends that you migrate to using Centrally Managed Users (CMU). This feature enables you to directly connect with Microsoft Active Directory without an intervening directory service for enterprise user authentication and authorization to the database. If your Oracle Database is in the cloud, you can also choose to move to one of the newer integrations with a cloud identity provider.

Property	Description
Parameter type	X.500 Distinguished Name
Default value	There is no default value.
Modifiable	No
Modifiable in a PDB	No
Range of values	All X.500 Distinguished Name format values
Basic	No

Note:

The RDBMS_SERVER_DN parameter is deprecated in Oracle Database 12c Release 1 (12.1.0.2). It is replaced by the LDAP_DIRECTORY_ACCESS initialization parameter.

If you do not want to use a directory for enterprise user and privilege management, but prefer to use SSL authentication alone, do not set this parameter.



- Oracle Database Enterprise User Security Administrator's Guide for more information on enterprise roles and the enterprise directory service
- "LDAP_DIRECTORY_ACCESS"

2.325 READ_ONLY

READ ONLY enables or disables read-only mode for a user session.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	Different instances can use different values.

The READ_ONLY parameter allows you to prohibit write operations in a user session, without having to put an entire CDB or container into read-only mode. You can set this parameter in a session created by any type of user, such as a CDB common user, an application common user, or a local user. This parameter is effective in any type of container, such as the CDB root, a PDB, an application root, or an application PDB, when the container is open in read/write mode.

- To enable read-only mode for the current session, set the READ_ONLY parameter to true. You cannot set this parameter to true when there are active transactions in the session.
- To disable read-only mode for the current session, set the READ_ONLY parameter to false.

The READ_ONLY parameter value for a user session does not affect the value of the OPEN_MODE column for the session's container, as displayed in views such as V\$CONTAINER_TOPOLOGY, V\$CONTAINERS, V\$DATABASE, and V\$PDBS.



This parameter is available starting with Oracle Database 23ai.

2.326 READ_ONLY_OPEN_DELAYED

READ ONLY OPEN DELAYED determines when datafiles in read-only tablespaces are accessed.

Property	Description
Parameter type	Boolean



Property	Description
Default value	false
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No

Values

t.rue

The datafiles are accessed for the first time only when an attempt is made to read data stored within them.

false

The datafiles are accessed at database open time.

You can use this parameter to speed up some operations (primarily opening the database) for very large databases when substantial portions of the database are stored in read-only tablespaces. Consider setting this parameter to true for such databases, especially if portions of the read-only data are stored on slow-access devices or hierarchical storage.



Oracle Database Administrator's Guide for information on the consequences of delaying access of datafiles in read-only tablespaces

2.327 RECOVERY_PARALLELISM

RECOVERY_PARALLELISM specifies the number of processes to participate in instance or crash recovery.

Property	Description
Parameter type	Integer
Default value	System-determined parallel recovery
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	System-determined, but cannot exceed PARALLEL_MAX_SERVERS initialization parameter setting
Basic	No

To force serial crash and instance recovery, set the RECOVERY_PARALLELISM parameter to 0 or 1.0 or 1 disable parallel instance and crash recovery on a system that has multiple CPUs. An alert log displays the degree of parallelism that was chosen when the database instance/recovery starts.



 Oracle Real Application Clusters Administration and Deployment Guide for information on setting this parameter in an Oracle RAC environment

2.328 RECYCLEBIN

RECYCLEBIN is used to control whether the Flashback Drop capability is turned on or off.

Property	Description
Parameter type	String
Syntax	<pre>RECYCLEBIN = { on off }</pre>
Default value	on
Modifiable	ALTER SESSION, ALTER SYSTEM DEFERRED
Modifiable in a PDB	Yes
Basic	No

If the parameter is set to off, then dropped tables do not go into the recycle bin. If this parameter is set to on, then dropped tables go into the recycle bin and can be recovered.



Oracle Database Administrator's Guide for more information about using Flashback Drop and managing the recycle bin

2.329 REDO_TRANSPORT_USER

REDO_TRANSPORT_USER specifies the name of the user whose password verifier is used when a remote login password file is used for redo transport authentication. This user must have the SYSOPER privilege and must have the same password in the database that initiates the redo transport session and in the database that is the target of the redo transport session.

Property	Description
Parameter type	String
Syntax	REDO_TRANSPORT_USER = user_name
Default value	There is no default value.
Range of values	Any character string that matches the name of a user who has been granted the SYSOPER privilege
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	Every instance should use the same value



The value of this parameter is case sensitive and must exactly match the value of the USERNAME column of a row in the V\$PWFILE_USERS view. The value of the SYSOPER column of the row must also be TRUE.

If this parameter is not specified, then the password verifier of the SYS user will be used when a remote login password file is used for redo transport authentication.

See Also:

Oracle Data Guard Concepts and Administration for more information about this parameter

2.330 REMOTE_DEPENDENCIES_MODE

REMOTE_DEPENDENCIES_MODE specifies how Oracle should handle dependencies upon remote PL/SQL stored procedures.

Property	Description
Parameter type	String
Syntax	<pre>REMOTE_DEPENDENCIES_MODE = { TIMESTAMP SIGNATURE }</pre>
Default value	TIMESTAMP
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Values

TIMESTAMP

The client running the procedure compares the timestamp recorded on the server-side procedure with the current timestamp of the local procedure and executes the procedure only if the timestamps match.

SIGNATURE

Oracle allows the procedure to execute as long as the signatures are considered safe. This setting allows client PL/SQL applications to be run without recompilation.

See Also:

Oracle Database Development Guide for suggestions for managing dependencies

2.331 REMOTE_LISTENER

REMOTE_LISTENER specifies a network name that resolves to an address or address list of Oracle Net remote listeners (that is, listeners that are not running on the same system as this instance). The address or address list is specified in the TNSNAMES.ORA file or other address repository as configured for your system.

Property	Description
Parameter type	String
Syntax	REMOTE_LISTENER = network_name
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	Yes

- Oracle Database Concepts for more information about listener processes and dispatcher processes
- Oracle Database Net Services Administrator's Guide and your operating systemspecific Oracle documentation for more information about specifying network addresses for the protocols on your system
- Oracle Clusterware Administration and Deployment Guide for information about SCAN addresses

2.332 REMOTE_LOGIN_PASSWORDFILE

REMOTE LOGIN PASSWORDFILE specifies whether Oracle checks for a password file.

Description
String
<pre>REMOTE_LOGIN_PASSWORDFILE = { shared exclusive none }</pre>
exclusive
No
No
Yes
Multiple instances must have the same value.

Values

shared

One or more databases can use the password file. The password file can contain SYS and non-SYS users.

When REMOTE_LOGIN_PASSWORDFILE is set to shared:

 The SYS password cannot be changed. If you try, the password change operation fails with "ORA-28046: Password change for SYS disallowed."



- The password of any user who has SYS* admin privileges (SYSDBA, SYSOPER, SYSASM, SYSBACKUP, SYSDG, SYSKM) cannot be changed. If you try, the password change operation fails with "ORA-01999: password file cannot be updated in SHARED mode."
- Grants of SYS* admin privileges (SYSDBA, SYSOPER, SYSASM, SYSBACKUP, SYSDG, SYSKM) to individual users are not allowed. For example, grant sysdba to scott fails with "ORA-01999: password file cannot be updated in SHARED mode." Similarly, revoke of SYS* admin privileges fails.
- If the password file does not exist, then the behavior is the same as setting REMOTE LOGIN PASSWORDFILE to none.

• exclusive

The password file can be used by only one database. The password file can contain <code>SYS</code> and non-<code>SYS</code> users.

When REMOTE_LOGIN_PASSWORDFILE is set to exclusive, if the password file does not exist, then the behavior is the same as setting REMOTE LOGIN PASSWORDFILE to none.

none

Oracle ignores any password file. Therefore, privileged users must be authenticated by the operating system.



If you change REMOTE_LOGIN_PASSWORDFILE to exclusive or shared from none, then ensure that the password file is synchronized with the dictionary passwords.

2.333 REMOTE OS ROLES

REMOTE OS ROLES specifies whether operating system roles are allowed for remote clients.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No

The default value, false, causes Oracle to identify and manage roles for remote clients.



REMOTE_OS_ROLES is deprecated in Oracle Database 23ai. The ability for a remote client to authenticate a user within Oracle Database was eliminated when the REMOTE_OS_AUTHENT initialization parameter was desupported in Oracle Database 21c. Therefore, REMOTE_OS_ROLES no longer has a supported use case and is deprecated.

2.334 REMOTE_RECOVERY_FILE_DEST

REMOTE_RECOVERY_FILE_DEST specifies a directory from which to read archive log files during a pluggable database (PDB) refresh operation if the source is not available.

Property	Description
Parameter type	String
Syntax	REMOTE_RECOVERY_FILE_DEST = string
Default value	None
Modifiable	ALTER SYSTEM SID='*'
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The directory location where archive redo logs are accessible can be different on different instances, so this parameter can be set to different values on different instances.

For a PDB refresh copy to be in sync with its source, redo is accessed from the source PDB over a database link. Sometimes the source PDB or the CDB to which the source PDB belongs is not accessible when the refresh copy needs to be updated. In those cases, if this parameter is set, an attempt will be made to read archive log files from the directory specified by this parameter.

2.335 REPLICATION_DEPENDENCY_TRACKING

REPLICATION_DEPENDENCY_TRACKING enables or disables dependency tracking for read/write operations to the database.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No

Dependency tracking is essential for propagating changes in a replicated environment in parallel.

Values

TRUE

Enables dependency tracking.

FALSE

Allows read/write operations to the database to run faster, but does not produce dependency information for Oracle to perform parallel propagation.



Do not specify this value unless you are sure that your application will not perform any read/write operations to the replicated tables.

2.336 RESOURCE_LIMIT

RESOURCE LIMIT determines whether resource limits are enforced in database profiles.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

Values

TRUE

Enables the enforcement of resource limits

• FALSE

Disables the enforcement of resource limits



Oracle Database Administrator's Guide and Oracle Database SQL Language Reference for more information on setting resource limits for profiles



2.337 RESOURCE_MANAGE_GOLDENGATE

RESOURCE_MANAGE_GOLDENGATE determines whether Oracle GoldenGate apply processes in the database are resource managed.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	All instances should use the same value.

To enable Resource Manager, set the RESOURCE MANAGER PLAN parameter.

By default, Oracle GoldenGate apply processes in the database are not resource managed. Given that replication to a PDB requires a separate Oracle GoldenGate apply process, it is possible that the apply processes for one PDB could end up consuming most of the CPU on the machine, even if there is a CPU resource management plan in place to limit CPU usage per PDB.

You can set the following values for the RESOURCE_MANAGE_GOLDENGATE parameter:

- TRUE: With this setting, Oracle GoldenGate apply processes in the database are resource managed based on the resources allocated to the GoldenGate apply user.
- FALSE: With this setting, Oracle GoldenGate apply processes are not resource managed.

2.338 RESOURCE_MANAGER_CPU_ALLOCATION

RESOURCE_MANAGER_CPU_ALLOCATION specifies the number of CPUs that the Resource Manager should use. The Resource Manager controls how a system's CPUs are utilized by its database's sessions.



The RESOURCE_MANAGER_CPU_ALLOCATION parameter is deprecated and its value is ignored. This parameter is retained only for backward compatibility. Use the CPU_COUNT parameter instead.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No



Property	Description
Range of values	0 to operating system-specific
Basic	No

The Resource Manager schedules database sessions on the CPUs according to a resource plan that has been configured and enabled by the DBA. Normally, the Resource Manager schedules enough database sessions to keep all CPUs utilized. However, in some scenarios, a DBA may only want to schedule enough database sessions to keep a subset of the CPUs utilized.

See Also:

2.339 RESOURCE MANAGER CPU SCOPE

RESOURCE_MANAGER_CPU_SCOPE controls the scope of CPU resource management.

In previous releases, Oracle offered one method of CPU resource management, which operated within a database instance. Starting with Oracle Database 23ai, Oracle offers an alternative method for CPU resource management, which operates across all database instances on the server. This parameter allows you to choose between the two methods.

Property	Description
Parameter type	String
Syntax	<pre>RESOURCE_MANAGER_CPU_SCOPE = { SERVER_WIDE INSTANCE_ONLY }</pre>
Default value	INSTANCE_ONLY
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	The same value must be used on all instances.

Values:

SERVER WIDE - Enables server-level inter-instance CPU resource management

Introduced in Oracle Database 23ai, this method can be used when multiple database instances share a Linux server. It enables CPU resource management across all database instances on the server. With this method, Oracle automatically configures Linux control groups (cgroups) with the desired CPU sharing and CPU utilization limits, and then places database sessions into the appropriate cgroups.

In order to use this method, the following system and database requirements must be met:

This method requires the Linux operating system. It is supported on Oracle
 Autonomous Database, Oracle Database Exadata Cloud@Customer, Oracle Database
 Exadata Cloud Service, and on-premises Oracle Exadata systems. This method is not
 supported on non-Exadata Linux systems.

 The server operating system must be Unbreakable Enterprise Kernel (UEK) version 5 or later.

You can configure this method as follows:

- Oracle Autonomous Database: All system and database requirements are installed and configured by default. RESOURCE_MANAGER_CPU_SCOPE is set to SERVER_WIDE by default.
- Oracle Database Exadata Cloud@Customer: All system and database requirements
 are installed and configured by default. RESOURCE_MANAGER_CPU_SCOPE is set to
 INSTANCE_ONLY by default. You must set it to SERVER_WIDE if you want to use this
 method.
- Oracle Database Exadata Cloud Service and on-premises Oracle Exadata systems:
 The Exadata image meets all system and database requirements.

 RESOURCE_MANAGER_CPU_SCOPE is set to INSTANCE_ONLY by default. You must set it to SERVER WIDE if you want to use this method.

When you specify the SERVER_WIDE setting, the value of the PROCESSOR_GROUP_NAME initialization parameter is ignored.

INSTANCE ONLY - Enables CPU resource management only within the database instance.

This is the method of CPU resource management used in releases prior to Oracle Database 23ai. It is supported on all Oracle Databases.



This parameter is available starting with Oracle Database 23ai.

See Also:

Oracle Database Administrator's Guide for more information about setting this parameter

2.340 RESOURCE_MANAGER_PLAN

RESOURCE MANAGER PLAN specifies the resource plan to use for a database (CDB or non-CDB).

Property	Description
Parameter type	String
Syntax	RESOURCE_MANAGER_PLAN = resource_plan_name
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	Any valid character string
Basic	No



In a CDB

In the root for a CDB, RESOURCE_MANAGER_PLAN specifies the CDB resource plan. A CDB resource plan allocates resources among PDBs.

A CDB resource plan is created using <code>DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN</code> and <code>CREATE_CDB_PLAN</code> directive.

See Also:

- Oracle Database Administrator's Guide for more information about using DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN and DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE
- Oracle Database PL/SQL Packages and Types Reference for more information about the CREATE_CDB_PLAN procedure and CREATE_CDB_PLAN_DIRECTIVE procedure for the DBMS RESOURCE MANAGER package

A session must be root to change the value of RESOURCE_MANAGER_PLAN for a CDB using the ALTER SYSTEM statement. For example, to enable and disable a CDB resource plan:

```
SQL> ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = CDB_resource_plan_name;
SQL> ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = '';
```

In a PDB, RESOURCE MANAGER PLAN specifies the PDB resource plan to use for the PDB.

A session must be in the PDB to enable or disable a PDB resource plan for that PDB. For example, to enable and disable a PDB resource plan:

```
SQL> ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = PDB_resource_plan_name; SQL> ALTER SYSTEM SET RESOURCE MANAGER PLAN = '';
```

In a PDB, the PDB resource plan has some restrictions compared to the resource plan of a non-CDB. The following restrictions apply to PDB resource plans:

- A PDB resource plan cannot have subplans.
- A PDB resource plan cannot have a multiple-level scheduling policy.

To enforce certain PDB resource plan policies, policies regarding resource allocation among PDBs should exist in the CDB resource plan. Without a CDB resource plan, certain PDB resource plan policies will not be enforced. If a PDB resource plan contains CPU or parallel statement queuing directives and a CDB resource plan is not specified, then Resource Manager will automatically enable the DEFAULT_CDB_PLAN plan. To prevent this behavior, set the RESOURCE_MANAGER_PLAN parameter at the root level to ORA\$INTERNAL_CDB_PLAN.

Note:

See *Oracle Database Administrator's Guide* for information on CPU, I/O bandwidth, and parallel execution servers requirements in CDB resource plans, and for a description of the results in the PDB resource plans when those requirements are not met.



In a Non-CDB

RESOURCE_MANAGER_PLAN specifies the top-level resource plan to use for an instance in a non-CDB. The resource manager will load this top-level resource plan along with all its descendants (subplans, directives, and consumer groups). If you do not specify this parameter, the resource manager is off by default.

You can change the setting of this parameter using the ALTER SYSTEM statement to turn on the resource manager (if it was previously off) or to turn off the resource manager or change the current resource plan (if it was previously on). If you specify a resource plan that does not exist in the data dictionary, Oracle returns an error message.

See Also:

- Oracle Database Administrator's Guide for information on using Oracle Resource Manager
- Oracle Database PL/SQL Packages and Types Reference for information on the DBMS RESOURCE MANAGER package
- Oracle Database PL/SQL Packages and Types Reference for information on the DBMS RESOURCE MANAGER PRIVS package
- "DBA_RSRC_PLANS", "DBA_RSRC_PLAN_DIRECTIVES", and the various V\$RSRC_* dynamic performance views in Dynamic Performance Views for information on existing resource plans

2.341 RESULT_CACHE_AUTO_BLOCKLIST

Use ${\tt RESULT_CACHE_AUTO_BLOCKLIST}$ to enable or disable adaptive result cache object exclusion.

Property	Description
Parameter type	String
Syntax	RESULT_CACHE_AUTO_BLOCKLIST = { OFF ON }
Default value	ON
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different instances can use different values.

When this parameter is set to ON, adaptive result cache object exclusion is enabled.

When this parameter is set to OFF, adaptive result cache object exclusion is disabled.



This parameter is available starting with Oracle Database 23ai.

2.342 RESULT_CACHE_EXECUTION_THRESHOLD

RESULT_CACHE_EXECUTION_THRESHOLD specifies the maximum number of times a PL/SQL function can be executed before its result is stored in the result cache.

Property	Description
Parameter type	Integer
Default value	2
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to operating system-dependent
Basic	No
Oracle RAC	Multiple instances can have different values.

2.343 RESULT_CACHE_INTEGRITY

RESULT CACHE INTEGRITY specifies whether the database enforces result cache integrity.

Property	Description
Parameter type	String
Syntax	<pre>RESULT_CACHE_INTEGRITY = { ENFORCED TRUSTED }</pre>
Default value	Oracle Autonomous Database: ENFORCED
	All other Oracle databases: TRUSTED
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different instances can have different values.

Values:

ENFORCED

Regardless of the setting of the RESULT_CACHE_MODE initialization parameter or any specified hints, only deterministic constructs will be eligible for result caching. For example, queries using PL/SQL functions that are not explicitly declared as deterministic will not be cached. This setting enforces result cache integrity.

TRUSTED

The database honors the setting of the RESULT_CACHE_MODE initialization parameter and any specified hints, and will consider queries using possibly nondeterministic constructs as candidates for result caching. For example, queries using PL/SQL functions that are not explicitly declared as deterministic may be cached. Results that are certain to be nondeterministic (for example, SYSDATE or constructs involving SYSDATE) will not be cached. This setting does not enforce result cache integrity, which matches the behavior in releases prior to Oracle Database 23ai.



Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

"RESULT_CACHE_MODE"

2.344 RESULT_CACHE_MAX_RESULT

RESULT_CACHE_MAX_RESULT specifies the percentage of RESULT_CACHE_MAX_SIZE that any single result can use.

Property	Description
Parameter type	Integer
Default value	5 percent
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 100
Basic	No
Oracle RAC	Multiple instances can have different values.

2.345 RESULT_CACHE_MAX_SIZE

RESULT_CACHE_MAX_SIZE specifies the maximum amount of SGA memory (in bytes) that can be used by the result cache.

Property	Description
Parameter type	Big integer
Syntax	RESULT_CACHE_MAX_SIZE = integer [K M G]
Default value	Derived from the values of SHARED_POOL_SIZE, SGA_TARGET, and MEMORY_TARGET
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to operating system-dependent
Basic	No
Oracle RAC	You must either set this parameter to 0 on all instances to disable the result cache, or use a nonzero value on all instances. Disabling the result cache on some instances may lead to incorrect results.

Values of this parameter greater than 0 are rounded up to the next multiple of 32 KB. If the value of this parameter is 0, then the feature is disabled.



Oracle Database Performance Tuning Guide for information about tuning the result cache

2.346 RESULT CACHE MAX TEMP RESULT

RESULT_CACHE_MAX_TEMP_RESULT specifies the percentage of RESULT_CACHE_MAX_TEMP_SIZE that any single cached query result can consume.

Property	Description
Parameter type	Integer
Default value	5 percent
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 100
Basic	No
Oracle RAC	Multiple instances can have different values.

2.347 RESULT_CACHE_MAX_TEMP_SIZE

 ${\tt RESULT_CACHE_MAX_TEMP_SIZE} \ \ specifies \ the \ maximum \ amount \ of \ temporary \ tablespace \ (in bytes) \ that \ can be \ consumed \ by \ the \ result \ cache.$

Property	Description
Parameter type	Big integer
Syntax	RESULT_CACHE_MAX_TEMP_SIZE = integer [K M G]
Default value	RESULT_CACHE_MAX_SIZE * 10
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to operating system-dependent
Basic	No
Oracle RAC	You must either set this parameter to 0 on all instances to disable the result cache, or use a nonzero value on all instances. Disabling the result cache on some instances may lead to incorrect results.

If you attempt to set this parameter to a value that is less than 5% of the size of the SGA, then this parameter will be set to 5% of the size of the SGA. If you attempt to set this parameter to a value that is more than 10% of the current total unused space available for temporary tablespaces in the SYS schema, then this parameter will be set to 10% of the current total unused space available for temporary tablespaces in the SYS schema.

If the value of this parameter is 0, then the result cache is disabled for the PDB.



Oracle Database Performance Tuning Guide for information about tuning the result cache

2.348 RESULT_CACHE_MODE

RESULT_CACHE_MODE specifies which queries are eligible to store result sets in the result cache. Only query execution plans with the result cache operator will attempt to read from or write to the result cache.

Property	Description
Parameter type	String
Syntax	<pre>RESULT_CACHE_MODE = { MANUAL MANUAL_TEMP FORCE FORCE_TEMP }</pre>
Default value	Oracle Autonomous Database: FORCE
	All other Oracle databases: MANUAL
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different instances can have different values.

Values

MANUAL

Query results can only be stored in the result cache by using a query hint or table annotation.

MANUAL TEMP

Query results can only be stored in the result cache by using a query hint or table annotation. All hinted queries are allowed to leverage temporary segments on disk unless explicitly prohibited by using the $/*+\ RESULT_CACHE\ (TEMP=FALSE)\ */\ hint.$

FORCE

All query results are stored in the result cache. If a query result is not in the cache, then the database executes the query and stores the result in the cache. Subsequent executions of the same SQL statement that include the /*+ RESULT_CACHE */ hint will retrieve data from the cache. Sessions will use these results, if possible. To exclude query results from the cache, use the /*+ NO RESULT CACHE */ hint.

FORCE_TEMP

All query results are stored in the result cache. All queries are allowed to leverage temporary segments on disk unless explicitly prohibited by a hint.

A

Caution:

If this parameter is set to <code>FORCE</code> or <code>FORCE_TEMP</code>, then Oracle recommends setting the <code>RESULT_CACHE_INTEGRITY</code> parameter to <code>ENFORCED</code>, which ensures that only deterministic constructs are eligible for result caching. This prevents the caching of nondeterministic constructs, which can potentially cause material changes to results. See <code>"RESULT_CACHE_INTEGRITY"</code> for more information.

See Also:

- Oracle Database SQL Tuning Guide for more information on how the result cache handles PL/SQL functions before changing the value of this initialization parameter
- Oracle Database SQL Language Reference for more information about the NO RESULT CACHE hint

2.349 RESULT_CACHE_REMOTE_EXPIRATION

RESULT_CACHE_REMOTE_EXPIRATION specifies the number of minutes that a result using a remote object is allowed to remain valid.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to operating system-dependent
Basic	No

Setting this parameter to 0 implies that results using remote objects should not be cached. Setting this parameter to a nonzero value can produce stale answers (for example, if the remote table used by a result is modified at the remote database).



See Also:

Oracle Database Performance Tuning Guide for information about tuning the result cache

2.350 RESUMABLE_TIMEOUT

RESUMABLE_TIMEOUT enables or disables resumable statements and specifies resumable timeout at the system level.

Property	Description
Parameter type	Integer
Default value	0 (seconds)
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 2 ³¹ - 1 (in seconds)
Basic	No
Oracle RAC	Multiple instances can have different values.

See Also:

Oracle Database Administrator's Guide for more information about enabling resumable space allocation, what conditions are correctable, and what statements can be made resumable

2.351 ROLLBACK_SEGMENTS

ROLLBACK SEGMENTS allocates one or more rollback segments by name to this instance.

Property	Description
Parameter type	String
Syntax	ROLLBACK_SEGMENTS =
	<pre>(segment_name [, segment_name])</pre>
Default value	If you do not specify this parameter, the instance uses public rollback segments by default, unless the <code>UNDO_MANAGEMENT</code> initialization parameter is set to <code>AUTO</code> . In that case, the <code>ROLLBACK_SEGMENTS</code> parameter is ignored and automatic undo management is used.
Modifiable	No
Modifiable in a PDB	Yes
Range of values	Any rollback segment names listed in DBA_ROLLBACK_SEGS except SYSTEM
Basic	No
Oracle RAC	Multiple instances must have different values.

If you set this parameter, the instance acquires all of the rollback segments named in this parameter, even if the number of rollback segments exceeds the minimum number required by the instance (calculated as TRANSACTIONS / TRANSACTIONS_PER_ROLLBACK_SEGMENT).

You cannot change the value of this parameter dynamically, but you can change its value and then restart the instance. Although this parameter usually specifies private rollback segments, it can also specify public rollback segments if they are not already in use.

To find the name, segment ID number, and status of each rollback segment in the database, query the data dictionary view DBA ROLLBACK SEGS.

When undo management is set to auto, rollback segments is ignored.



See Also:

- Oracle Real Application Clusters Administration and Deployment Guide for information on setting this parameter in an Oracle RAC environment
- "DBA_ROLLBACK_SEGS"

2.352 SAGA_HIST_RETENTION

SAGA_HIST_RETENTION specifies the length of time (in minutes) that completed sagas are retained in the DBA HIST SAGAS and USER HIST SAGAS views.

Property	Description
Parameter type	Integer
Default value	43200 minutes (equivalent to 30 days)
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	1 - 2147483647 minutes
Basic	No
Oracle RAC	Multiple instances must have the same value.

Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

- "ALL_HIST_SAGAS"
- "DBA_HIST_SAGAS"
- "USER HIST SAGAS"

2.353 SEC_MAX_FAILED_LOGIN_ATTEMPTS

SEC_MAX_FAILED_LOGIN_ATTEMPTS specifies the number of authentication attempts that can be made by a client on a connection to the server process.

Property	Description
Parameter type	Integer
Default value	3
Modifiable	No



Property	Description
Modifiable in a PDB	No
Range of values	An integer greater than or equal to 1 .
Basic	No

These login attempts can be for multiple user accounts in the same connection. After the specified number of failure attempts, the connection will be automatically dropped by the server process, and the server process is terminated.



Oracle Database Security Guide for more information about this parameter

2.354 SEC_PROTOCOL_ERROR_FURTHER_ACTION

SEC_PROTOCOL_ERROR_FURTHER_ACTION specifies the further execution of a server process when receiving bad packets from a possibly malicious client.

Property	Description
Parameter type	String
Syntax	<pre>SEC_PROTOCOL_ERROR_FURTHER_ACTION = { CONTINUE (DELAY, integer) (DROP, integer) }</pre>
Default value	(DROP, 3)
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

Values

CONTINUE

The server process continues execution. The database server may be subject to a Denial of Service (DoS) if bad packets continue to be sent by a malicious client.

(DELAY, integer)

The client experiences a delay of *integer* seconds before the server process accepts the next request from the same client connection. Malicious clients are prevented from excessive consumption of server resources while legitimate clients experience a degradation in performance but can continue to function.

• (DROP, integer)

The server forcefully terminates the client connection after *integer* cumulative bad packets. The server protects itself at the expense of the client (for example, a client transaction may be lost). The client may reconnect and attempt the same operation.



See Also:

Oracle Database Security Guide for examples of using this parameter

2.355 SEC_PROTOCOL_ERROR_TRACE_ACTION

SEC_PROTOCOL_ERROR_TRACE_ACTION specifies the kind of logging the database server does when bad packets are received from a possibly malicious client, apart from the client receiving the error.

Property	Description
Parameter type	String
Syntax	<pre>SEC_PROTOCOL_ERROR_TRACE_ACTION = { NONE TRACE LOG ALERT }</pre>
Default value	TRACE
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

Values

NONE

The database server does not record it in any of the trace files.

TRACE

A short entry is made in the alert log file and a detailed incident file is generated. The server trace file also will have an entry about the protocol error and incident file.

LOG

The server trace file will have an entry about the protocol violation.

ALERT

A short entry is made in the alert log file and in the server trace file about the protocol violation.

2.356 SEC_RETURN_SERVER_RELEASE_BANNER

SEC_RETURN_SERVER_RELEASE_BANNER specifies whether the server returns complete database software information to unauthenticated clients.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No



Property	Description
Range of values	true false
Basic	No

Values

true

Returns complete database version information to the client.

false

Returns a generic version string to the client.



Oracle Call Interface Developer's Guide and Oracle Database Security Guide for more information on controlling the display of the database version banner.

2.357 SERIAL_REUSE

SERIAL_REUSE specifies which types of cursors make use of the serial-reusable memory feature. This feature allocates private cursor memory in the SGA so that it can be reused (serially, not concurrently) by sessions executing the same cursor.

Property	Description
Parameter type	String
Syntax	<pre>SERIAL_REUSE = { disable all select dml plsql force }</pre>
Default value	disable
Modifiable	No
Modifiable in a PDB	No
Basic	No



The ${\tt SERIAL_REUSE}$ parameter is deprecated. It is retained for backward compatibility only.

Values

disable

Disables the option for all SQL statement types. This value overrides any other values included in the list.

• all

Enables the option for both DML and SELECT statements. Equivalent to setting select, dml, and plsql.

• select

Enables the option for SELECT statements.

dml

Enables the option for DML statements.

plsql

Currently has no effect (although PL/SQL packages do support the serial-reuse memory option using PL/SQL pragmas).



If CURSOR_SPACE_FOR_TIME is set to true, then the value of SERIAL_REUSE is ignored and treated as if it were set to disable.

See Also:

"CURSOR SPACE FOR TIME"

2.358 SERVICE NAMES

 ${\tt SERVICE_NAMES} \ \ \textbf{specifies one or more names by which clients can connect to the instance}.$

Property	Description
Parameter type	String
Syntax	SERVICE_NAMES =
	<pre>db_service_name [, db_service_name []]</pre>
Default value	DB_UNIQUE_NAME.DB_DOMAIN if defined
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Any ASCII string or comma-separated list of string names
Basic	No
Oracle RAC	Do not set the SERVICE_NAMES parameter for Oracle RAC environments. Instead, define services using Oracle Enterprise Manager and manage those services using Server Control (SRVCTL) utility.

The instance registers its service names with the listener. When a client requests a service, the listener determines which instances offer the requested service and routes the client to the appropriate instance.

You can specify multiple service names to distinguish among different uses of the same database. For example:

SERVICE_NAMES = sales.example.com, widgetsales.example.com



You can also use service names to identify a single service that is available from two different databases through the use of replication.

If you do not qualify the names in this parameter with a domain, Oracle qualifies them with the value of the $\mbox{DB_DOMAIN}$ parameter. If $\mbox{DB_DOMAIN}$ is not specified, then no domain will be applied to the non-qualified $\mbox{SERVICE}$ NAMES values.

When you specify additional service names with this parameter, the default service name is not overridden. The default service name plus the additional service names specified with this parameter are the service names that clients can use to connect to the database.

Note:

The SERVICE_NAMES initialization parameter is deprecated in Oracle Database 19c and may be desupported in a future release.

Use of the SERVICE_NAMES parameter is no longer actively supported. It must not be used for high availability (HA) deployments and it is not supported for HA operations. This restriction includes FAN, load balancing, FAILOVER_TYPE, FAILOVER_RESTORE, SESSION STATE CONSISTENCY, and any other uses.

To manage your services, Oracle recommends that you instead use the SRVCTL command-line utility, the GDSCTL command-line utility, or the DBMS_SERVICE PL/SQL package.

See Also:

- Oracle Database Net Services Administrator's Guide for more information on this parameter and its settings
- Oracle Real Application Clusters Administration and Deployment Guide for information about services administration in an Oracle RAC environment
- "DB DOMAIN"

2.359 SESSION CACHED CURSORS

SESSION CACHED CURSORS specifies the number of session cursors to cache.

Property	Description
Parameter type	Integer
Default value	50
Modifiable	ALTER SESSION, ALTER SYSTEM DEFERRED
Modifiable in a PDB	Yes
Range of values	0 to operating system-dependent
Basic	No
Oracle RAC	Multiple instances can have different values.



Repeated parse calls of the same SQL (including recursive SQL) or PL/SQL statement cause the session cursor for that statement to be moved into the session cursor cache. Oracle uses a least recently used algorithm to remove entries in the session cursor cache to make room for new entries when needed.

See Also:

Oracle Database Performance Tuning Guide for information about enabling the session cursor cache

2.360 SESSION_EXIT_ON_PACKAGE_STATE_ERROR

Use <code>SESSION_EXIT_ON_PACKAGE_STATE_ERROR</code> to force a hard session exit when a session's state has been invalidated.

Exiting sessions after state invalidation avoids errors that can occur when applications mishandle an invalid state.

Property	Description
Parameter type	Boolean
Default value	FALSE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	TRUE FALSE
Basic	No
Oracle RAC	Different instances can use different values.

Set SESSION_EXIT_ON_PACKAGE_STATE_ERROR to TRUE to force a hard session exit under the following circumstances:

- When a stateful PL/SQL package undergoes modification, the sessions that have an active instantiation of the package receive an ORA-4068 error when they attempt to run it. When SESSION_EXIT_ON_PACKAGE_STATE_ERROR is set to TRUE, the session immediately exits instead of raising ORA-04068. This can be advantageous because many applications are better equipped to handle a session being discarded, and failing to catch errors related to session state, such as ORA-04068, can lead to silent data corruption.
- When an Oracle Database Multilingual Engine (MLE) module or environment is modified via DDL, sessions that have an active instantiation of the object receive an ORA-4106 or ORA-4107 error. Setting SESSION_EXIT_ON_PACKAGE_STATE_ERROR to TRUE forces sessions to be disconnected if the session state is invalidated. Because many applications capture session disconnect, this option can help simplify the recovery from the invalidation of an existing session state.

Note:

This parameter is available starting with Oracle Database 23ai.

2.361 SESSION_MAX_OPEN_FILES

SESSION_MAX_OPEN_FILES specifies the maximum number of BFILEs that can be opened in any session. Once this number is reached, subsequent attempts to open more files in the session by using DBMS_LOB.FILEOPEN() or OCILobFileOpen() will fail. The maximum value for this parameter depends on the equivalent parameter defined for the underlying operating system.

Property	Description
Parameter type	Integer
Default value	10
Modifiable	No
Modifiable in a PDB	No
Range of values	1 to either 50 or the value of MAX_OPEN_FILES defined at the operating system level, whichever is less
Basic	No

See Also:

- Oracle Database SecureFiles and Large Objects Developer's Guide for information on LOBs in general
- Oracle Database SecureFiles and Large Objects Developer's Guide for information on BFILEs
- Oracle Database PL/SQL Packages and Types Reference for information on the DBMS LOB.FILEOPEN() procedure
- Oracle Call Interface Developer's Guide for information on the <code>OCILobFileOpen()</code> procedure

2.362 SESSIONS

SESSIONS specifies the maximum number of sessions that can be created in the system.

Property	Description
Parameter type	Integer
Default value	Derived: (1.5 * PROCESSES) + 22
Modifiable	ALTER SYSTEM can be used in a PDB only to change the value of the SESSIONS parameter for that PDB.
	ALTER SYSTEM cannot be used to change the value of the SESSIONS parameter in a non-CDB or in CDB\$ROOT.
Modifiable in a PDB	Yes
Range of values	1 to (2 ¹⁸ - 1), which is 1 to 262143
Basic	Yes



Because every login requires a session, this parameter effectively determines the maximum number of concurrent users in the system. You should always set this parameter explicitly to a value equivalent to your estimate of the maximum number of concurrent users, plus the number of background processes, plus approximately 10% for recursive sessions.

Oracle uses the default value of this parameter as its minimum. Values between 1 and the default do not trigger errors, but Oracle ignores them and uses the default instead.

The default values of the <code>ENQUEUE_RESOURCES</code> and <code>TRANSACTIONS</code> parameters are derived from <code>SESSIONS</code>. Therefore, if you increase the value of <code>SESSIONS</code>, you should consider whether to adjust the values of <code>ENQUEUE_RESOURCES</code> and <code>TRANSACTIONS</code> as well. (Note that <code>ENQUEUE_RESOURCES</code> is obsolete as of Oracle Database 10g release 2 (10.2).)

In a shared server environment, the value of PROCESSES can be quite small. Therefore, Oracle recommends that you adjust the value of SESSIONS to approximately 1.1 * total number of connections.

For a CDB, the root container's SESSIONS parameter specifies the total number of sessions for the database.

The SESSIONS parameter for a PDB specifies the total number of sessions that can be used by that PDB. Its value defaults to the root container's SESSIONS value. If the PDB tries to use more sessions than configured by its SESSIONS parameter, an ORA-00018 error message is generated. For PDBs, the SESSIONS parameter does not count recursive sessions and hence does not require the 10% adjustment.

The SESSIONS parameter for a PDB can only be modified by the PDB. It cannot be set higher than the CDB's SESSIONS value.

See Also:

- Oracle Database Concepts for more information on memory structures
- Oracle Database Concepts for more information on processes
- Oracle Multitenant Administrator's Guide for more information about CDBs and PDBs

2.363 SGA_MAX_SIZE

SGA MAX SIZE specifies the maximum size of the SGA for the lifetime of the instance.

Property	Description
Parameter type	Big integer
Syntax	SGA_MAX_SIZE = integer [K M G]
Default value	Initial size of SGA at startup, dependent on the sizes of different pools in the SGA, such as buffer cache, shared pool, large pool, and so on.
Modifiable	No
Modifiable in a PDB	No
Range of values	0 to operating system-dependent



On 64-bit platforms and non-Windows 32-bit platforms, when either MEMORY_TARGET or MEMORY_MAX_TARGET is specified, the default value of SGA_MAX_SIZE is set to the larger of the two parameters. This causes more address space to be reserved for expansion of the SGA.

On Windows 32-bit platforms, the default value of SGA_MAX_SIZE is the largest of the following values:

- 60% of MEMORY TARGET, if specified
- 60% of MEMORY MAX TARGET, if specified
- 25% of the total available virtual address space



Oracle Database Performance Tuning Guide for more information about this parameter

2.364 SGA MIN SIZE

 SGA_MIN_SIZE indicates a possible minimum value for SGA usage of a pluggable database (PDB).

Property	Description
Parameter type	Big integer
Syntax	SGA_MIN_SIZE = integer [K M G]
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 50% of SGA_TARGET
Basic	No
Oracle RAC	The same value must be used for all instances.

Setting this parameter at the CDB level has no effect.



Note:

To be able to use Resource Manager in a CDB to control the amount of memory each PDB can use:

- The NONCDB_COMPATIBLE initialization parameter must be set to FALSE at the CDB level (in the root of the CDB).
- The MEMORY TARGET initialization parameter must not be set at the CDB level.
- A value for SGA TARGET must be set at the CDB level.

If SGA_TARGET is not set at the CDB level, then setting SGA_MIN_SIZE in a PDB has no effect. You will not receive an error message, and the PDB's SGA_MIN_SIZE value will not be enforced.

- You must set the SGA MIN SIZE value to a value that meets these requirements:
 - In a PDB, to a value that is less than or equal to 50% of the value of SGA TARGET in the PDB
 - In a PDB, to a value that is less than or equal to 50% of the value of SGA TARGET at the CDB level
 - In a PDB, to a value that is less than the value of SGA_MAX_SIZE at the CDB level
 - Across all the open PDBs in a CDB, the sum of SGA_MIN_SIZE values must be less than or equal to 50% of the managed SGA total at the CDB level.
 Managed SGA is DB CACHE SIZE plus SHARED POOL SIZE.

When you set SGA_MIN_SIZE in a PDB to a value that does not meet these requirements, you receive an error.

See Also:

Oracle Multitenant Administrator's Guide for more information about the initialization parameters that control the memory usage of PDBs

2.365 SGA TARGET

SGA TARGET specifies the total size of all SGA components.

Property	Description
Parameter type	Big integer
Syntax	SGA_TARGET = integer [K M G]
Default value	0 (SGA autotuning is disabled for DEFERRED mode autotuning requests, but allowed for IMMEDIATE mode autotuning requests)
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes



Property	Description
Range of values	64 MB to SGA_MAX_SIZE
	Note that if MEMORY_MAX_TARGET > 0, then the value of SGA_TARGET also cannot exceed the value of MEMORY_MAX_TARGET.
Basic	Yes

If SGA TARGET is specified, then 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)
- Data transfer cache (DATA TRANSFER CACHE SIZE)

If these automatically tuned memory pools are set to nonzero values, then those values are used as minimum levels by Automatic Shared Memory Management. You would 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 Automatic Shared Memory Management:

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

The memory allocated to these pools is deducted from the total available for SGA_TARGET when Automatic Shared Memory Management computes the values of the automatically tuned memory pools.

In the **Default value** field, IMMEDIATE mode autotuning requests are necessary to avoid ORA-04031 errors. The DEFERRED and IMMEDIATE modes are reflected in the OPER_MODE column of the V\$MEMORY RESIZE OPS view.

If Automatic Memory Management is enabled (MEMORY_TARGET is set to a positive value) and SGA_TARGET is also set to a positive value, the SGA_TARGET value acts as the minimum value for the Size of the SGA.



Note:

This parameter is optional for pluggable databases (PDBs). When this parameter is set for a PDB, it specifies the maximum SGA that the PDB can use at any time. When this parameter is not set at the PDB level, the PDB has no limit for the amount of SGA it can use, other than the CDB's SGA size.

To be able to use Resource Manager in a CDB to control the amount of memory each PDB can use:

- The NONCDB_COMPATIBLE initialization parameter must be set to FALSE at the CDB level (in the root of the CDB).
- The MEMORY_TARGET initialization parameter must not be set at the CDB level.
- You must set the SGA TARGET initialization parameter at the CDB level.
- You must set SGA TARGET in a PDB.

If you set a PDB's SGA_TARGET value and SGA_TARGET is not set at the CDB level, you will not receive an error message and the PDB's SGA_TARGET value will not be enforced.

- When setting SGA TARGET in a PDB, the following requirements must be met:
 - The value of SGA_TARGET in the PDB must be less than or equal to the SGA_TARGET value at the CDB level.
 - The value of SGA_TARGET in the PDB must be less than or equal to the SGA_MAX_SIZE value at the CDB level.
 - The value of SGA_TARGET in the PDB must be twice the PDB's DB_CACHE_SIZE value, if the PDB's DB_CACHE_SIZE value is set.
 - The value of SGA_TARGET in the PDB must be twice the PDB's SHARED_POOL_SIZE value, if the PDB's SHARED_POOL_SIZE value is set.
 - The value of SGA_TARGET in the PDB must be twice the PDB's SGA_MIN_SIZE value, if the PDB's SGA_MIN_SIZE value is set.

When you set SGA_TARGET in a PDB to a value that does not meet these requirements, you receive an error.



See Also:

- "DB_CACHE_SIZE"
- "SHARED_POOL_SIZE"
- "LARGE_POOL_SIZE"
- "JAVA_POOL_SIZE"
- "STREAMS_POOL_SIZE"
- "V\$MEMORY_RESIZE_OPS"
- "MEMORY_TARGET"
- "PGA_AGGREGATE_TARGET"
- Oracle Multitenant Administrator's Guide for more information about the initialization parameters that control the memory usage of PDBs
- Oracle Database Administrator's Guide for information on automatic memory management
- Oracle Database Administrator's Guide for information on managing the SGA manually

2.366 SHADOW_CORE_DUMP

SHADOW_CORE_DUMP specifies whether Oracle includes the SGA in the core file for foreground (client) processes.

Property	Description
Parameter type	String
Syntax	SHADOW_CORE_DUMP = { partial full none }
Default value	partial. On Windows, the default value is none.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Values

partial

Oracle does not include the SGA in the core dump.

full

Oracle includes the SGA in the core dump.

none

No core files will be generated for foreground processes.



See Also:

"BACKGROUND_CORE_DUMP"

2.367 SHARD_ENABLE_RAFT_FOLLOWER_READ

Use <code>SHARD_ENABLE_RAFT_FOLLOWER_READ</code> to enable or disable reads from follower replication units in a shard.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No



This parameter is available starting with Oracle Database 23ai.

2.368 SHARD_RAFT_LOGFILE_SIZE

SHARD RAFT LOGFILE SIZE specifies (in bytes) the size of Raft log files.

Property	Description
Parameter type	Big integer
Syntax	SHARD_RAFT_LOGFILE_SIZE = integer [K M G]
Default value	1G
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to operating system-dependent
Basic	No
Oracle RAC	Different instances can use different values.

Each replication unit (RU) has 3 Raft log files. Therefore, the log capacity of an RU is 3 times the value of this parameter.

When you modify the value of this parameter in the current instance, the change takes effect the first time a Raft log file is subsequently overwritten.





This parameter is available starting with Oracle Database 23ai.

2.369 SHARED MEMORY ADDRESS

SHARED_MEMORY_ADDRESS and HI_SHARED_MEMORY_ADDRESS specify the starting address at run time of the system global area (SGA). This parameter is ignored on the many platforms that specify the SGA's starting address at linktime.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Modifiable in a PDB	No
Basic	No

Use this parameter to specify the entire address on 32-bit platforms and to specify the low-order 32 bits of a 64-bit address on 64-bit platforms. Use <code>HI_SHARED_MEMORY_ADDRESS</code> to specify the high-order 32 bits of a 64-bit address on 64-bit platforms. If both parameters are 0 or unspecified, the SGA address defaults to a platform-specific location.



"HI_SHARED_MEMORY_ADDRESS"

2.370 SHARED_POOL_RESERVED_SIZE

SHARED_POOL_RESERVED_SIZE specifies (in bytes) the shared pool space that is reserved for large contiguous requests for shared pool memory.

Property	Description
Parameter type	Big integer
Syntax	SHARED_POOL_RESERVED_SIZE = integer [K M G]
Default value	5% of the value of SHARED_POOL_SIZE
Modifiable	No
Modifiable in a PDB	No
Range of values	Minimum: 5000
	Maximum: half of the value of SHARED_POOL_SIZE
Basic	No



You can use this parameter to avoid performance degradation in the shared pool in situations where pool fragmentation forces Oracle to search for and free chunks of unused pool to satisfy the current request.

See Also:

- "SHARED POOL SIZE"
- Oracle Database Performance Tuning Guide for information on sizing the shared pool

2.371 SHARED_POOL_SIZE

SHARED POOL SIZE specifies (in bytes) the size of the shared pool.



Oracle recommends against setting this parameter in a PDB. The only shared memory sizing parameter that should be set in a PDB is SGA_TARGET , which specifies the maximum SGA that the PDB can use at any time.

Property	Description
Parameter type	Big integer
Syntax	SHARED_POOL_SIZE = integer [K M G]
Default value	If SGA_TARGET is set: If the parameter is not specified, then the default is 0 (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool.
	If SGA_TARGET is not set (32-bit platforms): 64 MB, rounded up to the nearest granule size.
	If SGA_TARGET is not set (64-bit platforms): 128 MB, rounded up to the nearest granule size.
	For considerations when dealing with database instances using Oracle ASM, see "SHARED_POOL_SIZE and Automatic Storage Management".
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	Minimum: the granule size
	Maximum: operating system-dependent
Basic	No

The shared pool contains shared cursors, stored procedures, control structures, and other structures. If SGA_TARGET is not set, then Oracle also allocates parallel execution message buffers from the shared pool. Larger values improve performance in multiuser systems. Smaller values use less memory.

You can monitor utilization of the shared pool by querying the view V\$SGASTAT.

See Also:

- "SGA_TARGET"
- "V\$SGASTAT"
- Oracle Multitenant Administrator's Guide for more information about the initialization parameters that control the memory usage of PDBs
- Oracle Database Performance Tuning Guide for more information on setting this parameter

SHARED_POOL_SIZE and Automatic Storage Management

On a database instance using Oracle Automatic Storage Management (Oracle ASM), additional memory is required to store extent maps. As a general guideline, you can aggregate the values from the following queries to obtain current database storage size that is either already on Oracle ASM or will be stored in Oracle ASM. Then determine the redundancy type that is used (or will be used), and calculate the value for <code>SHARED_POOL_SIZE</code>, using the aggregated value as input.

```
SELECT SUM(BYTES)/(1024*1024*1024) FROM V$DATAFILE;

SELECT SUM(BYTES)/(1024*1024*1024) FROM V$LOGFILE a, V$LOG b

WHERE a.group#=b.group#;

SELECT SUM(BYTES)/(1024*1024*1024) FROM V$TEMPFILE WHERE

status='ONLINE';
```

Additionally, keep the following guidelines in mind:

- For disk groups using external redundancy:
 (Every 100G of space needs 1M of extra shared pool) + 2M
- For disk groups using normal redundancy:
 (Every 50G of space needs 1M of extra shared pool) + 4M
- For disk groups using high redundancy:
 (Every 33G of space needs 1M of extra shared pool) + 6M

2.372 SHARED_SERVER_SESSIONS

SHARED SERVER SESSIONS specifies the total number of shared server sessions to allow.

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	If SHARED_SERVER_SESSIONS is specified, then it should be less than SESSIONS. If SHARED_SERVER_SESSIONS is not specified, then a shared server session may be created as long as there is a free session slot.
Basic	No



Setting this parameter enables you to reserve user sessions for dedicated servers.

See Also:

Oracle Database Concepts for more information on sessions

2.373 SHARED_SERVERS

SHARED_SERVERS specifies the number of server processes that you want to create when an instance is started. If system load decreases, then this minimum number of servers is maintained. Therefore, you should take care not to set SHARED_SERVERS too high at system startup.

Property	Description
Parameter type	Integer
Default value	0, meaning that shared server is not on.
	If you are using shared server architecture or if the DISPATCHERS parameter is set such that the total number of dispatchers is more than 0, then the default value is 1.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	The value of this parameter should be less than MAX_SHARED_SERVERS. If it is greater than or equal to MAX_SHARED_SERVERS, then the number of servers will not be self-tuned but will remain constant, as specified by SHARED_SERVERS.
Basic	Yes

Starting with Oracle Database 12c Release 1 (12.1.0.2), the SHARED_SERVERS parameter can be set on PDBs. However, unlike most other parameters that can be set on a PDB, this parameter can only be used within a PDB to enable or disable use of shared servers for that PDB. Therefore, in a PDB, the DBA can either set SHARED_SERVERS to 0 to disable use of shared servers for that PDB or use ALTER SYSTEM RESET SHARED_SERVERS to re-enable shared servers for the PDB. The configuration of shared servers for the CDB can only be done in the root.

See Also:

- "DISPATCHERS"
- "MAX_SHARED_SERVERS"
- Oracle Database Administrator's Guide for more information on setting this parameter



2.374 SHRD_DUPL_TABLE_REFRESH_RATE

SHRD DUPL TABLE REFRESH RATE displays the duplicated table refresh rate (in seconds).

Property	Description
Parameter type	Integer
Default value	60
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	20 to 2 ³¹ - 1
Basic	No
Oracle RAC	All instances should use the same instance.

You can change the value of this parameter to change the duplicated table refresh rate.

A duplicated table is a table that has the same contents on all shards in a sharded database. Duplicated tables are implemented using materialized views that are refreshed at a certain rate.

See Also:

- Oracle Globally Distributed Database Guide for an introduction to sharding
- Oracle Globally Distributed Database Guide for more information about duplicated tables

2.375 SKIP_UNUSABLE_INDEXES

SKIP_UNUSABLE_INDEXES enables or disables the use and reporting of tables with unusable indexes or index partitions.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

If a SQL statement uses a hint that forces the usage of an unusable index, then this hint takes precedence over initialization parameter settings, including <code>SKIP_UNUSABLE_INDEXES</code>. If the optimizer chooses an unusable index, then an <code>ORA-01502</code> error will result. (See *Oracle Database Administrator's Guide* for more information about using hints.)



Values

true

Disables error reporting of indexes and index partitions marked UNUSABLE. This setting allows all operations (inserts, deletes, updates, and selects) on tables with unusable indexes or index partitions.



If an index is used to enforce a UNIQUE constraint on a table, then allowing insert and update operations on the table might violate the constraint. Therefore, this setting does not disable error reporting for unusable indexes that are unique.

false

Enables error reporting of indexes marked UNUSABLE. This setting does not allow inserts, deletes, and updates on tables with unusable indexes or index partitions.



Oracle Database SQL Language Reference for more information about hints

2.376 SMTP_OUT_SERVER

 ${\tt SMTP_OUT_SERVER}$ specifies the SMTP host and port to which ${\tt UTL_MAIL}$ delivers out-bound Email.

Property	Description
Parameter type	String
Syntax	<pre>SMTP_OUT_SERVER = server_clause [, server_clause]</pre>
Syntax	server_clause::=
	host_name[:port]
Default value	There is no default value.
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Multiple servers may be specified, separated by commas.

If the first server in the list is unavailable, then $\mathtt{UTL}\ \mathtt{MAIL}\ \mathsf{tries}$ the second server, and so on.

If SMTP_OUT_SERVER is not specified, then the SMTP server name defaults to the value of DB_DOMAIN, the port number defaults to 25, and the SMTP domain defaults to the suffix of DB_DOMAIN.





Oracle Database PL/SQL Packages and Types Reference for information on the ${\tt UTL}\ {\tt MAIL}\ package$

2.377 SORT AREA RETAINED SIZE

SORT_AREA_RETAINED_SIZE specifies (in bytes) the maximum amount of the user global area (UGA) memory retained after a sort run completes. The retained size controls the size of the read buffer, which Oracle uses to maintain a portion of the sort in memory. This memory is released back to the UGA, not to the operating system, after the last row is fetched from the sort space.

Property	Description
Parameter type	Integer
Default value	Derived from SORT_AREA_SIZE
Modifiable	ALTER SESSION, ALTER SYSTEM DEFERRED
Modifiable in a PDB	Yes
Range of values	From the value equivalent of two database blocks to the value of SORT_AREA_SIZE
Basic	No

Note:

Oracle does not recommend using the <code>SORT_AREA_RETAINED_SIZE</code> parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting <code>PGA_AGGREGATE_TARGET</code> instead. <code>SORT_AREA_RETAINED_SIZE</code> is retained for backward compatibility.

Oracle may allocate multiple sort spaces of this size for each query. Usually, only one or two sorts occur simultaneously, even for complex queries. In some cases, however, additional concurrent sorts are required, and each sort keeps its own memory area. If the shared server is used, allocation is to the SGA until the value in <code>SORT_AREA_RETAINED_SIZE</code> is reached. The difference between <code>SORT_AREA_RETAINED_SIZE</code> and <code>SORT_AREA_SIZE</code> is allocated to the PGA.

Note:

The default value as reflected in the V\$PARAMETER dynamic performance view is 0. However, if you do not explicitly set this parameter, Oracle actually uses the value of the SORT AREA SIZE parameter.



See Also:

- "SORT_AREA_SIZE"
- Oracle Database Performance Tuning Guide for information on setting the values
 of this parameter and the SORT_AREA_SIZE parameter to tune sort operations
 using shared servers

2.378 SORT AREA SIZE

SORT_AREA_SIZE specifies (in bytes) the maximum amount of memory Oracle will use for a sort.

Property	Description
Parameter type	Integer
Default value	65536
Modifiable	ALTER SESSION, ALTER SYSTEM DEFERRED
Modifiable in a PDB	Yes
Range of values	Minimum: the value equivalent of six database blocks Maximum: operating system-dependent
Basic	No

Note:

Oracle does not recommend using the <code>SORT_AREA_SIZE</code> parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting <code>PGA_AGGREGATE_TARGET</code> instead. <code>SORT_AREA_SIZE</code> is retained for backward compatibility.

After the sort is complete, but before the rows are returned, Oracle releases all of the memory allocated for the sort, except the amount specified by the <code>SORT_AREA_RETAINED_SIZE</code> parameter. After the last row is returned, Oracle releases the remainder of the memory.

Increasing SORT AREA SIZE size improves the efficiency of large sorts.

Each sort in a query can consume memory up to the amount specified by <code>SORT_AREA_SIZE</code>, and there can be multiple sorts in a query. Also, if a query is executed in parallel, each PX server can consume memory up to the amount specified by <code>SORT_AREA_SIZE</code> for each sort it does.

SORT_AREA_SIZE is also used for inserts and updates to bitmap indexes. Setting this value appropriately results in a bitmap segment being updated only once for each DML operation, even if more than one row in that segment changes.

Larger values of SORT_AREA_SIZE permit more sorts to be performed in memory. If more space is required to complete the sort than will fit into the memory provided, then temporary segments on disk are used to hold the intermediate sort runs.



The default is adequate for most OLTP operations. You might want to adjust this parameter for decision support systems, batch jobs, or large <code>CREATE INDEX</code> operations.

See Also:

- Oracle Database Concepts for information on sort areas
- Your operating system-specific Oracle documentation for the default value on your system
- "SORT_AREA_RETAINED_SIZE"

2.379 SPATIAL_VECTOR_ACCELERATION

SPATIAL VECTOR ACCELERATION enables or disables the spatial vector acceleration.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Multiple instances should use the same value

The setting of true for this parameter improves spatial vector query performance.



Oracle Spatial Developer's Guide for more information about this parameter

2.380 SPFILE

The value of this parameter is the name of the current server parameter file (SPFILE) in use.

Property	Description
Parameter type	String
Syntax	SPFILE = spfile_name
Default value	ORACLE_BASE_CONFIG/dbs/spfile.ora
Modifiable	No
Modifiable in a PDB	No
Range of values	Any valid SPFILE
Basic	No

Property	Description
Oracle RAC	Multiple instances must have the same value.

This parameter can be defined in a client side PFILE to indicate the name of the server parameter file to use.

When the default server parameter file is used by the server, the value of SPFILE is internally set by the server.

The SPFILE resides in the <code>ORACLE_BASE_CONFIG/dbs</code> directory; however, users can place it anywhere on their system as long as it is specified in an initialization parameter file.



Oracle Database Administrator's Guide for more information about creating the server parameter file

2.381 SQL_ERROR_MITIGATION

Use SQL_ERROR_MITIGATION to enable or disable automatic error mitigation for SQL statements.

Property	Description
Parameter type	String
Syntax	SQL_ERROR_MITIGATION = { ON OFF }
Default value	ON
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different instances can use different values.

Set this parameter to ON to enable automatic error mitigation for SQL statements; set it to OFF to disable automatic error mitigation for SQL statements.



This parameter is available starting with Oracle Database 23ai.



2.382 SQL_HISTORY_ENABLED

Use SQL HISTORY ENABLED to enable or disable SQL history monitoring.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Different instances can use different values.

To enable SQL history monitoring, set this parameter to true. This feature monitors user-issued SQL statements in each user session on a best effort, depending on memory capacity. DDL, DML, and query statements are monitored; SQL statements issued in the background and recursive SQL statements are excluded. The monitored information is exposed by the VSQL HISTORY dynamic performance view.

To disable SQL history monitoring, set this parameter to false.



This parameter is available starting with Oracle Database 23ai.

See Also:

"V\$SQL_HISTORY"

2.383 SQL TRACE

SQL TRACE enables or disables the SQL trace facility.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No



Setting this parameter to true provides information on tuning that you can use to improve performance.

Note:

Using this initialization parameter to enable the SQL trace facility for the entire instance can have a severe performance impact. Enable the facility for specific sessions using the ALTER SESSION statement. If you must enable the facility on an entire production environment, then you can minimize performance impact by:

- Maintaining at least 25% idle CPU capacity
- Maintaining adequate disk space for the USER DUMP DEST location
- · Striping disk space over sufficient disks

Note:

The SQL_TRACE parameter is deprecated. Oracle recommends that you use the DBMS_MONITOR and DBMS_SESSION packages instead. SQL_TRACE is retained for backward compatibility only.

See Also:

- "USER DUMP DEST"
- Oracle Database PL/SQL Packages and Types Reference for more information about the DBMS MONITOR package
- Oracle Database PL/SQL Packages and Types Reference for more information about the DBMS SESSION package
- Oracle Database Performance Tuning Guide for more information about performance diagnostic tools

2.384 SQL TRANSPILER

SQL_TRANSPILER enables or disables the SQL Transpiler.

Property	Description
Parameter type	String
Syntax	SQL_TRANSPILER = { ON OFF }
Default value	OFF
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Different instances can use different values.



When this parameter is set to ON, the SQL Transpiler feature is enabled and PL/SQL functions are automatically transpiled into SQL expressions whenever possible, without user intervention. When this parameter is set to OFF, the SQL Transpiler feature is disabled.



This parameter is available starting with Oracle Database 23ai.

2.385 SQL92 SECURITY

SQL92_SECURITY specifies whether users must have been granted the SELECT privilege on a table in order to specify certain clauses in a DELETE, INSERT, MERGE, or UPDATE Statement.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	No
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

The SQL standard specifies that security administrators should be able to require that users have the SELECT privilege on a table when executing a DELETE, INSERT, MERGE, or UPDATE statement that references table column values in a RETURNING, SET, or WHERE clause.

Values

true

The user must have the SELECT privilege on the target table in order to reference its columns in:

- The where clause of a delete or update statement
- The right hand side of an assignment in the SET clause of an UPDATE statement.
- The RETURNING clause of a DELETE, INSERT, MERGE, or UPDATE statement
- false
 - A user with the DELETE privilege on the target table of a DELETE statement may reference any column of that target table in the WHERE clause of the DELETE statement.
 - A user with the UPDATE privilege on the target table of an UPDATE statement may
 reference any column of that target table in the WHERE clause or on the right hand side
 of any assignment in the SET clause of the UPDATE statement.
 - A user with the READ or SELECT privilege on the target table of a DELETE, INSERT, MERGE, or UPDATE statement may reference any column of that target table in the RETURNING clause.



2.386 SQLTUNE_CATEGORY

SQLTUNE_CATEGORY specifies the category name for use by sessions to qualify the lookup of SQL profiles during SQL compilation.

Property	Description
Parameter type	String
Syntax	SQLTUNE_CATEGORY = category_name
Default value	DEFAULT
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No



Oracle Database PL/SQL Packages and Types Reference for information on the ${\tt DBMS_SQLTUNE}$ package

2.387 STANDBY_DB_PRESERVE_STATES

STANDBY_DB_PRESERVE_STATES controls whether user sessions, buffers, and other internal states of the instance are retained when a readable physical standby database is converted to a primary database. This parameter is meaningful on a physical standby database that is open in real-time query mode.

Property	Description
Parameter type	String
Syntax	STANDBY_DB_PRESERVE_STATES = { NONE SESSION BUFFER ALL }
Default value	NONE
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	The same value must be used on all instances.

Values

NONE

Nothing is retained. All sessions are disconnected and all buffers are flushed. This is the default value.

SESSION

The role transition waits 10 seconds for active sessions to finish the current statement or top-level PL/SQL call. The sessions that are still active after 10 seconds are terminated.

The sessions that complete their work in this lap of time are retained, along with inactive sessions. If the database is reopened as the primary within 300 seconds, the retained sessions resume their operations as if nothing had happened. If the database (or individual PDB) is not opened in the primary role within 300 seconds, the sessions will be terminated.

BUFFER

All current buffers are retained. Media recovery buffers are converted to current buffers and retained, if possible. When the database is reopened as the primary, the retained buffers are available for use, which may enable queries to run faster during the first few minutes of operation. If the database (or an individual PDB) is not opened in the primary role, the buffers will be flushed.

ALL

This value is equivalent to setting both the SESSION and BUFFER values.



Sessions that have long running queries or are using database links will not be retained regardless of the setting of this parameter.

See Also:

- Oracle Data Guard Concepts and Administration for more information about realtime query mode
- Oracle Data Guard Concepts and Administration for more information about preserving user sessions and buffers when a standby database is converted to a primary database

2.388 STANDBY_FILE_MANAGEMENT

STANDBY FILE MANAGEMENT enables or disables automatic standby file management.

Property	Description
Parameter type	String
Syntax	STANDBY_FILE_MANAGEMENT = { MANUAL AUTO }
Default value	MANUAL
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No

When automatic standby file management is enabled, operating system file additions and deletions on the primary database are replicated on the standby database.

STANDBY FILE MANAGEMENT is only applicable to physical standby databases.

Values

MANUAL

Disables automatic standby file management

AUTO

Enables automatic standby file management

Setting STANDBY_FILE_MANAGEMENT to AUTO causes Oracle to automatically create files on the standby database and, in some cases, overwrite existing files. Care must be taken when setting STANDBY_FILE_MANAGEMENT and DB_FILE_NAME_CONVERT so that existing standby files will not be accidentally overwritten.

If the standby database is on the same system as the primary database, then ensure that the primary and standby systems do not point to the same files.



Oracle Data Guard Concepts and Administration for more information about setting this parameter

2.389 STANDBY_PDB_SOURCE_FILE_DBLINK

STANDBY_PDB_SOURCE_FILE_DBLINK specifies the name of a database link that will be used to try to copy the datafiles from a source PDB to which the database link points.

Property	Description
Parameter type	String
Syntax	STANDBY_PDB_SOURCE_FILE_DBLINK = database-link
Default value	NULL
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	Different values can be set on different instances.

In Oracle Database releases prior to Oracle Database 18c, if a PDB was created by cloning a PDB from the same CDB (a local clone), on a standby the datafiles were automatically copied from the source PDB. However, when the PDB was created as a remote clone, the user was responsible for copying datafiles to the Oracle Managed Files (OMF) location on the standby.

To address that deficiency, the <code>STANDBY_PDB_SOURCE_FILE_DBLINK</code> parameter specifies the name of a database link that will be used to try to copy the datafiles from a source PDB to which the database link points. The file copy is done only if the database link points to the source PDB and the source PDB is open in read only mode. Otherwise, the user is still responsible for copying datafiles to the OMF location on the standby.

This parameter can also be used to try to copy files in cases where the source PDB's files are not present on the standby. For example, the source PDB could have been created with standbys=NONE. In this case, the STANDBY_PDB_SOURCE_FILE_DBLINK parameter can enable the standby to copy files from the primary if there is a database link set up to the primary.



See Also:

"STANDBY_PDB_SOURCE_FILE_DIRECTORY"

2.390 STANDBY_PDB_SOURCE_FILE_DIRECTORY

STANDBY_PDB_SOURCE_FILE_DIRECTORY specifies a directory location on the standby where source datafiles for instantiating the standby PDB may be found.

Property	Description
Parameter type	String
Syntax	STANDBY_PDB_SOURCE_FILE_DIRECTORY = datafiles-directory
Default value	NULL
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	Different values can be set on different instances.

In Oracle Database releases prior to Oracle Database 18c, if a PDB was plugged into a CDB, on a standby, the datafiles were expected to be in PDB's Oracle Managed Files (OMF) directory location. If they were not found there, the user had to copy the datafiles to the OMF location and then restart redo apply on the standby.

To address that deficiency, the <code>STANDBY_PDB_SOURCE_FILE_DIRECTORY</code> specifies a directory location on the standby where source datafiles for instantiating the PDB may be found. If the datafiles are not found there, an attempt will be made to locate them in the OMF location on the standby.

This parameter can also be used to try to copy files in cases where the source PDB's files are not present on the standby. For example, the source PDB could have been created with standbys=NONE. In this case, the STANDBY_PDB_SOURCE_FILE_DIRECTORY parameter can enable the standby to copy files from a location if they are made available.

See Also:

"STANDBY PDB SOURCE FILE DBLINK"

2.391 STAR_TRANSFORMATION_ENABLED

STAR_TRANSFORMATION_ENABLED determines whether a cost-based query transformation will be applied to star queries.

Property	Description
Parameter type	String



Property	Description
Syntax	STAR_TRANSFORMATION_ENABLED = { FALSE TRUE TEMP_DISABLE }
Default value	FALSE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	Yes

Values

FALSE

The transformation will not be applied.

TRUE

The optimizer will consider performing a cost-based query transformation on the star query.

TEMP DISABLE

The optimizer will consider performing a cost-based query transformation on the star query but will not use temporary tables in the star transformation.

See Also:

Oracle Database SQL Tuning Guide for information on enabling star queries

2.392 STATISTICS LEVEL

STATISTICS_LEVEL specifies the level of collection for database and operating system statistics. The Oracle Database collects these statistics for a variety of purposes, including making self-management decisions.

Property	Description
Parameter type	String
Syntax	STATISTICS_LEVEL = { ALL TYPICAL BASIC }
Default value	TYPICAL
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

The default setting of ${\tt TYPICAL}$ ensures collection of all major statistics required for database self-management functionality and provides best overall performance. The default value should be adequate for most environments.

When the STATISTICS_LEVEL parameter is set to ALL, additional statistics are added to the set of statistics collected with the TYPICAL setting. The additional statistics are timed operating system statistics and plan execution statistics.

Setting the STATISTICS_LEVEL parameter to BASIC disables the collection of many of the important statistics required by Oracle Database features and functionality, including:

- All server-generated alerts
- Automatic Database Diagnostic Monitor (ADDM)
- Automatic optimizer statistics collection
- Automatic SGA Memory Management
- Automatic Workload Repository (AWR) Snapshots
- Buffer cache advisory
- Database time distribution statistics (V\$SESS TIME MODEL and V\$SYS TIME MODEL)
- End to End Application Tracing (V\$CLIENT STATS)
- · Monitoring of statistics
- MTTR advisory
- Object Activity Tracking System (OATS)
- Object level statistics
- PGA Target advisory
- Segment level statistics
- Service level statistics
- Shared pool sizing advisory
- Timed statistics

Note:

Oracle strongly recommends that you do not disable these important features and functionality.

When the STATISTICS_LEVEL parameter is modified by ALTER SYSTEM, all advisories or statistics are dynamically turned on or off, depending on the new value of STATISTICS_LEVEL. When modified by ALTER SESSION, the following advisories or statistics are turned on or off in the local session only. Their systemwide state is not changed:

- Timed statistics
- Timed operating system statistics
- Plan execution statistics

The V\$STATISTICS_LEVEL view displays information about the status of the statistics or advisories controlled by the STATISTICS LEVEL parameter. See "V\$STATISTICS_LEVEL".

See Also:

Oracle Database Performance Tuning Guide for more information about this parameter



2.393 STREAMS_POOL_SIZE

The STREAMS POOL SIZE value helps determine the size of the Streams pool.

Property	Description
Parameter type	Big integer
Syntax	STREAMS_POOL_SIZE = integer [K M G]
Default value	0
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Minimum: 0 (values greater than zero are rounded up to the nearest granule size)
	Maximum: operating system-dependent
Basic	No

Oracle's Automatic Shared Memory Management feature manages the size of the Streams pool when the SGA_TARGET initialization parameter is set to a nonzero value. If the STREAMS_POOL_SIZE initialization parameter also is set to a nonzero value, then Automatic Shared Memory Management uses this value as a minimum for the Streams pool.

If SGA_TARGET is set to a nonzero value and STREAMS_POOL_SIZE is not specified or is set to a null value, Automatic Shared Memory Management uses 0 (zero) bytes as a minimum for the Streams pool.

If the <code>STREAMS_POOL_SIZE</code> initialization parameter is set to a nonzero value, and the <code>SGA_TARGET</code> parameter is set to 0 (zero), then the Streams pool size is the value specified by the <code>STREAMS_POOL_SIZE</code> parameter, in bytes.

If both the STREAMS_POOL_SIZE and the SGA_TARGET initialization parameters are set to 0 (zero), then, by default, on the first request for Streams pool memory in a database, an amount of memory equal to 10% of the shared pool is transferred from the buffer cache to the Streams pool. Products and features that use the Streams pool include Oracle GoldenGate, XStream, Oracle Advanced Queuing, and Oracle Data Pump.

The Streams pool is a shared resource, and the amount of memory a process can use from the Streams pool is determined by the application. The capture or apply parameter MAX_SGA_SIZE can be controlled for Oracle GoldenGate or XStream. For Oracle Advanced Queuing, use the procedures in the dbms_aqadm package to control the amount of Streams Pool needed.



See Also:

- Oracle Database XStream Guide for information on configuring the Streams pool for an XStream Out configuration
- Oracle Database XStream Guide for information on configuring the Streams pool for an XStream In configuration
- Oracle Database PL/SQL Packages and Types Reference for more information about the dbms agadm package

2.394 SYSDATE AT DBTIMEZONE

SYSDATE_AT_DBTIMEZONE enables special handling for the date and time value returned in calls to SYSDATE and SYSTIMESTAMP.

Depending on the value of SYSDATE_AT_DBTIMEZONE, you see either the date and time based on the default Autonomous Database time zone, Coordinated Universal Time (UTC), or based on the time zone that you set in your database.

Note:

This parameter is valid only for Oracle Autonomous Database Serverless. On other types of Oracle databases, the value of this parameter is ignored.

See Also:

Using Oracle Autonomous Database Serverless for more information about this parameter

2.395 TABLESPACE ENCRYPTION

TABLESPACE ENCRYPTION specifies the tablespace encryption policy for a database.

Property	Description
Parameter type	String
Syntax	TABLESPACE_ENCRYPTION = { AUTO_ENABLE MANUAL_ENABLE DECRYPT_ONLY }
Default value	Cloud databases: AUTO_ENABLE
	On-premises databases: MANUAL_ENABLE
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	The same value should be specified for all instances.



The Tablespace_encryption parameter is an alternative to the deprecated encrypt_new_tablespaces parameter. Similar to encrypt_new_tablespaces, this parameter allows you to specify whether to encrypt newly created user tablespaces. However, the Tablespace_encryption parameter offers an additional benefit in Data Guard environments. In previous releases, tablespaces in Data Guard environments were required to use the same encryption policy on the primary and standby databases. For example, if all tablespaces were encrypted on the primary database, then all tablespaces were required to be encrypted on the standby database. The Tablespace_encryption parameter eliminates this requirement and allows you to use different tablespace encryption policies on primary and standby databases. See Oracle Database Transparent Data Encryption Guide for more information.

If the behavior specified by the <code>ENCRYPT_NEW_TABLESPACES</code> setting conflicts with the behavior specified by the <code>TABLESPACE_ENCRYPTION</code> setting, then the <code>TABLESPACE_ENCRYPTION</code> behavior takes precedence.

Values:

AUTO ENABLE

Use this setting if you would like to encrypt all tablespaces in the database.

All newly created tablespaces will be encrypted. In the CREATE TABLESPACE statement, if you specify the DECRYPT clause, it will be ignored and the tablespace will be encrypted. If you specify the ENCRYPTION USING algorithm ENCRYPT clause, the specified algorithm will be used to encrypt the tablespace. If you specify the ENCRYPTION ENCRYPT clause and do not specify an algorithm, the tablespace will be encrypted with the algorithm specified by the TABLESPACE ENCRYPTION DEFAULT ALGORITHM parameter.

If an existing tablespace is unencrypted, the database writes a warning to the alert log. See *Oracle Database Transparent Data Encryption Guide* for information about encrypting existing tablespaces.

If you attempt to decrypt an existing encrypted tablespace, an error will occur.

This is the default setting for Cloud databases. Moreover, because all tablespaces must be encrypted in the Cloud, setting this parameter to MANUAL_ENABLE or DECRYPT_ONLY for a Cloud database will be ignored and the Cloud database will behave as if the setting is AUTO ENABLE.

MANUAL ENABLE

Use this setting if you would like to manually control which tablespaces are encrypted.

To encrypt a tablespace, specify the ENCRYPTION ... ENCRYPT clause in the CREATE TABLESPACE statement. If you specify the ENCRYPTION USING algorithm ENCRYPT clause, the specified algorithm will be used to encrypt the tablespace. If you specify the ENCRYPTION ENCRYPT clause and do not specify an algorithm, the tablespace will be encrypted with the algorithm specified by the TABLESPACE ENCRYPTION DEFAULT ALGORITHM parameter.

This is the default setting for on-premises databases. This setting is ignored for Cloud databases.

DECRYPT_ONLY

Use this setting if you do not want any encrypted tablespaces in the database.

All newly created tablespaces must be unencrypted. If you specify the ENCRYPT clause in the CREATE TABLESPACE statement, an error will occur.

If an existing tablespace is encrypted, the database writes a warning to the alert log. See *Oracle Database Transparent Data Encryption Guide* for information about decrypting existing tablespaces.



If you attempt to encrypt an existing unencrypted tablespace, an error will occur.

This setting is valid only for on-premises databases. This setting is ignored for Cloud databases.

See Also:

"ENCRYPT NEW TABLESPACES"

2.396 TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM

TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM specifies the default algorithm the database uses when encrypting a tablespace.

Property	Description
Parameter type	String
Syntax	TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM = { AES128 AES192 AES256 ARIA128 ARIA192 ARIA256 3DES168 }
	Note: 3DES168 must be enclosed in single quotation marks when specified in the ${\tt ALTER}$ SYSTEM command.
Default value	AES256
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value should be specified for all instances.

The value of this initialization parameter determines the algorithm the database uses if an encryption algorithm is not specified when creating an encrypted tablespace or when encrypting an existing tablespace. For example, the value of this parameter takes effect when:

- You create a tablespace with the CREATE TABLESPACE statement and you specify the ENCRYPTION ... ENCRYPT clause, but you do not specify the USING keyword and an algorithm.
- You create a tablespace with the CREATE TABLESPACE statement and you do not specify the ENCRYPTION ... ENCRYPT clause. However, the value of the ENCRYPT_NEW_TABLESPACES initialization parameter instructs the database to encrypt the tablespace.
- You encrypt an existing tablespace with the ALTER TABLESPACE statement, either by specifying the ENCRYPTION OFFLINE ENCRYPT clause or the ENCRYPTION ONLINE ... ENCRYPT clause, and you do not specify the USING keyword and an algorithm.



Note:

GOST 28147-89 has been deprecated by the Russian government, and SEED has been deprecated by the South Korean government. If you need South Korean government-approved TDE cryptography, then use ARIA instead. If you are using GOST 28147-89, then you must decrypt and encrypt with another supported TDE algorithm. The decryption algorithms for GOST 28147-89 and SEED are included in Oracle Database 23ai, but are deprecated, and the GOST encryption algorithm is desupported with Oracle Database 23ai. If you are using GOST or SEED for TDE encryption, then Oracle recommends that you online re-key to another algorithm before upgrading to Oracle Database 23ai. However, with the exception of the HP Itanium platform, the GOST and SEED decryption libraries are available with Oracle Database 23ai, so you can also decrypt after upgrading.

See Also:

- "ENCRYPT NEW TABLESPACES"
- Oracle Database SQL Language Reference for more information about the CREATE TABLESPACE statement
- Oracle Database SQL Language Reference for more information about the ALTER TABLESPACE statement

2.397 TAPE_ASYNCH_IO

TAPE_ASYNCH_IO controls whether I/O to sequential devices (for example, backup or restore of Oracle data to or from tape) is asynchronous—that is, whether parallel server processes can overlap I/O requests with CPU processing during table scans.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No

If your platform supports asynchronous I/O to sequential devices, Oracle recommends that you leave this parameter set to its default. However, if the asynchronous I/O implementation is not stable, you can set <code>TAPE_ASYNCH_IO</code> to <code>false</code> to disable asynchronous I/O. If your platform does not support asynchronous I/O to sequential devices, this parameter has no effect.





Oracle Database VLDB and Partitioning Guide for more information about this parameter

2.398 TDE CONFIGURATION

Use <code>TDE_CONFIGURATION</code> to set the type of keystore that is used for Transparent Data Encryption (TDE) by the root container; united PDBs inherit the value from the root container, isolated PDBs can be set individually.

Before Oracle Database 18c, each PDB stored its separate encryption keys in the CDB's keystore (united mode). Starting with Oracle Database 18c Cloud environments, a PDB can optionally store its encryption keys in a separate keystore (isolated mode), thus allowing protection by a separate keystore password. Starting with Oracle Database 19c, version 19.14, isolated mode is available for all Oracle database deployments, including on-premises databases, legacy deployments, and engineered systems.

The WALLET_ROOT initialization parameter must be set in order for TDE_CONFIGURATION to take effect.

Property	Description
Parameter type	String
Syntax	<pre>TDE_CONFIGURATION = "{ KEYSTORE_CONFIGURATION = value [; CONTAINER = pdb-name] }"</pre>
Syntax	value ::=
	{
	FILE
	OKV
	HSM
	FILE OKV
	FILE HSM
	OKV FILE
	HSM FILE
	}
	Notes:
	 The KEYSTORE_CONFIGURATION value is case-insensitive. For example, you can specify FILE or file.
	 FILE OKV, FILE HSM, OKV FILE, and HSM FILE are values. The vertical bars they contain are not separators in the syntax shown above.
Default value	None
Modifiable	ALTER SYSTEM ¹
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value must be specified on all instances using the ALTER SYSTEM SET TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=value" SCOPE=BOTH SID='*'; statement.

In some cases, when this parameter is set using ALTER SYSTEM SCOPE=SPFILE, the SHOW PARAMETER TDE_CONFIGURATION statement does not show the correct value. However, the value set for TDE_CONFIGURATION can be derived from information shown in the V\$ENCRYPTION_WALLET view.

The following attributes can be specified:

- KEYSTORE_CONFIGURATION attribute. This attribute is required. The value specified with this
 attribute configures the keystore type for the specified PDB. The following values can be
 specified for this attribute:
 - FILE: This value configures a wallet keystore.
 - OKV: This value configures the database to use Oracle Key Vault (OKV) for TDE key management.
 - This value is also used to disable an auto-login OKV configuration and cause any existing <code>cwallet.sso</code> files, containing the credentials to the OKV server as the <code>OKV PASSWORD</code> client secret, to be ignored.
 - HSM: This value configures a hardware security module (HSM).
 - Oracle does not support the use of HSMs for TDE key management. For more information, see My Oracle Support note 2310066.1 "Oracle TDE Support With 3rd Party HSM Vendors" at the following URL: https://support.oracle.com/rs?type=doc&id=2310066.1
 - FILE | OKV: This value configures a reverse migration from OKV to a wallet keystore.
 - FILE | HSM: This value configures a reverse migration from a HSM to a wallet keystore.
 - OKV|FILE: This value configures a migration from a wallet to OKV.
 - This value is also used in an auto-login OKV configuration, because in this configuration a <code>cwallet.sso</code> file, containing the <code>OKV_PASSWORD</code> client secret, must be used by the Oracle server to obtain the credentials to log in to the OKV server.
 - HSM|FILE: This value configures a migration from a wallet to a HSM.
 - Oracle does not support the use of HSMs for TDE key management. For more information, see My Oracle Support note 2310066.1 "Oracle TDE Support With 3rd Party HSM Vendors" at the following URL: https://support.oracle.com/rs?type=doc&id=2310066.1

Some of the KEYSTORE_CONFIGURATION attribute values consist of a single word, for example, the FILE, OKV, and HSM values. The other KEYSTORE_CONFIGURATION attribute values consist of two words separated by the "|" character that is a required part of the value's syntax, for example, the FILE | OKV, FILE | HSM, OKV | FILE, and HSM | FILE values.

• CONTAINER attribute: This optional attribute can be used only when setting the parameter in the CDB\$ROOT of a CDB. The CONTAINER attribute can be specified only when the CDB\$ROOT is in MOUNTED state. With this attribute, you must specify the name of the PDB for which you are setting the parameter. When you specify the CONTAINER attribute, you must use a semicolon ";" as the separation character between the KEYSTORE_CONFIGURATION and CONTAINER attributes.

Examples

The following statement configures a wallet keystore for the open PDB from which the statement is issued:

ALTER SYSTEM SET TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=FILE" SCOPE=BOTH SID='*';



The following statement configures an OKV keystore for the PDB in MOUNTED state from which the statement is issued:

ALTER SYSTEM SET TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=OKV" SCOPE=SPFILE SID='*';

The following statement configures an isolated standby PDB for an auto-open Oracle Key Vault setup; standby PDBs are MOUNTED, so the command must be executed in the standby root CDB:

ALTER SYSTEM SET TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=OKV|FILE; CONTAINER=FINANCIALS" SCOPE=both SID='*';



In an Oracle Data Guard configuration, during keystore migration to OKV or OCI Key Management Service (KMS), or reverse migration to FILE, after the configuration change to <code>OKV|FILE</code> or <code>FILE|OKV</code> respectively, the wallet must be closed and reopened on the standby.

See Also:

- WALLET ROOT
- V\$ENCRYPTION WALLET
- Oracle Database Transparent Data Encryption Guide for information about managing keystores and encryption keys in united mode
- Oracle Database Transparent Data Encryption Guide for information about managing keystores and encryption keys in isolated mode

2.399 TDE_KEY_CACHE

Use TDE_KEY_CACHE to enable or disable sharing of Transparent Data Encryption (TDE) master encryption keys across Oracle processes.



This parameter is available only for databases in Oracle Cloud Infrastructure (OCI) that use the OCI Key Management Service (KMS) for TDE key management.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false



Property	Description
Basic	No
Oracle RAC	The same value must be used on all instances.

Values:

true

Enables sharing of TDE master encryption keys across Oracle processes

false

Disables sharing of TDE master encryption keys across Oracle processes

See Also:

- Oracle Database Transparent Data Encryption Guide for more information about setting the TDE master encryption key in a hardware keystore in united mode
- Oracle Database Transparent Data Encryption Guide for more information about setting the TDE master encryption key in a hardware keystore in isolated mode

2.400 TEMP_UNDO_ENABLED

TEMP_UNDO_ENABLED determines whether transactions within a particular session can have a temporary undo log.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Each session of each instance can have its own value or not set any value at all

The default choice for database transactions has been to have a single undo log per transaction. This parameter, at the session level / system level scope, lets a transaction split its undo log into temporary undo log (for changes on temporary objects) and permanent undo log (for changes on persistent objects).

By splitting the undo stream of a transaction into two streams (temporary and permanent), a database can provide separate storage and retention model for these. This results in overall reduction in the size of undo log and redo log in the database

If database applications make use of temporary objects (using global temporary tables or temporary table transformations), it is advisable to set this parameter's value to true.



When TEMP_UNDO_ENABLED is set to true and the COMPATIBLE initialization parameter is set to 12.0.0, this feature is enabled. The temporary undo feature is enabled for the session in which it is set. Setting it across the system will affect all existing and upcoming sessions. If the value is set in the init.ora file, all upcoming sessions will inherit this value unless overwritten by an explicit ALTER SESSION or ALTER SYSTEM statement. All undo for operations on temporary objects is deemed temporary.

If TEMP_UNDO_ENABLED is not set to true, existing applications that make use of temporary objects run as is without any change.

Once the value of the parameter is set, it cannot be changed for the lifetime of the session. If the session has temporary objects using temporary undo, the parameter cannot be disabled for the session. Similarly, if the session already has temporary objects using regular undo, setting this parameter will have no effect.

This parameter is only applicable for the primary database. For a standby database, this parameter is ignored because temporary undo is enabled by default on the standby database.



Oracle Database Administrator's Guide for information on managing temporary undo

2.401 THREAD

THREAD has been superseded by the INSTANCE_NAME and INSTANCE_NUMBER parameters, and will be made obsolete in a future release of the Oracle Database.

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Modifiable in a PDB	No
Range of values	0 to the maximum number of enabled threads
Basic	No
Oracle RAC	If specified, multiple instances must have different values.



"INSTANCE NAME" and "INSTANCE NUMBER"

2.402 THREADED EXECUTION

THREADED EXECUTION specifies whether to enable the multithreaded Oracle model.



Property	Description
Parameter type	Boolean
Default value	On UNIX and Linux: false
	On Windows: true
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	If specified, all instances must use the same value

THREADED_EXECUTION on UNIX and Linux

Oracle Database 12c introduced the <code>THREADED_EXECUTION</code> initialization parameter for databases running on UNIX and Linux systems. On such databases, the default value for <code>THREADED_EXECUTION</code> is <code>false</code>. Setting this parameter to <code>true</code> enables the multithreaded Oracle model, which allows Oracle processes on UNIX and Linux to run as operating system threads in separate address spaces.

By default, some background processes on UNIX and Linux always use threaded execution; the remaining Oracle processes run as operating system processes. Thus, an "Oracle process" is not always equivalent to an "operating system process."

Note:

When <code>THREADED_EXECUTION</code> is set to <code>true</code> on databases running on UNIX and Linux systems, which enables the multithreaded Oracle model, operating system authentication is not supported. Attempts to connect to the database using operating system authentication (for example, <code>CONNECT / AS SYSDBA</code> or <code>CONNECT /)</code> when this initialization parameter is set to <code>true</code> receive an <code>ORA-01031 "insufficient privileges"</code> error.

The solution to this error is to always use the password when connecting to the database.

Also, when this initialization parameter is set to true on databases running on UNIX and Linux systems, the <code>DEDICATED_THROUGH_BROKER_listener-name=ON</code> parameter should be added to the listener.ora file, where <code>listener-name</code> is the name of the Oracle Net listener and the <code>LOCAL_LISTENER</code> initialization parameter should be set to a TNS name entry corresponding to your instance service. This enables the server to spawn threads when connections to the database are requested through the listener.

THREADED_EXECUTION on Windows

Starting with Oracle Database 21c, the THREADED_EXECUTION initialization parameter is supported for databases running on Microsoft Windows. On such databases, this parameter must remain set to its default value of true; setting this parameter to false on databases running on Windows is not supported. The introduction of this parameter does not change the threading model for Oracle database on Windows. As in previous releases, an Oracle Database server instance on Windows is a service with a single multi-threaded operating system process. However, starting with Oracle Database 21c, you will observe changes to the



V\$PROCESS view, the dedicated connection broker, and the naming of trace files. For details about these changes, refer to *Oracle Database Administrator's Reference for Microsoft Windows*.

Note:

Unlike on UNIX and Linux systems, when <code>THREADED_EXECUTION</code> is set to <code>true</code> on Windows, operating system authentication is supported.

See Also:

- Oracle Database Concepts for more information about multithreaded Oracle
- Table F-1 for more information about background processes that run as threads instead of as operating system processes when multithreaded Oracle is enabled
- Oracle Database Net Services Reference for more information about the DEDICATED THROUGH BROKER listener-name parameter in the listener.ora file
- Oracle Database Net Services Administrator's Guide for an overview of Oracle Net listener

2.403 TIME AT DBTIMEZONE

TIME_AT_DBTIMEZONE specifies whether time-dependent database operations use the time zone of the database host system or the time zone of the PDB.

Some examples of time-dependent database operations are running the SQL functions SYSDATE and SYSTIMESTAMP, querying table columns of type DATE or TIMESTAMP, using Oracle Job scheduler, performing materialized view refresh, and using Oracle Flashback Technology.

Property	Description
Parameter type	String
Syntax	TIME_AT_DBTIMEZONE = { OFF USER_SQL DATABASE }
Default value	OFF
Modifiable	No
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Multiple instances must use the same value.

Values:

- OFF For all time-dependent operations, the database uses the time zone of the database host system that was in effect when the database was started. This is the behavior in releases prior to Oracle Database 23ai.
- USER_SQL The SQL functions SYSDATE and SYSTIMESTAMP return the date and time based on the DBTIMEZONE setting for the PDB. For all other time-dependent operations, the

database uses the time zone of the database host system that was in effect when the database was started.

• DATABASE - For all time-dependent operations, the database uses the time zone specified by the DBTIMEZONE setting for the PDB.



This parameter is available starting with Oracle Database 23ai.

See Also:

- Oracle Database SQL Language Reference for information about the SYSDATE function
- Oracle Database SQL Language Reference for information about the SYSTIMESTAMP function

2.404 TIMED_OS_STATISTICS

TIMED_OS_STATISTICS specifies (in seconds) the interval at which Oracle collects operating system statistics when a request is made from the client to the server or when a request completes.

Property	Description
Parameter type	Integer
Default value	If STATISTICS_LEVEL is set to ALL, then 60
	If STATISTICS_LEVEL is set to BASIC or TYPICAL, then 0
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	Unlimited
Basic	No

On dedicated servers, Oracle collects operating system statistics at user logon and after each subsequent client invocation through the OCI into the Oracle server as a remote procedure call message.

On shared servers, Oracle collects statistics when client calls to Oracle are processed.

A value of zero specifies that operating system statistics are not gathered. To collect statistics, set a value meaningful for your application and site needs.



Note:

Gathering operating system statistics is very expensive. Oracle recommends that you set this parameter in an ALTER SYSTEM statement rather than in the initialization parameter file, and that you reset the value to zero as soon as the needed statistics have been gathered.

See Also:

Oracle Database Performance Tuning Guide for more information about this parameter

2.405 TIMED_STATISTICS

TIMED STATISTICS specifies whether statistics related to time are collected.

Property	Description
Parameter type	Boolean
Default value	If STATISTICS_LEVEL is set to TYPICAL or ALL, then true
	If STATISTICS_LEVEL is set to BASIC, then false
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No

Values

true

The statistics are collected and stored in trace files or displayed in the V\$SESSTATS and V\$SYSSTATS dynamic performance views.

false

The value of all time-related statistics is set to zero. This setting lets Oracle avoid the overhead of requesting the time from the operating system.

Starting with release 11.1.0.7.0, the value of the <code>TIMED_STATISTICS</code> parameter cannot be set to <code>false</code> if the value of <code>STATISTICS</code> LEVEL is set to <code>TYPICAL</code> or <code>ALL</code>.

On some systems with very fast timer access, Oracle might enable timing even if this parameter is set to false. On these systems, setting the parameter to true can sometimes produce more accurate statistics for long-running operations.



See Also:

- Oracle Database SQL Tuning Guide for more information on setting this parameter
- Statistics Descriptions indicates which statistics depend on the setting of this parameter.

2.406 TIMEZONE_VERSION_UPGRADE_ONLINE

TIMEZONE_VERSION_UPGRADE_ONLINE enables you to keep the database running in normal mode while upgrading time zone data using the DBMS DST package.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	true false
Basic	No
Oracle RAC	Multiple instances must have the same value.

In releases prior to Oracle Database 21c, you were required to put the database in <code>UPGRADE</code> mode in order to initiate a time zone data upgrade using the <code>DBMS_DST</code> package. Starting with Oracle Database 21c, you can perform such an upgrade while the database is in normal mode by setting the value of <code>TIMEZONE VERSION UPGRADE ONLINE</code> to <code>true</code>.



Setting the value of this parameter is only one step in a series of steps for upgrading your time zone data. Refer to *Oracle Database Globalization Support Guide* for the complete set of steps before you set this parameter value.

2.407 TRACE ENABLED

TRACE_ENABLED controls tracing of the execution history, or code path, of Oracle. Oracle Support Services uses this information for debugging.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No



Property	Description
Range of values	true false
Basic	No
Oracle RAC	The default value is ${\tt TRUE}.$ Oracle recommends that multiple instances have the same value.

When TRACE_ENABLED is set to true, Oracle records information in specific files when errors occur.

Oracle records this information for all instances, even if only one instance terminates. This allows Oracle to retain diagnostics for an entire cluster.

Although the overhead incurred from this processing is not excessive, you can improve performance by setting <code>TRACE_ENABLED</code> to <code>false</code>. You might do this, for example, to meet highend benchmark requirements. However, if you leave this parameter set to <code>false</code>, you may lose valuable diagnostic information. Therefore, always set <code>TRACE_ENABLED</code> to <code>true</code> to trace system problems and to reduce diagnostic efforts when unexplained instance failures occur.

2.408 TRACEFILE_IDENTIFIER

TRACEFILE_IDENTIFIER specifies a custom identifier that becomes part of the Oracle Trace file name. Such a custom identifier is used to identify a trace file simply from its name and without having to open it or view its contents.

Property	Description
Parameter type	String
Syntax	TRACEFILE_IDENTIFIER = "traceid"
Default value	There is no default value.
Modifiable	ALTER SESSION
Modifiable in a PDB	No
Range of values	Any characters that can occur as part of a file name on the customer platform
Basic	No

Each time this parameter is dynamically modified, the next trace dump will be written to a trace file which has the new parameter value embedded in its name. Trace file continuity information is automatically added to both the old and new trace files to indicate that these trace files belong to the same process.

This parameter can only be used to change the name of the foreground process' trace file; the background processes continue to have their trace files named in the regular format. For foreground processes, the <code>TRACEID</code> column of the <code>V\$PROCESS</code> view contains the current value of the <code>TRACEFILE_IDENTIFIER</code> parameter. When this parameter value is set, the trace file name has the following format:

sid ora pid traceid.trc

In this example, sid is the Oracle instance ID, pid is the process ID, and traceid is the value of the TRACEFILE IDENTIFIER parameter.



See Also:

- Oracle Database SQL Tuning Guide for more information about this parameter
- This parameter is not supported on all operating systems. See your operating system-specific Oracle documentation for more information.

2.409 TRANSACTION_RECOVERY

TRANSACTION_RECOVERY specifies whether transaction recovery is enabled or disabled for a PDB.

Transaction recovery refers to the series of steps that the database performs in the event of a crash (either an instance crash or a transaction crash) to roll back uncommitted transactions.

Property	Description
Parameter type	String
Syntax	TRANSACTION_RECOVERY = { ENABLED DISABLED }
Default value	ENABLED
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Multiple instances must use the same value.

Values:

- ENABLED Transaction recovery is enabled for a PDB.
- DISABLED Transaction recovery is disabled for a PDB.

During normal operation, the value of this parameter is set to ENABLED by the database and this setting should not be modified by a DBA. However, there are two circumstances under which it is permissible for a DBA to modify the value of this parameter:

- A quarantined transaction is a transaction that is isolated and for which transaction recovery will not be attempted while the quarantine on it remains. It requires intervention by the DBA to first resolve the issue that caused the transaction to be quarantined, and then to drop the transaction quarantine so that transaction recovery is attempted again. If transaction recovery becomes disabled on a PDB due to an excessive number of quarantined transactions (a process known as "transaction quarantine escalation"), then the database automatically sets the value of this parameter to DISABLED. When this happens, the DBA should take corrective action for the quarantined transactions, then restart transaction recovery for the PDB by setting the value of this parameter to ENABLED. Refer to Oracle Database Administrator's Guide to learn about this entire process before you set this parameter.
- While working to resolve an issue, Oracle Support may instruct a DBA to temporarily
 disable transaction recovery by setting this parameter to DISABLED. After the issue has
 been resolved, this parameter should ultimately be set back to ENABLED so that normal
 database operation can resume.



Caution:

You should never set this parameter to <code>DISABLED</code> unless specifically directed to do so by Oracle Support. Setting this parameter to <code>DISABLED</code> could lead to severe problems



This parameter is available starting with Oracle Database 23ai.

2.410 TRANSACTIONS

TRANSACTIONS specifies how many rollback segments to online when UNDO_MANAGEMENT = MANUAL.

Property	Description
Parameter type	Integer
Default value	Derived: (1.1 * SESSIONS)
Modifiable	No
Modifiable in a PDB	No
Range of values	4 to 2 ³¹ - 1
Basic	No
Oracle RAC	Multiple instances can have different values.

The maximum number of concurrent transactions is now restricted by undo tablespace size (UNDO_MANAGEMENT = AUTO) or the number of online rollback segments (UNDO_MANAGEMENT = MANUAL).

2.411 TRANSACTIONS_PER_ROLLBACK_SEGMENT

TRANSACTIONS_PER_ROLLBACK_SEGMENT specifies the number of concurrent transactions you expect each rollback segment to have to handle.

Property	Description
Parameter type	Integer
Default value	5
Modifiable	No
Modifiable in a PDB	No
Range of values	1 to operating system-dependent
Basic	No
Oracle RAC	Multiple instances can have different values.

The minimum number of rollback segments acquired at startup is TRANSACTIONS divided by the value for this parameter. For example, if TRANSACTIONS is 101 and this parameter is 10, then

the minimum number of rollback segments acquired would be the ratio 101/10, rounded up to 11.

You can acquire more rollback segments by naming them in the parameter ${\tt ROLLBACK}\ {\tt SEGMENTS}.$



Your operating system-specific Oracle documentation for the range of values for this parameter.

2.412 TRUE_CACHE

Use TRUE CACHE to specify that an instance is a True Cache.

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Modifiable in a PDB	No
Range of values	true false
Basic	No

Set this parameter to true on a True Cache. Otherwise, leave this parameter set to false.



This parameter is available starting with Oracle Database 23ai.

See Also:

Oracle True Cache User's Guide for more information about this parameter

2.413 TXN_PRIORITY

TXN PRIORITY specifies the priority for all user transactions in a session.

Property	Description
Parameter type	String
Syntax	TXN_PRIORITY = { HIGH MEDIUM LOW }
Default value	HIGH
Modifiable	ALTER SESSION



Property	Description
Modifiable in a PDB	No
Basic	No

The Priority Transactions feature automatically rolls back low-priority transactions that are blocking higher priority transactions from obtaining row locks.

You can use the TXN_PRIORITY parameter to assign a priority (HIGH, MEDIUM, or LOW) to all transactions in a particular user session. Priority Transactions then operates as follows for the three priorities:

- HIGH Priority Transactions never rolls back a HIGH priority transaction in order to release a row lock for another transaction. This is the default.
- MEDIUM Priority Transactions will roll back a MEDIUM priority transaction if it is holding a row lock required by a HIGH priority transaction, and the HIGH priority transaction has been waiting for the row lock longer than the time specified by the PRIORITY TXNS HIGH WAIT TARGET parameter.
- LOW Priority Transactions will roll back a LOW priority transaction if it is holding a row lock required by a HIGH or MEDIUM priority transaction, and the HIGH or MEDIUM priority transaction has been waiting for the row lock longer than the time specified by the PRIORITY_TXNS_HIGH_WAIT_TARGET or PRIORITY_TXNS_MEDIUM_WAIT_TARGET parameter, respectively.

If a LOW priority transaction is blocked waiting for a row lock, Priority Transactions will not roll back the transaction holding the row lock, regardless of its priority.

You can view the priority and maximum wait time for a transaction by querying the TXN PRIORITY AND PRIORITY TXNS WAIT TARGET columns in the V\$TRANSACTION view.

Note:

This parameter is available starting with Oracle Database 23ai.

See Also:

- "PRIORITY_TXNS_HIGH_WAIT_TARGET"
- "PRIORITY TXNS MEDIUM WAIT TARGET"
- "PRIORITY_TXNS_MODE"
- "V\$TRANSACTION"
- Oracle Database Administrator's Guide for more information about Priority Transactions

2.414 UNDO_MANAGEMENT

UNDO MANAGEMENT specifies which undo space management mode the system should use.

Property	Description
Parameter type	String
Syntax	UNDO_MANAGEMENT = { MANUAL AUTO }
Default value	AUTO
Modifiable	No
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	Multiple instances must have the same value.

When <code>UNDO_MANAGEMENT</code> is set to <code>AUTO</code>, the instance starts in automatic undo management mode. In manual undo management mode, undo space is allocated externally as rollback segments.



In a CDB, the <code>UNDO_MANAGEMENT</code> initialization parameter must be set to <code>AUTO</code>, and an undo tablespace is required to be created to manage the undo data.

See Also:

- Oracle Database Administrator's Guide for more information about this parameter
- Oracle Database Administrator's Guide for more information about managing undo data

2.415 UNDO_RETENTION

UNDO RETENTION specifies (in seconds) the low threshold value of undo retention.

Property	Description
Parameter type	Integer
Default value	900
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	0 to 2 ³¹ - 1
Basic	No
Oracle RAC	Oracle recommends that multiple instances have the same value.

For both AUTOEXTEND undo tablespaces and fixed-size undo tablespaces, the system retains undo for at least the time specified in this parameter, and automatically tunes the undo retention period to satisfy the undo requirements of the queries.



Automatic tuning of undo retention is not supported for LOBs. The RETENTION value for LOB columns is set to the value of the UNDO RETENTION parameter.

The <code>UNDO_RETENTION</code> parameter is honored only if the current undo tablespace has enough space. If an active transaction requires undo space and the undo tablespace does not have available space, then the system starts reusing unexpired undo space. This action can potentially cause some queries to fail with a "snapshot too old" message.

The amount of time for which undo is retained for the Oracle Database for the current undo tablespace can be obtained by querying the <code>TUNED_UNDORETENTION</code> column of the <code>V\$UNDOSTAT</code> dynamic performance view.

Recommendations for Modifying UNDO_RETENTION

Oracle generally recommends that you leave UNDO_RETENTION set to its default value. Modifying this parameter is recommended only in the following situations:

- You may want to increase the value of UNDO_RETENTION when using flashback features, such as Oracle Flashback Query, which require undo to be retained for longer than the longest running query in the system.
- In Oracle Active Data Guard environments, you may want to increase the value of UNDO_RETENTION on the primary instance in order to accommodate undo retention requirements on the standby instances. This allows the primary instance to retain undo for a longer period of time to serve queries on the standby instances. For more information, see Oracle Database Administrator's Guide and Oracle Data Guard Concepts and Administration.

Modifying UNDO_RETENTION in a CDB

You can set the value of <code>UNDO_RETENTION</code> in the CDB root (<code>CDB\$ROOT</code>) and in individual PDBs. The CDB root and PDBs can have the same value or different values. Note that you can modify this parameter in a PDB only if the PDB is in local undo mode; you cannot modify this parameter if the PDB is in shared undo mode.

Starting with Oracle Database release 19c, version 19.9, the value of <code>UNDO_RETENTION</code> is not inherited in a CDB. Therefore, if you change the value for <code>UNDO_RETENTION</code> in the CDB root, the PDBs do not inherit the new value. If you want to change the value for <code>UNDO_RETENTION</code> at the PDB level, you must explicitly do so. For example, if your CDB contains two PDBs (PDB1 and PDB2), you can set <code>UNDO_RETENTION</code> to 2000 in the CDB root and both PDBs as follows:

```
ALTER SESSION SET CONTAINER = CDB$ROOT;

ALTER SYSTEM SET UNDO_RETENTION=2000 SCOPE=BOTH;

ALTER SESSION SET CONTAINER = PDB1;

ALTER SYSTEM SET UNDO_RETENTION=2000 SCOPE=BOTH;

ALTER SESSION SET CONTAINER = PDB2;

ALTER SYSTEM SET UNDO_RETENTION=2000 SCOPE=BOTH;
```



See Also:

- Oracle Database SQL Language Reference for information about creating undo tablespaces
- Oracle Database Administrator's Guide for information about managing undo data

2.416 UNDO_TABLESPACE

UNDO_TABLESPACE specifies the undo tablespace to be used when an instance starts. If this parameter is specified when the instance is in manual undo management mode, then an error will occur and startup will fail.

Property	Description
Parameter type	String
Syntax	UNDO_TABLESPACE = undoname
Default value	The first available undo tablespace in the database.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	Legal name of an existing undo tablespace
Basic	Yes
Oracle RAC	Each instance must have a unique value for this parameter, when it is set.

If the <code>UNDO_TABLESPACE</code> parameter is omitted, the first available undo tablespace in the database is chosen. If no undo tablespace is available, the instance will start without an undo tablespace. In such cases, user transactions will be executed using the <code>SYSTEM</code> rollback segment. You should avoid running in this mode under normal circumstances.

You can replace an undo tablespace with another undo tablespace while the instance is running.



When you update this parameter on the primary database in an Oracle Data Guard configuration, you must also update it on all the physical standby databases in the configuration. This ensures that the standby databases can find the undo tablespace when they become the primary database.

See Also:

Oracle Database SQL Language Reference for information about creating undo tablespaces



2.417 UNIFIED_AUDIT_COMMON_SYSTEMLOG

 ${\tt UNIFIED_AUDIT_COMMON_SYSTEMLOG}\ specifies\ whether\ key\ fields\ of\ unified\ audit\ records\ generated\ due\ to\ common\ audit\ policies\ will\ be\ written\ to\ the\ SYSLOG\ utility.$



This initialization parameter is supported only on UNIX platforms.

Property	Description
Parameter type	String
Syntax	<pre>UNIFIED_AUDIT_COMMON_SYSTEMLOG = 'facility_clause.priority_clause'</pre>
	facility_clause::=
	{ USER LOCAL[0 1 2 3 4 5 6 7] }
	<pre>priority_clause::=</pre>
	{ NOTICE INFO DEBUG WARNING ERR CRIT ALERT EMERG }
Default value	None
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	The same value must be used on all instances.

When this parameter is set, key fields of unified audit records generated due to common audit policies are written to SYSLOG. These fields uniquely identify the detailed unified audit records in the <code>UNIFIED_AUDIT_TRAIL</code> view. Only a subset of unified audit record fields are written to ensure that the audit record entries do not exceed the maximum allowed size for a SYSLOG entry (typically 1024 bytes).

Do not set this parameter if you do not want key fields of unified audit records generated due to common audit policies written to SYSLOG.

This parameter differs from the <code>UNIFIED_AUDIT_SYSTEMLOG</code> parameter in that it is set at the CDB level and enables all unified audit records from common unified audit policies to be consolidated into a single destination, whereas <code>UNIFIED_AUDIT_SYSTEMLOG</code> is set at the PDB level and enables the logging of unified audit records on a per-PDB basis.

See Also:

- "UNIFIED AUDIT SYSTEMLOG"
- "UNIFIED_AUDIT_TRAIL"
- Oracle Database Security Guide for a table that maps the names given to the unified audit records fields that are written to SYSLOG to the corresponding column names in the UNIFIED AUDIT TRAIL view

2.418 UNIFIED_AUDIT_SYSTEMLOG

UNIFIED_AUDIT_SYSTEMLOG specifies whether key fields of unified audit records will be written to the SYSLOG utility (on UNIX platforms) or to the Windows Event Viewer (on Windows). In a CDB, this parameter is a per-PDB static initialization parameter.

Property	Description
Parameter type	String for UNIX platforms, Boolean for Windows
Syntax	On UNIX:
	<pre>UNIFIED_AUDIT_SYSTEMLOG = 'facility_clause.priority_clause'</pre>
	<pre>facility_clause::=</pre>
	{ USER LOCAL[0 1 2 3 4 5 6 7] }
	<pre>priority_clause::=</pre>
	{ NOTICE INFO DEBUG WARNING ERR CRIT ALERT EMERG }
	On Windows:
	UNIFIED_AUDIT_SYSTEMLOG = { FALSE TRUE }
Default value	No default on UNIX platforms
	FALSE on Windows
Modifiable	No
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value must be used on all instances.

When this parameter is set on UNIX, key fields of unified audit records are written to SYSLOG. When this parameter is set on Windows, key fields of unified audit records are written to the Windows Event Viewer.

Do not set this parameter (or set it to FALSE on Windows) if you do not want key fields of unified audit records written to SYSLOG or the Windows Event Viewer.

When UNIFIED_AUDIT_SYSTEMLOG is enabled, the key fields of the unified audit records that are written to SYSLOG or Windows Event Viewer uniquely identify the detailed unified audit records in the UNIFIED_AUDIT_TRAIL view. Only a subset of the unified audit record fields are written to ensure that the audit record entries do not exceed the maximum allowed size for a SYSLOG entry (typically 1024 bytes).

This parameter differs from the <code>UNIFIED_AUDIT_COMMON_SYSTEMLOG</code> parameter in that it is set at the PDB level and enables the logging of unified audit records on a per-PDB basis, whereas <code>UNIFIED_AUDIT_COMMON_SYSTEMLOG</code> is set at the CDB level and enables all unified audit records from common unified audit policies to be consolidated into a single destination.

See Also:

- "UNIFIED_AUDIT_COMMON_SYSTEMLOG"
- "UNIFIED AUDIT TRAIL"
- Oracle Database Security Guide for a table that maps the names given to the
 unified audit records fields that are written to SYSLOG and the Windows Event
 Viewer to the corresponding column names in the UNIFIED_AUDIT_TRAIL view

2.419 UNIFORM_LOG_TIMESTAMP_FORMAT

UNIFORM_LOG_TIMESTAMP_FORMAT specifies that a uniform timestamp format be used in Oracle Database trace (.trc) files and log files (such as the alert log).

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	true false
Basic	No
Oracle RAC	Multiple instances should use the same value.

When the value of UNIFORM_LOG_TIMESTAMP_FORMAT is TRUE, the format used for timestamps in trace files is standardized on universal time with millisecond precision. For example:

2012-09-26 00:16:47.154

When the value of UNIFORM_LOG_TIMESTAMP_FORMAT is FALSE, trace files include a mix of timestamps using different precisions, with some timestamps showing local time and other timestamps showing universal time.

2.420 USE_DEDICATED_BROKER

USE DEDICATED BROKER determines how dedicated servers are spawned.



Starting with Oracle Database 23ai, the type for this parameter is changed from Boolean to String, the NONE and CDB values are introduced, and the default value is changed from FALSE to NONE. However, there is no change to the functionality of this parameter, because the NONE value is equivalent to FALSE, and the CDB value is equivalent to TRUE.



Property	Description
Parameter type	String
Syntax	<pre>USE_DEDICATED_BROKER = { NONE FALSE CDB TRUE }</pre>
Default value	NONE unless multithreaded Oracle is enabled ¹
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Basic	No
Oracle RAC	If specified, all instances must use the same value

¹ Multithreaded Oracle is enabled by setting the THREADED EXECUTION initialization parameter to TRUE.

When this parameter is set to NONE or FALSE, the listener spawns a dedicated server directly.

When this parameter is set to CDB or TRUE, the listener hands the connection to a dedicated connection broker that spawns the dedicated server. Unlike the listener, the dedicated connection broker is a database process, and hence it can implement policies leveraging database information before the spawn.

To enable the dedicated connection broker using the <code>USE_DEDICATED_BROKER</code> initialization parameter, you must:

- 1. Set the use dedicated broker initialization parameter to true.
- 2. Set the DEDICATED THROUGH BROKER listener-name parameter to on in the listener.ora file.

Performing these two steps configures the Connection Broker Process (Nnnn).



When multithreaded Oracle is enabled, the dedicated connection broker process is also enabled and used.

Configuration of the brokers is controlled by the CONNECTION BROKERS initialization parameter.

See Also:

- "CONNECTION_BROKERS" for more information about configuring brokers
- Table F-1 for more information about the Connection Broker Process (Nnnn)
- "THREADED_EXECUTION" for more information about enabling multithreaded Oracle
- Oracle Database Net Services Reference for more information about the DEDICATED_THROUGH_BROKER_listener-name parameter in the listener.ora file



2.421 USE_LARGE_PAGES

USE LARGE PAGES is used to manage the database's use of large pages for SGA memory.



This parameter is applicable only on the Linux operating system. However, setting this parameter to FALSE can cause performance degradation on any platform.

The exception is when <code>USE_LARGE_PAGES</code> is set to <code>FALSE</code> automatically in an Oracle ASM instance because <code>MEMORY_TARGET</code> is enabled (set to a non-zero value). In this case, the <code>FALSE</code> setting does not cause performance degradation.

Property	Description
Parameter type	String
Syntax	USE_LARGE_PAGES = { TRUE FALSE AUTO ONLY AUTO_ONLY }
Default value	ONLY for Exadata Database Service in the cloud, Base Database Service in the cloud, and databases created using Exadata automation tools such as Oracle Exadata Deployment Assistant (OEDA) and OEDA Command Line Utility (OEDACLI) AUTO_ONLY for all other Exadata systems TRUE for all other systems
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	Multiple instances can use different values

This parameter does not affect process-private memory allocations.

Values

These values can be specified for USE LARGE PAGES:

TRIIF

Specifies that the instance can use large pages if large pages are configured on the system.

In Oracle Database 11g Release 2 (11.2.0.2), if there are not enough large pages configured on the system, then regular sized pages will be used to allocate SGA memory. This can cause the free large pages to go unused, and the operating system can allocate a huge amount of memory to create page tables to map SGA into physical pages for the Oracle processes. This may lead to ORA-04030 errors and severe performance degradation on an instance.

In Oracle Database 11*g* Release 2 (11.2.0.3) and later releases, Oracle allocates as much of the SGA as it can in large pages, and if it runs out, it will allocate the rest of the SGA using regular sized pages. This can cause the instance to create additional shared memory segments for the SGA, but the total SGA size will be unchanged. In this supported mixed page mode allocation, the database will exhaust the available large pages before switching to regular sized pages.



FALSE

Specifies that the instance will not use large pages. This setting is not recommended because it can cause severe performance degradation for the instance. Although the USE_LARGE_PAGES initialization parameter applies only to Linux systems, setting this parameter to FALSE can cause performance degradation on any platform.

• AUTO

Specifies that, during startup, the instance will calculate and request the number of large pages it requires. If the operating system cannot fulfill this request, then the instance will start with a combination of large and regular pages.

ONLY

Specifies that the instance will fail to start if large pages cannot be used for the entire SGA memory. Oracle recommends this setting for consistent performance. This setting is the default setting for Exadata Database Service in the cloud and Base Database Service in the cloud.

When USE_LARGE_PAGES is set to ONLY, you must not explicitly set values for MEMORY_TARGET and MEMORY_MAX_TARGET in the initialization parameter file. Doing so will prevent the instance from starting.

AUTO_ONLY

Specifies that, during startup, the instance will calculate and request the number of large pages it requires. If the operating system can fulfill this request, then the instance will start successfully. If the operating system cannot fulfill this request, then the instance will fail to start. This ensures that no instances will run with under-provisioned large pages. This setting is available starting with Oracle Database 19c and it is the default setting for onpremises Exadata systems.

See Also:

- "MEMORY_TARGET"
- "MEMORY_MAX_TARGET"

2.422 USER_DUMP_DEST

USER_DUMP_DEST specifies the pathname for a directory where the server will write debugging trace files on behalf of a user process.

Property	Description
Parameter type	String
Syntax	<pre>USER_DUMP_DEST = { pathname directory }</pre>
Default value	Operating system-dependent
Modifiable	ALTER SYSTEM
Modifiable in a PDB	No
Range of values	Any valid local path, directory, or disk
Basic	No



Note:

The USER DUMP DEST initialization parameter is deprecated.

For example, this directory might be set as follows:

On MS-DOS: C:\ORACLE\UTRC

On UNIX: /oracle/utrc

On VMS: DISK\$UR3: [ORACLE.UTRC]

Note:

This parameter is ignored by the diagnosability infrastructure introduced in Oracle Database 11g Release 1 (11.1), which places trace and core files in a location controlled by the <code>DIAGNOSTIC</code> <code>DEST</code> initialization parameter.

See Also:

- Oracle Database SQL Tuning Guide for more information about the use of trace files
- Your operating system-specific Oracle documentation for the range of values

2.423 WALLET ROOT

WALLET_ROOT specifies the path to the root of a directory tree containing a subdirectory for each pluggable database (PDB).

Property	Description
Parameter type	String
Syntax	WALLET_ROOT = wallet-root-directory-path
Default value	There is no default value.
Modifiable	No
Modifiable in a PDB	No
Basic	No
Oracle RAC	Multiple instances must have the same value.

The name of the various wallet files is always the same, regardless of the component they are associated with. The wallets for each component are stored under each PDB GUID directory within the WALLET_ROOT directory structure in a directory whose name is based on the component name. For example, for the TDE component, the subdirectory name is tide.

The TDE_CONFIGURATION initialization parameter cannot be set unless the WALLET_ROOT parameter is also set.

For example, the contents of the directory at the location specified by the <code>WALLET_ROOT</code> initialization parameter could look as follows, where <code>wallet-root</code> is the directory specified by the <code>WALLET_ROOT</code> parameter:

```
wallet-root/eus/ewallet.p12
wallet-root/tde/ewallet.p12
wallet-root/tde/ewallet_2016120918333644.p12
wallet-root/tde_seps/cwallet.sso
wallet-root/tde_seps/cwallet.sso
wallet-root/xdb_wallet/ewallet.p12
wallet-root/3FD1C95B48205D0FE053C5A0E40AEF8C/tde/ewallet.p12
wallet-root/3FD1C95B48205D0FE053C5A0E40AEF8C/tde/ewallet_2016110918331622.p12
wallet-root/3FD1C95B48205D0FE053C5A0E40AEF8C/tde/ewallet_2016110918332363.p12
wallet-root/3FD1C95B48205D0FE053C5A0E40AEF8C/tde_seps/cwallet.sso
wallet-root/3FD1C95B48205D0FE053C5A0E40AEF8C/tls/cwallet.sso
wallet-root/3FD1C95B48205D0FE053C5A0E40AEF8C/tls/ewallet.p12
```

When the WALLET_ROOT parameter is set, you can omit the path from some ADMINISTER KEY MANAGEMENT commands.

The <code>WALLET_ROOT</code> value can include references to environment variables. The following example shows how to use <code>WALLET_ROOT</code> when multiple TDE-enabled databases are installed into the same <code>ORACLE_HOME</code>:

```
WALLET_ROOT=/etc/ORACLE/KEYSTORES/$ORACLE_SID
```

The <code>ORACLE_SID</code> environment variable (or the <code>DB_UNIQUE_NAME</code> environment variable for Oracle RAC) makes sure that each database that is installed into the same <code>ORACLE_HOME</code> has its own set of wallets and TDE keys. This sets the root of the wallet directory hierarchy to the directory specified by <code>wallet-root-directory-path</code>.



Note:

The normalized length of the wallet-root-directory-name that is specified with the WALLET_ROOT parameter cannot exceed 255 characters, otherwise one of the following sets of error messages is displayed:

```
ORA-46693: The WALLET_ROOT location is missing or invalid.
ORA-32021: parameter value longer than 255 characters
ORA-01078: failure in processing system parameters

ORA-46693: The WALLET_ROOT location is missing or invalid.
ORA-07204: sltln: name translation failed due to lack of output buffer space.
ORA-01078: failure in processing system parameters
```

The normalized length includes the length of expanded environment variables specified with the \mathtt{WALLET}_{ROOT} parameter. The values of the environment variables of the user who starts the instance are used in the normalization of the \mathtt{WALLET}_{ROOT} parameter.

The SHOW PARAMETER WALLET_ROOT command always displays the normalized value (with all the environment variables expanded).

For non-ASM file systems, the PDB GUID-extended paths for the TDE component are created automatically under the directory specified by the WALLET_ROOT parameter when any Transparent Data Encryption (TDE) wallet is created for a PDB.

Enabling Automatic Creation of Directories Under WALLET ROOT

By using the specific configuration of WALLET_ROOT described in each of the following subsections, Oracle Database can be configured to automatically create the necessary *pdb-guid* and *component name* directories under the WALLET_ROOT directory path. Other settings of WALLET_ROOT are allowed, but would not result in the automatic creation of the necessary subdirectories by the ASM OMF layer.

Required setting to enable auto-directory creation for a database not using Oracle ASM

For a database not using Oracle ASM filesystems, the \mathtt{WALLET}_{ROOT} parameter needs to be set as follows:

```
WALLET ROOT=wallet-root-directory-path
```

This sets the root of the wallet directory hierarchy to the directory specified by wallet-root-directory-path. For example:

```
/etc/ORACLE/KEYSTORES/FINANCE
```

When this is done, Oracle Database automatically creates the directory for the TDE wallet of a CDB\$ROOT at the following location:

```
/etc/ORACLE/KEYSTORES/FINANCE/tde
```

The directories, that Oracle Database automatically creates for holding the TDE wallets of isolated PDBs, will include the *pdb-guid*. For example:



/etc/ORACLE/KEYSTORES/FINANCE/3FD1C95B48205D0FE053C5A0E40AEF8C/tde

Required setting to enable auto-directory creation for a database using Oracle ASM with Oracle Managed Files

For a database using ASM with OMF, the WALLET_ROOT parameter needs to begin with a plus sign followed by a disk group name and the value of the DB_UNIQUE_NAME initialization parameter. In the example below, DATA is the name of a disk group and FINRAC is the value of the DB_UNIQUE_NAME initialization parameter:

WALLET ROOT=+DATA/FINRAC

When this is done, Oracle Database automatically creates the necessary directory within the ASM filesystem at the following location when the ADMINISTER KEY MANAGEMENT CREATE KEYSTORE command is run:

+DATA/FINRAC/tde

For isolated PDBs, the directories that Oracle Database automatically creates for holding the TDE wallets of PDBs will include the *pdb-guid*. For example:

+DATA/FINRAC/3FD1C95B48205D0FE053C5A0E40AEF8C/tde

Required setting to enable auto-directory creation for RAC-enabled databases

For a RAC-enabled database, only shared TDE wallets are supported (as opposed to individual TDE-wallets per RAC instance). WALLET_ROOT can either point to an ASM disk group, or a directory in ACFS. If the WALLET_ROOT parameter points to +diskgroup/dbname, then the /tde sub-directory is automatically created when issuing an ADMINISTER KEY MANAGEMENT CREATE KEYSTORE command. For example, when WALLET_ROOT is set to +DATA/FINANCE, the directory +DATA/FINANCE/tde is automatically created. This guarantees that when multiple databases are installed, their TDE-wallets are kept separate.

See Also:

"TDE_CONFIGURATION"

2.424 WORKAREA SIZE POLICY

WORKAREA_SIZE_POLICY specifies the policy for sizing work areas. This parameter controls the mode in which working areas are tuned.

Property	Description
Parameter type	String
Syntax	WORKAREA_SIZE_POLICY = { AUTO MANUAL }
Default value	AUTO
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No



Values

You can specify the following values for WORKAREA SIZE POLICY:

• AUTO

When AUTO is specified, work areas used by memory-intensive operators are sized automatically, based on the PGA memory used by the system, the target PGA memory set in PGA AGGREGATE TARGET, and the requirement of each individual operator.

MANUAL

When MANUAL is specified, the sizing of work areas is manual and based on the values of the *_AREA_SIZE parameter corresponding to the operation (for example, a sort uses SORT_AREA_SIZE). Specifying MANUAL may result in sub-optimal performance and poor PGA memory utilization.



Oracle Database Performance Tuning Guide for additional information on setting this parameter

2.425 XML CLIENT SIDE DECODING

Use XML CLIENT SIDE DECODING to enable or disable XML client-side decoding.

Property	Description
Parameter type	String
Syntax	<pre>XML_CLIENT_SIDE_DECODING = { TRUE FALSE }</pre>
Default value	TRUE
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Values:

- TRUE Enable XML client-side decoding. This is the default setting.
- FALSE Disable XML client-side decoding. This setting may be beneficial if you are using CSX binary XML storage and have millions or billions of distinct tokens with no XML namespaces in their token tables. Setting this parameter to FALSE may improve the performance of queries on those no-namespace tokens.



This parameter is available starting with Oracle Database 23ai.



2.426 XML_DB_EVENTS

XML DB EVENTS enables or disables XML DB events.

Property	Description
Parameter type	String
Syntax	<pre>XML_DB_EVENTS = { enable disable }</pre>
Default value	enable
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No

Changing this parameter through an ALTER SESSION statement affects only the current session. Only users with the XDBADMIN role are allowed to change this parameter in a session.

Changing this parameter through an ALTER SYSTEM statement will make a systemwide change of this parameter; however, the change is only registered by new sessions started after the change. Currently running sessions need to be restarted to pick up the new change. Users must have the ALTER SYSTEM privilege to make a systemwide change.

See Also:

Oracle XML DB Developer's Guide for more information about this parameter

2.427 XML PARAMS

XML PARAMS lets you adjust the behavior for XMLType data.

Property	Description
Parameter type	String
Syntax	<pre>XML_PARAMS = "parameter=value [, parameter=value]"</pre>
Default value	None
Modifiable	ALTER SESSION, ALTER SYSTEM
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	If specified, Oracle recommends that multiple instances have the same value.

You can specify one of the following for parameter=value:

xml default storage tbx=true

Sets the default $\mathtt{XMLType}$ storage to Transportable Binary XML (TBX) when no storage type is explicitly specified. Starting with Oracle Database 23ai, this is the default and recommended storage type for all $\mathtt{XMLType}$ data, even if this parameter is not set. Note that Transportable Binary XML must be stored as Securefiles LOBs.



xml_default_storage_tbx=false

Sets the default XMLType storage to Not-Transportable Binary XML (CSX) when no storage type is explicitly specified. This setting can be used as a workaround for applications that still rely on the CSX storage type. Not-Transportable Binary XML can be stored as Basicfiles LOBs or Securefiles LOBs.

Example:

ALTER SYSTEM SET XML_PARAMS = "xml_default_storage_tbx=true";



This parameter is available starting with Oracle Database 23ai.



Part II

Static Data Dictionary Views

This part describes data dictionary tables and views. These tables and views are called **static**, because they change only when a change is made to the data dictionary (for example, when a new table is created or when a user is granted new privileges).

This part contains the following chapters:

- Static Data Dictionary Views: ALL_ALL_TABLES to ALL_OUTLINES
- Static Data Dictionary Views: ALL_PART_COL_STATISTICS to DATABASE PROPERTIES
- Static Data Dictionary Views: DBA_2PC_NEIGHBORS to DBA_HIST_JAVA_POOL_ADVICE
- Static Data Dictionary Views: DBA_HIST_LATCH to DBA_STORED_SETTINGS
- Static Data Dictionary Views: DBA STREAMS ADD COLUMN to USER ZONEMAPS

Note:

Oracle also maintains views that monitor ongoing database activity. These **dynamic performance views** are described in **Dynamic Performance Views** .

Note:

A multitenant container database is the only supported architecture in Oracle Database 21c and later releases. While the documentation is being revised, legacy terminology may persist. In most cases, "database" and "non-CDB" refer to a CDB or PDB, depending on context. In some contexts, such as upgrades, "non-CDB" refers to a non-CDB from a previous release.

