Oracle8*i*

Reference

Release 2 (8.1.6)

December 1999

Part No. A76961-01

ORACLE

Reference, Release 2 (8.1.6)

Part No. A76961-01

Primary Author: Diana Lorentz

Contributing Authors: Jason Durbin, Paul Lane

Contributors: Alex Tsukerman, Alok Pareek, Amit Ganesh, Andy Witkowski, Angela Amor, Ann Rhee, Aravind Yalamanchi, Archna Johnson, Ari Mozes, Arvind Nithrakashyap, Bill Lee, Bob Jenkins, Bob Thome, Brian Wright, Bruce Ernst, Cetin Ozbutun, Christopher Racicot, Daniel Wong, Debbie Steiner, Diana Rehfield, Ekrem Soylemez, Eric Voss, Eugene Chong, George Candea, Graham Wood, Hakkan Jakobsson, Harlan Sexton, Harvey Eneman, Helen Slattery, Ira Greenberg, Jonathan Klein, Juan Loaiza, Juan Tellez, Karuna Muthiah, Lance Ashdown, Lilian Hobbs, Lois Price, Mark Bauer, Mark Johnson, Mark Ramacher, Mary Ann Davidson, Michelle Cyran, Mohamed Ziauddin, Muralidhar Krishnaprasad, Namit Jain, Neerja Bhatt, Neil MacNaughton, Nina Lewis, Nipun Agarwal, Qin Yu, Randy Urbano, Ravi Murthy, Rick anderson, Rita Moran, Ronald Obermarck, Ruth Baylie, Sandy Dreskin, Sashikanth Chandrasekaran, Seema Sundara, Shankar Raman, Steve Harris, Steve Vivian, Steve Wertheimer, Subramanian Muralidhar, Sumanta Chatterjee, Suvendu Ray, Tak Wang, Thuvan Hoang, Tirthankar Lahiri, Vinay Srihari, Wilson Chan

Contents

2 Static Data Dictionary Views

3 Dynamic Performance (V$) Views

Send Us Your Comments

Oracle8*i* Reference, Release 2 (8.1.6)

Part No. A76961-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to us in the following ways:

- E-mail - infodev@us.oracle.com
- FAX - (650) 506-7228. Attn: Information Development
- Postal service:
  Oracle Corporation
  Server Technologies Documentation Manager
  500 Oracle Parkway
  Redwood Shores, CA 94065
  U.S.A.

If you would like a reply, please give your name, address, and telephone number below.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This reference provides reference information about Oracle8*i* for all operating systems.

Features and Functionality

Oracle8i Reference contains information about the features and functionality of the Oracle8*i* and the Oracle8*i* Enterprise Edition products. Oracle8*i* and Oracle8*i* Enterprise Edition have the same basic features. However, several advanced features are available only with the Enterprise Edition, and some of these are optional.

> **See Also:** *Getting to Know Oracle8i* for information about the differences between Oracle8*i* and the Oracle8*i* Enterprise Edition and the available features and options. That book also describes all the features that are new in Oracle8*i*.

Audience

This reference is intended for database administrators, system administrators, and database application developers.

This reference is not an installation or migration guide. If your primary interest is installation, refer to your operating system-specific Oracle documentation. If your primary interest is database and application migration, refer to *Oracle8i Migration*.

This reference describes the architecture, processes, structures, and other concepts of Oracle8*i*, but it does not explain how to administer an Oracle server. For that information, see the *Oracle8i Administrator's Guide*.

Experienced users of Oracle and advanced database application designers will find information in this reference useful. However, database application developers should also refer to the *Oracle8i Application Developer's Guide - Fundamentals* and to the documentation for the tool or language product they are using to develop Oracle database applications.

Knowledge Assumed of the Reader

Readers of this reference should be familiar with relational database concepts, basic Oracle concepts, and with the operating system environment in which they are running Oracle.

What's New in Oracle8*i*

Release 2 (8.1.6)

The following initialization parameters are new in this release of the documentation:

- ACTIVE_INSTANCE_COUNT on page 1-12
- CURSOR_SHARING on page 1-23
- JAVA_MAX_SESSIONSPACE_SIZE on page 1-48
- JAVA_POOL_SIZE on page 1-48
- JAVA_SOFT_SESSIONSPACE_LIMIT on page 1-48
- LOG_ARCHIVE_TRACE on page 1-61
- MTS_CIRCUITS on page 1-67
- MTS_SESSIONS on page 1-73

The following data dictionary views are new in this release of the documentation:

- ALL_MVIEWS on page 2-43 (and the related DBA_MVIEWS and USER_MVIEWS)
- ALL_TAB_MODIFICATIONS on page 2-78 (and the related DBA_TAB_MODIFICATIONS and USER_TAB_MODIFICATIONS)
- DBA_ORPHAN_KEY_TABLE on page 2-118
- DBA_REPAIR_TABLE on page 2-125

- V$BSP on page 3-15
- V$HS_PARAMETER on page 3-49

The following SQL scripts are new in this release of the documentation:

- INITJVM.SQL on page 5-7
- RMJVM.SQL on page 5-7
- UTLJAVARM.SQL on page 5-7

How Oracle8*i* Reference Is Organized

This manual is organized as follows:

Chapter 1, "Initialization Parameters"

This chapter describes the database initialization parameters you can specify in the parameter file to start or configure an instance.

Chapter 2, "Static Data Dictionary Views"

This chapter describes the Oracle data dictionary tables and views, also known as static views.

Chapter 3, "Dynamic Performance (V$) Views"

This chapter describes the dynamic performance views, also known as the V$ views.

Chapter 4, "Database Limits"

This chapter lists the limits of values associated with database functions and objects.

Chapter 5, "SQL Scripts"

This chapter describes the SQL scripts that are required for optimal operation of the Oracle server.

Appendix A, "Oracle Wait Events"

This appendix describes some event names, wait times, and parameters for wait events displayed by the V$SESSION_WAIT and V$SYSTEM_EVENT views.

Appendix B, "Oracle Enqueue Names"

This appendix lists some enqueues used by Oracle8*i*.

Appendix C, "Statistics Descriptions"
This appendix describes some statistics stored in the V$SESSION_WAIT and V$SYSSTAT dynamic performance table.

Conventions Used in This Manual

The following sections describe the conventions used in this manual.

Text of the Manual

The text of this manual uses the following conventions.

UPPERCASE Characters Uppercase text is used to call attention to command keywords, database object names, parameters, filenames, and so on.

For example, "After inserting the default value, Oracle checks the foreign key integrity constraint defined on the DEPTNO column," or "If you create a private rollback segment, the name must be included in the ROLLBACK_SEGMENTS initialization parameter."

***Italicized* Characters** Italicized words within text are book titles or emphasized words.

Code Examples

Commands or statements of SQL, Oracle Enterprise Manager line mode, and SQL*Plus appear in a monospaced font.

For example:

```
INSERT INTO emp (empno, ename) VALUES (1000, 'SMITH');
ALTER TABLESPACE users ADD DATAFILE 'users2.ora' SIZE 50K;
```

Example statements may include punctuation, such as commas or quotation marks. All punctuation in example statements is required. All example statements terminate with a semicolon (;). Depending on the application, a semicolon or other terminator may or may not be required to end a statement.

UPPERCASE in Code Examples Uppercase words in example statements indicate the keywords within Oracle SQL. When you issue statements, however, keywords are not case sensitive.

`lowercase` in Code Examples Lowercase words in example statements indicate words supplied only for the context of the example. For example, lowercase words may indicate the name of a table, column, or file.

Your Comments Are Welcome

We value and appreciate your comment as an Oracle user and reader of our manuals. As we write, revise, and evaluate our documentation, your opinions are the most important feedback we receive.

You can send comments and suggestions about this reference to the Information Development department at the following e-mail address:

infodev@us.oracle.com

If you prefer, you can send letters or faxes containing your comments to:

Server Technologies Documentation Manager
Oracle Corporation
500 Oracle Parkway Redwood Shores, CA 94065
Fax: (650) 506-7228 Attn.: Oracle8*i* Reference

1

Initialization Parameters

This chapter contains detailed descriptions of the database initialization parameters and includes the following topics:

Initialization Parameter File

The initialization parameter file is a text file that contains a list of parameters and a value for each parameter. The file should be written in the client's default character set. Specify in the parameter file values that reflect your installation.

The following are sample entries in a parameter file:

```
PROCESSES = 100
OPEN_LINKS = 12
GLOBAL_NAMES = TRUE
```

The name of the 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. As the database administrator, you can choose a different filename for your parameter file.

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

Sample parameter files are provided on the Oracle Server 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 system for best performance. Any changes will take effect after you completely shut down the instance and then restart it.

Database administrators can use initialization parameters to:

- Optimize performance by adjusting memory structures, such as the number of database buffers in memory
- Set some 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

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 Corporation Worldwide Support staff.

> **Note:** If you are using Trusted Oracle, see your documentation for Oracle security-related products for more details.

Specifying Values in the Parameter File

This section describes several aspects of setting parameter values in the parameter file. The following topics are included:

- Rules Governing Parameter Files
- Using Special Characters in Parameter Values
- Changing Parameter Values
- Displaying Current Parameter Values
- Uses of Parameters
- Types of Parameters
- Parameters You Should Not Specify in the Parameter File
- When Parameters Are Set Incorrectly

Rules Governing Parameter Files

The following rules govern the specification of parameters in the parameter file:

- All parameters are optional. The server has a default value for each required parameter. This value may be operating system dependent, depending on the parameter.
- The parameter file should contain only parameters and comments. A pound sign (#) starts a comment. The rest of the line is ignored.
- 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 value entries. Enter multiple values enclosed in parentheses and separated by commas. For example:

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

  Alternatively, you can enter multiple values without parentheses and commas. For example:

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

  Either syntax is valid.

  If you enter values for one parameter in multiple entries, the entries must be on consecutive lines. If they are not, the first entry will not be processed properly. For example, in the following entry:

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

  the setting for SEG3 and SEG4 will override the setting for SEG1 and SEG2.

- A backslash or **escape character** (\) indicates continuation of the parameter specification. If a backslash continues a line, the continued line must have no leading spaces. For example:

  ```
  ROLLBACK_SEGMENTS = (SEG1, SEG2, \
  ```

```
SEG3, SEG4, SEG5)
```

- You can use the `IFILE` parameter to call another parameter file, which must be in the same format as the original parameter file. See "IFILE" on page 1-45.
- 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"
  ```

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

- Enclose in double quotation marks any parameter value that contains a special character (described in the section that follows).

See your operating system specific Oracle documentation for more information on parameter files.

Using Special Characters in Parameter Values

If a parameter value contains a special character, then the special character must be preceded by an escape character or the entire parameter value must be enclosed in double quotation marks. For example:

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

or

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

Table 1–1 lists the special characters you can use in parameter files.

Table 1–1 Special Characters in the Initialization Parameter File

Character	Description
#	Comment
(	Start list of values
)	End list of values
"	Start or end of quoted string

Table 1–1 Special Characters in the Initialization Parameter File

Character	Description
'	Start or end of quoted string
=	Separator of keyword and value(s)
,	Separator of elements
-	Precedes UNIX-style keywords
\	Escape character

If a special character must be treated literally in the initialization parameter file, it must either preceded by the escape character or the entire string that contains the special character must be enclosed in single or double quotation marks.

Using the Escape Character As described in "Rules Governing Parameter Files" on page 1-3, the escape character (\) can also signify a line continuation. If the escape character is followed by an alphanumeric character, then the escape character is treated as a normal character in the input. If it is not followed by an alphanumeric, then the escape character is treated either as an escape character or as a continuation character.

Using Quotes Quotes can be nested in any of three ways. 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'
```

The third method is to precede the inner quotation marks with an escape character. For example:

```
NLS_DATE_FORMAT = "\"Today is\" MM/DD/YYYY"
```

Changing Parameter Values

You change the value of a parameter by editing the parameter file. In most cases, the new value takes effect the next time you start an instance of the database. However, you can change the value of some parameters for the duration of the current session, as discussed in the section that follows.

Dynamic Parameters Some initialization parameters are **dynamic**, that is, they can be modified using the ALTER SESSION or ALTER SYSTEM command while an instance is running.

Use this syntax for dynamically altering the initialization parameters:

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

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

The ALTER SESSION command changes the value of the specified parameter for the duration of the session that invokes this command. The value of this parameter does not change for other sessions in the instance. The value of the initialization parameters listed in Table 1–2 can be changed with ALTER SESSION.

The ALTER SYSTEM command 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 initialization parameters listed in Table 1–3 can be changed with ALTER SYSTEM.

The ALTER SYSTEM...DEFERRED command 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 initialization parameters listed in Table 1–4 can be changed with ALTER SYSTEM...DEFERRED.

Table 1–2 Initialization Parameters Alterable with ALTER SESSION

CURSOR_SHARING	NLS_TERRITORY
DB_BLOCK_CHECKING	OBJECT_CACHE_MAX_SIZE_PERCENT
DB_FILE_MULTIBLOCK_READ_COUNT	OBJECT_CACHE_OPTIMAL_SIZE
FAST_START_IO_TARGET	OPTIMIZER_INDEX_CACHING
GLOBAL_NAMES	OPTIMIZER_INDEX_COST_ADJ
HASH_AREA_SIZE	OPTIMIZER_MAX_PERMUTATIONS
HASH_JOIN_ENABLED	OPTIMIZER_MODE
HASH_MULTIBLOCK_IO_COUNT	OPTIMIZER_PERCENT_PARALLEL
LOG_ARCHIVE_DEST_n	PARALLEL_BROADCAST_ENABLED
LOG_ARCHIVE_DEST_STATE_n	PARALLEL_INSTANCE_GROUP
LOG_ARCHIVE_MIN_SUCCEED_DEST	PARALLEL_MIN_PERCENT
MAX_DUMP_FILE_SIZE	PARTITION_VIEW_ENABLED
NLS_CALENDAR	PLSQL_V2_COMPATIBILITY
NLS_COMP	QUERY_REWRITE_ENABLED
NLS_CURRENCY	QUERY_REWRITE_INTEGRITY
NLS_DATE_FORMAT	REMOTE_DEPENDENCIES_MODE
NLS_DATE_LANGUAGE	SESSION_CACHED_CURSORS
NLS_DUAL_CURRENCY	SORT_AREA_RETAINED_SIZE
NLS_ISO_CURRENCY	SORT_AREA_SIZE
NLS_LANGUAGE	SORT_MULTIBLOCK_READ_COUNT
NLS_NUMERIC_CHARACTERS	STAR_TRANSFORMATION_ENABLED
NLS_SORT	TIMED_STATISTICS

Table 1–3 Initialization Parameters Alterable with ALTER SYSTEM

AQ_TM_PROCESSES	LOG_ARCHIVE_MAX_PROCESSES
BACKGROUND_DUMP_DEST	LOG_ARCHIVE_MIN_SUCCEED_DEST
CONTROL_FILE_RECORD_KEEP_TIME	LOG_ARCHIVE_TRACE
CORE_DUMP_DEST	LOG_CHECKPOINT_INTERVAL
CURSOR_SHARING	LOG_CHECKPOINT_TIMEOUT
DB_BLOCK_CHECKSUM	MAX_DUMP_FILE_SIZE
DB_BLOCK_MAX_DIRTY_TARGET	MTS_DISPATCHERS
DB_FILE_MULTIBLOCK_READ_COUNT	MTS_SERVERS
FAST_START_IO_TARGET	OPTIMIZER_MAX_PERMUTATIONS
FAST_START_PARALLEL_ROLLBACK	PARALLEL_ADAPTIVE_MULTI_USER
FIXED_DATE	PARALLEL_INSTANCE_GROUP
GC_DEFER_TIME	PARALLEL_THREADS_PER_CPU
GLOBAL_NAMES	PLSQL_V2_COMPATIBILITY
HASH_MULTIBLOCK_IO_COUNT	QUERY_REWRITE_ENABLED
HS_AUTOREGISTER	QUERY_REWRITE_INTEGRITY
JOB_QUEUE_PROCESSES	REMOTE_DEPENDENCIES_MODE
LICENSE_MAX_SESSIONS	RESOURCE_LIMIT
LICENSE_MAX_USERS	RESOURCE_MANAGER_PLAN
LICENSE_SESSIONS_WARNING	STANDBY_ARCHIVE_DEST
LOG_ARCHIVE_DEST	TIMED_OS_STATISTICS
LOG_ARCHIVE_DEST_n	TIMED_STATISTICS
LOG_ARCHIVE_DEST_STATE_n	USER_DUMP_DEST
LOG_ARCHIVE_DUPLEX_DEST	

Table 1–4 Initialization Parameters Alterable with ALTER SYSTEM . . . DEFERRED

BACKUP_TAPE_IO_SLAVES	PLSQL_V2_COMPATIBILITY
DB_BLOCK_CHECKING	SORT_AREA_RETAINED_SIZE
DB_FILE_DIRECT_IO_COUNT	SORT_AREA_SIZE
MAX_DUMP_FILE_SIZE	SORT_MULTIBLOCK_READ_COUNT
OBJECT_CACHE_MAX_SIZE_PERCENT	TRANSACTION_AUDITING
OBJECT_CACHE_OPTIMAL_SIZE	

Displaying Current Parameter Values

To see the current settings for initialization parameters, use the following SQL*Plus statement:

```
SHOW PARAMETERS
```

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

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

```
SHOW PARAMETERS BLOCK
```

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

Uses of Parameters

Initialization parameters fall into various functional groups. For example, parameters perform the following functions:

- Set limits for the entire database
- Set user or process limits
- Name files or directories required by a database system
- Set limits on database resources
- Affect performance (these are called **variable parameters**)

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

Types of Parameters

An Oracle server has the following types of initialization parameters:

- Derived Parameters
- Global Cache Parameters with the Prefix GC
- Operating System Dependent Parameters
- Variable Parameters (these can be dynamic parameters or any of the above)
- Heterogeneous Services Parameters

Derived Parameters Some initialization parameters are called **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, 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, the default value of SESSIONS changes as well, unless you override it with a specified value.

Global Cache Parameters with the Prefix GC Initialization parameters with the prefix GC, such as GC_DEFER_TIME, apply to systems using the Oracle Parallel Server. The prefix GC stands for "global cache." The settings of these parameters determine how the Oracle Parallel Server coordinates multiple instances. The settings you choose have an effect on the use of specific operating system resources.

See Also:

- *Oracle8i Parallel Server Concepts* for more information about the Oracle Parallel Server.
- Your system release bulletins or other Oracle documentation on your operating system for information on Oracle Parallel Server parameters.

Operating System Dependent Parameters The valid values or value ranges of some initialization parameters depend upon the host operating system. For example, the parameter DB_BLOCK_BUFFERS 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 DB_BLOCK_SIZE, has a system dependent default value.

See Also: The Oracle documentation for your operating system for more information on operating system dependent Oracle parameters and operating system parameters.

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 OPEN_CURSORS 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 DB_BLOCK_BUFFERS 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.

Heterogeneous Services Parameters A number of parameters are specific to Oracle Heterogeneous Services. These parameters must be set at gateways using the `DBMS_HS` package.

> **See Also:** *Oracle8i Distributed Database Systems* for information about specifying these parameters.

Parameters You Should Not Specify in the Parameter File

Normally you should not specify two types of parameters in the parameter file:

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

When Parameters Are Set Incorrectly

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, 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, you should shut down the instance, adjust the relevant parameter, and restart the instance.

Reading the Parameter Descriptions

The parameter descriptions in this chapter follow the format shown below.

PARAMETER_NAME

Parameter type:	specifies the type of the parameter value: integer, Boolean, string, and so on.
Syntax:	for string parameters, specifies the valid syntax for specifying the string.
Parameter class:	specifies whether the parameter is dynamic or static. If dynamic, then it also specifies whether its value can be changed by an `ALTER SYSTEM` or `ALTER SESSION` statement. `ALTER SYSTEM` sets or changes the value of the parameter for all sessions. `ALTER SESSION` overrides the system setting of the parameter for the current session only. You can restore the system setting for that session only by issuing another `ALTER SESSION` statement.
Default value:	specifies the value this parameter assumes if not explicitly specified.
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.
Oracle Parallel Server:	specifies how the values for this parameter must be specified for multiple instances in an Oracle Parallel Server. Not applicable to all parameters.

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

Parameter Descriptions

Descriptions of the individual initialization parameters follow in alphabetical order. Initialization parameter values apply to the entire database, not to an individual user, unless otherwise specified. Parameters that have become obsolete in this release are not documented.

> **See Also:** Your system release bulletins or other operating system specific Oracle documentation.

ACTIVE_INSTANCE_COUNT

Parameter type:	Integer
Parameter class:	Static

Default value:	None
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.)
Oracle Parallel Server:	You must set this parameter for every instance, and multiple instances must have the same value.

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

ACTIVE_INSTANCE_COUNT lets you 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.

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 now-primary instance fails.

See Also: *Oracle8i Parallel Server Setup and Configuration Guide* for more information on setting this parameter.

ALWAYS_ANTI_JOIN

Parameter type:	String
Syntax:	ALWAYS_ANTI_JOIN = {NESTED_LOOPS | MERGE | HASH}
Parameter class:	Static
Default value:	NESTED_LOOPS

An **anti-join** returns rows that match a NOT IN subquery. ALWAYS_ANTI_JOIN sets the type of anti-join that Oracle uses. The system verifies that it is legal to perform an anti-join, and if it is, processes the subquery depending on the value of this parameter.

- NESTED_LOOPS: Oracle uses a nested loop algorithm.
- MERGE: Oracle uses a sort merge algorithm.
- HASH: Oracle uses a hash algorithm to evaluate the subquery. This value allows the evaluation to run in parallel.

See Also: *Oracle8i Concepts* and *Oracle8i Designing and Tuning for Performance* for more information on optimizing anti-joins.

ALWAYS_SEMI_JOIN

Parameter type:	String
Syntax	ALWAYS_SEMI_JOIN = {NESTED_LOOPS | MERGE | HASH}
Parameter class:	Static
Default value:	NESTED_LOOPS

A **semi-join** returns rows that match an EXISTS subquery. Semi-joins can improve query performance if no index has been defined on the column that constrains the subquery. ALWAYS_SEMI_JOIN specifies the type of semi-join that Oracle uses. The system verifies that it is legal to perform a semi-join, and if it is, processes the subquery depending on the value of this parameter.

- NESTED_LOOPS: Oracle uses a nested loop algorithm.
- MERGE: Oracle uses the sort merge algorithm.
- HASH: Oracle converts a correlated EXISTS subquery into a view query block and semi-join, which it evaluates in parallel.

See Also: *Oracle8i Concepts* for more information on optimizing semi-joins.

AQ_TM_PROCESSES

Parameter type:	Integer
Parameter class:	Dynamic. Scope= ALTER SYSTEM.
Default value:	0
Range of values:	0 to 10

AQ_TM_PROCESSES enables time monitoring on queue messages. The times can be used in messages that specify delay and expiration properties. Values from 1 to 10 specify the number of queue monitor processes created to monitor the messages. If AQ_TM_PROCESSES is not specified or is set to 0, then the queue monitor is not created.

See Also: *Oracle8i Application Developer's Guide - Advanced Queuing* for more information about this parameter and Advanced Queuing.

AUDIT_FILE_DEST

Parameter type:	String
Syntax:	AUDIT_FILE_DEST = 'directory'

Parameter class:	Static
Default value:	`$oracle_home/rdbms/audit`

`AUDIT_FILE_DEST` specifies the directory where Oracle stores auditing files.

AUDIT_TRAIL

Parameter type:	String
Syntax	`AUDIT_TRAIL = {NONE | FALSE | DB | TRUE | OS}`
Parameter class:	Static
Default value:	`NONE`

`AUDIT_TRAIL` enables or disables the automatic writing of rows to the audit trail.

- `NONE` or `FALSE`: Audit records are not written.
- `OS`: enables system-wide auditing and causes audited records to be written to the operating system's audit trail.
- `DB` or `TRUE`: enables system-wide auditing and causes audited records to be written to the database audit trail (the `SYS.AUD$` table).

You can use the `SQL` statements `AUDIT` *sql_statements* and `AUDIT` *schema_objects* to set auditing options regardless of the setting of this parameter.

See Also: *Oracle8i Administrator's Guide*.

BACKGROUND_CORE_DUMP

Parameter type:	String
Syntax:	`BACKGROUND_CORE_DUMP = {FULL | PARTIAL}`
Parameter class:	Static
Default value:	`PARTIAL`

`BACKGROUND_CORE_DUMP` is primarily a UNIX parameter. It specifies whether Oracle includes the SGA in the core file for Oracle background processes.

- `FULL`: Oracle dumps the SGA as part of the generated core file.
- `PARTIAL`: Oracle does not dump the SGA as part of the generated core file.

See Also: "SHADOW_CORE_DUMP" on page 1-110.

BACKGROUND_DUMP_DEST

Parameter type:	String
Syntax:	BACKGROUND_DUMP_DEST = {pathname | directory}
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	Operating system dependent
Range of values:	Any valid local path, directory, or disk

BACKGROUND_DUMP_DEST specifies the pathname (directory or disc) where debugging trace files for the background processes (LGWR, DBW*n*, and so on) are written during Oracle operations. (For information on setting a destination for server process trace files, see "USER_DUMP_DEST" on page 1-121.)

An **alert file** in the directory specified by BACKGROUND_DUMP_DEST logs significant database events and messages. Anything that affects the database for an instance or globally is recorded here. The alert file is a normal text file. Its filename is operating system dependent. For platforms that support multiple instances, it takes the form ALERT_sid.LOG (where *sid* 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.

See Also:

- *Oracle8i Administrator's Guide* for more information on setting this parameter and on alert files.
- Your operating system specific Oracle documentation for the default value of this parameter.

BACKUP_TAPE_IO_SLAVES

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SYSTEM ... DEFERRED.
Default value:	FALSE
Range of values:	TRUE | FALSE

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

Typically, I/O server processes are used to simulate asynchronous I/O on platforms that either do not support asynchronous I/O or implement it inefficiently. However, you can use I/O server processes even when asynchronous I/O is available. In that case, the I/O server processes will use asynchronous I/O. See also "DBWR_IO_SLAVES" on page 1-34.

Only one process can access a tape device at any given time. Therefore, this parameter is Boolean, and allows or disallows deployment of an I/O server process to access a tape device.

Note: 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: *Oracle8i Recovery Manager User's Guide and Reference* for more information on duplexed backups.

BITMAP_MERGE_AREA_SIZE

Parameter type:	Integer
Parameter class:	Static
Default value:	1048576 (1 MB)
Range of values:	Operating system dependent

`BITMAP_MERGE_AREA_SIZE` is relevant only for systems containing bitmap indexes. It specifies the amount of memory Oracle uses to merge bitmaps retrieved from a range scan of the index. The default value is 1 MB. A larger value usually improves performance, because the bitmap segments must be sorted before being merged into a single bitmap.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.

BLANK_TRIMMING

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

BLANK_TRIMMING specifies the data assignment semantics of character datatypes.

- 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 is all blanks. This value complies with the semantics of SQL92 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.

BUFFER_POOL_KEEP

Parameter type:	String
Syntax:	BUFFER_POOL_KEEP = {integer | (BUFFERS: integer [, LRU_LATCHES: integer]) } where *integer* is the number of buffers and, optionally, the number of LRU latches.
Parameter class:	Static
Default value:	None

BUFFER_POOL_KEEP lets you save objects in the buffer cache by setting aside a portion of the total number of buffers (the value of the DB_BLOCK_BUFFERS parameter) as a KEEP buffer pool. You can also allocate to the KEEP buffer pool a specified portion of the total number of LRU latches (the value of the DB_BLOCK_LRU_LATCHES parameter).

The string value can take one of two forms. You can simply specify a buffer count:

```
BUFFER_POOL_KEEP = 5
```

or you can specify a combination of buffers and LRU latches. Specifying LRU latches is optional:

```
BUFFER_POOL_KEEP = (BUFFERS: 400 [, LRU_LATCHES:3] )
```

This parameter is most useful in combination with BUFFER_POOL_RECYCLE.

See Also: *Oracle8i Designing and Tuning for Performance* for information on setting these parameters and on using multiple buffer pools.

BUFFER_POOL_RECYCLE

Parameter type:	String
Syntax:	BUFFER_POOL_RECYCLE = integer | (BUFFERS: integer [, LRU_LATCHES: integer]) where *integer* is the number of buffers and, optionally, the number of LRU latches.
Parameter class:	Static
Default value:	None

BUFFER_POOL_RECYCLE lets you limit the size of objects in the buffer cache by setting aside a portion of the total number of buffers (the value of the DB_BLOCK_BUFFERS parameter) as a RECYCLE buffer pool. You can also allocate to the RECYCLE buffer pool a specified portion of the total number of LRU latches (the value of the DB_BLOCK_LRU_LATCHES parameter).

The string value can take one of two forms. You can simply specify a buffer count:

```
BUFFER_POOL_RECYCLE = 5
```

or you can specify a combination of buffers and LRU latches. Specifying LRU latches is optional:

```
BUFFER_POOL_RECYCLE = (BUFFERS: 50 [, LRU_LATCHES:1] )
```

This parameter is most useful in combination with BUFFER_POOL_KEEP.

See Also: *Oracle8i Designing and Tuning for Performance* for information on setting these parameters and on using multiple buffer pools.

COMMIT_POINT_STRENGTH

Parameter type:	Integer
Parameter class:	Static
Default value:	Operating system dependent
Range of values:	0 to 255

COMMIT_POINT_STRENGTH is relevant only in distributed database systems. It specifies a value that determines the **commit point site** in a distributed transaction. The node in the transaction with the highest value for COMMIT_POINT_STRENGTH 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, COMMIT_POINT_STRENGTH 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:

- *Oracle8i Concepts* and *Oracle8i Distributed Database Systems* for information on two-phase commit.
- Your operating system specific Oracle documentation for the default value of this parameter.

COMPATIBLE

Parameter type:	String
Syntax:	COMPATIBLE = release_number
Parameter class:	Static
Default value:	8.0.0
Range of values:	Default release to current release
Oracle Parallel Server:	Multiple instances must have the same value

COMPATIBLE lets you use a new release, while at the same time guaranteeing backward compatibility with an earlier release. This ability is helpful in case it becomes necessary to revert to the earlier release.

This parameter specifies the release with which the Oracle server must maintain compatibility. It allows you to take advantage of the maintenance improvements of a new release immediately in your production systems without testing the new functionality in your environment. Some features of the current release may be restricted.

When using the standby database feature, this parameter must have the same value on the primary and standby databases, and the value must be 7.3.0.0.0 or higher. The default value is the earliest release with which compatibility can be guaranteed.

The value of COMPATIBLE must be set to 8.1 or higher in order to use stored columns of UROWID type. You can use ROWID pseudocolumns for index-organized tables regardless of the parameter setting.

See Also: *Oracle8i Migration* for more information on setting this parameter.

CONTROL_FILE_RECORD_KEEP_TIME

Parameter type:	Integer
Parameter class:	Dynamic. Scope = `ALTER SYSTEM`.
Default value:	7 (days)
Range of values:	0 to 365 (days)

`CONTROL_FILE_RECORD_KEEP_TIME` specifies the minimum number of days before a reusable record in the control file can be reused. In the event a new record needs to 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.

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 datafile, tablespace, and redo thread records, which are never reused unless the corresponding object is dropped from the tablespace.

See Also: "CONTROL_FILES" on page 1-21, *Oracle8i Concepts*, and *Oracle8i Backup and Recovery Guide* for more information on control files.

CONTROL_FILES

Parameter type:	String
Syntax:	`CONTROL_FILES = filename [, filename [...] ]`
Parameter class:	Static
Default value:	Operating system dependent
Range of values:	1 to 8 filenames

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 and redo files). `CONTROL_FILES` specifies one or more names of control files, separated by commas.

In an Oracle Parallel Server environment, you must set this parameter for every instance.

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

See Also: *Oracle8i Designing and Tuning for Performance.*

CORE_DUMP_DEST

Parameter type:	String
Syntax:	`CORE_DUMP_DEST = directory`
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	`$ORACLE_HOME/DBS/`

CORE_DUMP_DEST is primarily a UNIX parameter and may not be supported on your platform. It specifies the directory where Oracle dumps core files.

CPU_COUNT

Parameter type:	Integer
Parameter class:	Static
Default value:	Set automatically by Oracle
Range of values:	0 to unlimited

WARNING: On most platforms, Oracle automatically sets the value of `CPU_COUNT` to the number of CPUs available to your Oracle instance. Do not change the value of `CPU_COUNT`.

`CPU_COUNT` specifies the number of CPUs available to Oracle. Oracle uses it to set the default value of the `LOG_SIMULTANEOUS_COPIES` parameter. On single-CPU computers, the value of `CPU_COUNT` is 1.

If your system experiences heavy contention for latches, change the value of `LOG_SIMULTANEOUS_COPIES` to twice the number of CPUs you have. Do not change the value of `CPU_COUNT`.

See Also: Your operating system specific Oracle documentation for more information about this parameter.

CREATE_BITMAP_AREA_SIZE

Parameter type:	Integer
Parameter class:	Static
Default value:	8388608 (8 MB)
Range of values:	Operating system dependent

CREATE_BITMAP_AREA_SIZE is relevant only for systems containing bitmap indexes. It specifies the amount of memory allocated for bitmap creation. The default value is 8 MB. 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 on the order of kilobytes rather than megabytes. As a general rule, the higher the cardinality, the more memory is needed for optimal performance.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on using bitmap indexes.

CURSOR_SHARING

Parameter type:	String
Syntax:	CURSOR_SHARING = {FORCE | EXACT}
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM.
Default value:	EXACT

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

- EXACT causes only identical SQL statements to share a cursor.
- FORCE forces statements that may differ in some literals, but are otherwise identical, to share a cursor, unless the literals affect the meaning of the statement.

Note: Forcing cursor sharing among similar (but not identical) statements can have unexpected results in some DSS applications and if your applications use stored outlines.

See Also: *Oracle8i Designing and Tuning for Performance* for information on setting this parameter in these and other environments.

CURSOR_SPACE_FOR_TIME

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

CURSOR_SPACE_FOR_TIME lets you use more space for cursors in order to save time. It affects both the shared SQL area and the client's private SQL area.

- 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.

Note: If this parameter is set to TRUE, the SERIAL_REUSE parameter is disabled. See "SERIAL _REUSE" on page 1-107.

See Also: *Oracle8i Concepts* and *Oracle8i Designing and Tuning for Performance*.

DB_BLOCK_BUFFERS

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived: 48 MB / DB_BLOCK_SIZE
Range of values:	4 to an operating system specific maximum
Oracle Parallel Server:	Multiple instances can have different values, and you can change the values as needed.

DB_BLOCK_BUFFERS specifies the number of database buffers in the buffer cache. It is one of several parameters that contribute to the total memory requirements of the SGA of an instance.

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

The buffers for the "keep" pool and the "recycle" pool come out of the buffer cache. In addition, each LRU list in the buffer cache must have at least 50 buffers. Therefore, if you specify BUFFER_POOL_KEEP and BUFFER_POOL_RECYCLE, be sure the value of DB_BLOCK_BUFFERS conforms to the following formula:

```
DB_BLOCK_BUFFERS > #_buffers_in_"keep"_buffer_pool
                 + #_BUFFERS_IN_"recycle"_pool
                 + 50*(DB_BLOCK_LRU_LATCHES
                       - #_lru_latches_in_"keep"_buffer_pool
                       - #_lru_latches_in_"recycle"_buffer_pool)
```

If you did not specify any LRU latches in BUFFER_POOL_KEEP and BUFFER_POOL_RECYCLE, the values defaults to 1.

This parameter affects the probability of data block pinging when Oracle Parallel Server is enabled: the more buffers, the more chance of pinging.

Note: The checkpoint process is enabled automatically if the system configuration is such that moving checkpoint processing from the logwriter to a separate process is deemed to improve performance. Currently, the criteria are DB_FILES >= 50 or DB_BLOCK_BUFFERS >= 10000.

See Also:

- *Oracle8i Designing and Tuning for Performance*, "BUFFER_POOL_KEEP" on page 1-18, and "BUFFER_POOL_RECYCLE" on page 1-19 for more information on setting this parameter.
- *Oracle8i Parallel Server Concepts* for more information on data block pinging.
- Your operating system specific Oracle documentation.

DB_BLOCK_CHECKING

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM ... DEFERRED.
Default value:	FALSE
Range of values:	TRUE | FALSE

DB_BLOCK_CHECKING controls whether Oracle performs block checking for data blocks. When this parameter is set to TRUE, Oracle performs block checking for all data blocks. When it is set to FALSE, Oracle does not perform block checking for blocks in the user tablespaces. However, block checking for the SYSTEM tablespace is always turned on.

Oracle checks a block by walking through the data on the block, making sure it is self-consistent. Block checking can often prevent memory and data corruption. Block checking typically causes 1% to 10% overhead, depending on workload. The more updates or inserts in a workload, the more expensive it is to turn on block checking. Oracle recommends that you set DB_BLOCK_CHECKING to TRUE if the performance overhead is acceptable.

DB_BLOCK_CHECKSUM

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	FALSE
Range of values:	TRUE | FALSE

DB_BLOCK_CHECKSUM determines whether DBW*n* 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. Checksums are verified when a block is read only if this parameter is TRUE and the last write of the block stored a checksum. In addition, Oracle will give every log block a checksum before writing it to the current log.

If this parameter is set to FALSE, DBW*n* calculates checksums only for the SYSTEM tablespace, but not for user tablespaces.

Checksums allow Oracle to detect corruption caused by underlying disks, storage systems, or I/O systems. Turning on this feature typically causes only 1% to 2% overhead. Therefore, Oracle Corporation recommends that you set DB_BLOCK_CHECKSUM to TRUE.

DB_BLOCK_LRU_LATCHES

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived: CPU_COUNT / 2
Range of values:	1 to either (CPU_COUNT x 2 x 3) or (DB_BLOCK_BUFFERS / 50), whichever is less

DB_BLOCK_LRU_LATCHES specifies the maximum number of LRU latch sets. The buffers of a buffer pool are equally divided among the working LRU latch sets of the buffer pool so that each buffer is protected by one LRU latch. Normally, the more latches you specify, the less contention exists for those latches. However, too many latches may result in small LRU lists, potentially reducing the cache life of a database block. The maximum of (CPU_COUNT x 2 x 3) ensures that the number of latches does not exceed twice the product of the number of CPUs and the number of buffer pools.

Typically you should set this parameter to the number of CPUs or a multiple of that number. Each working set is handled entirely by one database writer (DBW*n*) process. Therefore, if multiple DBW*n* processes are running, the number of LRU latches should be greater than or equal to the number of DBW*n* processes. To balance the load evenly between the DBW*n* processes, the number of LRU latches in each buffer pool should be a multiple of the number of DBW*n* processes.

If you do not set this parameter, Oracle uses the value `CPU_COUNT/2`. This value is usually adequate. Increase this value only if misses are higher than 3%, as calculated from values in `V$LATCH`. When you increase the value, Oracle decides whether to use this value or reduce it based on a number of internal checks.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.

DB_BLOCK_MAX_DIRTY_TARGET

Parameter type:	Integer
Parameter class:	Dynamic. Scope = `ALTER SYSTEM`.
Default value:	All buffers in the cache
Range of values:	1000 to all buffers in the cache. (Setting to 0 disables limiting dirty buffers.)
Oracle Parallel Server:	Multiple instances can have different values

`DB_BLOCK_MAX_DIRTY_TARGET` specifies the number of buffers that can be dirty (modified and different from what is on disk) in the buffer cache. Therefore, it indirectly controls the duration of instance recovery by placing a rough limit on the number of blocks that must be read during crash and instance recovery.

Note: The `FAST_START_IO_TARGET` parameter (available only with Oracle Enterprise Edition) can be used to enforce a a more accurate bound on the number of I/Os during recovery.

When you specify this parameter, DBW*n* writes out buffers to attempt to keep the number of dirty buffers in the cache below the specified value. Setting this value to 0 disables the mechanism that limits the number of dirty buffers in the cache.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.

DB_BLOCK_SIZE

Parameter type:	Integer
Parameter class:	Static
Default value:	Operating system dependent
Range of values:	2048 to 32768, but your operating system may have a narrower range
Oracle Parallel Server:	You must set this parameter for every instance, and multiple instances must have the same value

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

DB_BLOCK_SIZE specifies the size in bytes of Oracle database blocks. Typical values are 2048 and 4096. The value for DB_BLOCK_SIZE in effect at CREATE DATABASE time determines the size of the blocks. The value must remain set to its original value.

If you are using Oracle Parallel Server, 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.

See Also:

- *Oracle8i Designing and Tuning for Performance* and *Oracle8i Administrator's Guide* for information on setting this parameter.
- *Oracle8i SQL Reference* for information on freelist groups.
- *Oracle8i Designing and Tuning for Performance* for more information on the DSS and data warehouse environments.
- *Oracle8i Concepts* for information on block size in general.

DB_DOMAIN

Parameter type:	String
Syntax:	DB_DOMAIN = domain_name
Parameter class:	Static
Default value:	WORLD
Range of values:	Any legal string of name components, separated by periods and up to 128 characters long (including the periods). See valid characters below. This value cannot be null.

Oracle Parallel Server: You must set this parameter for every instance, and multiple instances must have the same value

In a distributed database system, DB_DOMAIN specifies the logical location of the database within the network structure. 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, separated by periods. Oracle Corporation recommends that you specify DB_DOMAIN 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 DB_DOMAIN is "JAPAN.ACME.COM", then their "SALES" database (SALES.JAPAN.ACME.COM) is uniquely distinguished from another database with DB_NAME = "SALES" but with DB_DOMAIN = "US.ACME.COM".

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. See also the data dictionary view "GLOBAL_NAME" on page 2-145.

The following characters are valid in a database domain name: alphabetic characters, numbers, underscore (_), and pound (#).

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- *Oracle8i Distributed Database Systems* for information on setting this parameter in a distributed database.

DB_FILE_DIRECT_IO_COUNT

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM ... DEFERRED.
Default value:	64
Range of values:	Operating system dependent

DB_FILE_DIRECT_IO_COUNT specifies the buffer size (in number of blocks) used for I/O operations performed by backup, restore, or direct path read and write functions. The I/O buffer size is a product of DB_FILE_DIRECT_IO_COUNT * DB_BLOCK_SIZE. Assuming a block size of 2048, the resulting buffer size is 128K. The I/O buffer size cannot exceed the maximum size of an I/O operation for your platform.

Assigning a high value to this parameter results in greater use of PGA and SGA memory.

Note: If you have previously used the CCF_IO_SIZE parameter and are migrating to DB_FILE_DIRECT_IO_COUNT, remember that CCF_IO_SIZE was specified in bytes while DB_FILE_DIRECT_IO_COUNT must be specified in database blocks.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.

DB_FILE_MULTIBLOCK_READ_COUNT

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM, ALTER SESSION.
Default value:	8
Range of values:	Operating system dependent

DB_FILE_MULTIBLOCK_READ_COUNT is one of the parameters you can use to minimize I/O during table scans. It specifies the maximum number of blocks read in one I/O operation during a sequential scan. 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.

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 always less than 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, Oracle uses the maximum.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- Your operating system specific Oracle documentation for the default value of this parameter.

DB_FILE_NAME_CONVERT

Parameter type:	String
Syntax:	`DB_FILE_NAME_CONVERT = [()'string1' , 'string2'())]` where `string1` is the pattern of the primary database filenames `string2` is the pattern of the standby database filenames You can use single or double quotation marks. The parentheses are optional.
Parameter class:	Static
Default value:	None

`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. 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 two strings. The first string is the pattern found in the datafile names on the primary database. The second string is the pattern found in the datafile names on the standby database.

You can also use `DB_FILE_NAME_CONVERT` to rename the datafiles in the clone controlfile when setting up a clone database during tablespace point-in-time recovery.

See Also:

- *Oracle8i Backup and Recovery Guide.*
- *Oracle8i Standby Database Concepts and Administration.*
- *Oracle8i Recovery Manager User's Guide and Reference.*

DB_FILES

Parameter type:	Integer
Parameter class:	Static
Default value:	Operating system dependent

Range of values:	**Minimum**: the current actual number of datafiles in the database **Maximum**: the value that was specified in the MAXDATAFILES clause the last time CREATE DATABASE or CREATE CONTROLFILE was executed
Oracle Parallel Server:	You must set this parameter for every instance, and multiple instances must have the same value

DB_FILES specifies the maximum number of database files that can be opened for this database. 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 DB_FILES, 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, they should have the same value for this parameter.

Note: The checkpoint process is enabled automatically if DB_FILES >= 50 or DB_BLOCK_BUFFERS >= 10000. At these values, switching checkpoint processing from the logwriter to a separate process is deemed to improve performance.

See Also:

- *Oracle8i Parallel Server Administration, Deployment, and Performance* for information on setting this parameter in an Oracle Parallel Server environment.
- Your operating system specific Oracle documentation for the default value of this parameter.

DB_NAME

Parameter type:	String
Syntax:	DB_NAME = database_name
Parameter class:	Static
Default value:	None
Oracle Parallel Server:	You must set this parameter for every instance. Multiple instances must have the same value, or the same value must be specified in the SQL*Plus command STARTUP OPEN *db_name* or the SQL statement ALTER DATABASE *db_name* MOUNT.

DB_NAME specifies a database identifier of up to 8 characters. If specified, it must correspond to the name specified in the CREATE DATABASE statement. Although the use of DB_NAME is optional, you should generally set it before issuing the CREATE DATABASE statement, and then reference it in that 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.

If you do not specify this parameter, you must specify a database name in either the STARTUP command or the ALTER DATABASE ... MOUNT statement for each instance of the Oracle Parallel Server.

The following characters are valid in a database name: alphabetic characters, numbers, underscore (_), pound (#), and dollar symbol ($). No other characters are valid. 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.

See Also: *Oracle8i Administrator's Guide* and *Oracle8i Parallel Server Administration, Deployment, and Performance* for more information on setting this parameter.

DB_WRITER_PROCESSES

Parameter type:	Integer
Parameter class:	Static
Default value:	1
Range of values:	1 to 10

DB_WRITER_PROCESSES is useful for systems that modify data heavily. It specifies the initial number of database writer processes for an instance. If you increase the value from the default of 1, you should adjust the DB_BLOCK_LRU_LATCHES parameter so that each database writer process (DBW*n*) has the same lumber of latches (LRU buffer lists).

Note: If you set the DBWR_IO_SLAVES parameter, Oracle uses only one database writer process, regardless of the setting for DB_WRITER_PROCESSES.

See Also:

- *Oracle8i Concepts* for more information on the database writer processes.
- *Oracle8i Designing and Tuning for Performance* for information on setting the DBW*n* parameters.

DBLINK_ENCRYPT_LOGIN

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

When you attempt to connect to a database using a password, Oracle encrypts the password before sending it to the database. DBLINK_ENCRYPT_LOGIN specifies whether attempts to connect to other Oracle servers through database links should use encrypted passwords.

When a connection fails, this parameter has the following effect:

- TRUE: Oracle does not reattempt the connection.
- FALSE: Oracle reattempts the connection using an unencrypted version of the password.

DBWR_IO_SLAVES

Parameter type:	Integer
Parameter class:	Static
Default value:	0
Range of values:	0 to operating system dependent value

DBWR_IO_SLAVES is relevant only on systems with only one database writer process (DBW0). It specifies the number of I/O server processes used by the DBW0 process. 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 DBWR_IO_SLAVES to a non-zero value, the numbers of I/O server processes used by the ARCH process, LGWR process, and Recovery Manager are set to 4.

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" on page 1-16.

DISK_ASYNCH_IO

Parameter type:	Boolean
Parameter class:	Static
Default value:	TRUE
Range of values:	TRUE | FALSE

DISK_ASYNCH_IO controls whether I/O to datafiles, control files, and logfiles are asynchronous (that is, whether parallel server processes can overlap I/O requests with CPU processing during table scans). If your platform supports asynchronous I/O to disk, Oracle Corporation 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 DISK_ASYNCH_IO to FALSE, then you should also set DBWR_IO_SLAVES to a value other than its default of zero in order to simulate asynchronous I/O.

See Also:

- "DBWR_IO_SLAVES" on page 1-34.
- *Oracle8i Designing and Tuning for Performance* for more information on asynchronous I/O and on setting this parameter.

DISTRIBUTED_TRANSACTIONS

Parameter type:	Integer
Parameter class:	Static
Default value:	Operating system dependent
Range of values:	0 to the value of the TRANSACTIONS parameter

DISTRIBUTED_TRANSACTIONS is relevant only if you are using Oracle's Distributed Systems feature. It specifies the maximum number of distributed transactions in which this database can participate at one time. The value of this parameter cannot exceed the value of the parameter TRANSACTIONS.

If network failures occur at an abnormally high rate, causing many in-doubt transactions, you can decrease this parameter's value temporarily. Doing so limits the number of concurrent distributed transactions, which then reduces the number of in-doubt transactions. The reduction of in-doubt transactions, in turn, reduces the amount of blocked data and possible heuristic decision making.

If you set DISTRIBUTED_TRANSACTIONS to 0:

- No distributed transactions are allowed for the database.
- The recovery (RECO) process, which resolves failures involving distributed transactions, does not start when the instance starts up.

See Also: *Oracle8i Distributed Database Systems* for more information on setting this parameter.

DML_LOCKS

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived: (4 * TRANSACTIONS)
Range of values:	20 to unlimited. A setting of 0 disables enqueues.
Oracle Parallel Server:	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). DML_LOCKS specifies the maximum number of DML locks—one for each table modified in a transaction. The 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 per 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 DML_LOCKS to 0, enqueues are disabled and performance is slightly increased. However, you cannot use DROP TABLE, CREATE INDEX, or explicit lock statements such as LOCK TABLE IN EXCLUSIVE MODE.

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:

- *Oracle8i Concepts* for a discussion of lock and enqueue resources needed for parallel DML.
- *Oracle8i Parallel Server Administration, Deployment, and Performance*, *Oracle8i Concepts*, and *Oracle8i Distributed Database Systems* for more information on data concurrency.

ENQUEUE_RESOURCES

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived from SESSIONS parameter
Range of values:	10 to unlimited

ENQUEUE_RESOURCES sets the number of resources that can be concurrently locked by the lock manager. An **enqueue** is a sophisticated locking mechanism that permits several concurrent processes to share known resources to varying degrees. Any object that can be used concurrently can be protected with enqueues. For example, Oracle allows varying levels of sharing on tables: two processes can lock a table in share mode or in share update mode.

One difference between enqueues and latches is that latches do not entail an ordered queue of waiting processes as do enqueues. Processes waiting for latches can either use timers to wake up and retry or (in multiprocessors) spin.

The default value of ENQUEUE_RESOURCES is derived from the SESSIONS parameter and is usually adequate, as long as its value is greater than DML_LOCKS + 20. For three or fewer sessions, the default value is 20. For 4 to 10 sessions, the default value is ((SESSIONS - 3) * 5) + 20. For more than 10 sessions, it is ((SESSIONS - 10) * 2) + 55.

If you explicitly set ENQUEUE_RESOURCES to a value higher than DML_LOCKS + 20, then Oracle uses the value you provide.

Caution: **Increase this parameter only if Oracle returns an error specifying that enqueues are exhausted. If you have many shared resources, increase the value to one per resource (regardless of the number of sessions or cursors using that resource), not one per lock.**

See Also:

- *Oracle8i Parallel Server Administration, Deployment, and Performance.*
- *Oracle8i Concepts.*
- *Oracle8i Distributed Database Systems.*

EVENT

Parameter type:	String
Parameter class:	Static
Default value:	None

EVENT is a parameter used only to debug the system. Do not alter the value of this parameter except at the direction of Oracle technical support personnel.

FAST_START_IO_TARGET

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM.
Default value:	All the buffers in the cache
Range of values:	1000 to all buffers in the cache. A setting of 0 disables limiting recovery I/Os.
Oracle Parallel Server:	Multiple instances can have different values, and you can change the values at runtime.

FAST_START_IO_TARGET (available only with the Oracle Enterprise Edition) specifies the number of I/Os that should be needed during crash or instance recovery. It gives you more precise control over the duration of recovery than DB_BLOCK_MAX_DIRTY_TARGET.

When you set this parameter, DBW*n* writes dirty buffers out more aggressively, so that the number of blocks that must be processed during recovery stays below the value specified in the parameter. However, this parameter does not impose a hard limit on the number of recovery I/Os. Under transient workload situations, the number of I/Os needed during recovery may be greater than the value specified in this parameter. In such situations, DBW*n* will not slow down database activity.

Smaller values for this parameter result in faster recovery times. This improvement in recovery performance is achieved at the expense of additional writing activity during normal processing.

Setting this parameter to 0 disables fast-start checkpointing, which is the mechanism that limits the number of I/Os that need to be performed during recovery. All other writing activity is unaffected.

Note: Recovery I/O can also be limited by setting the LOG_CHECKPOINT_TIMEOUT or 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:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- *Oracle8i Backup and Recovery Guide* and V$INSTANCE_RECOVERY on page 3-51 for information on fast-start checkpointing.

FAST_START_PARALLEL_ROLLBACK

Parameter type:	String
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	LOW
Range of values:	FALSE | LOW | HIGH

FAST_START_PARALLEL_ROLLBACK determines the maximum number of processes that can exist for performing parallel rollback. This parameter is useful on systems in which some or all of the transactions are long running.

- FALSE indicates that parallel rollback is disabled
- LOW limits the number of rollback processes to 2 * CPU_COUNT
- HIGH limits the number of rollback processes to 4 * CPU_COUNT

FIXED_DATE

Parameter type:	String
Syntax:	YYYY-MM-DD-HH24:MI:SS (or the default Oracle date format)
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	None

FIXED_DATE lets you set a constant date that SYSDATE will always return instead of the current date. 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.

Specify the value without quotation marks or with double quotation marks. **Do not use single quotation marks.**

GC_DEFER_TIME

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	10
Range of values:	Any positive integer
Oracle Parallel Server:	Multiple instances can have different values.

GC_DEFER_TIME is an Oracle Parallel Server parameter that specifies the time (in 100ths of a second) that the server waits, or "defers," before responding to forced-write requests for hot blocks from other instances. Specifying the GC_DEFER_TIME parameter makes it more likely that buffers will be properly cleaned out before being written. Doing so makes the buffers easier for other instances to read and also improves the chance of hot blocks being used multiple times within an instance between forced writes.

Setting the value to 0 disables the feature. In this case Oracle does not defer forced-write requests at all.

GC_FILES_TO_LOCKS

Parameter type:	String
Syntax:	GC_FILES_TO_LOCKS = '{file_list=lock_count[!blocks][R][EACH])[:...])' Spaces are not allowed within the quotation marks.
Parameter class:	Static
Default value:	None
Oracle Parallel Server:	You must set this parameter for every instance, and multiple instances must have identical values. To change the value, you must shut down all instances in the cluster, change the value for each instance, and then start up each instance.

GC_FILES_TO_LOCKS is an Oracle Parallel Server parameter that has no effect on an instance running in exclusive mode. It controls the mapping of parallel cache management (PCM) locks to datafiles.

The syntax elements have the following meaning:

`file_list` is one or more datafiles listed by their file numbers, or ranges of file numbers, with comma separators:

```
filenumber[-filenumber][,filenumber[-filenumber]]...
```

To find the correspondence between filenames and file numbers, query FILE_NAME and FILE_ID columns of the data dictionary view "DBA_DATA_FILES" on page 2-104.

By default, any datafiles that you do not specify in this parameter are covered by releasable locks. For information on releasable locks, see "GC_RELEASABLE_LOCKS" on page 1-41.

`lock_count` is the number of PCM locks assigned to *file_list*. By default these locks are fixed. If you set *lock_count* to 0, then Oracle uses fine-grain locking for these files and takes locks as needed from the pool of releasable locks

!blocks	specifies the number of contiguous blocks covered by one lock. The default is noncontiguous blocks.
R	indicates that these locks are releasable and are drawn as needed from the pool of releasable locks. If no blocks from *file_list* are in the buffer cache, then no locks are taken from the releasable pool.
EACH	indicates that each datafile in *file-list* is assigned a separate set of *lock_count* PCM locks.

The value of the parameter should be set to cover as many files as possible. Therefore, to avoid performance problems, you should always change GC_FILES_TO_LOCKS when the size of datafiles change or when new datafiles are added. Doing so requires you to shut down and restart your Parallel Server.

If the number of PCM locks allocated to a datafile is less than or equal to the number of blocks in a datafile, each of these locks will cover a number of contiguous blocks within the datafile equal to *!blocks*. If the number of PCM locks assigned to the datafile is larger than its number of blocks, resources will be wasted, because some locks will not be covering any blocks.

A colon (:) separates each clause that assigns a number of PCM locks to *file_list*.

See Also: *Oracle8i Parallel Server Administration, Deployment, and Performance* for more information on setting this parameter.

GC_RELEASABLE_LOCKS

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived: value of DB_BLOCK_BUFFERS
Range of values:	50 or more
Oracle Parallel Server:	Multiple instances can have different values, and you can change the values as needed.

This parameter is specific to the Oracle Parallel Server in shared mode. Lock elements can be fixed or releasable. Fixed lock elements are used by hashed parallel cache management (PCM) locks, in which the lock element name is preassigned. Releasable lock elements are used with fine-grain locking. If you set the GC_RELEASABLE_LOCKS parameter, then Oracle uses its value to allocate space for fine-grain locking. The maximum value is imposed only by memory restrictions.

See Also: *Oracle8i Parallel Server Administration, Deployment, and Performance* for more information on setting this parameter.

GC_ROLLBACK_LOCKS

Parameter type:	String
Syntax:	GC_ROLLBACK_LOCKS = '{rs_list=lock_count[!blocks][R][EACH])[:...])' Spaces are not allowed within the quotation marks.
Parameter class:	Static
Default value:	0-128=32!8REACH
Oracle Parallel Server:	You must set this parameter for every instance, and multiple instances must have identical values. To change the value, you must shut down all instances in the cluster, change the value for each instance, and then start up each instance.

GC_ROLLBACK_LOCKS is an Oracle Parallel Server parameter that specifies, for each rollback segment, the number of distributed locks available for simultaneously modified rollback segment blocks. The default is adequate for most applications.

These instance locks are acquired in exclusive mode by the instance that acquires the rollback segment. The locks force the instance to write rollback segment blocks to disk when another instance needs a read-consistent version of a block.

The syntax elements have the following meaning:

rs_list	is one or more rollback segments listed by their segment numbers, or ranges of segment numbers, with comma separators: segnumber[-segnumber][,segnumber[-segnumber]]... To find the correspondence between segment names and segment numbers, query SEGMENT_NAME and SEGMENT_ID columns of the data dictionary view "DBA_ROLLBACK_SEGS" on page 2-127. By default, any rollback segments that you do not specify in this parameter are covered by releasable locks. For information on releasable locks, see "GC_RELEASABLE_LOCKS" on page 1-41.
lock_count	is the number of PCM locks assigned to *rs_list*. By default these locks are fixed. If you set *lock_count* to 0, then Oracle uses fine-grain locking for these segments and takes locks as needed from the pool of releasable locks
!blocks	specifies the number of contiguous blocks covered by one lock. The default is noncontiguous blocks.
R	indicates that these locks are releasable and are drawn as needed from the pool of releasable locks. If no blocks from *rs_list* are in the buffer cache, then no locks are taken from the releasable pool.

EACH	indicates that each rollback segment in *rs_list* is assigned a separate set of *lock_count* PCM locks.

A colon (:) separates each clause that assigns a number of PCM locks to *rs_list*.

See Also: *Oracle8i Parallel Server Administration, Deployment, and Performance* for more information on setting this parameter.

GLOBAL_NAMES

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM.
Default value:	TRUE
Range of values:	TRUE | FALSE

GLOBAL_NAMES specifies whether a database link is required to have the same name as the database to which it connects. If the value of GLOBAL_NAMES is FALSE, then no check is performed. If you use or plan to use distributed processing, Oracle Corporation recommends that you set this parameter to TRUE to ensure the use of consistent naming conventions for databases and links in a networked environment.

See Also: *Oracle8i Distributed Database Systems* for more information on setting this parameter.

HASH_AREA_SIZE

Parameter type:	Integer
Parameter class:	Dynamic. Scope= ALTER SESSION.
Default value:	Derived: 2 * value of SORT_AREA_SIZE parameter
Range of values:	0 to operating system dependent value

HASH_AREA_SIZE is relevant to parallel execution operations and to the query portion of DML or DDL statements. It specifies the maximum amount of memory, in bytes, to be used for hash joins.

See Also:

- *Oracle8i Concepts* for information on hash joins in general.
- *Oracle8i Designing and Tuning for Performance* for information on calculating an appropriate value for this parameter.

HASH_JOIN_ENABLED

Parameter type:	Boolean
Parameter class:	Dynamic. Scope= ALTER SESSION.
Default value:	TRUE
Range of values:	TRUE | FALSE

HASH_JOIN_ENABLED specifies whether the optimizer should consider using a hash join as a join method. When set to FALSE, hashing is not available as a join method. When set to TRUE, the optimizer compares the cost of a hash join with other types of joins, and chooses hashing if it gives the best cost. Oracle Corporation recommends that you set this parameter to TRUE for all data warehousing applications.

HASH_MULTIBLOCK_IO_COUNT

Parameter type:	Integer
Parameter class:	Dynamic. Scope= ALTER SESSION, ALTER SYSTEM.
Default value:	Query dependent. Appears as 0 in V$PARAMETER.
Range of values:	Operating system dependent

HASH_MULTIBLOCK_IO_COUNT specifies how many sequential blocks a hash join reads and writes in one I/O. The maximum value is operating system dependent. It is always less than the maximum I/O size of the operating system expressed as Oracle blocks (MAX_IO_SIZE / DB_BLOCK_SIZE).

You need not set or change the value of this parameter, because Oracle computes the value individually for every query. If you let Oracle do the automatic computation, the value of the parameter appears as 0 in the V$PARAMETER dynamic performance view.

Oracle Corporation does not recommend that you set or change the value of this parameter. If you must set it to investigate its effect on performance, make sure that the following formula remains true:

```
R / M <= Po2(M/C)
```

where:

R = number of bytes in the smaller relation to be joined. The number of bytes is the product of the size of each column in the smaller relation times the number of rows in that relation.

```
M = HASH_AREA_SIZE * 0.9
C = HASH_MULTIBLOCK_IO_COUNT * DB_BLOCK_SIZE
```

Po2(n) = a function that returns the largest power of 2 that is smaller than its argument.

Note: If you are using Oracle's multi-threaded server architecture, Oracle ignores any value you set for this parameter, and instead uses a value of 1.

HI_SHARED_MEMORY_ADDRESS

Parameter type:	Integer
Parameter class:	Static
Default value:	0

HI_SHARED_MEMORY_ADDRESS specify the starting address at runtime of the system global area (SGA). It is ignored on platforms that specify the SGA's starting address at linktime.

On 64-bit platforms, use HI_SHARED_MEMORY_ADDRESS to specify the high-order 32 bits of a 64-bit address. Use SHARED_MEMORY_ADDRESS to specify the low-order 32 bits of the address (see "SHARED_MEMORY_ADDRESS" on page 1-110). If both parameters are 0 or unspecified, the SGA address defaults to a platform-specific location.

HS_AUTOREGISTER

Parameter type:	Boolean
Parameter class:	Dynamic. Scope= ALTER SYSTEM.
Default value:	TRUE
Range of values:	TRUE | FALSE

HS_AUTOREGISTER enables or disables automatic self-registration of Heterogeneous Services (HS) agents. When enabled, information is uploaded into the server's data dictionary to describe a previously unknown agent class or a new agent version.

Oracle Corporation recommends that you use 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.

See Also: *Oracle8i Distributed Database Systems* for more information on HS agents.

IFILE

Parameter type:	String
Syntax:	IFILE = parameter_file_name
Parameter class:	Static

Default value:	None
Range of values:	Valid parameter filenames
Oracle Parallel Server:	Multiple instances can have different values.

Use IFILE to embed another parameter file within the current parameter file. 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.

INSTANCE_GROUPS

Parameter type:	String
Syntax	INSTANCE_GROUPS = group_name [, group_name ...]
Parameter class:	Static
Default value:	None
Range of values:	One or more instance group name, separated by commas
Oracle Parallel Server:	Multiple instances can have different values.

INSTANCE_GROUPS is an Oracle Parallel Server parameter that you can specify only in parallel mode. Used in conjunction with the PARALLEL_INSTANCE_GROUP parameter, it lets you restrict parallel query operations to a limited number of instances. See "PARALLEL_INSTANCE_GROUP" on page 1-94.

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: *Oracle8i Parallel Server Administration, Deployment, and Performance* for more information on parallel query execution in an Oracle Parallel Server environment.

INSTANCE_NAME

Parameter type:	String
Syntax	`INSTANCE_NAME = instance_id`
Parameter class:	Static
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.
Allowable values:	Any alphanumeric characters

In an Oracle Parallel Server 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.

In a single-instance database system, the instance name is usually the same as the database name.

See Also: *Oracle8i Parallel Server Administration, Deployment, and Performance and Net8 Administrator's Guide* for more information.

INSTANCE_NUMBER

Parameter type:	Integer
Parameter class:	Static
Default value:	Lowest available number (depends on instance startup order and on the `INSTANCE_NUMBER` values assigned to other instances). 0 if not configured for Oracle Parallel Server.
Range of values:	1 to maximum number of instances specified when the database was created
Oracle Parallel Server:	Multiple instances need not specify, but must have different values if they do specify. You can change the values if necessary.

INSTANCE_NUMBER is an Oracle Parallel Server parameter that can be specified in parallel mode or exclusive mode. It specifies a unique number that maps the instance to one free list group for each database object created with storage parameter FREELIST GROUPS.

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.

See Also: *Oracle8i Parallel Server Administration, Deployment, and Performance* for more information.

JAVA_MAX_SESSIONSPACE_SIZE

Parameter type:	Integer
Parameter class:	Static
Default value:	0
Range of values:	0 to 4 GB

Java session space is the memory that holds Java state from one database call to another. JAVA_MAX_SESSIONSPACE_SIZE specifies in bytes the maximum amount of session space made available to a Java program executing in the server. When a user's session-duration Java state attempts to exceed this amount, the Java virtual machine kills the session with an out-of-memory failure.

See Also:

- *Oracle8i Java Developer's Guide.*
- "JAVA_SOFT_SESSIONSPACE_LIMIT" on page 1-49.

JAVA_POOL_SIZE

Parameter type:	String
Parameter class:	Static
Default value:	20000K
Range of values:	1000000 to 1000000000

JAVA_POOL_SIZE specifies the size in bytes of the Java pool, from which the Java memory manager allocates most Java state during runtime 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.

See Also: *Oracle8i Java Developer's Guide* for information on adjusting this parameter.

JAVA_SOFT_SESSIONSPACE_LIMIT

Parameter type:	Integer
Parameter class:	Static
Default value:	0
Range of values:	0 to 4 GB

Java session space is the memory that holds Java state from one database call to another. 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. When a user's session-duration Java state exceeds this size, Oracle generates a warning that goes into the trace files.

See Also:

- *Oracle8i Java Developer's Guide* for more information on this parameter.
- "JAVA_MAX_SESSIONSPACE_SIZE" on page 1-48.

JOB_QUEUE_INTERVAL

Parameter type:	Integer
Parameter class:	Static
Default value:	60
Range of values:	1 to 3600
Oracle Parallel Server:	Multiple instances can have different values.

JOB_QUEUE_INTERVAL is relevant in replication environments. It specifies in seconds how frequently each SNP*n* background process of the instance wakes up.

See Also: *Oracle8i Replication* for more information on setting this parameter.

JOB_QUEUE_PROCESSES

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	0
Range of values:	0 to 36
Oracle Parallel Server:	Multiple instances can have different values.

JOB_QUEUE_PROCESSES is relevant in replication environments. It specifies the number of SNP*n* job queue processes per instance (SNP0, ... SNP9, SNPA, ... SNPZ). Job queue processes process requests created by DBMS_JOB.

Some job queue requests are created automatically. An example is refresh support for materialized views. If you wish to have your materialized views updated automatically, you must set JOB_QUEUE_PROCESSES to a value of one or higher.

See Also: *Oracle8i Replication* and *Oracle8i Data Warehousing Guide* for more information on managing materialized views.

LARGE_POOL_SIZE

Parameter type:	String
Syntax	LARGE_POOL_SIZE = integer [K | M]
Parameter class:	Static
Default value:	0 if ■ The pool is not required by parallel execution **and** ■ DBWR_IO_SLAVES is not set. Otherwise, derived from the values of PARALLEL_MAX_SERVERS, PARALLEL_THREADS_PER_CPU, PARALLEL_SERVER_INSTANCES, MTS_DISPATCHERS, and DBWR_IO_SLAVES.
Range of values:	600K to at least 2 GB (actual maximum is operating system specific)

LARGE_POOL_SIZE lets you specify the size in bytes of the large pool allocation heap. The large pool allocation heap is used in multi-threaded 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 PARALLEL_AUTOMATIC_TUNING is set to TRUE.)

CAUTION: When Oracle derives a default value, it adds 250K per session for the multi-threaded server if MTS_DISPATCHERS is configured. The final derived value also includes a port-specific amount of memory for backup I/O buffers. The total derived default value can either be too large to allocate or can cause performance problems. In that case, set LARGE_POOL_SIZE to a number sufficiently small so that the database can start.

You can specify the value of this parameter using a numeric value or number, optionally followed by K or M to specify kilobytes or megabytes, respectively.

See Also: *Oracle8i Designing and Tuning for Performance* and *Oracle8i Migration* for more information on setting this parameter.

LICENSE_MAX_SESSIONS

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	0
Range of values:	0 to number of session licenses
Oracle Parallel Server:	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.

LICENSE_MAX_SESSIONS specifies the maximum number of concurrent user sessions allowed. When this limit 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 LICENSE_SESSIONS_WARNING (see "LICENSE_SESSIONS_WARNING" on page 1-52).

Do not enable both concurrent usage licensing and user licensing. Set either LICENSE_MAX_SESSIONS or LICENSE_MAX_USERS to zero.

See Also: *Oracle8i Administrator's Guide* for more information on setting this parameter.

LICENSE_MAX_USERS

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	0

Range of values:	0 to number of user licenses
Oracle Parallel Server:	Multiple instances should have the same values. If different instances specify different values for this parameter, the value of the first instance to mount the database takes precedence.

LICENSE_MAX_USERS specifies the maximum number of users you can create in the database. When you reach this limit, you cannot create more users. You can, however, increase the limit.

Do not enable both concurrent usage (session) licensing and user licensing. Set either LICENSE_MAX_SESSIONS or LICENSE_MAX_USERS to zero.

See Also: *Oracle8i Administrator's Guide* for more information on setting this parameter.

LICENSE_SESSIONS_WARNING

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	0
Range of values:	0 to value of LICENSE_MAX_SESSIONS parameter
Oracle Parallel Server:	Multiple instances can have different values.

LICENSE_SESSIONS_WARNING specifies a warning limit on the number of concurrent user sessions. When this limit is reached, additional users can connect, but Oracle writes a message in the alert file 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" on page 1-51).

See Also: *Oracle8i Administrator's Guide* for more information on setting this parameter.

LM_LOCKS

Parameter type:	Integer
Parameter class:	Static
Default value:	12000

Range of values: 512 to a number limited by

- The shared memory available in the operating system,
- The maximum size of contiguous shared memory segment, or
- The address space.

Oracle Parallel Server: You must set this parameter for every instance, and multiple instances must have the same value.

LM_LOCKS is an Oracle Parallel Server parameter that specifies the number of locks that will be configured for the lock manager. The number of locks can be represented by the following equation:

```
L = R + (R*(N - 1))/N
```

where

L = the total number of locks

R = the number of resources (as determined by the LM_RESS parameter). See "LM_RESS" on page 1-53.

N = the total number of nodes

Oracle configures locks per lock manager instance. Thus the value of LM_LOCKS must be the same for all lock manager instances. With DBA locking, up to 2 * GC_RELEASABLE_LOCKS could be required, at most.

See Also: *Oracle8i Parallel Server Administration, Deployment, and Performance* for more information on setting this parameter.

LM_RESS

Parameter type: Integer

Parameter class: Static

Default value: 6000

Range of values: 256 to a number limited by

- The shared memory available in the operating system,
- The maximum size of contiguous shared memory segment, or
- The address space.

Oracle Parallel Server: You must set this parameter for every instance, and multiple instances must have the same value.

LM_RESS controls the number of resources that can be locked by each lock manager instance. These resources include lock resources allocated for DML, DDL (data dictionary locks), data dictionary, and library cache locks plus the file and log management locks.

In most cases you should specify a value less than 2 * DML_LOCKS plus an overhead of about 20 locks.

See Also: *Oracle8i Parallel Server Administration, Deployment, and Performance* for more information on calculating an appropriate value for this parameter.

LOCAL_LISTENER

Parameter type:	String
Syntax:	LOCAL_LISTENER = network_name
Parameter class:	Static
Default value:	(ADDRESS = (PROTOCOL=TCP)(HOST=)(PORT=1521))

LOCAL_LISTENER specifies a network name that resolves to an address or address list of Net8 local listeners (that is, listeners that are running on the same machine as this instance). The address or address list is specified in the TNSNAMES.ORA file or other address repository as configured for your system.

See Also:

- *Oracle8i Concepts* for more information about instances, listener processes, and dispatcher processes.
- *Net8 Administrator's Guide* and your operating system specific Oracle documentation for more information about specifying network addresses for the protocols on your system.

LOCK_NAME_SPACE

Parameter type:	String
Syntax:	LOCK_NAME_SPACE = namespace
Parameter class:	Static
Default:	None
Range of values:	Up to 8 alphanumeric characters. No special characters allowed.

LOCK_NAME_SPACE specifies the namespace that the distributed lock manager (DLM) uses to generate lock names. Consider setting this parameter if a standby or clone database has the same database name on the same cluster as the primary database.

LOCK_SGA

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

LOCK_SGA locks the entire SGA into physical memory. 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.

LOG_ARCHIVE_DEST

Parameter type:	String
Syntax:	LOG_ARCHIVE_DEST = filespec
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	None
Range of values:	Any valid path or device name, except raw partitions
Oracle Parallel Server:	Multiple instances can have different values.

Note: 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 command ARCHIVE LOG START *filespec* for automatic archiving, where *filespec* is the new archive destination. To permanently change the destination, use the command ALTER SYSTEM SET LOG_ARCHIVE_DEST = *filespec*, where *filespec* is the new archive destination.

See Also:

- *Oracle8i Backup and Recovery Guide*.
- "LOG_ARCHIVE_DUPLEX_DEST" on page 1-58, "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-60, and "V$ARCHIVE_DEST" on page 3-4 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 filename using LOG_ARCHIVE_DEST.

LOG_ARCHIVE_DEST_*n*

Parameter type	String
Syntax:	LOG_ARCHIVE_DEST_[1 | 2 | 3 | 4 | 5] = "null_string" | ((SERVICE=tnsnames_service | LOCATION=local_pathname) [MANDATORY | OPTIONAL] [REOPEN[=integer]])
Parameter class:	Dynamic. Scope = ALTER SYSTEM, ALTER SESSION.
Default value:	None
Range of values:	Valid keyword definitions (see syntax below)

Note: This parameter is valid only if you have installed Oracle Enterprise Edition. You may continue to use LOG_ARCHIVE_DEST if you have installed Oracle Enterprise Edition. However, you cannot use both LOG_ARCHIVE_DEST_*n* and LOG_ARCHIVE_DEST, as they are not compatible.

The LOG_ARCHIVE_DEST_*n* parameters (where *n* = 1,2,3,4,5) define up to five archive log destinations. The parameter integer suffix is defined as the **handle** displayed by the V$ARCHIVE_DEST dynamic performance view.

- SERVICE specifies a standby destination. Net8 (IPC or TCP) transmits the archivelog. A standby instance must be associated with the destination. tnsnames_service corresponds to an appropriate service-name in TNSNAMES.ORA.
- LOCATION specifies a local file-system destination. You must specify this parameter for at least one destination.
- MANDATORY specifies that archiving to the destination must succeed before the REDO logfile can be made available for reuse.
- OPTIONAL specifies that successful archiving to the destination is not required before the REDO log file can be made available for reuse. If the "must succeed count" (LOG_ARCHIVE_MIN_SUCCEED_DEST) is met, the REDO logfile is marked for reuse. This is the default.
- REOPEN specifies an interval of time (in seconds) that must pass after an error has been encountered during archiving to the destination before future archives to the destination can be attempted. Future attempts are made when the next REDO logfile is archived. If a destination is MANDATORY, Oracle Corporation recommends that you specify a REOPEN time that reduces the possibility of primary database shutdown due to lack of available online REDO logfiles.

  If you do not specify *integer*, the default value is 300 seconds. If you do not specify REOPEN, the archiver process will never reopen a destination after an error.

See Also: *Oracle8i Backup and Recovery Guide* and *Oracle8i Administrator's Guide* for more information and examples.

LOG_ARCHIVE_DEST_STATE_*n*

Parameter type:	String
Syntax:	LOG_ARCHIVE_DEST_STATE_*n* = {ENABLE | DEFER}
Parameter class:	Dynamic. Scope = ALTER SYSTEM, ALTER SESSION.
Default value:	ENABLE

The LOG_ARCHIVE_DEST_STATE_*n* parameters (where n = 1,2,3,4,5) specify the availability state of the corresponding destination. The parameter suffix (1 through 5) specifies one of the five corresponding LOG_ARCHIVE_DEST_*n* destination parameters.

- ENABLE specifies that a valid log archive destination can be used for a subsequent archiving operation (automatic or manual).
- DEFER specifies that valid destination information and attributes are preserved, but the destination is excluded from archiving operations until re-enabled.

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$ARCHIVE_DEST dynamic performance view shows values in use for the current session. The DEST_ID column of that view corresponds to the "n" archive destination suffix. See "V$ARCHIVE_DEST" on page 3-4.

See Also: *Oracle8i Backup and Recovery Guide* and *Oracle8i Administrator's Guide* for more information and examples.

LOG_ARCHIVE_DUPLEX_DEST

Parameter type:	String
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	None
Range of values:	Either a null string or any valid path or device name, except raw partitions

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.

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).

The default setting of a null string ("") or (' ') indicates that a duplex archive destination does not exist.

See Also:

- "LOG_ARCHIVE_DEST_n" on page 1-56.
- "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-60.
- "V$ARCHIVE_DEST" on page 3-4.

LOG_ARCHIVE_FORMAT

Parameter type:	String
Syntax:	LOG_ARCHIVE_FORMAT = filename
Parameter class:	Static
Default value:	Operating system dependent

Range of values:	Any string that resolves to a valid filename
Oracle Parallel Server:	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

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 is:

```
LOG_ARCHIVE_FORMAT = "LOG%s_%t.ARC"
```

See Also:

- *Oracle8i Backup and Recovery Guide* and *Oracle8i Parallel Server Administration, Deployment, and Performance* for more information on setting this parameter.
- Your operating system specific Oracle documentation for the default value and range of values for LOG_ARCHIVE_FORMAT.

LOG_ARCHIVE_MAX_PROCESSES

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	1
Range of values:	Any integer from 1 to 10 inclusive

LOG_ARCHIVE_MAX_PROCESSES specifies the number of archiver background processes (ARC0 through ARC9) Oracle initially invokes.

- If the LOG_ARCHIVE_START initialization parameter has the value TRUE, this value is evaluated at instance startup.
- Otherwise, this parameter is evaluated when the archiver process is first invoked by SQL*Plus or SQL syntax.

The actual number of archiver processes in use may vary subsequently based on archive workload.

See Also: *Oracle8i Concepts*.

LOG_ARCHIVE_MIN_SUCCEED_DEST

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM.
Default value:	1
Range of values:	1 to 5 if you are using LOG_ARCHIVE_DEST_*n* 1 or 2 if you are using LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_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.

- 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 you are using the LOG_ARCHIVE_DEST_*n* parameters and automatic archiving is enabled, the value of this parameter cannot exceed

  - The total number of destinations, or

  - The number of enabled, valid destinations specified as MANDATORY plus the number of enabled, valid non-standby destinations specified as OPTIONAL.

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, some of the enabled, valid OPTIONAL non-standby destinations are treated as MANDATORY.

You cannot set a value for this parameter dynamically if LOG_ARCHIVE_DEST or LOG_ARCHIVE_DUPLEX_DEST are in use. However, you can switch dynamically from one system to the other using ALTER SYSTEM, as follows:

- Set LOG_ARCHIVE_MIN_SUCCEED_DEST to 1.
- Set the value of LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST to the null string.
- Set the desired number of destinations for the LOG_ARCHIVE_DEST_*n* parameters.
- Reset LOG_ARCHIVE_MIN_SUCCEED_DEST to the desired value.

See Also:

- *Oracle8i Administrator's Guide* for more information on setting this parameter.
- "LOG_ARCHIVE_DEST_n" on page 1-56, "LOG_ARCHIVE_DUPLEX_DEST" on page 1-58, and "V$ARCHIVE_DEST" on page 3-4 for information on related parameters.

LOG_ARCHIVE_START

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE
Oracle Parallel Server:	Multiple instances can have different values.

LOG_ARCHIVE_START is applicable only when you use the redo log in ARCHIVELOG mode. It indicates whether archiving should be automatic or manual when the instance starts up.

- TRUE indicates that archiving is automatic.
- FALSE indicates that the database administrator will archive filled redo log files manually.

The SQL*Plus command ARCHIVE LOG START or STOP overrides this parameter.

In ARCHIVELOG mode, if all online redo log files fill without being archived, Oracle issues an error and suspends instance operations until the necessary archiving is performed. This delay is more likely if you use manual archiving. You can reduce its likelihood by increasing the number of online redo log files.

To use ARCHIVELOG mode while creating a database, set this parameter to TRUE. Normally, a database is created in NOARCHIVELOG mode and then altered to ARCHIVELOG mode after creation.

See Also: *Oracle8i Administrator's Guide* and *Oracle8i Parallel Server Administration, Deployment, and Performance.*

LOG_ARCHIVE_TRACE

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	0
Range of values:	0, 1, 2, 4, 8, 16, 32, 64
Oracle Parallel Server:	Multiple instances can have different values.

LOG_ARCHIVE_TRACE controls output generated by the archivelog process. This process can be initiated by

- An ARC*n* background process (designated as ARC*n* in the output logs)
- An explicit session-invoked foreground process (designated as ARCH in the output logs), or
- The remote file server (RFS) process of Managed Standby.

The valid values have the following meanings:

- 0: Disable archivelog tracing (this is the default)
- 1: Track archival of REDO log file
- 2: Track archival status of each archivelog destination
- 4: Track archival operational phase
- 8: Track archivelog destination activity
- 16: Track detailed archivelog destination activity
- 32: Track archivelog destination parameter modifications
- 64: Track ARC*n* process state activity

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 changes will take effect at the start of the next archivelog operation.

See Also: *Oracle8i Backup and Recovery Guide.*

LOG_BUFFER

Parameter type:	Integer
Parameter class:	Static
Default value:	Operating system specific. Maximum: 500K or 128K * CPU_COUNT, whichever is greater
Range of values:	Operating system dependent

LOG_BUFFER specifies the amount of memory, in bytes, that Oracle uses when buffering redo entries to a redo log file. 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.

In general, larger values for LOG_BUFFER reduce redo log file I/O, particularly if transactions are long or numerous. In a busy system, a value 65536 or higher is reasonable.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- Your operating system specific Oracle documentation for the default value and range of values.

LOG_CHECKPOINT_INTERVAL

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	Operating system dependent
Range of values:	Unlimited
Oracle Parallel Server:	Multiple instances can have different values.

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.

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.

Notes:

- 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 FAST_START_IO_TARGET or 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:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- "FAST_START_IO_TARGET" on page 1-38, "LOG_CHECKPOINT_TIMEOUT" on page 1-64, and "V$INSTANCE_RECOVERY" on page 3-51.

LOG_CHECKPOINT_TIMEOUT

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	Oracle8*i*: 900 seconds. Enterprise Edition: 1800 seconds.
Range of values:	0 to unlimited
Oracle Parallel Server:	Multiple instances can have different values.

LOG_CHECKPOINT_TIMEOUT specifies that the incremental checkpoint is at the position where the last write to the redo log (sometimes called the "tail of the log") was *integer* seconds ago. This parameter also signifies that no buffer will remain dirty (in the cache) for more than *integer* seconds. The value is specified in seconds.

Specifying a value of 0 for the timeout disables time-based checkpoints. Hence, setting the value to 0 is not recommended.

Notes:

- 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 FAST_START_IO_TARGET or 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:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- "FAST_START_IO_TARGET" on page 1-38, "LOG_CHECKPOINT_TIMEOUT" on page 1-64, and "V$INSTANCE_RECOVERY" on page 3-51.

LOG_CHECKPOINTS_TO_ALERT

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

LOG_CHECKPOINTS_TO_ALERT lets you log your checkpoints to the alert file. Doing so is useful for determining whether checkpoints are occurring at the desired frequency.

LOG_FILE_NAME_CONVERT

Parameter type:	String
Syntax:	LOG_FILE_NAME_CONVERT = [()'string1','string2'[)]
	where
	string1 is the pattern of the primary database log filenames
	string2 is the pattern of the standby database log filenames
	You can use single or double quotation marks. The parentheses are optional.
Parameter class:	Static
Default value:	None
Range of values:	Character strings

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. If you add a log file to the primary database, you must add a corresponding file to 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.

Set the value of this parameter to two 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.

You should also use LOG_FILE_NAME_CONVERT to rename the logfiles in the clone controlfile when setting up the clone database during tablespace point-in-time recovery.

See Also: *Oracle8i Backup and Recovery Guide* and *Oracle8i Standby Database Concepts and Administration*.

MAX_COMMIT_PROPAGATION_DELAY

Parameter type:	Integer
Parameter class:	Static
Default value:	700
Range of values:	0 to 90000
Oracle Parallel Server:	You must set this parameter for every instance, and multiple instances must have identical values.

WARNING: Change this parameter only when it is absolutely necessary to see the most current version of the database when doing a query.

MAX_COMMIT_PROPAGATION_DELAY is an Oracle Parallel Server parameter. This initialization parameter should not be changed except under a limited set of circumstances specific to the Oracle Parallel Server.

This parameter specifies the maximum amount of time allowed before the system change number (SCN) held in the SGA of an instance is refreshed by the log writer process (LGWR). It determines whether the local SCN should be refreshed from the lock value when getting the snapshot SCN for a query. Units are in hundredths of seconds.

Under very unusual circumstances involving rapid updates and queries of the same data from different instances, the SCN might not be refreshed in a timely manner. Setting the parameter to zero causes the SCN to be refreshed immediately after a commit. The default value (700 hundredths of a second, or seven seconds) is an upper bound that allows the preferred existing high performance mechanism to remain in place.

See Also: *Oracle8i Parallel Server Administration, Deployment, and Performance* for more information on setting this parameter.

MAX_DUMP_FILE_SIZE

Parameter type:	String
Syntax	MAX_DUMP_FILE_SIZE = {integer | UNLIMITED}
Parameter class:	Dynamic. Scope = ALTER SYSTEM [... DEFERRED], ALTER SESSION.
Default value:	UNLIMITED
Range of values:	0 to unlimited, or UNLIMITED

MAX_DUMP_FILE_SIZE specifies the maximum size of trace files (excluding the alert file). Change this limit if you are concerned that trace files may use too much space.

- A numerical value for MAX_DUMP_FILE_SIZE specifies the maximum size in operating system blocks.
- A number followed by a "K" or "M" suffix specifies the file size in kilobytes or megabytes.
- 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.

See Also: *Oracle8i Administrator's Guide* and *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.

MAX_ENABLED_ROLES

Parameter type:	Integer
Parameter class:	Static
Default value:	20
Range of values:	0 to 148

MAX_ENABLED_ROLES specifies the maximum number of database roles that users can enable, including roles contained within other roles.

The actual number of roles users can enable is 2 plus the value of MAX_ENABLED_ROLES, because each user has two additional roles, PUBLIC and the user's own role. For example, if MAX_ENABLED_ROLES is set to 5, user SCOTT can have seven roles enabled: the five enabled by MAX_ENABLED_ROLES plus PUBLIC and SCOTT.

See Also: *Oracle8i Administrator's Guide* for more information on setting this parameter.

MAX_ROLLBACK_SEGMENTS

Parameter type:	Integer
Parameter class:	Static
Default value:	`MAX(30, TRANSACTIONS/TRANSACTIONS_PER_ROLLBACK_SEGMENT)`
Range of values:	2 to 65535

`MAX_ROLLBACK_SEGMENTS` specifies the maximum size of the rollback segment cache in the SGA. The number specified signifies the maximum number of rollback segments that can be kept online (that is, status of `ONLINE`) simultaneously by one instance.

See Also: *Oracle8i Administrator's Guide* for more information on setting this parameter.

MTS_CIRCUITS

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived: the value of the `SESSIONS` parameter if you are using the multi-threaded server architecture; 0 otherwise.

`MTS_CIRCUITS` specifies the total number of virtual circuits that are available for inbound and outbound network sessions. It is one of several parameters that contribute to the total SGA requirements of an instance.

See Also:

- *Oracle8i Concepts* for more information on memory structures and processes.
- "DB_BLOCK_BUFFERS" on page 1-24 and "TRANSACTIONS" on page 1-119, which also contribute to SGA requirements.

MTS_DISPATCHERS

Parameter type:	String
Syntax:	`MTS_DISPATCHERS = 'dispatch_clause'`

dispatch_clause::=

```
(PROTOCOL = protocol) |
(ADDRESS = address) |
(DESCRIPTION = description )
[options_clause]
```

options_clause::=

```
(DISPATCHERS = integer |
 SESSIONS = integer |
 CONNECTIONS = integer |
 TICKS = seconds |
 POOL = { 1 | ON | YES | TRUE | BOTH |
   ({IN|OUT} = ticks) | 0 | OFF | NO | FALSE |
   ticks} |
 MULTIPLEX = {1 | ON | YES | TRUE |
   0 | OFF |NO | FALSE | BOTH | IN | OUT} |
 LISTENER = tnsname |
 SERVICE = service |
 INDEX = integer)
```

Parameter class: Dynamic. Scope = ALTER SYSTEM .

Default value: None

MAX_DISPATCHERS configures dispatcher processes in the multi-threaded server (MTS) architecture. The parsing software supports a name-value syntax to enable the specification of attributes in a position-independent case-insensitive manner. For example:

```
MTS_DISPATCHERS = "(PROTOCOL=TCP)(DISPATCHERS=3)"
```

Specify only one of the following attributes: ADDRESS, DESCRIPTION, or PROTOCOL. If you specify either ADDRESS or DESCRIPTION, you can specify additional network attributes. Doing so supports multi-homed hosts.

ADDRESS (ADD or ADDR)	is the network address (in Net8 syntax) of the end point which the dispatchers will listen on. (Includes the protocol.)
DESCRIPTION (DES or DESC)	is the network description (in Net8 syntax, including the protocol) of the end point on which the dispatchers will listen.
PROTOCOL (PRO or PROT)	is the network protocol for which the dispatchers will generate a listening end point.

The following attributes are optional:

DISPATCHERS (DIS or DISP)	is an integer specifying the initial number of dispatchers to start. Default is 1.
SESSIONS (SES or SESS)	specifies the maximum number of network sessions to allow for each dispatcher. The default is set by Net8 and is platform specific.
CONNECTIONS (CON or CONN)	is an integer specifying the maximum number of network connections to allow for each dispatcher. The default is set by Net8 and is platform specific.
TICKS (TIC or TICK)	specifies the size of a network tick in seconds. The default is set by Net8 and is platform specific.
POOL (POO)	enables the Net8 "Connection Pooling" feature. ■ An integer indicates that Connection Pooling is enabled for both incoming and outgoing network connections. The number specified is the timeout in ticks for both incoming and outgoing network connections. ■ ON, YES, TRUE, and BOTH indicate that Connection Pooling is enabled for both incoming and outgoing network connections. The default timeout (set by Net8) will be used for both incoming and outgoing network connections. ■ IN indicates that Connection Pooling is enabled for incoming network connections. The default timeout (set by Net8) will be used for incoming network connections. ■ OUT indicates that Connection Pooling is enabled for outgoing network connections. The default timeout (set by Net8) will be used for outgoing network connections. ■ NO, OFF, and FALSE indicate that Connection Pooling is disabled for both incoming and outgoing network connections. This is the default.
MULTIPLEX (MUL or MULT)	enables the Net8 "Network Session Multiplex" 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 (LIS, LIST)	specifies the network name of an address or address list of the Net8 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 overrides the LOCAL_LISTENER parameter. See "LOCAL_LISTENER" on page 1-54. POOL can also be assigned a name-value string such as: "(IN=10)", "(OUT=20)", or "(IN=10)(OUT=20)". In such cases: ■ If an IN numeric value is specified, then Connection Pooling is enabled for incoming connections, and the number specified is the timeout in ticks for incoming network connections. ■ If an OUT numeric value is specified, then Connection Pooling is enabled for outgoing network connections, and the number specified is the timeout in ticks for outgoing network connections. ■ If the numeric value of a specified timeout is 0 or 1, then the default value (set by Net8) will be used.
SERVICE (SER, SERV)	specifies the service name which the dispatchers register with the Net8 listeners.
INDEX	Use this parameter in an ALTER SYSTEM SET MTS_DISPATCHER statement to indicate which dispatcher you want to modify. (If you specify INDEX in the initialization parameter file, Oracle 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 process) to 1 less than the total number of dispatchers you define. For example, if you specify 3 dispatchers in the initialization parameter file, you would modify the third dispatcher by specify INDEX=2 in the ALTER SYSTEM statement. You could also add an additional dispatcher in the ALTER SYSTEM statement by specifying INDEX=3.

See Also: *Net8 Administrator's Guide* for more information on setting this parameter.

MTS_MAX_DISPATCHERS

Parameter type:	Integer
Parameter class:	Static
Default value:	5 or the number of dispatchers configured, whichever is greater

Range of values:	Operating system dependent

MTS_MAX_DISPATCHERS specifies the maximum number of dispatcher processes allowed to be running simultaneously. The default value applies only if dispatchers have been configured for the system.

The value of MTS_MAX_DISPATCHERS should at least equal the maximum number of concurrent sessions divided by the number of connections per dispatcher. For most systems, a value of 250 connections per dispatcher provides good performance.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- Your operating system specific Oracle documentation for the default value and range of values.

MTS_MAX_SERVERS

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived from MTS_SERVERS (either 20 or 2*MTS_SERVERS)
Range of values:	Operating system dependent

MTS_MAX_SERVERS specifies the maximum number of shared server processes allowed to be running simultaneously. If artificial deadlocks occur too frequently on your system, you should increase the value of MTS_MAX_SERVERS.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- *Oracle8i Concepts* for information on artificial deadlocks.
- Your operating system specific Oracle documentation for the default value and range of values.

MTS_SERVERS

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	1 if you are using multi-threaded server architecture; 0 otherwise.

Range of values:	Operating system dependent

MTS_SERVERS specifies the number of server processes that you want to create when an instance is started up. If system load decreases, this minimum number of servers is maintained. Therefore, you should take care not to set MTS_SERVERS to too high at system startup.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.

MTS_SESSIONS

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived: the lesser of MTS_CIRCUITS and SESSIONS - 5
Range of values:	0 to SESSIONS - 5

MTS_SESSIONS specifies the total number of multi-threaded server architecture user sessions to allow. Setting this parameter enables you to reserve user sessions for dedicated servers.

See Also: *Oracle8i Concepts* for more information on memory structures and processes.

NLS_CALENDAR

Parameter type:	String
Syntax:	NLS_CALENDAR = "calendar_system"
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	None
Range of values:	Any valid calendar format name

NLS_CALENDAR specifies which calendar system Oracle uses. It can have one of the following values:

- Arabic Hijrah
- English Hijrah
- Gregorian
- Japanese Imperial
- Persian
- ROC Official (Republic of China)
- Thai Buddha

For example, suppose NLS_CALENDAR is set to "Japanese Imperial", the date format is "E YY-MM-DD". ("E" is the date format element for the abbreviated era name.) If the date is May 15, 1997, then the SYSDATE is displayed as follows:

```
SELECT SYSDATE FROM DUAL;
SYSDATE
--------
H 09-05-15
```

See Also: *Oracle8i National Language Support Guide* for a listing of available calendar systems.

NLS_COMP

Parameter type:	String
Syntax:	NLS_COMP = {BINARY | ANSI}
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	BINARY

Normally, comparisons in the WHERE clause of queries are binary unless you specify the NLSSORT function. By setting this parameter to ANSI, you indicate that comparisons in the WHERE clause of queries should use the linguistic sort specified in the NLS_SORT parameter. You must also define an index on the column for which you want linguistic sorts.

See Also: *Oracle8i National Language Support Guide* for more information on setting this parameter.

NLS_CURRENCY

Parameter type:	String
Syntax:	`NLS_CURRENCY = currency_symbol`
Parameter class:	Dynamic. Scope = `ALTER SESSION`.
Default value:	Derived from `NLS_TERRITORY`
Range of values:	Any valid character string, with a maximum of 10 bytes (not including null)

`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`.

See Also:

- *Oracle8i National Language Support Guide* for more information on setting this parameter.
- *Oracle8i SQL Reference* for information on number format elements.

NLS_DATE_FORMAT

Parameter type:	String
Syntax:	`NLS_DATE_FORMAT = "fmt"`
Parameter class:	Dynamic. Scope = `ALTER SESSION`.
Default value:	Derived from `NLS_TERRITORY`
Range of values:	Any valid date format mask but not exceeding a fixed length

`NLS_DATE_FORMAT` specifies the default date format to use with the `TO_CHAR` and `TO_DATE` functions. The default value of this parameter is determined by `NLS_TERRITORY`. The value of this parameter can be any valid date format mask, and the value must be surrounded by double quotation marks. For example:

```
NLS_DATE_FORMAT = "MM/DD/YYYY"
```

See Also:

- *Oracle8i National Language Support Guide* for more information on setting this parameter.
- *Oracle8i Distributed Database Systems* for information on setting this parameter in heterogeneous systems.

NLS_DATE_LANGUAGE

Parameter type:	String
Syntax:	NLS_DATE_LANGUAGE = language
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	Derived from NLS_LANGUAGE
Range of values:	Any valid NLS_LANGUAGE value

NLS_DATE_LANGUAGE specifies the language to use for the spelling of day and month names and date abbreviations (AM, PM, AD, BC) returned by the TO_DATE and TO_CHAR functions.

See Also:

- *Oracle8i National Language Support Guide* for more information on setting this parameter.
- *Oracle8i SQL Reference* for information on the TO_DATE and TO_CHAR functions.
- *Oracle8i Distributed Database Systems* for information on setting this parameter in heterogeneous systems.

NLS_DUAL_CURRENCY

Parameter type:	String
Syntax:	NLS_DUAL_CURRENCY = currency_symbol
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	Derived from NLS_TERRITORY
Range of values:	Any valid format name up to 10 characters

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.

See Also: *Oracle8i National Language Support Guide* for more information on setting this parameter.

NLS_ISO_CURRENCY

Parameter type:	String
Syntax:	NLS_ISO_CURRENCY = territory

Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	Derived from NLS_TERRITORY
Range of values:	Any valid NLS_TERRITORY value

NLS_ISO_CURRENCY specifies the string to use as the international currency symbol for the C number format element.

Local currency symbols can be ambiguous. For example, a dollar sign ($) can refer to US dollars or Australian dollars. ISO Specification 4217 1987-07-15 defines unique "international" currency symbols for the currencies of specific territories or countries.

See Also:

- *Oracle8i National Language Support Guide* for more information on setting this parameter.
- *Oracle8i SQL Reference* for information on number format elements.

NLS_LANGUAGE

Parameter type:	String
Syntax:	NLS_LANGUAGE = language
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	Operating system dependent, derived from the NLS_LANG environment variable
Range of values:	Any valid language name

NLS_LANGUAGE specifies the default language of the database. This language is used for messages, day and month names, symbols for AD, BC, AM, and PM, and the default sorting mechanism. This parameter also determines the default values of the parameters NLS_DATE_LANGUAGE and NLS_SORT.

See Also:

- *Oracle8i National Language Support Guide* for a complete list of languages and additional information on this parameter.
- Your operating system specific Oracle documentation and the release notes for your country.

NLS_NUMERIC_CHARACTERS

Parameter type:	String
Syntax:	NLS_NUMERIC_CHARACTERS = "decimal_character group_separator"
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	Derived from NLS_TERRITORY

NLS_NUMERIC_CHARACTERS specifies the characters to use as the group separator and decimal character. It 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 (+), hyphen (-), less than sign (<), greater than sign (>). Either character can be a space.

For example, if you wish 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 = ", "
```

See Also: *Oracle8i National Language Support Guide* for more information on setting this parameter.

NLS_SORT

Parameter type:	String
Syntax:	NLS_SORT = {BINARY | linguistic_definition}
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	Derived from NLS_LANGUAGE
Range of values:	BINARY or any valid linguistic definition name

NLS_SORT specifies the collating sequence for ORDER BY queries.

- If the value is BINARY, then the collating sequence for ORDER BY queries is based on the numeric value of characters (a binary sort that requires less system overhead).
- If the value is a named linguistic sort, sorting is based on the order of the defined linguistic sort. Most (but not all) languages supported by the NLS_LANGUAGE parameter also support a linguistic sort with the same name.

Note: Setting NLS_SORT to anything other than BINARY causes a sort to use a full table scan, regardless of the path chosen by the optimizer. BINARY is the exception because indexes are built according to a binary order of keys. Thus the optimizer can use an index to satisfy the ORDER BY clause when NLS_SORT is set to BINARY. If NLS_SORT is set to any linguistic sort, the optimizer must include a full table scan and a full sort in the execution plan.

You must use the NLS_SORT operator with comparison operations if you want the linguistic sort behavior.

See Also:

- "Linguistic definitions" in *Oracle8i National Language Support Guide* for a current listing of values you can specify for this parameter.
- *Oracle8i Concepts* for more information on this parameter.

NLS_TERRITORY

Parameter type:	String
Syntax:	NLS_TERRITORY = territory
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	Operating system dependent
Range of values:	Any valid territory name

NLS_TERRITORY specifies the name of the territory whose conventions are to be followed for day and week numbering.

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" on page 1-75, "NLS_NUMERIC_CHARACTERS" on page 1-78, "NLS_CURRENCY" on page 1-75, and "NLS_ISO_CURRENCY" on page 1-76.

See Also:

- *Oracle8i National Language Support Guide* for a complete list of territories.
- Your operating system specific Oracle documentation for the territory-dependent default values for these parameters.

O7_DICTIONARY_ACCESSIBILITY

Parameter type:	Boolean
Parameter class:	Static

Default value:	TRUE
Range of values:	TRUE | FALSE

O7_DICTIONARY_ACCESSIBILITY is intended for use when you migrate from Oracle7 to Oracle8*i*. It controls restrictions on SYSTEM privileges. If the parameter is set to TRUE, access to objects in the SYS schema is allowed (Oracle7 behavior). The default setting of FALSE ensures that system privileges that allow access to objects in "any schema" do not allow access to objects in SYS schema.

For example, if O7_DICTIONARY_ACCESSIBILITY=FALSE, then the SELECT ANY TABLE privilege allows access to views or tables in any schema except the SYS schema (data dictionary tables cannot be accessed). The system privilege EXECUTE ANY PROCEDURE allows access on the procedures in any schema except the SYS schema.

If this parameter is set to FALSE and you need to access objects in the SYS schema, then you must be granted explicit object privilege. Also, the following roles, which can be granted to the database administrator, also allow access to dictionary objects: SELECT_CATALOG_ROLE, EXECUTE_CATALOG_ROLE, and DELETE_CATALOG_ROLE.

See Also:

- *Oracle8i Migration* for more information on this parameter and the roles mentioned above.
- *Oracle8i SQL Reference* for information on granting roles.

OBJECT_CACHE_MAX_SIZE_PERCENT

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM ...DEFERRED.
Default value:	10
Range of values:	0 to operating system dependent maximum

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. 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" on page 1-81 for a description of the object cache.
- *Oracle8i Concepts*, *Pro*C/C++ Precompiler Programmer's Guide*, and *Oracle Call Interface Programmer's Guide* for information on precompiler use of the object cache.

OBJECT_CACHE_OPTIMAL_SIZE

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM ...DEFERRED.
Default value:	102400 (100 KB)
Range of values:	10 KB to operating system dependent maximum

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_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.

See Also: *Oracle8i Concepts, Pro*C/C++ Precompiler Programmer's Guide,* and *Oracle Call Interface Programmer's Guide* for information on precompiler use of the object cache.

OPEN_CURSORS

Parameter type:	Integer
Parameter class:	Static
Default value:	50
Range of values:	1 to 4294967295 (4 GB -1)

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. This parameter also constrains the size of the PL/SQL cursor cache which PL/SQL uses to avoid having to reparse as statements are reexecuted by a user.

It is important to set the value of OPEN_CURSORS 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 OPEN_CURSORS, there is no added overhead to setting this value higher than actually needed.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- Your operating system specific Oracle documentation for the range of values.

OPEN_LINKS

Parameter type:	Integer
Parameter class:	Static
Default value:	4
Range of values:	0 to 255

OPEN_LINKS specifies the maximum number of concurrent open connections to remote databases in one session. These connections include the schema objects called database links, as well as external procedures and cartridges, each of which uses a separate process.

Oracle counts one open link

- For each user that references a public or private database link and
- 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 OPEN_LINKS 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.

See Also: "OPEN_LINKS_PER_INSTANCE" on page 1-82 for information on setting open connections globally for a database instance.

OPEN_LINKS_PER_INSTANCE

Parameter type:	Integer
Parameter class:	Static
Default value:	4
Range of values:	0 to 4294967295 (4 GB -1)
Oracle Parallel Server:	Multiple instances can have different values.

OPEN_LINKS_PER_INSTANCE specifies the maximum number of migratable open connections globally per database instance. 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" on page 1-82.

OPTIMIZER_FEATURES_ENABLE

Parameter type:	String
Parameter class:	Static
Default value:	8.1.6
Range of values:	8.0.0, 8.0.3, 8.0.4, 8.0.5, 8.0.6, 8.1.0, 8.1.3, 8.1.4, 8.1.5, 8.1.6, 8.1.7

OPTIMIZER_FEATURES_ENABLE lets you change the behavior of the Oracle optimizer based on a release number. For example, if you migrate your database from 8.0.5 to 8.1.5, but you want to keep the 8.0.5 optimizer behavior, you can do so by setting this parameter to 8.0.5. At another time, you can try the new enhancements introduced up to release 8.1.5 by setting the parameter to 8.1.5.

Table 1–5 lists some optimizer features and indicates which release settings enables those features. The features listed in Table 1–5 are described in the documents indicated.

Table 1–5 Enabling Optimizer Features Based on Release Number

Features	Release										
	8.0.0	8.0.3	8.0.4	8.0.5	8.0.6	8.1.0	8.1.3	8.1.4	8.1.5	8.1.6	8.1.7
Use of index fast full scans (*Oracle8i Designing and Tuning for Performance*, *Oracle8i Concepts*)			X	X	X	X	X	X	X	X	
Consideration of bitmap access paths for tables with only B-tree indexes			X	X	X						
Complex view merging (*Oracle8i Concepts*)			X	X	X						

Table 1–5 Enabling Optimizer Features Based on Release Number

Features	Release										
	8.0.0	8.0.3	8.0.4	8.0.5	8.0.6	8.1.0	8.1.3	8.1.4	8.1.5	8.1.6	8.1.7
Push-join predicate feature (*Oracle8i Designing and Tuning for Performance*)			X	X	X						
Subquery unnesting (*Oracle8i SQL Reference*)											X
Common subexpression elimination											X
Index joins (*Oracle8i Designing and Tuning for Performance, Oracle8i Concepts*)											X

OPTIMIZER_INDEX_CACHING

Parameter type:	Integer
Parameter class:	Dynamic. Scope = `ALTER SESSION`.
Default value:	0
Range of values:	0 to 100

`OPTIMIZER_INDEX_CACHING` lets you adjust the behavior of cost-based optimization to favor nested loops joins and IN-list iterators.

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: *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.

OPTIMIZER_INDEX_COST_ADJ

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	100
Range of values:	1 to 10000

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.

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: *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter, and on its relationship to "OPTIMIZER_INDEX_CACHING" on page 1-84.

OPTIMIZER_MAX_PERMUTATIONS

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM.
Default value:	80000
Range of values:	4 to 2^{32} (~4.3 billion)

OPTIMIZER_MAX_PERMUTATIONS restricts the number of permutations of the tables the optimizer will consider in queries with joins. Such a restriction ensures that the parse time for the query stays within acceptable limits. However, a slight risk exists that the optimizer will overlook a good plan it would otherwise have found.

The default value for this parameter is 80000, which corresponds to the no limitation at all. Setting this parameter to a value less than 1000 normally ensures parse times of a few seconds or less.

OPTIMIZER_MODE

Parameter type:	String
Syntax:	OPTIMIZER_MODE = {RULE | CHOOSE | FIRST_ROWS | ALL_ROWS}
Parameter class:	Dynamic. Scope=ALTER SESSION.
Default value:	CHOOSE

OPTIMIZER_MODE specifies the goal of optimization of SQL statements.

- RULE specifies that rule-based optimization is to be used unless hints are specified in the query.
- CHOOSE specifies cost-based optimization for a SQL statement if the data dictionary contains statistics for at least one table accessed in the statement. If no such statistics are present, the optimizer uses the rule-based approach.
- FIRST_ROWS causes the optimizer to use cost-based optimization to choose execution plans that minimize response time. This setting overrides a nonzero setting for "OPTIMIZER_PERCENT_PARALLEL" on page 1-86.
- ALL_ROWS causes the optimizer to use cost-based optimization to choose execution plans that minimize total execution time.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- *Oracle8i Concepts* and *Oracle8i Designing and Tuning for Performance* for more information about the optimizer.

OPTIMIZER_PERCENT_PARALLEL

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	0
Range of values:	0 to 100

OPTIMIZER_PERCENT_PARALLEL specifies the amount of parallelism that the optimizer uses in its cost functions.

The default of 0 means that the optimizer chooses the best serial plan. A value of 100 means that the optimizer uses each object's degree of parallelism in computing the cost of a full table scan operation. Low values favor indexes, and high values favor table scans.

Cost-based optimization is always used for queries that reference an object with a nonzero degree of parallelism. For such queries, a RULE hint or optimizer mode or goal will be ignored.

Note: Use of a FIRST_ROWS hint or of optimizer mode override a non-zero setting of OPTIMIZER_PERCENT_PARALLEL.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.

ORACLE_TRACE_COLLECTION_NAME

Parameter type:	String
Syntax	ORACLE_TRACE_COLLECTION_NAME = collection_name
Parameter class:	Static
Default value:	None
Range of values:	Valid collection name up to 16 characters long (except for platforms that enforce 8-character file names)

A **collection** is data collected for events that occurred while an instrumented product was running. ORACLE_TRACE_COLLECTION_NAME specifies the Oracle Trace collection name for this instance. Oracle also uses this parameter in the output file names (collection definition file .cdf and data collection file .dat). If you set ORACLE_TRACE_ENABLE to TRUE, setting this value to a non-null string will start a default Oracle Trace collection that will run until this value is set to null again.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on the Oracle Trace facility and on setting this parameter.
- "ORACLE_TRACE_ENABLE" on page 1-88.

ORACLE_TRACE_COLLECTION_PATH

Parameter type:	String
Syntax:	ORACLE_TRACE_COLLECTION_PATH = pathname
Parameter class:	Static
Default value:	Operating system specific
Range of values:	Full directory pathname

ORACLE_TRACE_COLLECTION_PATH specifies the directory pathname where the Oracle Trace collection definition (.cdf) and data collection (.dat) files are located. If you accept the default, the Oracle Trace .cdf and .dat files will be located in **$ORACLE_HOME/otrace/admin/cdf**.

You must specify this parameter if you set the ORACLE_TRACE_FACILITY_NAME parameter.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on the Oracle Trace facility and on setting this parameter.

ORACLE_TRACE_COLLECTION_SIZE

Parameter type:	Integer
Parameter class:	Static
Default value:	5242880
Range of values:	0 to 4294967295

ORACLE_TRACE_COLLECTION_SIZE specifies, in bytes, the maximum size of the Oracle Trace collection file (.dat). Once the collection file reaches this maximum, the collection is disabled. A value of 0 means that the file has no size limit.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on the Oracle Trace facility and on setting this parameter.

ORACLE_TRACE_ENABLE

Parameter type:	Boolean
Syntax:	ORACLE_TRACE_ENABLE = {TRUE | FALSE}
Parameter class:	Static

Default value: FALSE

To enable Oracle Trace collections for the server, set ORACLE_TRACE_ENABLE to TRUE. This setting alone does not start an Oracle Trace collection, but it allows Oracle Trace to be used for the server.

You can then start an Oracle Trace collection by using the Oracle Trace Manager application (supplied with the Oracle Diagnostics Pack) or by including a name in the ORACLE_TRACE_COLLECTION_NAME parameter.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on the Oracle Trace facility and on setting this parameter.
- "ORACLE_TRACE_COLLECTION_NAME" on page 1-87.

ORACLE_TRACE_FACILITY_NAME

Parameter type: String

Syntax: ORACLE_TRACE_FACILITY_NAME =
{ORACLED | ORACLEE | ORACLESM | ORACLEC}

Parameter class: Static

Default value: ORACLED

ORACLE_TRACE_FACILITY_NAME specifies the event set that Oracle Trace collects. The value of this parameter, followed by the .fdf extension, is the name of the Oracle Trace product definition file. That file must be located in the directory specified by the ORACLE_TRACE_FACILITY_PATH parameter. The product definition file contains definition information for all the events and data items that can be collected for products that use the Oracle Trace data collection API.

The Oracle server has multiple event sets and therefore multiple product definition files:

- ORACLE is the ALL event set
- ORACLED is the DEFAULT event set
- ORACLEE is the EXPERT event set
- ORACLESM is the SUMMARY event set
- ORACLEC is the CACHEIO event set

Oracle Corporation recommends that you use the DEFAULT event set for Oracle server collections, ORACLED.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on the Oracle Trace facility and on setting this parameter.
- "ORACLE_TRACE_COLLECTION_PATH" on page 1-88.

ORACLE_TRACE_FACILITY_PATH

Parameter type:	String
Syntax:	`ORACLE_TRACE_FACILITY_PATH = pathname`
Parameter class:	Static
Default value:	Operating system specific
Range of values:	Full directory pathname

`ORACLE_TRACE_FACILITY_PATH` specifies the directory pathname where Oracle TRACE facility definition files are located. On Solaris, the default path is ?/otrace/admin/fdf/. On NT, the default path is %OTRACE80%\ADMIN\FDF\.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on the Oracle Trace facility and on setting this parameter.

OS_AUTHENT_PREFIX

Parameter type:	String
Syntax:	`OS_AUTHENT_PREFIX = authentication_prefix`
Parameter class:	Static
Default value:	Operating system specific (typically "OPS$")

`OS_AUTHENT_PREFIX` specifies a prefix that Oracle uses to authenticate users attempting to connect to the server. Oracle concatenates the value of this parameter to the beginning of the user's operating system account name and password. When a connection request is attempted, Oracle compares the prefixed username with Oracle usernames 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.

Note: The text of the `OS_AUTHENT_PREFIX` parameter is case sensitive on some operating systems.

See Also:

- *Oracle Advanced Security Administrator's Guide* for more information on setting this parameter.
- Your operating system specific Oracle documentation for the default value.

OS_ROLES

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

OS_ROLES determines whether Oracle or the operating system identifies and manages the roles of each username.

- 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. A user can subsequently enable as many roles identified by the operating system as specified by the parameter MAX_ENABLED_ROLES.

  Revocation by Oracle of roles granted by the operating system are ignored, as are any roles previously granted by Oracle.
- FALSE: Oracle identifies and manages the roles.

See Also:

- *Oracle8i Administrator's Guide* and *Oracle Advanced Security Administrator's Guide* for more information on roles and on setting this parameter.
- "MAX_ENABLED_ROLES" on page 1-67 and "REMOTE_OS_ROLES" on page 1-103.

PARALLEL_ADAPTIVE_MULTI_USER

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	Derived from the value of PARALLEL_AUTOMATIC_TUNING
Range of values:	TRUE | FALSE

PARALLEL_ADAPTIVE_MULTI_USER, when set to TRUE, enables an adaptive algorithm designed to improve performance in multi-user environments that use parallel execution. 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 single-user environment,
- The parameter PARALLEL_AUTOMATIC_TUNING parameter is set to TRUE, and
- Tables and hint use the default degree of parallelism.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on tuning parallel execution and on the algorithm described here.

PARALLEL_AUTOMATIC_TUNING

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

Note: This parameter applies to parallel execution in exclusive mode as well as in the Oracle Parallel Server environment.

When PARALLEL_AUTOMATIC_TUNING is set to TRUE, Oracle determines the default values for parameters that control parallel execution. In addition to setting this parameter, you must specify the PARALLEL clause for the target tables in the system. Oracle then tunes all subsequent parallel operations automatically.

If you used parallel execution in a previous release and are now enabling PARALLEL_AUTOMATIC_TUNING, you should reduce the amount of memory allocated from the shared pool to account for the decreased demand on that pool. This memory will be now be allocated from the large pool, and will be computed automatically if LARGE_POOL_SIZE is left unset.

As part of the automatic tuning, Oracle will enable the PARALLEL_ADAPTIVE_MULTI_USER parameter. You can override any of the system-provided defaults if desired.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on tuning parallel execution.

PARALLEL_BROADCAST_ENABLED

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	FALSE
Range of values:	TRUE | FALSE

Note: This parameter refers to parallel execution, not to Oracle Parallel Server.

PARALLEL_BROADCAST_ENABLED lets you improve performance of hash and merge join operations in which a very large join result set is joined with a very small result set (size being measured in bytes, rather than number of rows).

When this parameter is set to TRUE, the optimizer can choose to copy all the source rows of the small result set and broadcast a copy to each parallel server that is processing some rows of the larger set.

PARALLEL_EXECUTION_MESSAGE_SIZE

Parameter type:	Integer
Parameter class:	Static
Default value:	Operating system dependent.
Range of values:	2148 to 65535 (64 K - 1)
Oracle Parallel Server:	Multiple instances must have the same value.

PARALLEL_EXECUTION_MESSAGE_SIZE specifies the size of messages for parallel execution (formerly referred to as parallel query, PDML, Parallel Recovery, replication).

On most platforms, the default value is 2148 if PARALLEL_AUTOMATIC_TUNING is set to FALSE, and 4096 if PARALLEL_AUTOMATIC_TUNING is TRUE. 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.

Note: When PARALLEL_AUTOMATIC_TUNING is set to TRUE, message buffers are allocated out of the large pool. In this case, the default is generally higher.

PARALLEL_INSTANCE_GROUP

Parameter type:	String
Syntax:	PARALLEL_INSTANCE_GROUP = group_name
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM.
Default value:	A group consisting of all instances currently active
Range of values:	Any group name specified in the INSTANCE_GROUPS parameter of any active instance
Oracle Parallel Server:	Different instances can have different values.

PARALLEL_INSTANCE_GROUP is an Oracle Parallel Server parameter that you can specify in parallel mode only. Used in conjunction with the INSTANCE_GROUPS parameter, it lets you restrict parallel query operations to a limited number of instances. See "INSTANCE_GROUPS" on page 1-46.

This parameter identifies the parallel instance group Oracle will use for spawning parallel execution processes. Parallel operations will spawn parallel execution processes only on instances that specify a matching group in their INSTANCE_GROUPS parameter.

If the value of PARALLEL_INSTANCE_GROUP does not correspond to an instance group name specified for an active instance, Oracle returns an error.

See Also: *Oracle8i Parallel Server Administration, Deployment, and Performance* for more information on parallel query execution in an Oracle Parallel Server environment.

PARALLEL_MAX_SERVERS

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived from the values of CPU_COUNT, PARALLEL_AUTOMATIC_TUNING, and PARALLEL_ADAPTIVE_MULTI_USER
Range of values:	0 to 3599
Oracle Parallel Server:	All instances must have the same value (which can be zero).

Note: This parameter applies to parallel execution in exclusive mode as well as in the Oracle Parallel Server environment.

PARALLEL_MAX_SERVERS specifies the maximum number of parallel execution processes and parallel recovery processes for an instance. As demand increases, Oracle increases the number of processes from the number created at instance startup up to this value.

If you set this parameter too low, some queries may not have a parallel execution process available to them during query processing. If you set it too high, memory resource shortages may occur during peak periods, which can degrade performance.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on tuning parallel execution.

PARALLEL_MIN_PERCENT

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	0
Range of values:	0 to 100
Oracle Parallel Server:	Different instances can have different values

PARALLEL_MIN_PERCENT operates in conjunction with PARALLEL_MAX_SERVERS and PARALLEL_MIN_SERVERS. It lets you specify the minimum percentage of parallel execution processes (of the value of PARALLEL_MAX_SERVERS) required for parallel execution. Setting this parameter ensures that parallel operations will not execute sequentially 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 in conjunction with PARALLEL_ADAPTIVE_MULTI_USER. In a multi-user environment, a 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:

- *Oracle8i Designing and Tuning for Performance* for more information on tuning parallel execution.
- "PARALLEL_MAX_SERVERS" on page 1-94, "PARALLEL_MIN_SERVERS" on page 1-96, and "PARALLEL_ADAPTIVE_MULTI_USER" on page 1-91.

PARALLEL_MIN_SERVERS

Parameter type:	Integer
Parameter class:	Static
Default value:	0
Range of values:	0 to value of PARALLEL_MAX_SERVERS
Oracle Parallel Server:	Different instances can have different values.

Note: This parameter applies to parallel execution in exclusive mode as well as in the Oracle Parallel Server environment.

PARALLEL_MIN_SERVERS specifies the minimum number of parallel execution processes for the instance. This value is the number of parallel execution processes Oracle creates when the instance is started.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on tuning parallel execution.

PARALLEL_SERVER

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE
Oracle Parallel Server:	Multiple instances must have the same value.

PARALLEL_SERVER is an Oracle Parallel Server parameter that specifies whether Oracle Parallel Server is enabled.

See Also: *Oracle8i Parallel Server Administration, Deployment, and Performance.*

PARALLEL_SERVER_INSTANCES

Parameter type:	Integer
Parameter class:	Static
Default value:	1
Range of values:	Any non-zero value

PARALLEL_SERVER_INSTANCES is an Oracle Parallel Server parameter that specifies the number of instances currently configured. You must set this parameter for every instance. Normally you should set this parameter to the number of instances in your Oracle Parallel Server environment. A proper setting for this parameter can improve memory use.

Oracle uses the value of this parameter to compute the default value of the LARGE_POOL_SIZE parameter when the PARALLEL_AUTOMATIC_TUNING parameter is set to TRUE.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on tuning parallel execution.
- *Oracle8i Parallel Server Administration, Deployment, and Performance* for information on Oracle Parallel Server.

PARALLEL_THREADS_PER_CPU

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	Operating system dependent, usually 2
Range of values:	Any non-zero number

Note: This parameter applies to parallel execution in exclusive mode as well as in the Oracle Parallel Server environment.

PARALLEL_THREADS_PER_CPU specifies the default degree of parallelism for the instance and determines the parallel adaptive and load balancing algorithms. The parameter describes the number of parallel execution processes or "threads" that a CPU can handle during parallel execution.

The default is platform dependent and is adequate in most cases. You should decrease the value of this parameter if the machine appears to be overloaded when a representative parallel query is executed. You should increase the value if the system is I/O bound.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on tuning parallel execution.

PARTITION_VIEW_ENABLED

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	FALSE

Range of values:	TRUE | FALSE

Note: Oracle Corporation recommends that you use partitioned tables (available beginning with Oracle8) rather than partition views. Partition views are supported for backward compatibility only.

PARTITION_VIEW_ENABLED specifies whether the optimizer uses partition views. If you set it to TRUE, the optimizer prunes (or skips) unnecessary table accesses in a partition view and alters the way it computes statistics on a partition view from statistics on underlying tables.

See Also: *Oracle8i Concepts* for information on the advantages of partitioned tables, and how to convert partition views to partitioned tables.

PLSQL_V2_COMPATIBILITY

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM [...DEFERRED].
Default value:	FALSE
Range of values:	TRUE | FALSE

PL/SQL Version 2 allows some abnormal 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, PL/SQL V8 behavior is enforced and V2 behavior is not allowed.

See Also: *PL/SQL User's Guide and Reference* for a description of the differences between PL/SQL Versions 2 and 8, and for more information on setting this parameter.

PRE_PAGE_SGA

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	FALSE | TRUE

PRE_PAGE_SGA determines whether all SGA pages are brought into memory at instance startup. Operating system page table entries are then prebuilt for each page of the SGA. The resulting decrease in disk I/O may be offset by performance degradation in other areas. Therefore, this parameter is most useful on systems that have sufficient memory to hold all the SGA pages without degrading performance in other areas.

See Also: *Oracle8i Designing and Tuning for Performance* for a discussion of the advantages and disadvantages of setting this parameter to TRUE.

PROCESSES

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived from PARALLEL_MAX_SERVERS
Range of values:	6 to operating system dependent
Oracle Parallel Server:	Multiple instances can have different values.

PROCESSES specifies the maximum number of operating system user processes that can simultaneously connect to Oracle. Its value 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:

- *Oracle8i Administrator's Guide* for information on setting this parameter in exclusive mode.
- *Oracle8i Parallel Server Administration, Deployment, and Performance* for information on setting this parameter in an Oracle Parallel Server environment.
- Your operating system specific Oracle documentation for the range of values.

QUERY_REWRITE_ENABLED

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SYSTEM, ALTER SESSION.
Default value:	FALSE
Range of values:	TRUE | FALSE
Oracle Parallel Server:	Multiple instances can have different values.

QUERY_REWRITE_ENABLED lets you to enable or disable query rewriting globally for the database.

To take advantage of query rewrite for a particular materialized view, you must also enable query rewrite for that materialized view, and you must enable cost-based optimization.

See Also:

- *Oracle8i Data Warehousing Guide* for information on query rewrite of materialized views.
- *Oracle8i Designing and Tuning for Performance* and "OPTIMIZER_MODE" on page 1-86 for information on cost-based optimization.

QUERY_REWRITE_INTEGRITY

Parameter type:	String
Syntax:	QUERY_REWRITE_INTEGRITY = {ENFORCED | TRUSTED | STALE_TOLERATED}
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM.
Default value:	ENFORCED
Oracle Parallel Server:	Multiple instances can have different values.

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.

- 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.

See Also: *Oracle8i Data Warehousing Guide* for more information on query rewrite of materialized views.

RDBMS_SERVER_DN

Parameter type:	X.500 Distinguished Name
Parameter class:	Static
Default value:	None

Range of values:	All X.500 Distinguished Name format values

RDBMS_SERVER_DN specifies the Distinguished Name of the Oracle server. It is used for retrieving Enterprise Roles from an enterprise directory service.

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.

See Also: *Oracle Advanced Security Administrator's Guide* for more information on enterprise roles and the enterprise directory service.

READ_ONLY_OPEN_DELAYED

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

READ_ONLY_OPEN_DELAYED determines when datafiles in read-only tablespaces are accessed:

- TRUE: 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.

See Also: *Oracle8i Administrator's Guide* for information on the consequences of delaying access of datafiles in read-only tablespaces.

RECOVERY_PARALLELISM

Parameter type:	Integer
Parameter class:	Static
Default value:	Operating system dependent
Range of values:	Operating system dependent, but cannot exceed PARALLEL_MAX_SERVERS

RECOVERY_PARALLELISM specifies the number of processes to participate in instance or crash recovery. A value of 0 or 1 indicates that recovery is to be performed serially by one process.

See Also:

- *Oracle8i Designing and Tuning for Performance* for information on setting this parameter in exclusive mode.
- *Oracle8i Parallel Server Administration, Deployment, and Performance* for information on setting this parameter in an Oracle Parallel Server environment.

REMOTE_DEPENDENCIES_MODE

Parameter type:	String
Syntax:	REMOTE_DEPENDENCIES_MODE = {TIMESTAMP | SIGNATURE}
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM.
Default value:	TIMESTAMP

REMOTE_DEPENDENCIES_MODE specifies how Oracle should handle dependencies upon remote PL/SQL stored procedures:

- 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: *Oracle8i Application Developer's Guide - Fundamentals* for information about the consequences of the settings of this parameter.

REMOTE_LOGIN_PASSWORDFILE

Parameter type:	String
Syntax:	REMOTE_LOGIN_PASSWORDFILE= {NONE | SHARED | EXCLUSIVE}
Parameter class:	Static
Default value:	NONE
Oracle Parallel Server:	Multiple instances must have the same value.

REMOTE_LOGIN_PASSWORDFILE specifies whether Oracle checks for a password file and how many databases can use the password file.

- NONE: Oracle ignores any password file. Therefore, privileged users must be authenticated by the operating system.
- EXCLUSIVE: The password file can be used by only one database and the password file can contain names other than SYS and INTERNAL. **This setting is required for Oracle Parallel Server.**
- SHARED: More than one database can use a password file. However, the only users recognized by the password file are SYS and INTERNAL.

See Also: *Oracle8i Administrator's Guide* for more information about secure connections for privileged users.

REMOTE_OS_AUTHENT

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

REMOTE_OS_AUTHENT specifies whether remote clients will be authenticated with the value of the OS_AUTHENT_PREFIX parameter.

See Also:

- *Oracle Advanced Security Administrator's Guide* for more information on setting this parameter.
- "OS_AUTHENT_PREFIX" on page 1-90.

REMOTE_OS_ROLES

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

REMOTE_OS_ROLES specifies whether operating system roles are allowed for remote clients. The default value, FALSE, causes Oracle to identify and manage roles for remote clients.

See Also:

- *Oracle8i Administrator's Guide* for more information on setting this parameter.
- "OS_ROLES" on page 1-91.

REPLICATION_DEPENDENCY_TRACKING

Parameter type:	Boolean
Parameter class:	Static
Default value:	TRUE
Range of values:	TRUE | FALSE

REPLICATION_DEPENDENCY_TRACKING enables or disables dependency tracking for read / write operations to the database. Dependency tracking is essential for propagating changes in a replicated environment in parallel.

- 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 perform no read/write operations to the replicated tables.**

See Also: *Oracle8i Replication* for more information on parallel propagation dependency tracking.

RESOURCE_LIMIT

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	FALSE
Range of values:	TRUE | FALSE

RESOURCE_LIMIT determines whether resource limits are enforced in database profiles:

- TRUE: enables the enforcement of resource limits.
- FALSE: disables the enforcement of resource limits.

See Also: *Oracle8i Administrator's Guide* and *Oracle8i SQL Reference* for more information on setting resource limits for profiles.

RESOURCE_MANAGER_PLAN

Parameter type:	String
Syntax:	RESOURCE_MANAGER_PLAN = plan_name
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	None
Range of values:	Any valid character string

RESOURCE_MANAGER_PLAN specifies the top-level resource plan to use for this instance. The resource manager will load this top-level 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 plan (if it was previously on). If you specify a plan that does not exist in the data dictionary, Oracle returns an error message.

See Also:

- *Oracle8i Administrator's Guide* for information on resource plans.
- *Oracle8i Supplied PL/SQL Packages Reference* for information on the DBMS_RESOURCE_MANAGER and DBMS_RESOURCE_MANAGER_PRIVS packages.
- "DBA_RSRC_PLANS" on page 2-129, "DBA_RSRC_PLAN_DIRECTIVES" on page 2-130, and the various V$RSRC_ dynamic performance views in Chapter 3 for information on existing resource plans.

ROLLBACK_SEGMENTS

Parameter type:	String
Syntax:	ROLLBACK_SEGMENTS = (segment_name [, segment_name] ...)
Parameter class:	Static
Default value:	None (the instance uses public rollback segments by default if you do not specify this parameter)
Range of values:	Any rollback segment names listed in DBA_ROLLBACK_SEGS except SYSTEM
Oracle Parallel Server:	Multiple instances must have different values.

ROLLBACK_SEGMENTS allocates one or more rollback segments by name to this instance. 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.

See Also:

- *Oracle8i Administrator's Guide* for more information on setting this parameter.
- *Oracle8i Parallel Server Administration, Deployment, and Performance* for information on setting this parameter in an Oracle Parallel Server environment.
- "DBA_ROLLBACK_SEGS" on page 2-127.

ROW_LOCKING

Parameter type:	String
Syntax:	ROW_LOCKING = {ALWAYS | DEFAULT | INTENT}
Parameter class:	Static
Default value:	ALWAYS
Oracle Parallel Server:	You must set this parameter for every instance, and multiple instances must have the same value.

ROW_LOCKING specifies whether row locks are acquired during UPDATE operations.

- ALWAYS and DEFAULT: Row locks are acquired. Table locks are not acquired.
- INTENT: In addition to row locks, restricted table-level locks are acquired during UPDATE operations.

The acquisition of table-level locks is a way to ensure data concurrency. However, Oracle Corporation recommends that you use the SQL statement LOCK TABLE for that purpose rather than changing the setting of this parameter from its default setting of ALWAYS.

SERIAL _REUSE

Parameter type:	String
Syntax:	SERIAL_REUSE = {DISABLE | SELECT | DML | PLSQL | ALL}
Parameter class:	Static
Default Value:	DISABLE

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.

- DISABLE: disables the option for all SQL statement types. This value overrides any other values included in the list.
- SELECT: enables the option for SELECT statements.
- DML: enables the option for DML statements.
- PLSQL: currently has no effect (although PLSQL packages do support the serial-reuse memory option using PLSQL pragmas).
- ALL: enables the option for both DML and SELECT statements. Equivalent to setting SELECT, DML, and PLSQL.

Note: 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" on page 1-23.

SERVICE_NAMES

Parameter type:	String
Syntax:	SERVICE_NAMES = db_service_name [, db_service_name [,...]]
Parameter class:	Static
Default value:	DB_NAME.DB_DOMAIN if defined
Range of values:	Any ASCII string or comma-separated list of string names

SERVICE_NAMES specifies one or more names for the database service to which this instance connects. You can specify multiple services names in order to distinguish among different uses of the same database. For example:

```
SERVICE_NAMES = sales.acme.com, widgetsales.acme.com
```

You can also use service names to identify a single service that is available from two different databases through the use of replication.

In an Oracle Parallel Server environment, you must set this parameter for every instance.

If you do not qualify the names in this parameter with a domain, Oracle qualifies them with the value of the DB_DOMAIN parameter. If DB_DOMAIN is not specified, Oracle uses the domain of your local database as it currently exists in the data dictionary.

See Also:

- *Net8 Administrator's Guide* for more information on this parameter and its settings.
- "DB_DOMAIN" on page 1-28.

SESSION_CACHED_CURSORS

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default:	0
Range of values:	0 to operating system dependent
Oracle Parallel Server:	Multiple instances can have different values.

SESSION_CACHED_CURSORS lets you specify the number of session cursors to cache. Repeated parse calls of the same SQL statement cause the session cursor for that statement to be moved into the session cursor cache. Subsequent parse calls will find the cursor in the cache and need not reopen the cursor. 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: *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter and its uses.

SESSION_MAX_OPEN_FILES

Parameter type:	Integer
Parameter class:	Static
Default value:	10

Range of values:	1 to either 50 or the value of MAX_OPEN_FILES defined at the OS level, whichever is less.

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 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.

See Also:

- *Oracle8i Application Developer's Guide - Large Objects (LOBs)* for information on LOBs in general and BFILEs in particular.
- *Oracle8i Supplied PL/SQL Packages Reference* for information on the DBMS_LOB.FILEOPEN() procedure.
- *Oracle Call Interface Programmer's Guide* for information on the OCILobFileOpen() procedure.

SESSIONS

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived: 1.1 * PROCESSES + 5
Range of values:	1 to 2^{31}

SESSIONS specifies the maximum number of sessions that can be created in the system. 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 ENQUEUE_RESOURCES and TRANSACTIONS parameters are derived from SESSIONS. Therefore, if you increase the value of SESSIONS, you should consider whether to adjust the values of ENQUEUE_RESOURCES and TRANSACTIONS as well.

In a multi-threaded server environment, the value of PROCESSES can be quite small. Therefore, Oracle Corporation recommends that you adjust the value of SESSIONS to approximately 1.1 * total number of **connections**.

See Also:

- *Oracle8i Concepts* for more information on memory structures and processes.
- "ENQUEUE_RESOURCES" on page 1-37 and "TRANSACTIONS" on page 1-119.

SHADOW_CORE_DUMP

Parameter type:	String
Syntax:	SHADOW_CORE_DUMP = {PARTIAL | FULL}
Parameter class:	Static
Default value:	PARTIAL

SHADOW_CORE_DUMP is primarily a UNIX parameter and is not useful on other platforms. It specifies whether Oracle includes the SGA in the core file for foreground (client) processes.

- FULL: The SGA is included in core dumps.
- PARTIAL: The SGA is not dumped.

See Also: "BACKGROUND_CORE_DUMP" on page 1-15.

SHARED_MEMORY_ADDRESS

Parameter type:	Integer
Parameter class:	Static
Default value:	0

SHARED_MEMORY_ADDRESS and HI_SHARED_MEMORY_ADDRESS specify the starting address at runtime of the system global area (SGA). This parameter is ignored on the many platforms that specify the SGA's starting address at linktime.

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 HI_SHARED_MEMORY_ADDRESS 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.

See Also: "HI_SHARED_MEMORY_ADDRESS" on page 1-45.

SHARED_POOL_RESERVED_SIZE

Parameter type:	String
Syntax:	SHARED_POOL_RESERVED_SIZE = integer [K | M]
Parameter class:	Static
Default value:	5% of the value of SHARED_POOL_SIZE
Range of values:	Minimum: value of SHARED_POOL_RESERVED_MIN_ALLOC Maximum: one half of the value of SHARED_POOL_SIZE (in bytes)

SHARED_POOL_RESERVED_SIZE specifies in bytes the shared pool space that is reserved for large contiguous requests for shared pool memory. 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:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- "SHARED_POOL_SIZE" on page 1-111.

SHARED_POOL_SIZE

Parameter type:	String
Syntax:	SHARED_POOL_SIZE = integer [K | M]
Parameter class:	Static
Default value:	If 64 bit, 64MB, else 16MB
Range of values:	300 Kbytes to operating system dependent

SHARED_POOL_SIZE specifies in bytes the size of the shared pool. The shared pool contains shared cursors, stored procedures, control structures, and other structures. If you set PARALLEL_AUTOMATIC_TUNING to FALSE, 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:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- *Oracle8i Migration* for information on parallel execution message buffers.
- "PARALLEL_AUTOMATIC_TUNING" on page 1-92 and "V$SGASTAT" on page 3-99.

SORT_AREA_RETAINED_SIZE

Parameter type:	Integer
Parameter class:	Dynamic. Scope= ALTER SESSION, ALTER SYSTEM ... DEFERRED.
Default value:	Derived from SORT_AREA_SIZE
Range of values:	from the value equivalent of two database blocks to the value of SORT_AREA_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.

Oracle may allocate multiple sort spaces of this size for each query. Usually, only one or two sorts occur at one time, even for complex queries. In some cases, however, additional concurrent sorts are required, and each sort keeps its own memory area. If the multi-threaded server is used, allocation is to the SGA until the value in SORT_AREA_RETAINED_SIZE is reached. The difference between SORT_AREA_RETAINED_SIZE and SORT_AREA_SIZE 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:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- "SORT_AREA_SIZE" on page 1-112.

SORT_AREA_SIZE

Parameter type:	Integer
Parameter class:	Dynamic. Scope= ALTER SESSION, ALTER SYSTEM ... DEFERRED.

Default value:	Operating system dependent
Range of values:	Minimum: the value equivalent of six database blocks Maximum: operating system dependent

SORT_AREA_SIZE specifies in bytes the maximum amount of memory Oracle will use for a sort. After the sort is complete, but before the rows are returned, Oracle releases memory down to the size specified by the SORT_AREA_RETAINED_SIZE 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. Multiple allocations never occur. Only one memory area of SORT_AREA_SIZE exists for each user process at any time.

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 CREATE INDEX operations.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- *Oracle8i Concepts* for information on logical storage structures such as sort areas.
- Your operating system specific Oracle documentation for the default value on your system.
- "SORT_AREA_RETAINED_SIZE" on page 1-112.

SORT_MULTIBLOCK_READ_COUNT

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SESSION, ALTER SYSTEM ...DEFERRED.
Default value:	2
Range of values:	1 to operating system dependent value

SORT_MULTIBLOCK_READ_COUNT specifies the number of database blocks Oracle reads each time a sort performs a read from a temporary segment.

Oracle performs sort operations using an amount of memory specified by the SORT_AREA_SIZE parameter. If the data being sorted does not fit into that amount of memory, Oracle writes sections of the data to temporary segments in the form of sorted runs.

After all the data has been partially sorted to these runs, Oracle merges the runs by reading pieces of them from the temporary segment into memory to produce the final sorted output. If SORT_AREA_SIZE is not large enough to merge all the runs at once, subsets of the runs are merged in a number of merge passes.

If you increase the value of SORT_MULTIBLOCK_READ_COUNT, Oracle reads a larger section of each run into memory during a merge pass. Doing so reduces the number of runs that can be merged in one merge pass. Any increase in I/O throughput obtained by increasing SORT_MULTIBLOCK_READ_COUNT may be offset by an increase in total I/O due to an increase in the number of merge passes.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter.
- "SORT_AREA_SIZE" on page 1-112.

SQL92_SECURITY

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

The SQL92 standards specify that security administrators should be able to require that users have SELECT privilege on a table when executing an UPDATE or DELETE statement that references table column values in a WHERE or SET clause. SQL92_SECURITY lets you specify whether users must have been granted the SELECT object privilege in order to execute such UPDATE or DELETE statements.

SQL_TRACE

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

The value of SQL_TRACE disables or enables the SQL trace facility. Setting this parameter to TRUE provides information on tuning that you can use to improve performance. You can change the value using the DBMS_SYSTEM package.

CAUTION: Enabling the SQL trace facility for the entire instance using this initialization parameter can have a severe performance impact. Try to enable the facility for specific sessions using the ALTER SESSION statement. If you must enable the facility on an entire production environment, 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: If you change the value of this parameter by using the ALTER SESSION SET SQL_TRACE statement, the resulting change will not be reflected in the V$PARAMETER dynamic performance view. Therefore, this parameter is not considered dynamic. For information on the ALTER SESSION statement, see also *Oracle8i SQL Reference*.

See Also: *Oracle8i Designing and Tuning for Performance* for more information about performance diagnostic tools.

STANDBY_ARCHIVE_DEST

Parameter type:	String
Syntax:	STANDBY_ARCHIVE_DEST = filespec
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	Operating system specific
Range of values:	A valid path or device name other than RAW

STANDBY_ARCHIVE_DEST is relevant only for a standby database in managed recovery mode. It specifies the location of archivelogs arriving from a primary database. Oracle uses STANDBY_ARCHIVE_DEST and LOG_ARCHIVE_FORMAT to fabricate the fully qualified standby log filenames and stores the filenames in the standby control file.

You can see the value of this parameter by querying the V$ARCHIVE_DEST data dictionary view.

See Also:

- *Oracle8i Standby Database Concepts and Administration* for more information on setting this parameter and on managed recovery mode in general.
- "LOG_ARCHIVE_DEST" on page 1-55 and "V$ARCHIVE_DEST" on page 3-4.

STAR_TRANSFORMATION_ENABLED

Parameter type:	String
Syntax:	STAR_TRANSFORMATION_ENABLED = {TRUE | FALSE | TEMP_DISABLE}
Parameter class:	Dynamic. Scope = ALTER SESSION.
Default value:	FALSE

STAR_TRANSFORMATION_ENABLED determines whether a cost-based query transformation will be applied to star queries.

- TRUE: The optimizer will consider performing a cost-based query transformation on the star query.
- FALSE: The transformation will not be applied.
- 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:

- *Oracle8i Concepts* for information on star transformation.
- *Oracle8i Designing and Tuning for Performance* for information on enabling star transformation.

TAPE_ASYNCH_IO

Parameter type:	Boolean
Parameter class:	Static
Default value:	TRUE
Range of values:	TRUE | FALSE

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. If your platform supports asynchronous I/O to sequential devices, Oracle Corporation recommends that you leave this parameter set to its default. However, if the asynchronous I/O implementation is not stable, you can set TAPE_ASYNCH_IO to false to disable asynchronous I/O. If your platform does not support asynchronous I/O to sequential devices, this parameter has no effect.

See Also: *Oracle8i Designing and Tuning for Performance* for more information on asynchronous I/O and on setting this parameter.

THREAD

Parameter type:	Integer
Parameter class:	Static
Default value:	0
Range of values:	0 to the maximum number of enabled threads
Oracle Parallel Server:	If specified, multiple instances must have different values.

THREAD is an Oracle Parallel Server parameter that specifies the number of the redo thread to be used by this instance.

When you create a database, Oracle creates and enables thread 1 as a **public thread** (one that can be used by any instance). You must create and enable subsequent threads using the ADD LOGFILE THREAD clause and ENABLE THREAD clause of the ALTER DATABASE statement. The number of threads you create is limited by the MAXINSTANCES parameter specified in the CREATE DATABASE statement.

In exclusive mode, thread 1 is the default thread. However, you can specify THREAD for an instance running in exclusive mode if you want to use the redo log files in a thread other than thread 1.

In parallel mode, you can specify any available redo thread number, as long as that thread number is enabled and is not in use by another instance.

A value of zero specifies that this instance can use any available, enabled public thread.

See Also: *Oracle8i Parallel Server Administration, Deployment, and Performance* and *Oracle8i SQL Reference.*

TIMED_OS_STATISTICS

Parameter type:	Integer
Parameter class:	Dynamic. Scope = ALTER SYSTEM.
Default value:	0
Range of values:	Unlimited

TIMED_OS_STATISTICS specifies the interval (in seconds) at which Oracle collects operating system (OS) statistics when a request is made from the client to the server or a request completes.

- On dedicated servers, Oracle collects OS statistics at user logon and after each subsequent client invocation through the OCI into the Oracle server as a remote procedure call message.
- On multi-threaded servers, Oracle collects statistics when client calls to Oracle are processed.

A value of zero specifies that OS statistics are not gathered. To collect statistics, set a value meaningful for your application and site needs.

Note: Gathering OS statistics is very expensive. Oracle Corporation 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.

TIMED_STATISTICS

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SYSTEM, ALTER SESSION.
Default value:	FALSE
Range of values:	TRUE | FALSE

TIMED_STATISTICS specifies whether statistics related to time are collected or not. Appendix C, "Statistics Descriptions", indicates which statistics depend on the setting of this parameter.

- 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. Normally, TIMED_STATISTICS should be FALSE.

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: *Oracle8i Designing and Tuning for Performance* for more information on setting this parameter and on performance diagnostic tools in general.

TRANSACTION_AUDITING

Parameter type:	Boolean
Parameter class:	Dynamic. Scope = ALTER SYSTEM ... DEFERRED.
Default value:	TRUE
Range of values:	TRUE | FALSE

If TRANSACTION_AUDITING is TRUE, Oracle generates a special redo record that contains the user logon name, user name, the session ID, some operating system information, and client information. For each successive transaction, Oracle generates a record that contains only the session ID. These subsequent records link back to the first record, which also contains the session ID.

These records might be useful if you are using a redo log analysis tool. You can access the records by dumping the redo log.

If TRANSACTION_AUDITING is FALSE, no redo record will be generated.

TRANSACTIONS

Parameter type:	Integer
Parameter class:	Static
Default value:	Derived (1.1 * SESSIONS)
Range of values:	4 to 2^{32}
Oracle Parallel Server:	Multiple instances can have different values.

TRANSACTIONS specifies the maximum number of concurrent transactions. Greater values increase the size of the SGA and can increase the number of rollback segments allocated. The default value is greater than SESSIONS (and, in turn, PROCESSES) to allow for recursive transactions.

See Also:

- *Oracle8i Administrator's Guide* for information on setting this parameter.
- *Oracle8i Designing and Tuning for Performance* for information on using this parameter to tune parallel execution.
- *Oracle8i Parallel Server Administration, Deployment, and Performance* for the relationship of this parameter to number of rollback segments.

TRANSACTIONS_PER_ROLLBACK_SEGMENT

Parameter type:	Integer
Parameter class:	Static
Default value:	5
Range of values:	1 to operating system dependent
Oracle Parallel Server:	Multiple instances can have different values.

TRANSACTIONS_PER_ROLLBACK_SEGMENT specifies the number of concurrent transactions you expect each rollback segment to have to handle. 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 ROLLBACK_ SEGMENTS.

See Also:

- *Oracle8i Administrator's Guide* for information on setting this parameter.
- *Oracle8i Parallel Server Administration, Deployment, and Performance* for how Oracle acquires rollback segments.
- Your operating system specific Oracle documentation for the range of values for this parameter.

USE_INDIRECT_DATA_BUFFERS

Parameter type:	Boolean
Parameter class:	Static
Default value:	FALSE
Range of values:	TRUE | FALSE

USE_INDIRECT_DATA_BUFFERS controls how the system global area (SGA) uses memory. It enables or disables the use of the extended buffer cache mechanism for 32-bit platforms that can support more than 4GB of physical memory. On platforms that do not support this much physical memory, this parameter is ignored.

See Also:

- "LOCK_SGA" on page 1-55, "SHARED_MEMORY_ADDRESS" on page 1-110, and "HI_SHARED_MEMORY_ADDRESS" on page 1-45, which are other parameters that control how the SGA uses memory.
- *Oracle8i Concepts* for more information about the SGA.

USER_DUMP_DEST

Parameter type:	String
Syntax:	`USER_DUMP_DEST = {pathname | directory}`
Parameter class:	Dynamic. Scope = `ALTER SYSTEM`.
Default value:	Operating system dependent
Range of values:	Any valid local path, directory, or disk.

USER_DUMP_DEST specifies the pathname for a directory where the server will write debugging trace files on behalf of a user process.

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]`

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information about the use of trace files.
- Your operating system specific Oracle documentation for the range of values.

UTL_FILE_DIR

Parameter type:	String
Syntax:	`UTL_FILE_DIR = pathname`
Parameter class:	Static
Default value:	None
Range of values:	Any valid directory path

UTL_FILE_DIR lets you specify one or more directories that Oracle should use for PL/SQL file I/O. If you are specifying multiple directories, you must repeat the UTL_FILE_DIR parameter for each directory on separate lines of the initialization parameter file.

All users can read or write to all files specified by this parameter. Therefore all PL/SQL users must be trusted with the information in the directories specified by this parameter.

Note: If you list multiple values, all entries of this parameter must be on contiguous lines of the parameter file. If you separate them with other parameters, Oracle will read only the last (contiguous) lines.

2

Static Data Dictionary Views

This chapter 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 a user is granted new privileges). Oracle also maintains tables that monitor ongoing database activity. These "dynamic performance tables" are described in Chapter 3, "Dynamic Performance (V$) Views".

Data dictionary tables are not directly accessible, but you can access information in them through data dictionary views. To list the data dictionary views available to you, query the view DICTIONARY.

Many data dictionary tables have three corresponding views:

- An ALL_ view displays all the information accessible to the current user, including information from the current user's schema as well as information from objects in other schemas, if the current user has access to those objects by way of grants of privileges or roles.
- A DBA_ view displays all relevant information in the entire database. DBA_ views are intended only for administrators. They can be accessed only by users with the SELECT_ANY_TABLE privilege. (This privilege is assigned to the DBA role when the system is initially installed.)
- A USER_ view displays all the information from the schema of the current user. No special privileges are required to query these views.

The columns of the ALL_, DBA_, and USER_, views corresponding to a single data dictionary table are usually nearly identical. Therefore, these views are described in full only once in this chapter, at their first occurrence alphabetically, and are listed without full descriptions at their other occurrences.

A number of data dictionary views are accessible only if you are using Oracle Advanced Replication. Table 2–1 lists these views. The first part of the table lists

views for which there are ALL_, DBA_, and USER_ variations. The second part of the table lists additional Advanced Replication views.

See Also: *Oracle8i Replication Management API Reference*

Table 2–1 Oracle Advanced Replication Data Dictionary Views

ALL_, DBA_, USER_ views:		
ALL_REPCATLOG	DBA_REPCATLOG	USER_REPCATLOG
ALL_REPCAT_REFRESH_TEMPLATES	DBA_REPCAT_REFRESH_TEMPLATES	USER_REPCAT_REFRESH_TEMPLATES
ALL_REPCAT_TEMPLATE_OBJECTS	DBA_REPCAT_TEMPLATE_OBJECTS	USER_REPCAT_TEMPLATE_OBJECTS
ALL_REPCAT_TEMPLATE_PARMS	DBA_REPCAT_TEMPLATE_PARMS	USER_REPCAT_TEMPLATE_PARMS
ALL_REPCAT_TEMPLATE_SITES	DBA_REPCAT_TEMPLATE_SITES	USER_REPCAT_TEMPLATE_SITES
ALL_REPCAT_USER_AUTHORIZATIONS	DBA_REPCAT_USER_AUTHORIZATIONS	USER_REPCAT_USER_AUTHORIZATIONS
ALL_REPCAT_USER_PARM_VALUES	DBA_REPCAT_USER_PARM_VALUES	USER_REPCAT_USER_PARM_VALUES
ALL_REPCOLUMN	DBA_REPCOLUMN	USER_REPCOLUMN
ALL_REPCOLUMN_GROUP	DBA_REPCOLUMN_GROUP	USER_REPCOLUMN_GROUP
ALL_REPCONFLICT	DBA_REPCONFLICT	USER_REPCONFLICT
ALL_REPDDL	DBA-REPDDL	USER_REPDDL
ALL_REPGENERATED	DBA_REPGENERATED	USER_REPGENERATED
ALL_REPGENOBJECTS	DBA_REPGENOBJECTS	USER_REPGENOBJECTS
ALL_REPGROUP	DBA_REPGROUP	USER_REPGROUP
ALL_REPGROUPED_COLUMN	DBA_REPGROUPED_COLUMN	USER_REPGROUPED_COLUMN
ALL_REPKEY_COLUMNS	DBA_REPKEY_COLUMNS	USER_REPKEY_COLUMNS
ALL_REPOBJECT	DBA_REPOBJECT	USER_REPOBJECT
ALL_REPPARAMETER_COLUMN	DBA_REPPARAMETER_COLUMN	USER_REPPARAMETER_COLUMN
ALL_REPPRIORITY	DTA-REPPRIORITY	USER_REPPRIORITY
ALL_REPPRIORITY_GROUP	DBA_REPPRIORITY_GROUP	USER_REPPRIORITY_GROUP
ALL_REPPROP	DBA_REPPROP	USER_REPPROP
ALL_REPRESOLUTION	DBA_REPRESOLUTION	USER_REPRESOLUTION
ALL_REPRESOLUTION_METHOD	DBA_REPRESOLUTION_METHOD	USER_REPRESOLUTION_METHOD
ALL_REPRESOL_STATS_CONTROL	DBA_REPRESOL_STATS_CONTROL	USER_REPRESOL_STATS_CONTROL
ALL_REPSITES	DBA_REPSITES	USER_REPSITES

Table 2–1 Oracle Advanced Replication Data Dictionary Views

Additional views:		
DEFCALL	DEFERROR	DEFTRAN
DEFCALLDEST	DEWFLOB	DEFTRANDEST
DEFDEFAULTDEST	DEFPROPAGATOR	
DEFERRCOUNT	DEFSCHEDULE	

The remainder of this chapter lists the data dictionary views in alphabetical order.

ALL_ALL_TABLES

ALL_ALL_TABLES describes all object tables and relational tables accessible to the current user.

Related views:

- DBA_ALL_TABLES describes all object tables and relational tables in the database.
- USER_ALL_TABLES describes all object tables and relational tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the table
CLUSTER_NAME	VARCHAR2(30)		Name of the cluster, if any, to which the table belongs
IOT_NAME	VARCHAR2(30)		Name of the index organized table, if any, to which the overflow entry belongs
PCT_FREE	NUMBER		Minimum percentage of free space in a block
PCT_USED	NUMBER		Minimum percentage of used space in a block
INI_TRANS	NUMBER		Initial number of transactions
MAX_TRANS	NUMBER		Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment

Column	Datatype	NULL	Description
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute
BACKED_UP	VARCHAR2(1)		Whether the table has been backed up since the last modification
NUM_ROWS	NUMBER		Number of rows in the table
BLOCKS	NUMBER		Number of used blocks in the table
EMPTY_BLOCKS	NUMBER		Number of empty (never used) blocks in the table
AVG_SPACE	NUMBER		Average available free space in the table
CHAIN_CNT	NUMBER		Number of rows in the table that are chained from one data block to another or that have migrated to a new block, requiring a link to preserve the old rowid. This column is updated only after you ANALYZE the table.
AVG_ROW_LEN	NUMBER		Average row length, including row overhead
AVG_SPACE_FREELIST_BLOCKS	NUMBER		Average freespace of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		Number of blocks on the freelist
DEGREE	VARCHAR2(10)		Number of threads per instance for scanning the table
INSTANCES	VARCHAR2(10)		Number of instances across which the table is to be scanned
CACHE	VARCHAR2(5)		Whether the cluster is to be cached in the buffer cache (CACHE | NOCACHE)
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or disabled
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this table
LAST_ANALYZED	DATE		Date on which this table was most recently analyzed
PARTITIONED	VARCHAR2(3)		Whether this table is partitioned: YES or NO
IOT_TYPE	VARCHAR2(12)		If an index organized table, whether an overflow segment has been specified (OVERFLOW) or not (NULL)
OBJECT_ID_TYPE	VARCHAR2(16)		Whether the object ID is USER-DEFINED or SYSTEM GENERATED
TABLE_TYPE_OWNER	VARCHAR2(30)		If an object table, owner of the type from which the table is created
TABLE_TYPE	VARCHAR2(30)		If an object table, type of the table
TEMPORARY	VARCHAR2(1)		Whether the table is temporary

Column	Datatype	NULL	Description
SECONDARY	VARCHAR2(1)		Whether the table is a secondary object created by the ODCIIndexCreate method of the Oracle8*i* Data Cartridge to contain the contents of a domain index (Y or N)
NESTED	VARCHAR2(3)		Whether this is a nested table
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the table
ROW_MOVEMENT	VARCHAR2(8)		If a partitioned table, whether row movement is enabled or disabled
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether statistics were collected by analyzing the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO).
USER_STATS	VARCHAR2(3)		Whether statistics were entered directly by the user
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: SYS$SESSION: the rows are preserved for the duration of the session SYS$TRANSACTION: the rows are deleted after COMMIT. Null for a permanent table.
SKIP_CORRUPT	VARCHAR2(8)		Whether Oracle ignores blocks marked corrupt during table and index scans (ENABLED) or raises an error (DISABLED). To enable this feature, run the DBMS_REPAIR.skip_corrupt_blocks procedure.
MONITORING	VARCHAR2(3)		Whether the table has the MONITORING attribute set

ALL_ARGUMENTS

ALL_ARGUMENTS lists all arguments of procedures and functions that are accessible to the user.

Related view:

- USER_ARGUMENTS lists all arguments of procedures and functions that are owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the procedure or function
PACKAGE_NAME	VARCHAR2(30)		Package name

Column	Datatype	NULL	Description
OBJECT_ID	NUMBER	NOT NULL	Name of the package containing the procedure or function
OVERLOAD	VARCHAR2(40)		Overload unique identifier
ARGUMENT_NAME	VARCHAR2(30)		Name of the argument
POSITION	NUMBER	NOT NULL	Position in argument list, or NULL for function return value
SEQUENCE	NUMBER	NOT NULL	Argument sequence, including all nesting levels
DATA_LEVEL	NUMBER	NOT NULL	Nesting depth of argument for composite types
DATA_TYPE	VARCHAR2(14)		Datatype of the argument
DEFAULT_VALUE	LONG		Default value for the argument
DEFAULT_LENGTH	NUMBER		Length of default value for the argument
IN_OUT	VARCHAR2(9)		Argument direction (IN, OUT, or IN / OUT)
DATA_LENGTH	NUMBER		Length of the column in bytes
DATA_PRECISION	NUMBER		Length in decimal digits (NUMBER) or binary digits (FLOAT)
DATA_SCALE	NUMBER		Digits to right of decimal point in a number
RADIX	NUMBER		Argument radix for a number
CHARACTER_SET_NAME	VARCHAR2(44)		Character set name for the argument
TYPE_OWNER	VARCHAR2(30)		Username of the owner of the type of the argument
TYPE_NAME	VARCHAR2(30)		Name of the type of the argument. If the type is a package local type (that is, it is declared in a package specification), this column displays the name of the package.
TYPE_SUBNAME	VARCHAR2(30)		Relevant only for package local types. Displays the name of the type declared in the package identified in the TYPE_NAME column.
TYPE_LINK	VARCHAR2(128)		Relevant only for package local types when the package identified in the TYPE_NAME column is a remote package. This columns displays the database link used to refer to the remote package.
PLS_TYPE	VARCHAR2(30)		For numeric arguments, the name of the PL / SQL type of the argument. Null otherwise.

ALL_ASSOCIATIONS

ALL_ASSOCIATIONS describes user-defined statistics associated with objects accessible to the current user.

Related views:

- DBA_ASSOCIATIONS describes all user-defined statistics in the database.
- USER_ASSOCIATIONS describes user-defined statistics associated with objects owned by the current user.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object for which the association is being defined
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object for which the association is being defined
COLUMN_NAME	VARCHAR2(30)		Column name in the object for which the association is being defined
OBJECT_TYPE	VARCHAR2(9)		Kind of object with which statistics are being associated: column, type, package or function, indextype, or domain index.
STATSTYPE_SCHEMA	VARCHAR2(30)		Owner of the statistics type
STATSTYPE_NAME	VARCHAR2(30)		Name of statistics type that contains the cost, selectivity or statistics functions
DEF_SELECTIVITY	NUMBER		Default selectivity of the object, if any
DEF_CPU_COST	NUMBER		Default CPU cost of the object, if any
DEF_IO_COST	NUMBER		Default I/O cost of the object, if any
DEF_NET_COST	NUMBER		Default networking cost of the object, if any

ALL_CATALOG

ALL_CATALOG lists all schema objects accessible to the user.

Related views:

- DBA_CATALOG lists all schema objects in the entire database.
- USER_CATALOG lists all schema objects in the current user's schema. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
TABLE_TYPE	VARCHAR2(11)		Type of the object: INDEX, TABLE, CLUSTER, VIEW, SYNONYM, SEQUENCE, UNDEFINED

ALL_CLUSTERS

ALL_CLUSTERS describes all clusters accessible to the user.

Related views:

- DBA_CLUSTERS describes all clusters in the database.
- USER_CLUSTERS describes all clusters owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Name of the cluster
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the cluster
PCT_FREE	NUMBER		Minimum percentage of free space in a block
PCT_USED	NUMBER	NOT NULL	Minimum percentage of used space in a block
KEY_SIZE	NUMBER		Estimated size of cluster key plus associated rows
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENTS	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated to this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment
AVG_BLOCKS_PER_KEY	NUMBER		Number of blocks in the table divided by number of hash keys
CLUSTER_TYPE	VARCHAR2(5)		Type of cluster: B*-Tree index or hash

Column	Datatype	NULL	Description
FUNCTION	VARCHAR2(15)		If a hash cluster, the hash function
HASHKEYS	NUMBER		If a hash cluster, the number of hash keys (hash buckets)
DEGREE	VARCHAR2(10)		Number of threads per instance for scanning the cluster
INSTANCES	VARCHAR2(10)		Number of instances across which the cluster is to be scanned
CACHE	VARCHAR2(5)		Whether the cluster is to be cached in the buffer cache (CACHE | NOCACHE)
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the cluster
SINGLE_TABLE	VARCHAR2(5)		Whether this is a single-table cluster (Y or N)

ALL_CLUSTER_HASH_EXPRESSIONS

ALL_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters accessible to the current user.

Related views:

- DBA_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters in the database.
- USER_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Name of the cluster
HASH_EXPRESSION	LONG		Text of hash function of the hash cluster

ALL_COL_COMMENTS

ALL_COL_COMMENTS lists comments on columns of tables and views accessible to the current user.

Related views:

- DBA_COL_COMMENTS lists comments on all tables and views in the database.

- USER_COL_COMMENTS lists comments on tables and views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
COMMENTS	VARCHAR2(4000)		Comment on the column

ALL_COL_PRIVS

ALL_COL_PRIVS describes column object grants for which the current user or PUBLIC is the object owner, grantor, or grantee. This view displays OWNER but not TABLE_SCHEMA.

Related views:

- DBA_COL_PRIVS describes all column object grants in the database. This view displays OWNER but not TABLE_SCHEMA.
- USER_COL_PRIVS describes column object grants for which the current user is the object owner, grantor, or grantee. This view displays TABLE_SCHEMA but not OWNER.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object. Displayed only in the DBA_ and USER_ views.
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Schema of the object. Displayed only in the ALL_ view.
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2(3)		Whether privileges were granted with ADMIN OPTION (YES or NO)

ALL_COL_PRIVS_MADE

ALL_COL_PRIVS_MADE lists column object grants for which the current user is object owner or grantor.

Related view:

- USER_COL_PRIVS_MADE describes column object grants for which the current user is the grantor. This view does not display the OWNER column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2(3)		Whether the privilege was granted with ADMIN OPTION (YES or NO)

ALL_COL_PRIVS_RECD

ALL_COL_PRIVS_RECD describes column object grants for which the current user or PUBLIC is the grantee.

Related view:

- USER_COL_PRIVS_RECD describes column object grants for which the current user is the grantee. This view does not display the OWNER column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NUL	Name of the owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2(3)		Whether the privilege was granted with ADMIN OPTION (YES or NO)

ALL_COLL_TYPES

ALL_COLL_TYPES describes all named collection types (varrays and nested tables) accessible to the user.

Related views:

- DBA_COLL_TYPES describes all named collection types in the database.
- USER_COLL_TYPES describes all named collection types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the collection
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the collection
COLL_TYPE	VARCHAR2(30)	NOT NULL	Description of the collection, such as VARYING ARRAY, [nested] TABLE
UPPER_BOUND	NUMBER		For varrays only, maximum size
ELEM_TYPE_MOD	VARCHAR2(7)		Whether the collection
ELEM_TYPE_OWNER	VARCHAR2(30)		Owner of the type upon which the collection is based. This value is useful primarily in the case of a user-defined type.
ELEM_TYPE_NAME	VARCHAR2(30)		Name of the datatype or user-defined type upon which the collection is based
LENGTH	NUMBER		Length of CHAR elements or maximum length of VARCHAR or VARCHAR2 elements
PRECISION	NUMBER		Decimal precision of NUMBER or DECIMAL elements; binary precision of FLOAT elements

Column	Datatype	NULL	Description
SCALE	NUMBER		Scale of NUMBER or DECIMAL elements
CHARACTER_SET_NAME	VARCHAR2(44)		Name of the character set (CHAR_CS | NCHAR_CS)
ELEM_STORAGE	VARCHAR2(7)		[Obsolete column]
NULLS_STORED	VARCHAR2(3)		[Obsolete column]

ALL_CONS_COLUMNS

ALL_CONS_COLUMNS describes columns that are accessible to the current user and that are specified in constraints.

Related views:

- DBA_CONS_COLUMNS describes all columns in the database that are specified in constraints.
- USER_CONS_COLUMNS describes columns that are owned by the current user and that are specified in constraints.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the constraint definition
CONSTRAINT_NAME	VARCHAR2(30)	NOT NULL	Name of the constraint definition
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table with constraint definition
COLUMN_NAME	VARCHAR2(4000)		Name of the column or attribute of the object type column specified in the constraint definition **Note:** If you create a constraint on a user-defined REF column, the system creates the constraint on the attributes that make up the REF column. Therefore, the column names displayed in this view are the attribute names, with the REF column name as a prefix, in the following form: `"REF_name"."atttribute"`
POSITION	NUMBER		Original position of column or attribute in the definition of the object

ALL_CONSTRAINTS

ALL_CONSTRAINTS describes constraint definitions on tables accessible to the current user.

Related views:

- DBA_CONSTRAINTS describes all constraint definitions in the database.
- USER_CONSTRAINTS describes constraint definitions on tables in the current user's schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the constraint definition
CONSTRAINT_NAME	VARCHAR2(30)	NOT NULL	Name of the constraint definition
CONSTRAINT_TYPE	VARCHAR2(1)		Type of constraint definition: ■ C (check constraint on a table) ■ P (primary key) ■ U (unique key) ■ R (referential integrity) ■ V (with check option, on a view) ■ O (with read only, on a view)
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table with constraint definition
SEARCH_CONDITION	LONG		Text of search condition for a check constraint
R_OWNER	VARCHAR2(30)		Owner of table referred to in a referential constraint
R_CONSTRAINT_NAME	VARCHAR2(30)		Name of the unique constraint definition for referenced table
DELETE_RULE	VARCHAR2(9)		Delete rule for a referential constraint: CASCADE or NO ACTION
STATUS	VARCHAR2(8)		Enforcement status of constraint: ENABLED or DISABLED
DEFERRABLE	VARCHAR2(14)		Whether the constraint is deferrable
DEFERRED	VARCHAR2(9)		Whether the constraint was initially deferred
VALIDATED	VARCHAR2(13)		Whether all data obeys the constraint: VALIDATED or NOT VALIDATED
GENERATED	VARCHAR2(14)		Whether the name of the constraint is user or system generated
BAD	VARCHAR2(3)		A YES value indicates that this constraint specifies a century in an ambiguous manner. To avoid errors resulting from this ambiguity, rewrite the constraint using the TO_DATE function with a four-digit year. **See Also:** the TO_DATE function in *Oracle8i SQL Reference* and *Oracle8i Application Developer's Guide - Fundamentals.*
RELY	VARCHAR2(4)		Whether an enabled constraint is enforced or unenforced. **See Also:** the *constraint_clause* in *Oracle8i SQL Reference.*

Column	Datatype	NULL	Description
LAST_CHANGE	DATE		When the constraint was last enabled or disabled

ALL_CONTEXT

ALL_CONTEXT describes all context namespaces in the current session for which attributes and values have been specified using the DBMS_SESSION.SET_CONTEXT procedure.

Related view:

- DBA_CONTEXT describes all context namespaces defined the database, regardless whether any attributes have been specified for them using the DBMS_SESSION.SET_CONTEXT procedure.

Column	Datatype	NULL	Description
NAMESPACE	VARCHAR2(30)	NOT NULL	Name of the context namespace
SCHEMA	VARCHAR2(30)	NOT NULL	Schema name of the designated package that can set attributes using this namespace
PACKAGE	VARCHAR2(30)	NOT NULL	Package name of the designated package that can set attributes using this namespace

ALL_DB_LINKS

ALL_DB_LINKS describes database links accessible to the user. This view does not display the PASSWORD column.

Related views:

- DBA_DB_LINKS describes all database links defined in the database. This view does not display the PASSWORD column.
- USER_DB_LINKS describes database links in the current user's schema. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the database link
DB_LINK	VARCHAR2(12)	NOT NULL	Name of the database link
USERNAME	VARCHAR2(30)		Name of user when logging in

Column	Datatype	NULL	Description
PASSWORD	VARCHAR2(30)		Login password for current user.
HOST	VARCHAR2(200)		Net8 connect string
CREATED	DATE	NOT NULL	Creation time of the database link

ALL_DEF_AUDIT_OPTS

ALL_DEF_AUDIT_OPTS contains default object-auditing options that will be applied when objects are created. The output for each column takes one of the following forms:

- -/-: no default auditing
- S/-: auditing whenever successful
- -/S: auditing whenever not successful

Column	Datatype	NULL	Description
ALT	VARCHAR2(3)		Auditing ALTER WHENEVER SUCCESSFUL / UNSUCCESSFUL
AUD	VARCHAR2(3)		Auditing AUDIT WHENEVER SUCCESSFUL / UNSUCCESSFUL
COM	VARCHAR2(3)		Auditing COMMENT WHENEVER SUCCESSFUL / UNSUCCESSFUL
DEL	VARCHAR2(3)		Auditing DELETE WHENEVER SUCCESSFUL / UNSUCCESSFUL
GRA	VARCHAR2(3)		Auditing GRANT WHENEVER SUCCESSFUL / UNSUCCESSFUL
IND	VARCHAR2(3)		Auditing INDEX WHENEVER SUCCESSFUL / UNSUCCESSFUL
INS	VARCHAR2(3)		Auditing INSERT WHENEVER SUCCESSFUL / UNSUCCESSFUL
LOC	VARCHAR2(3)		Auditing LOCK WHENEVER SUCCESSFUL / UNSUCCESSFUL
REN	VARCHAR2(3)		Auditing RENAME WHENEVER SUCCESSFUL / UNSUCCESSFUL
SEL	VARCHAR2(3)		Auditing SELECT WHENEVER SUCCESSFUL / UNSUCCESSFUL
UPD	VARCHAR2(3)		Auditing UPDATE WHENEVER SUCCESSFUL / UNSUCCESSFUL
REF	VARCHAR2(3)		Auditing REFERENCES WHENEVER SUCCESSFUL / UNSUCCESSFUL (not used)
EXE	VARCHAR2(3)		Auditing EXECUTE WHENEVER SUCCESSFUL / UNSUCCESSFUL

ALL_DEPENDENCIES

ALL_DEPENDENCIES describes dependencies between procedures, packages, functions, package bodies, and triggers accessible to the user, including

dependencies on views created without any database links. This view does not display the SCHEMAID column.

Related views:

- DBA_DEPENDENCIES describes all dependencies between objects in the database. This view does not display the SCHEMAID column.
- USER_DEPENDENCIES describes dependencies between objects in the current user's schema. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of object
REFERENCED_OWNER	VARCHAR2(30)		Owner of the parent object
REFERENCED_NAME	VARCHAR2(64)		Type of parent object
REFERENCED_TYPE	VARCHAR2(12)		Type of referenced object
REFERENCED_LINK_NAME	VARCHAR2(128)		Name of the link to the parent object (if remote)
SCHEMAID	NUMBER		ID of the current schema
DEPENDENCY_TYPE	VARCHAR2(4)		Whether the dependency is a REF dependency (REF) or not (HARD)

ALL_DIMENSIONS

ALL_DIMENSIONS describes dimension objects accessible to the current user.

Related views:

- DBA_DIMENSIONS describes all dimensions in the database.
- USER_DIMENSIONS describes dimensions in the current user's schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
INVALID	VARCHAR2(1)		Whether the dimension is invalid (Y/N)
REVISION	NUMBER		Dimension revision level

ALL_DIM_ATTRIBUTES

ALL_DIM_ATTRIBUTES describes the relationship between a dimension level and a functionally dependent column. The level columns and the dependent column must be in the same table.

Related views:

- DBA_DIM_ATTRIBUTES describes all such dimension relationships in the database.
- USER_DIM_ATTRIBUTES describes all such dimension attributes in the current user's schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Dependent column name
INFERRED	CHAR(1)		Whether inferred

ALL_DIM_CHILD_OF

ALL_DIM_CHILD_OF describes hierarchical relationships of 1 to *n* between the pairs of levels in the dimensions accessible to the current user.

Related views:

- DBA_DIM_CHILD_OF describes all such hierarchical relationships in the database.
- USER_DIM_CHILD_OF describes all such hierarchical attributes in the current user's schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
HIERARCHY_NAME	VARCHAR2(30)		Hierarchy name
POSITION	NUMBER	NOT NULL	Hierarchical position within this hierarchy, position 1 being the most detailed

Column	Datatype	NULL	Description
CHILD_LEVEL_NAME	VARCHAR2(30)		Child side of 1:n relationship
JOIN_KEY_ID	VARCHAR2(40)		If non-null, then the child joins to the parent
PARENT_LEVEL_NAME	VARCHAR2(30)		Parent side of 1:n relationship in relation to the CHILD_LEVEL_NAME

ALL_DIM_HIERARCHIES

ALL_DIM_HIERARCHIES describes all dimension hierarchies accessible to the current user.

Related views:

- DBA_DIM_HIERARCHIES describes all such hierarchies in the database.
- USER_DIM_HIERARCHIES describes all such hierarchies owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
HIERARCHY_NAME	VARCHAR2(30)		Hierarchy name

ALL_DIM_JOIN_KEY

ALL_DIM_JOIN_KEY describes the joins between two dimension tables that are accessible to the current user. The join is always specified between a parent dimension level column and a child column.

Related views:

- DBA_DIM_JOIN_KEY describes all such joins in the database.
- USER_DIM_JOIN_KEY describes all such joins owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
DIM_KEY_ID	NUMBER	NOT NULL	Unique within a dimension

Column	Datatype	NULL	Description
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
KEY_POSITION	NUMBER	NOT NULL	Ordinal position of the key column within the level
HIERARCHY_NAME	VARCHAR2(30)		Name of the key column
CHILD_JOIN_COLUMN	VARCHAR2(30)	NOT NULL	Name of the join column

ALL_DIM_LEVELS

ALL_DIM_LEVELS describes the dimension levels accessible to the current user. All columns of a dimension level must come from the same relation.

Related views:

- DBA_DIM_LEVELS describes all dimension levels in the database.
- USER_DIM_LEVELS describes the levels of all dimensions owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
LEVEL_NAME	VARCHAR2(30)		Unique within a dimension
NUM_COLUMNS	NUMBER		Number of columns in the level definition
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Owner of the detail object that the keys of this level come from
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Name of the table that the keys of this level come from

ALL_DIM_LEVEL_KEY

ALL_DIM_LEVEL_KEY describes a column of a dimension level accessible to the current user. The position of a column within a level is specified by KEY_POSITION.

Related views:

- DBA_DIM_LEVEL_KEY describes all columns of dimension levels in the database.

- USER_DIM_LEVEL_KEY describes all columns of dimension levels owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
KEY_POSITION	NUMBER	NOT NULL	Ordinal position of the key column within the level
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the key column

ALL_DIRECTORIES

ALL_DIRECTORIES describes all directories accessible to the user.

Related view:

- DBA_DIRECTORIES describes all directories in the database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the directory (always SYS)
DIRECTORY_NAME	VARCHAR2(30)	NOT NULL	Name of the directory
DIRECTORY_PATH	VARCHAR2(4000)		Operating system pathname for the directory

ALL_ERRORS

ALL_ERRORS describes current errors on all stored objects (views, procedures, functions, packages, and package bodies) accessible to the current user.

Related views:

- DBA_ERRORS describes all current errors on all views, procedures, functions, packages, and package bodies in the database.
- USER_ERRORS describes all current errors on all views, procedures, functions, packages, and package bodies owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of object
SEQUENCE	NUMBER	NOT NULL	Sequence number, for ordering
LINE	NUMBER	NOT NULL	Line number at which this error occurs
POSITION	NUMBER	NOT NULL	Position in the line at which this error occurs
TEXT	VARCHAR2(4000)	NOT NULL	Text of the error

ALL_IND_COLUMNS

ALL_IND_COLUMNS describes the columns of indexes on all tables accessible to the current user.

Related views:

- DBA_IND_COLUMNS describes the columns of indexes on all tables in the database.
- USER_IND_COLUMNS describes the columns of indexes owned by the current user and columns of indexes on tables owned by the current user. This view does not display the INDEX_OWNER or TABLE_OWNER columns.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table or cluster
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table or cluster
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of object type column **Note:** If you create an index on a user-defined REF column, the system creates the index on the attributes that make up the REF column. Therefore, the column names displayed in this view are the attribute names, with the REF column name as a prefix, in the following form: `"REF_name"."atttribute"`
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within the index
COLUMN_LENGTH	NUMBER	NOT NULL	Indexed length of the column

Column	Datatype	NULL	Description
DESCEND	VARCHAR2(4)		Whether the column is sorted in descending order (Y / N)

ALL_IND_EXPRESSIONS

ALL_IND_EXPRESSIONS describes the expressions of function-based indexes on tables accessible to the current user.

Related views:

- DBA_IND_EXPRESSIONS describes the expressions of all function-based indexes in the database.
- USER_IND_EXPRESSIONS describes the expressions of function-based indexes on tables owned by the current user. This view does not display the INDEX_OWNER or TABLE_OWNER column.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Index owner
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table or cluster owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table or cluster name
COLUMN_EXPRESSION	LONG		Function-based index expression defining the column
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within index

ALL_IND_PARTITIONS

ALL_IND_PARTITIONS describes, for each index partition accessible to the current user, the partition-level partitioning information, the storage parameters for the partition, and various partition statistics collected by ANALYZE statements.

Related views:

- DBA_IND_PARTITIONS describes all index partitions in the database.
- USER_IND_PARTITIONS describes the index partitions owned by the current user. This view does not display the INDEX_OWNER column.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)		Owner of the index
INDEX_NAME	VARCHAR2(30)		Name of the index
COMPOSITE	VARCHAR2(3)		Whether the partition belongs to a local index on a composite-partitioned table (YES | NO)
PARTITION_NAME	VARCHAR2(30)		Name of the partition
SUBPARTITION_COUNT	NUMBER		If a local index on a composite-partitioned table, the number of subpartitions in the partition
HIGH_VALUE	LONG		Partition bound value expression
HIGH_VALUE_LENGTH	NUMBER		Length of partition bound value expression
PARTITION_POSITION	NUMBER		Position of the partition within the index
STATUS	VARCHAR2(8)		Whether index partition is USABLE or UNUSABLE
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the partition
PCT_FREE	NUMBER		Minimum percentage of free space in a block
INI_TRANS	NUMBER		Initial number of transactions
MAX_TRANS	NUMBER		Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER		Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of process freelist groups allocated in this segment
LOGGING	VARCHAR2(7)		Logging attribute of partition
COMPRESSION	VARCHAR2(8)		Indicates whether key compression is ENABLED or DISABLED for a partitioned index. NULL for a nonpartitioned index.
BLEVEL	NUMBER		B*-Tree level: depth of the index from its root block to its leaf blocks. A depth of 0 indicates that the root block and leaf block are the same.
LEAF_BLOCKS	NUMBER		Number of leaf blocks in the index partition
DISTINCT_KEYS	NUMBER		Number of distinct keys in the index partition

Column	Datatype	NULL	Description
AVG_LEAF_BLOCKS_PER_KEY	NUMBER		Average number of leaf blocks in which each distinct value in the index appears, rounded to the nearest integer. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is always 1.
AVG_DATA_BLOCKS_PER_KEY	NUMBER		Average number of data blocks in the table that are pointed to by a distinct value in the index rounded to the nearest integer. This statistic is the average number of data blocks that contain rows that contain a given value for the indexed columns.
CLUSTERING_FACTOR	NUMBER		Indicates the amount of order of the rows in the table based on the values of the index. ■ If the value is near the number of blocks, then the table is very well ordered. In this case, the index entries in a single leaf block tend to point to rows in the same data blocks. ■ If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
NUM_ROWS	NUMBER		Number of rows returned by the ANALYZE command
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this partition
LAST_ANALYZED	DATE		Date on which this partition was most recently analyzed
BUFFER_POOL	VARCHAR2(7)		The actual buffer pool for the partition
USER_STATS	VARCHAR2(3)		Whether the statistics were entered directly by the user (YES | NO)
PCT_DIRECT_ACCESS	NUMBER		If a secondary index on index-organized table, the percentage of rows with VALID guess
GLOBAL_STATS	VARCHAR2(3)		Indicates whether statistics for the partition were collected for the partition as a whole (YES) or were estimated from statistics on underlying subpartitions (NO).

ALL_IND_SUBPARTITIONS

ALL_IND_SUBPARTITIONS describes, for each index subpartition accessible to the current user, the partition-level partitioning information, the storage parameters for the subpartition, and various partition statistics collected by ANALYZE statements.

Related views:

- DBA_IND_SUBPARTITIONS describes all index subpartitions in the database.

- USER_IND_SUBPARTITIONS describes the index subpartitions owned by the current user. This view does not display the INDEX_OWNER column.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
PARTITION_NAME	VARCHAR2(30)		Name of the partition
SUBPARTITION_NAME	VARCHAR2(30)		Name of the subpartition
SUBPARTITION_POSITION	NUMBER	NOT NULL	Position of a subpartition within a partition
STATUS	VARCHAR2(8)		Whether index partition is usable or not
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the partition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of process freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of partition
BLEVEL	NUMBER		B-Tree level: depth of the index from its root block to its leaf blocks. A depth of 0 indicates that the root block and leaf block are the same.
LEAF_BLOCKS	NUMBER		Number of leaf blocks in the index
DISTINCT_KEYS	NUMBER		Number of distinct keys in the index partition
AVG_LEAF_BLOCKS_PER_KEY	NUMBER		Average number of leaf blocks in which each distinct value in the index appears, rounded to the nearest integer. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is always 1.
AVG_DATA_BLOCKS_PER_KEY	NUMBER		Average number of data blocks in the table that are pointed to by a distinct value in the index rounded to the nearest integer. This statistic is the average number of data blocks that contain rows that contain a given value for the indexed columns.

Column	Datatype	NULL	Description
CLUSTERING_FACTOR	NUMBER		Indicates the amount of order of the rows in the table based on the values of the index. ■ If the value is near the number of blocks, then the table is very well ordered. In this case, the index entries in a single leaf block tend to point to rows in the same data blocks. ■ If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
NUM_ROWS	NUMBER		Number of rows in this index subpartition
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this subpartition
LAST_ANALYZED	DATE		Date on which this partition was most recently analyzed
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the subpartition
USER_STATS	VARCHAR2(3)		Whether the statistics were entered directly by the user
GLOBAL_STATS	VARCHAR2(3)		Indicates whether column statistics for the subpartition statistics were collected by analyzing the table as a whole (YES) or estimated from statistics gathered on partitions and subpartitions (NO).

ALL_INDEXES

ALL_INDEXES describes indexes on all tables accessible to the user. To gather statistics for this view and the related views DBA_INDEXES and USER_INDEXES, use the SQL command ANALYZE.

Related views:

- DBA_INDEXES describes all indexes in the database.
- USER_INDEXES describes the indexes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
INDEX_TYPE	VARCHAR2(12)		Type of index (NORMAL, BITMAP, FUNCTION-BASED NORMAL, FUNCTION-BASED BITMAP, or DOMAIN)
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the indexed object

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the indexed object
TABLE_TYPE	CHAR(5)		Type of the indexed object (for example, TABLE, CLUSTER)
UNIQUENESS	VARCHAR2(9)		Whether the index is UNIQUE or NONUNIQUE
COMPRESSION	VARCHAR2(11)		Whether the index is ENABLED or DISABLED
PREFIX_LENGTH	NUMBER		Number of columns in the prefix of the compression key
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the index
INI_TRANS	NUMBER		Initial number of transactions
MAX_TRANS	NUMBER		Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent
NEXT_EXTENT	NUMBER		Size of secondary extents
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percentage increase in extent size
PCT_THRESHOLD	NUMBER		Threshold percentage of block space allowed per index entry
INCLUDE_COLUMN	NUMBER		Column ID of the last column to be included in index-organized table primary key (non-overflow) index. This column maps to the COLUMN_ID column of the *_TAB_COLUMNS data dictionary views.
FREELISTS	NUMBER		Number of process freelists allocated to this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment
PCT_FREE	NUMBER		Minimum percentage of free space in a block
LOGGING	VARCHAR2(3)		Logging information
BLEVEL	NUMBER		B*-Tree level: depth of the index from its root block to its leaf blocks. A depth of 0 indicates that the root block and leaf block are the same.
LEAF_BLOCKS	NUMBER		Number of leaf blocks in the index
DISTINCT_KEYS	NUMBER		Number of distinct indexed values. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is the same as the number of rows in the table (USER_TABLES.NUM_ROWS)
AVG_LEAF_BLOCKS_PER_KEY	NUMBER		Average number of leaf blocks in which each distinct value in the index appears, rounded to the nearest integer. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is always 1.

Column	Datatype	NULL	Description
AVG_DATA_BLOCKS_PER_KEY	NUMBER		Average number of data blocks in the table that are pointed to by a distinct value in the index rounded to the nearest integer. This statistic is the average number of data blocks that contain rows that contain a given value for the indexed columns.
CLUSTERING_FACTOR	NUMBER		Indicates the amount of order of the rows in the table based on the values of the index. ■ If the value is near the number of blocks, then the table is very well ordered. In this case, the index entries in a single leaf block tend to point to rows in the same data blocks. ■ If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
STATUS	VARCHAR2(8)		Whether a nonpartitioned index is VALID or UNUSABLE
NUM_ROWS	NUMBER		Number of rows in the index
SAMPLE_SIZE	NUMBER		Size of the sample used to analyze the index
LAST_ANALYZED	DATE		Date on which this index was most recently analyzed
DEGREE	VARCHAR2(40)		Number of threads per instance for scanning the index
INSTANCES	VARCHAR2(40)		Number of instances across which the indexes to be scanned
PARTITIONED	VARCHAR2(3)		Whether this index is partitioned (YES | NO)
TEMPORARY	VARCHAR2(1)		Whether the index is on a temporary table
GENERATED	VARCHAR2(1)		Whether the name of the index is system generated (Y|N)
SECONDARY	VARCHAR2(1)		Whether the index is a secondary object created by the ODCIIndexCreate method of the Oracle8*i* Data Cartridge (Y or N)
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool to be used for the index blocks
USER_STATS	VARCHAR2(3)		Whether the statistics were entered directly by the user
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: ■ SYS$SESSION: the rows are preserved for the duration of the session ■ SYS$TRANSACTION: the rows are deleted after COMMIT. Null for a permanent table.
PCT_DIRECT_ACCESS	NUMBER		For a secondary index on an index-organized table, the percentage of rows with VALID guess

Column	Datatype	NULL	Description
ITYP_OWNER	NUMBER		For a domain index, the owner of the indextype
ITYP_NAME	VARCHAR2(30)		For a domain index, the name of the indextype
PARAMETERS	VARCHAR2(1000)		For a domain index, the parameter string.
GLOBAL_STATS	VARCHAR2(3)		For partitioned indexes, indicates whether statistics were collected by analyzing index as a whole (YES) or were estimated from statistics on underlying index partitions and subpartitions (NO).
DOMIDX_STATUS	VARCHAR		Reflects the status of the domain index. ■ NULL: the specified index is not a domain index ■ VALID: the index is a valid domain index ■ IDXTYP_INVLD: the indextype of this domain index is invalid
DOMIDX_OPSTATUS	VARCHAR		Reflects the status of an operation that was performed on a domain index. ■ NULL: the specified index is not a domain index ■ VALID: the operation performed without errors ■ FAILED: the operation failed with an error
FUNCIDX_STATUS	VARCHAR		Indicates the status of a function-based index. ■ NULL: this is not a function-based index ■ ENABLED: the function-based index is enabled ■ DISABLED: the function-based index is disabled

ALL_INDEXTYPES

ALL_INDEXTYPES describes all the indextypes accessible to the current user.

Related views:

- DBA_INDEXTYPES describes all indextypes in the database.
- USER_INDEXTYPES describes the indextypes owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype
IMPLEMENTATION_SCHEMA	VARCHAR2(30)	NOT NULL	Name of the schema for indextype implementation (that is, containing the indextype operators)
IMPLEMENTATION_NAME	VARCHAR2(30)	NOT NULL	Name of indextype implementation type

Column	Datatype	NULL	Description
IMPLEMENTATION_VERSION	NUMBER	NOT NULL	Version of indextype implementation
NUMBER_OF_OPERATORS	NUMBER		Number of operators associated with the indextype

ALL_INDEXTYPE_OPERATORS

ALL_INDEXTYPE_OPERATORS lists all operators supported by indextypes accessible to the current user.

Related views:

- DBA_INDEXTYPE_OPERATORS lists all operators supported by indextypes in the database.
- USER_INDEXTYPE_OPERATORS lists the operators supported by indextypes owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator for which the indextype is defined
BINDING#	NUMBER	NOT NULL	Binding# associated with the operator

ALL_INTERNAL_TRIGGERS

ALL_INTERNAL_TRIGGERS describes internal triggers on tables accessible to the current user. Internal triggers are internal pieces of code executed when a particular flag is set for a table.

Related views:

- DBA_INTERNAL_TRIGGERS describes internal triggers on all tables in the database. This view does not display the OWNER_NAME column.
- USER_INTERNAL_TRIGGERS describes all internal triggers on tables owned by the current user. This view does not display the OWNER_NAME column.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Name of the table on which the trigger is defined

Column	Datatype	NULL	Description
OWNER_NAME	VARCHAR2(30)		Owner of the table
INTERNAL_TRIGGER_TYPE	VARCHAR2(19)		Indicates the type of internal trigger on the table

ALL_JOBS

ALL_JOBS describes all jobs in the database that are accessible to the current user.

See Also: *Oracle8i Administrator's Guide.*

Related views:

- DBA_JOBS describes all jobs in the database.
- USER_JOBS describes all jobs owned by the current user.

Column	Datatype	NULL	Description
JOB	NUMBER	NOT NULL	Identifier of job. Neither import/export nor repeated executions change this value.
LOG_USER	VARCHAR2(30)	NOT NULL	Login user when the job was submitted
PRIV_USER	VARCHAR2(30)	NOT NULL	User whose default privileges apply to this job
SCHEMA_USER	VARCHAR2(30)	NOT NULL	Default schema used to parse the job For example, if the SCHEMA_USER is SCOTT and you submit the procedure HIRE_EMP as a job, Oracle looks for SCOTT.HIRE_EMP
LAST_DATE	DATE		Date on which this job last successfully executed
LAST_SEC	VARCHAR2(8)		Same as LAST_DATE. This is when the last successful execution started.
THIS_DATE	DATE		Date that this job started executing (usually null if not executing)
THIS_SEC	VARCHAR2(8)		Same as THIS_DATE. This is when the last successful execution started.
NEXT_DATE	DATE	NOT NULL	Date that this job will next be executed
NEXT_SEC	VARCHAR2(8))		Same as MEXT_DATE. This is when the last successful execution started.
TOTAL_TIME	NUMBER		Total wall clock time spent by the system on this job, in seconds
BROKEN	VARCHAR2(1)		Y: no attempt is made to run this job N: an attempt is made to run this job

Column	Datatype	NULL	Description
INTERVAL	VARCHAR2(200)	NOT NULL	A date function, evaluated at the start of execution, becomes next NEXT_DATE
FAILURES	NUMBER		Number of times this job has started and failed since its last success
WHAT	VARCHAR2(4000)		Body of the anonymous PL/SQL block that this job executes
NLS_ENV	VARCHAR2(4000)		Session parameters describing the NLS environment of the job
MISC_ENV	RAW(32)		Other session parameters that apply to this job
INSTANCE	NUMBER		ID of the instance that can execute or is executing the job. The default is 0.

ALL_LIBRARIES

ALL_LIBRARIES describes all the libraries that are accessible to the current user.

Related views:

- DBA_LIBRARIES describes all libraries in the database.
- USER_LIBRARIES describes libraries owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the library
LIBRARY_NAME	VARCHAR2(30)	NOT NULL	Library name
FILE_SPEC	VARCHAR2(2000)		Operating system file specification associated with the library
DYNAMIC	VARCHAR2(1)		Whether the library is dynamically loadable (YES | NO)
STATUS	VARCHAR2(7)		Status of the library (VALID, INVALID, or N/A)

ALL_LOBS

ALL_LOBS displays the large objects (LOBs) contained in tables accessible to the user. LOBs include binary large objects (BLOBs) and character large objects (CLOBs). Binary files (BILEs) are stored outside the database, so they are not displayed by this view or the related views.

Related views:

- DBA_LOBS describes all LOBs in the database.
- USER_LOBS describes the LOBs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object containing the LOB
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object containing the LOB
COLUMN_NAME	VARCHAR2(4000)		Name of the LOB column or attribute
SEGMENT_NAME	VARCHAR2(30)	NOT NULL	Name of the LOB segment
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the LOB index
CHUNK	NUMBER		Size in bytes of the LOB chunk as a unit of allocation or manipulation
PCTVERSION	NUMBER	NOT NULL	Maximum percentage of the LOB space used for versioning
CACHE	VARCHAR2(3)		Whether and how the cluster is to be cached in the buffer cache (CACHE, NOCACHE, CACHEREADS)
LOGGING	VARCHAR2(3)		Whether changes to the LOB are logged
IN_ROW	VARCHAR2(3)		Whether some of the LOBs are stored inline with the base row

ALL_LOB_PARTITIONS

ALL_LOB_PARTITIONS displays LOB partitions contained in tables accessible to the user.

Related views:

- DBA_LOB_PARTITIONS describes all LOB partitions in the database.
- USER_LOB_PARTITIONS describes the LOB partitions owned by the current user. This view does not display the TABLE_OWNER column.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
COLUMN_NAME	VARCHAR2(30)		Name of the LOB column
LOB_NAME	VARCHAR2(30)		Name of the partitioned LOB item

Column	Datatype	NULL	Description
PARTITION_NAME	VARCHAR2(30)		Name of the table partition
LOB_PARTITION_NAME	VARCHAR2(30)		Name of the LOB data partition
LOB_INDPART_NAME	VARCHAR2(30)		Name of the corresponding LOB index partition
PARTITION_POSITION	NUMBER		Position of the LOB data partition within the LOB item
COMPOSITE	VARCHAR2(3)		Whether the partition is composite (YES | NO)
CHUNK	NUMBER		Value of the CHUNK attribute of the LOB data partition
PCTVERSION	NUMBER		Value of the PCTVERSION attribute of the LOB data partition
CACHE	VARCHAR2(3)		Whether and how the cluster is to be cached in the buffer cache (CACHE, NOCACHE, CACHEREADS)
IN_ROW	VARCHAR2(3)		Whether the STORAGE IN ROW attribute is enabled for the LOB data partition
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	VARCHAR2(40)		Size in bytes of the initial extent of the LOB data partition
NEXT_EXTENT	VARCHAR2(40)		Size in bytes of secondary extents of the LOB data partition
MIN_EXTENTS	VARCHAR2(40)		Minimum number of extents allowed in the segment of the LOB data partition
MAX_EXTENTS	VARCHAR2(40)		Maximum number of extents allowed in the segment of the LOB data partition
PCT_INCREASE	VARCHAR2(40)		Percentage increase in extent size for the LOB data partition
FREELISTS	VARCHAR2(40)		Number of process freelists allocated in the segment of the LOB data partition
FREELIST_GROUPS	VARCHAR2(40)		Number of freelist groups allocated in the segment of the LOB data partition
LOGGING	VARCHAR2(7)		Logging attribute of the LOB data partition
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the LOB partition blocks

ALL_LOB_SUBPARTITIONS

ALL_LOB_SUBPARTITIONS displays partition-level attributes of LOB data subpartitions accessible to the current user.

Related views:

- DBA_LOB_SUBPARTITIONS describes all LOB subpartitions in the database.

- USER_LOB_SUBPARTITIONS describes the LOB subpartitions owned by the current user. This view does not display the TABLE_OWNER column.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
COLUMN_NAME	VARCHAR2(30)		Name of the LOB column
LOB_NAME	VARCHAR2(30)		Name of the partitioned LOB item
LOB_PARTITION_NAME	VARCHAR2(30)		Name of LOB data partition to which this LOB data subpartition belongs
SUBPARTITION_NAME	VARCHAR2(30)		Name of the table subpartition to which this LOB subpartition corresponds
LOB_SUBPARTITION_NAME	VARCHAR2(30)		Name of the LOB subpartition
LOB_INDSUBPART_NAME	VARCHAR2(30)		Name of corresponding LOB index subpartition
SUBPARTITION_POSITION	NUMBER		Position of the LOB data partition within the LOB item
CHUNK	NUMBER		Value of the CHUNK attribute of the LOB data partition
PCTVERSION	NUMBER		Value of the PCTVERSION attribute of the LOB data partition
CACHE	VARCHAR2(3)		Whether and how the cluster is to be cached in the buffer cache (CACHE, NOCACHE, CACHEREADS)
IN_ROW	VARCHAR2(3)		Whether the STORAGE IN ROW attribute of the LOB data partition is enabled
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	VARCHAR2(40)		Size in bytes of the initial extent for the LOB data partition
NEXT_EXTENT	VARCHAR2(40)		Size in bytes of secondary extents for the LOB data partition
MIN_EXTENTS	VARCHAR2(40)		Minimum number of extents allowed in the segment of the LOB data partition
MAX_EXTENTS	VARCHAR2(40)		Maximum number of extents allowed in the segment of the LOB data partition
PCT_INCREASE	VARCHAR2(40)		Percentage increase in extent size for the LOB data partition
FREELISTS	VARCHAR2(40)		Number of process freelists allocated in the segment of the LOB data partition
FREELIST_GROUPS	VARCHAR2(40)		Number of freelist groups allocated in the segment of the LOB data partition
LOGGING	VARCHAR2(7)		The logging attribute of the LOB data partition

Column	Datatype	NULL	Description
BUFFER_POOL	VARCHAR2(7)		Default buffer pool to be used for the LOB data partition blocks

ALL_METHOD_PARAMS

ALL_METHOD_PARAMS describes method parameters of types accessible to the user.

Related views:

- DBA_METHOD_PARAMS describes all method parameters of all types in the database.
- USER_METHOD_PARAMS describes the method parameters of types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
METHOD_NAME	VARCHAR2(30)	NOT NULL	Name of the method
METHOD_NO	NUMBER	NOT NULL	For an overloaded method, a number distinguishing this method from others of the same. Do not confuse this number with the object ID.
PARAM_NAME	VARCHAR2(30)	NOT NULL	Name of the parameter
PARAM_NO	NUMBER	NOT NULL	Parameter number (position)
PARAM_MODE	VARCHAR2(6)		Mode of the parameter (IN, OUT, IN OUT)
PARAM_TYPE_MOD	VARCHAR2(7)		Whether this parameter is a REF to another object
PARAM_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the parameter
PARAM_TYPE_NAME	VARCHAR2(30)		Name of the type of the parameter
CHARACTER_SET_NAME	VARCHAR2(44)		Whether the character set or the method is fixed-length character set (CHAR_CS) or fixed-length national character set (NCHAR_CS), or a particular character set specified by the user

ALL_METHOD_RESULTS

ALL_METHOD_RESULTS describes return types of methods accessible to the user.

Related views:

- DBA_METHOD_RESULTS describes the return types of all methods in the database.
- USER_METHOD_RESULTS describes the return types of methods owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the method type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the method type
METHOD_NAME	VARCHAR2(30)	NOT NULL	Name of the method
METHOD_NO	NUMBER	NOT NULL	For an overloaded method, a number distinguishing this method from others of the same. Do not confuse this number with the object ID.
RESULT_TYPE_MOD	VARCHAR2(7)	NOT NULL	Whether this parameter is a REF to another object
RESULT_TYPE_OWNER	VARCHAR2(30)		Owner of the return type
RESULT_TYPE_NAME	VARCHAR2(30)		Name of the return type
CHARACTER_SET_NAME	VARCHAR2(44		Whether the character set or the method is fixed-length character set (CHAR_CS) or fixed-length national character set (NCHAR_CS), or a particular character set specified by the user

ALL_MVIEW_AGGREGATES

ALL_MVIEW_AGGREGATES describes the grouping functions (aggregate operations) that appear in the SELECT list of materialized aggregate views accessible to the current user.

Related views:

- DBA_MVIEW_AGGREGATES describes all such grouping functions defined for all materialized views in the database.
- USER_MVIEW_AGGREGATES describes such grouping functions defined for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that include references to a nonstatic value such as SYSDATE or USER. These views also exclude materialized views that were created as "snapshots" prior to Oracle8*i* and that were never altered to enable query rewrite.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
POSITION_IN_SELECT	NUMBER	NOT NULL	Ordinal position of this aggregation within the SELECT list. For the position of nonaggregate elements of the select list, see "ALL_MVIEW_KEYS" on page 2-42.
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	Name of this column in the container table
AGG_FUNCTION	VARCHAR2(8)		Aggregation function
DISTINCTFLAG	VARCHAR2(1)		Whether this aggregation is distinct (Y | N)
MEASURE	LONG		SQL text of the measure, excluding the aggregation function. Equal to * for COUNT(*).

ALL_MVIEW_ANALYSIS

ALL_MVIEW_ANALYSIS describes the materialized views accessible to the current user. It provides additional information for analysis by applications. Minimal information is displayed for materialized views that do not support query rewrite (such as materialized views with remote master tables or nondeterministic functions).

Related views:

- DBA_MVIEW_ANALYSIS describes all such materialized views in the database.
- USER_MVIEW_ANALYSIS describes all such materialized views owned by the current user.

> **Note:** All of the information in these views is also displayed in ALL_MVIEWS and its related views. Oracle Corporation recommends that you refer to ALL_MVIEWS for this information instead of these views.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
MVIEW_TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the container table (see next column)

Column	Datatype	Null	Description
CONTAINER_NAME	VARCHAR2(30)		Name of the internal container in which the materialized view data is held. Normally this is the same as MVIEW_NAME. For materialized views created in releases before 8.1 (or under pre-8.1 compatibility mode), Oracle attaches the 6-byte prefix SNAP$_. If MVIEW_NAME has more than 19 bytes, Oracle truncates the name to 19 bytes and adds a 4-byte sequence number as a suffix to produce a nonambiguous CONTAINER_NAME.
LAST_REFRESH_SCN	NUMBER		System change number (SCN) of the last refresh operation
LAST_REFRESH_DATE	DATE		SYSDATE of the last refresh
REFRESH_METHOD	VARCHAR2(8)		Default refresh method: FORCE, FAST, COMPLETE, or NONE
SUMMARY	VARCHAR2(1)		Whether this materialized view includes a GROUP BY clause or aggregation (Y | N)
FULLREFRESHTIM	NUMBER		Approximate refresh time, in seconds, for full refresh (defined only when SUMMARY = Y)
INCREFRESHTIM	NUMBER		Approximate refresh time, in seconds, for fast refresh (defined only when SUMMARY = Y)
CONTAINS_VIEWS	VARCHAR2(1)		Whether this materialized view contains a view in its definition (Y | N)
UNUSABLE	VARCHAR2(1)		Whether this materialized view is UNUSABLE (inconsistent data) (Y | N). A materialized view can be UNUSABLE if a system failure occurs during a full refresh.
RESTRICTED_SYNTAX	VARCHAR2(1)		Whether this materialized view had a restriction in its defining query that limits the use of query rewrite (Y | N). More complete information is provided by the REWRITE_CAPABILITY column of the ALL_, DBA_, and USER_MVIEWS views.
INC_REFRESHABLE	VARCHAR2(1)		Whether this materialized view can be fast refreshed (Y | N)
KNOWN_STALE	VARCHAR2(1)		Whether the data contained in the materialized view is known to be inconsistent with the master table data because that has been updated since the last successful refresh (Y | N)
INVALID	VARCHAR2(1)		Whether this materialized view is in an invalid state (inconsistent metadata) (Y | N)
REWRITE_ENABLED	VARCHAR2(1)		Whether this materialized view is currently enabled for query rewrite (Y | N)
QUERY_LEN	NUMBER	NOT NULL	The length (in bytes) of the query field
QUERY	LONG	NOT NULL	SELECT expression of the materialized view definition

Column	Datatype	Null	Description
REVISION	NUMBER		Reserved for internal use

ALL_MVIEW_DETAIL_RELATIONS

ALL_MVIEW_DETAIL_RELATIONS describes the named detail relations that are either specified in the FROM list of the subquery that defines a materialized view accessible to the current user, or that are indirectly referenced through views in that FROM list. Inline views in the materialized view definition are not represented in this view or the related views.

Related views:

- DBA_MVIEW_DETAIL_RELATIONS describes all such detail relations defined for all materialized views in the database.
- USER_MVIEW_DETAIL_RELATIONS describes such detail relations defined for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that includes references to a nonstatic value such as SYSDATE or USER. These views also exclude materialized views that were created as "snapshots" prior to Oracle8*i* and that were never altered to enable query rewrite.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Detail object owner
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Detail object name (that is, the name of a table or view)
DETAILOBJ_TYPE	VARCHAR2(9)		TABLE, VIEW, SNAPSHOT, CONTAINER, or UNDEFINED
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation

ALL_MVIEW_JOINS

ALL_MVIEW_JOINS describes joins between two columns in the WHERE clause of the subquery that defines a materialized view accessible to the current user.

Related views:

- DBA_MVIEW_JOINS describes all such joins for all materialized views in the database.
- USER_MVIEW_JOINS describes such joins for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that includes references to a nonstatic value such as SYSDATE or USER. These views also exclude materialized views that were created as "snapshots" prior to Oracle8*i* and that were never altered to enable query rewrite.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name
The next 8 rows relate only to materialized join views and materialized aggregate views. They describe the two detail objects of a materialized view join.			
DETAILOBJ1_OWNER	VARCHAR2(30)	NOT NULL	The owner of the first object in the join
DETAILOBJ1_RELATION	VARCHAR2(30)	NOT NULL	The name of the first object in the join
DETAILOBJ1_COLUMN	VARCHAR2(30)	NOT NULL	The join column of the first object in the join
OPERATOR	CHAR(1)		The join operator
OPERATOR_TYPE	VARCHAR2(1)		Whether the join is an inner or outer join
DETAILOBJ2_OWNER	VARCHAR2(30)	NOT NULL	The owner of the second object in the join
DETAILOBJ2_RELATION	VARCHAR2(30)	NOT NULL	The name of the second object in the join
DETAILOBJ2_COLUMN	VARCHAR2(30)	NOT NULL	The join column of the second object in the join

ALL_MVIEW_KEYS

ALL_MVIEW_KEYS describes the columns or expressions in the SELECT list upon which materialized views accessible to the current user are based.

Related views:

- DBA_MVIEW_KEYS describes such columns and expressions for all materialized views in the database.
- USER_MVIEW_KEYS describes such columns and expressions for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that includes references to a nonstatic value such as SYSDATE or USER. These views also exclude materialized views that were created as "snapshots" prior to Oracle8*i* and that were never altered to enable query rewrite.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name
POSITION_IN_SELECT	NUMBER	NOT NULL	Ordinal position of this key within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	Name of the column in the container table
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Detail object owner
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Detail object name (for example, the name of a table or view)
DETAILOBJ_TYPE	VARCHAR2(5)		Detail object type: VIEW | TABLE
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation
DETAILOBJ_COLUMN	VARCHAR2(30)	NOT NULL	Name of the detail relation column

ALL_MVIEWS

ALL_MVIEWS describes all materialized views accessible to the current user.

Related views:

- DBA_MVIEWS describes all materialized views in the database.
- USER_MVIEWS describes all materialized views owned by the current user.

See Also:

- *Oracle8i Replication* for more information on materialized views to support replication.
- *Oracle8i Data Warehousing Guide* for more information on materialized views to support data warehousing.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Schema in which the materialized view was created.
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view.

Column	Datatype	NULL	Description
CONTAINER_NAME	VARCHAR2(30)	NOT NULL	Name of the container in which the materialized view's data is held. Normally this is the same as MVIEW_NAME. For materialized views created in releases before 8.1 (or under pre-8.1 compatibility mode), Oracle attaches the 6-byte prefix SNAP$_. If MVIEW_NAME has more than 19 bytes, Oracle truncates the name to 19 bytes and adds a 4-byte sequence number as a suffix to produce a nonambiguous CONTAINER_NAME.
QUERY	LONG		The query that defines the materialized view
QUERY_LEN	INTEGER		Length in bytes of the defining query
UPDATABLE	VARCHAR2(1)		Whether the materialized view is updatable (Y | N)
UPDATE_LOG	VARCHAR2(30)		For updatable materialized views, the filename of the update log
MASTER_ROLLBACK_SEG	VARCHAR2(30)		Rollback segment for the master site
MASTER_LINK	VARCHAR2(128)		Database link for the master site
REWRITE_ENABLED	VARCHAR2(1)		Whether rewrite is enabled (Y | N)
REWRITE_CAPABILITY	VARCHAR2(19)		Whether the materialized view is eligible for rewrite, and if so, what rules must be followed: ■ NONE: The materialized view cannot be used for rewrite, because rewrite is disallowed or prevented. ■ TEXTMATCH: The defining query of the materialized view contained restrictions on the use of query rewrite. ■ GENERAL: The defining query of the materialized view contained no restrictions on the use of query rewrite, so Oracle can apply any rewrite rule that is supported.
REFRESH_MODE	VARCHAR2(6)		Refresh mode of the materialized view: ■ DEMAND: Oracle refreshes this materialized view whenever an appropriate refresh procedure is called. ■ COMMIT: Oracle refreshes this materialized view when a transaction on one of the materialized view's master tables commits. ■ NEVER: Oracle never refreshes this materialized view.

Column	Datatype	NULL	Description
REFRESH_METHOD	VARCHAR2(8)		Default method used to refresh the materialized view. (Can be overridden through the API.) ■ COMPLETE (C): The materialized view is completely refreshed from the master tables. ■ FORCE (?): Oracle performs a fast refresh if possible, otherwise a complete refresh. ■ FAST (F): Oracle performs an incremental refresh applying changes that correspond to changes in the master tables since the last refresh. ■ NEVER (N): The user specified that Oracle should not refresh this materialized view.
BUILD_MODE	VARCHAR2(9)		How the materialized view was populated during creation: ■ IMMEDIATE: Populated from the master tables during creation. ■ DEFERRED: Not populated during creation. Must be explicitly populated later by the user. ■ PREBUILT: Populated with an existing table during creation. The relationship of the contents of this prebuilt table to the materialized view's master tables is unknown to Oracle.
FAST_REFRESHABLE	VARCHAR2(18)		Whether or not the materialized view is eligible for incremental (fast) refresh. Oracle calculates this value statically, based on the materialized view definition query. ■ NO: The materialized view is not fast refreshable, and hence is complex. ■ DIRLOAD: Fast refresh is supported only for direct loads. ■ DML: Fast refresh is supported only for DML operations. ■ DIRLOAD_DML: Fast refresh is supported for both direct loads and DML operations. ■ DIRLOAD_LIMITEDDML: Fast refresh is supported for direct loads and a subset of DML operations.
LAST_REFRESH_TYPE	VARCHAR2(8)		The method used for the most recent refresh: ■ COMPLETE: The most recent refresh was complete. ■ FAST: The most recent refresh was fast (incremental). ■ NA: the materialized view has not yet been refreshed (for example, if it was created DEFERRED)..
LAST_REFRESH_DATE	DATE		Date on which the materialized view was most recently refreshed. Blank if not yet populated.

Column	Datatype	NULL	Description
STALENESS	VARCHAR2(9)		The relationship between the contents of the materialized view and the contents of the materialized view master tables. ■ FRESH: The materialized view is a read-consistent view of the current state of its master tables. ■ STALE: The materialized view is out of date because one or more of its master tables has changed. If the materialized view was FRESH before it became STALE, it is a read-consistent view of a former state of its master tables. ■ UNUSABLE: The materialized view is not a read-consistent view of its master tables from any point in time. ■ UNKNOWN: Oracle does not know whether the materialized view is in a read-consistent view of its master tables from any point in time. (This is the case for materialized views created on prebuilt tables.) ■ UNDEFINED: The materialized view has remote master tables. The concept of staleness is not defined for such materialized views.
AFTER_FAST_REFRESH	VARCHAR2(9)		Specifies the staleness value that will occur if a fast refresh is applied to this materialized view. Its values are the same as for the STALENESS column, plus the value NA, which is used when fast refresh is not applicable to this materialized view.
COMPILE_STATE	VARCHAR2(13)		Validity of the materialized view with respect to the object(s) upon which it depends: ■ VALID: The materialized view has been validated without error, and no object upon which it depends has changed since the last validation. ■ NEEDS_COMPILE: Some object upon which the materialized view depends has changed (other than normal DML changes). An ALTER MATERIALIZED VIEW ... COMPILE statement is required to validate this materialized view. ■ ERROR: The materialized view has been validated with one or more errors

ALL_NESTED_TABLES

ALL_NESTED_TABLES describes the nested tables in tables accessible to the user.

Related views:

- DBA_NESTED_TABLES describes all nested tables in the database.
- USER_NESTED_TABLES describes nested tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	Null	Description
OWNER	VARCHAR2(30)		Owner of the nested table
TABLE_NAME	VARCHAR2(30)		Name of the nested table
TABLE_TYPE_OWNER	VARCHAR2(30)		Owner of the type of which the nested table was created
TABLE_TYPE_NAME	VARCHAR2(30)		Name of the type of the nested table
PARENT_TABLE_NAME	VARCHAR2(30)		Name of the parent table containing the nested table
PARENT_TABLE_COLUMN	VARCHAR2(4000)		Column name of the parent table that corresponds to the nested table
STORAGE_SPEC	VARCHAR2(30)		Whether storage for the nested table is USER-SPECIFIED or DEFAULT
RETURN_TYPE	VARCHAR2(20)		Return type of the varray column (LOCATOR | VALUE)

ALL_OBJECT_TABLES

ALL_OBJECT_TABLES describes all object tables accessible to the user.

Related views:

- DBA_OBJECT_TABLES describes all object tables in the database.
- USER_OBJECT_TABLES describes all object tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the table
CLUSTER_NAME	VARCHAR2(30)		Name of the cluster, if any, to which the table belongs
IOT_NAME	VARCHAR2(30)		Name of the index-organized table, if any, to which the overflow entry belongs
PCT_FREE	NUMBER		Minimum percentage of free space in a block
PCT_USED	NUMBER		Minimum percentage of used space in a block
INI_TRANS	NUMBER		Initial number of transactions
MAX_TRANS	NUMBER		Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size in bytes of the initial extent
NEXT_EXTENT	NUMBER		Size in bytes of secondary extents

Column	Datatype	NULL	Description
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute
BACKED_UP	VARCHAR2(1)		Whether the table has been backed up since last modification
NUM_ROWS	NUMBER		Number of rows in the table
BLOCKS	NUMBER		Number of used blocks in the table
EMPTY_BLOCKS	NUMBER		Number of empty (never used) blocks in the table
AVG_SPACE	NUMBER		Average available free space in the table
CHAIN_CNT	NUMBER		Number of chained rows in the table
AVG_ROW_LEN	NUMBER		Average row length, including row overhead
AVG_SPACE_FREELIST_BLOCKS	NUMBER		Average free space of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		Number of blocks on the freelist
DEGREE	VARCHAR2(10)		Number of parallel execution processes per instance for scanning the table
INSTANCES	VARCHAR2(10)		Number of instances across which the table is to be scanned
CACHE	VARCHAR2(5)		Whether the cluster is to be cached in the buffer cache (CACHE | NOCACHE)
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or disabled
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this table
LAST_ANALYZED	DATE		Date on which this table was most recently analyzed
PARTITIONED	VARCHAR2(3)		Whether the table is partitioned (YES | NO)
IOT_TYPE	VARCHAR2(12)		If index organized table, then IOT_TYPE is IOT or IOT_OVERFLOW else NULL
OBJECT_ID_TYPE	VARCHAR2(16)		Whether the object ID (OID) is USER-DEFINED or SYSTEM-GENERATED

Column	Datatype	NULL	Description
TABLE_TYPE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the type of the table if the table is a typed table
TABLE_TYPE	VARCHAR2(30)	NOT NULL	Type of the table if the table is a typed table
TEMPORARY	VARCHAR2(1)		Whether this is a temporary table
SECONDARY	VARCHAR2(1)		Whether the object table is a secondary object created by the ODCIIndexCreate method of the Oracle8*i* Data Cartridge (Y | N)
NESTED	VARCHAR2(3)		Whether the table a nested table
BUFFER_POOL	VARCHAR2(7)		Default buffer pool to be used for table blocks
ROW_MOVEMENT	VARCHAR2(8)		Whether partitioned row movement is ENABLED or DiSABLED
GLOBAL_STATS	VARCHAR2(3)		For partitioned object tables, whether statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO).
USER_STATS	VARCHAR2(3)		Whether statistics were entered directly by the user
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: ■ SYS$SESSION: the rows are preserved for the duration of the session ■ SYS$TRANSACTION: the rows are deleted after COMMIT. Null for a permanent table.
SKIP_CORRUPT	VARCHAR2(8)		Whether Oracle ignores blocks marked corrupt during table and index scans (ENABLED) or raises an error (DISABLED). To enable this feature, run the DBMS_REPAIR.SKIP_CORRUPT_BLOCKS procedure.
MONITORING	VARCHAR2(3)		Whether the table has the MONITORING attribute set

ALL_OBJECTS

ALL_OBJECTS describes all objects accessible to the user.

Related views:

- DBA_OBJECTS describes all objects in the database.
- USER_OBJECTS describes all objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object
SUBOBJECT_NAME	VARCHAR2(30)		Name of the subobject (for example, partition)
OBJECT_ID	NUMBER	NOT NULL	Dictionary object number of the object
DATA_OBJECT_ID	NUMBER		Dictionary object number of the segment that contains the object
			Note: OBJECT_ID and DATA_OBJECT_ID display data dictionary metadata. Do not confuse these numbers with the unique 16-byte object identifier ("object ID") that Oracle assigns to row objects in object tables in the system.
OBJECT_TYPE	VARCHAR2(15)		Type of the object (such as TABLE, INDEX)
CREATED	DATE	NOT NULL	Timestamp for the creation of the object
LAST_DDL_TIME	DATE	NOT NULL	Timestamp for the last modification of the object resulting from a DDL command (including grants and revokes)
TIMESTAMP	VARCHAR2(20)		Timestamp for the specification of the object (character data)
STATUS	VARCHAR2(7)		Status of the object: VALID, INVALID, or N/A
TEMPORARY	VARCHAR2(1)		Whether the object is temporary (the current session can see only data that it placed in this object itself)
GENERATED	VARCHAR2(1)		Was the name of this object system generated?
SECONDARY	VARCHAR2(1)		Whether this is a secondary object created by the ODCIIndexCreate method of the Oracle8*i* Data Cartridge (Y | N)

ALL_OPANCILLARY

ALL_OPANCILLARY describes operators whose bindings are ancillary to other (primary) operators.

Related views:

- DBA_OPANCILLARY provides such information about all operators in the database.
- USER_OPANCILLARY provides such information about operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the ancillary operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the ancillary operator
BINDING#	NUMBER	NOT NULL	Binding number the of ancillary operator
PRIMOP_OWNER	VARCHAR2(30)	NOT NULL	Owner of the primary operator
PRIMOP_NAME	VARCHAR2(30)	NOT NULL	Name of the primary operator
PRIMOP_BIND#	NUMBER	NOT NULL	Binding number of the primary operator

ALL_OPARGUMENTS

ALL_OPARGUMENTS describes arguments for each operator binding accessible to the current user.

Related views:

- DBA_OPARGUMENTS describes arguments of all operator bindings in the database.
- USER_OPARGUMENTS describes arguments of all operator bindings owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator argument
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator argument
BINDING#	NUMBER	NOT NULL	Binding number of the operator argument
POSITION	NUMBER	NOT NULL	Position of the operator argument (1, 2, 3, ...)
ARGUMENT_TYPE	VARCHAR2(61)		Datatype of the operator argument

ALL_OPBINDINGS

ALL_OPBINDINGS describes operator bindings accessible to the current user.

Related views:

- DBA_OPBINDINGS describes bindings of all operators in the database.
- USER_OPBINDINGS describes bindings of all operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
BINDING#	NUMBER	NOT NULL	Binding number of the operator
FUNCTION_NAME	VARCHAR2(92)		Name of the binding function or method as specified by the user
RETURN_SCHEMA	VARCHAR2(30)		If the return type of the binding is an object type, the schema of the type
RETURN_TYPE	VARCHAR2(30)		Name of the return type
IMPLEMENTATION_TYPE_SCHEMA	VARCHAR2(30)		If the operator was created WITH INDEX CONTEXT, SCAN CONTEXT, this column displays the schema of the implementation type used by the functional implementation of the operator as a scan context. (Blank if the operator was created without this syntax.) **See Also:** the CREATE OPERTOR statement in *Oracle8i SQL Reference.*
IMPLEMENTATION_TYPE	VARCHAR2(30)		If the operator was created WITH INDEX CONTEXT, SCAN CONTEXT, this column displays the name of the implementation type used by the functional implementation of the operator as a scan context. (Blank if the operator was created without this syntax.) **See Also:** the CREATE OPERTOR statement in *Oracle8i SQL Reference.*

ALL_OPERATORS

ALL_OPERATORS describes operators accessible to the current user.

Related views:

- DBA_OPERATORS describes all operators in the database.
- USER_OPERATORS describes all operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
NUMBER_OF_BINDS	NUMBER	NOT NULL	Number of bindings associated with the operator

ALL_OUTLINE_HINTS

ALL_OUTLINE_HINTS describes the set of hints stored in outlines accessible to the user.

Related views:

- DBA_OUTLINE_HINTS describes such hints for all outlines in the database. This is the only one of the three views that displays the OWNER column.
- USER_OUTLINE_HINTS describes such hints for all outlines owned by the current user.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the outline
OWNER	VARCHAR2(30)		Name of the user who created the outline
NODE	NUMBER		ID of the query or subquery to which the hint applies. The top-level query is labeled 1. Subqueries are assigned sequentially numbered labels, starting with 2.
STAGE	NUMBER		Outline hints can be applied at three different stages during the compilation process. This column indicates the stage at which this hint was applied.
JOIN_POS	NUMBER		Position of the table in the join order. The value is 0 for all hints except access method hints, which identify a table to which the hint and the join position apply.
HINT	VARCHAR2(512)		Text of the hint

ALL_OUTLINES

ALL_OUTLINES describes all stored outlines accessible to the current user.

Related views:

- DBA_OUTLINES describes all stored outlines in the database. This is the only one of the three views that displays the OWNER column.
- USER_OUTLINES describes all stored outlines owned by the current user.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		User-specified or generated name of the stored outline. The name must be of a form that can be expressed in SQL

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Name of the user who created the outline
CATEGORY	VARCHAR2(30)		User-defined name of the category to which this outline belongs
USED	VARCHAR2(9)		Whether the outline has ever been used (USED, UNUSED, or UNDEFINED)
TIMESTAMP	DATE		Timestamp of outline creation
VERSION	VARCHAR2(64)		Oracle version that created the outline
SQL_TEXT	LONG		SQL text of the query, including any hints that were a part of the original statement. If bind variables are included, the variable names are stored as SQL text, not the values that are assigned to the variables. **Note:** This field may contain sensitive information about your database or application. Therefore, use discretion when granting SELECT or VIEW object privileges on these views.

ALL_PART_COL_STATISTICS

ALL_PART_COL_STATISTICS provides column statistics and histogram information for table partitions accessible to the current user.

Related views:

- DBA_PART_COL_STATISTICS provides such information for all table partitions in the database.
- USER_PART_COL_STATISTICS provides such information for all partitions of tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the partitioned table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the table partition
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
NUM_DISTINCT	NUMBER		Number of distinct values in the column
LOW_VALUE	RAW(32)		Low value in the column
HIGH_VALUE	RAW(32)		High value in the column

Column	Datatype	NULL	Description
DENSITY	NUMBER		Density of the column
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in histogram for the column
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
GLOBAL_STATS	VARCHAR2(3)		Indicates whether column statistics were collected for the partition as a whole (YES) or were estimated from statistics on underlying subpartitions (NO).
USER_STATS	VARCHAR2(3)		Whether the statistics were entered directly by the user
AVG_COL_LEN	NUMBER		Average length of the column, in bytes

ALL_PART_HISTOGRAMS

ALL_PART_HISTOGRAMS provides the histogram data (endpoints per histogram) for histograms on table partitions accessible to the current user.

Related views:

- DBA_PART_HISTOGRAMS provides such information for all table partitions in the database.
- USER_PART_HISTOGRAMS provides such information for all partitions of tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the table partition
COLUMN_NAME	VARCHAR2(30)		Name of the column
BUCKET_NUMBER	NUMBER		Bucket number of the histogram
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket
ENDPOINT_ACTUAL_VALUE	VARCHAR2(1000)		Actual (not normalized) string value of the endpoint for this bucket.

ALL_PART_INDEXES

ALL_PART_INDEXES provides object-level partitioning information for all partitioned indexes accessible to the current user.

Related views:

- DBA_PART_INDEXES provides such information for all partitioned indexes in the database.
- USER_PART_INDEXES provides such information for all partitioned indexes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of this partitioned index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of this partitioned index
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of this partitioned table
PARTITIONING_TYPE	VARCHAR2(7)		RANGE, HASH, SYSTEM, or UNKNOWN
SUBPARTITIONING_TYPE	VARCHAR2(7)		HASH, SYSTEM, or NONE
PARTITION_COUNT	NUMBER	NOT NULL	Number of partitions in this index
DEF_SUBPARTITION_COUNT	NUMBER	NOT NULL	For a composite-partitioned index, the **default** number of subpartitions, if specified
PARTITIONING_KEY_COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key
SUBPARTITIONING_KEY_COUNT	NUMBER	NOT NULL	For a composite-partitioned index, the number of columns in the subpartitioning key
LOCALITY	VARCHAR2(6)		Whether this partitioned index is LOCAL or GLOBAL
ALIGNMENT	VARCHAR2(12)		Whether this partitioned index is PREFIXED or NON-PREFIXED
DEF_TABLESPACE_NAME	VARCHAR2(30)		For a local index, the default tablespace to be used when adding or splitting a table partition
DEF_PCT_FREE	NUMBER	NOT NULL	For a local index, the default PCTFREE value to be used when adding a table partition
DEF_INI_TRANS	NUMBER	NOT NULL	For a local index, the default INITRANS to be used when adding a table partition
DEF_MAX_TRANS	NUMBER	NOT NULL	For a local index, the default MAXTRANS to be used when adding a table partition
DEF_INITIAL_EXTENT	VARCHAR2(40)	NOT NULL	For a local index, the default INITIAL value to be used when adding a table partition, or DEFAULT if no INITIAL value was specified.

Column	Datatype	NULL	Description
DEF_NEXT_EXTENT	VARCHAR2(40)	NOT NULL	For a local index, the default NEXT, or DEFAULT if no NEXT value was specified.
DEF_MIN_EXTENTS	VARCHAR2(40)	NOT NULL	For a local index, the default MINEXTENTS value to be used when adding a table partition, or DEFAULT if no MINEXTENTS value was specified.
DEF_MAX_EXTENTS	VARCHAR2(40)	NOT NULL	For a local index, the default MAXEXTENTS value to be used when adding a table partition, or DEFAULT if no MAXEXTENTS value was specified.
DEF_PCT_INCREASE	VARCHAR2(40)	NOT NULL	For a local index, the default PCTINCREASE value to be used when adding a table partition, or DEFAULT if no PCTINCREASE value was specified.
DEF_FREELISTS	NUMBER	NOT NULL	For a local index, the default FREELISTS value to be used when adding a table partition, or DEFAULT if no FREELISTS value was specified.
DEF_FREELIST_GROUPS	NUMBER	NOT NULL	For a local index, the default FREELIST GROUPS value to be used when adding a table partition, or DEFAULT if no FREELIST GROUPS value was specified.
DEF_LOGGING	VARCHAR2(7)		For a local index, the default LOGGING attribute to be used when adding a table partition, or DEFAULT if no LOGGING attribute was specified.
DEF_BUFFER_POOL	VARCHAR2(7)		For a local index, the default buffer pool to be used when adding a table partition

ALL_PART_KEY_COLUMNS

ALL_PART_KEY_COLUMNS describes the partitioning key columns for partitioned objects accessible to the current user.

Related views:

- DBA_PART_KEY_COLUMNS describes partitioning key columns for all partitions of all partitioned objects in the database.
- USER_PART_KEY_COLUMNS describes partitioning key columns for partitions of all objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the partitioned table or index
NAME	VARCHAR2(30)		Name of the partitioned table or index

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(11)		Object type (TABLE | NDEX)
COLUMN_NAME	VARCHAR2(30)		Name of the column
COLUMN_POSITION	NUMBER		Position of the column within the partitioning key

ALL_PART_LOBS

ALL_PART_LOBS provides table-level information about partitioned LOBs that are accessible to the current user, including default attributes for LOB data partitions.

Related views:

- DBA_PART_LOBS provides such information for all partitioned LOBs in the database.
- USER_PART_LOBS provides such information for all partitioned LOBs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the partitioned table containing LOB(s)
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned table containing LOB(s)
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the LOB column
LOB_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned LOB
LOB_INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned LOB index
DEF_CHUNK	NUMBER	NOT NULL	Default value of CHUNK for a LOB data partition to be used when adding a partition
DEF_PCTVERSION	NUMBER	NOT NULL	Default value of PCTVERSION for a LOB data partition to be used when adding a partition
DEF_CACHE	VARCHAR2(3)		Whether and how the cluster is to be cached by default in the buffer cache (CACHE, NOCACHE, CACHEREADS)
DEF_IN_ROW	VARCHAR2(3)		Whether LOB data < 4000 bytes is stored in the row (inline)—that is, whether ENABLE STORAGE IN ROW was specified when the LOB column was created or last altered
DEF_TABLESPACE_NAME	VARCHAR2(30)		Default tablespace for a LOB data partition to be used when adding a partition
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default value of INITIAL for a LOB data partition to be used when adding a partition

Column	Datatype	NULL	Description
DEF_NEXT_EXTENT	VARCHAR2(40)		Default value of NEXT for a LOB data partition to be used when adding a partition
DEF_MIN_EXTENTS	VARCHAR2(40)		Default value of MINEXTENT for a LOB data partition to be used when adding a partition
DEF_MAX_EXTENTS	VARCHAR2(40)		Default value of MAXEXTENTS for a LOB data partition to be used when adding a partition
DEF_PCT_INCREASE	VARCHAR2(40)		Default value of PCTINCREASE for a LOB data partition to be used when adding a partition
DEF_FREELISTS	VARCHAR2(40)		Default value of FREELISTS for a LOB data partition to be used when adding a partition
DEF_FREELIST_GROUPS	VARCHAR2(40)		Default value of FREELIST GROUPS for a LOB data partition to be used when adding a partition
DEF_LOGGING	VARCHAR2(7)		Default LOGGING attribute for a LOB data partition to be used when adding a partition
DEF_BUFFER_POOL	VARCHAR2(7)		Default buffer pool for a LOB data partition to be used when adding a partition

ALL_PART_TABLES

ALL_PART_TABLES provides object-level partitioning information for partitioned tables accessible to the current user.

Related views:

- DBA_PART_TABLES provides such information for all partitioned tables in the database.
- USER_PART_TABLES provides such information for all partitioned tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the partitioned table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned table
PARTITIONING_TYPE	VARCHAR2(7)		Type of partitioning (RANGE | HASH)
SUBPARTITIONING_TYPE	VARCHAR2(7)		Type of subpartitioning (HASH | NONE)
PARTITION_COUNT	NUMBER	NOT NULL	Number of partitions in this table
DEF_SUBPARTITION_COUNT	NUMBER	NOT NULL	For a composite-partitioned table, the default number of subpartitions, if specified

Column	Datatype	NULL	Description
PARTITIONING_KEY_COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key
SUBPARTITIONING_KEY_COUNT	NUMBER	NOT NULL	For a composite-partitioned table, the number of columns in the subpartitioning key
DEF_TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Default tablespace to be used when adding a partition
DEF_PCT_FREE	NUMBER	NOT NULL	Default value of PCTFREE to be used when adding a partition
DEF_PCT_USED	NUMBER	NOT NULL	Default value of PCTUSED to be used when adding a partition
DEF_INI_TRANS	NUMBER	NOT NULL	Default value of INITRANS to be used when adding a partition
DEF_MAX_TRANS	NUMBER	NOT NULL	Default value of MAXTRANS to be used when adding a partition
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default value of INITIAL to be used when adding a partition. DEFAULT if INITIAL was not specified.
DEF_NEXT_EXTENT	VARCHAR2(40)		Default value of NEXT to be used when adding a partition. DEFAULT if NEXT was not specified.
DEF_MIN_EXTENTS	VARCHAR2(40)		Default value of MINEXTENTS to be used when adding a partition. DEFAULT if MINEXTENTS was not specified.
DEF_MAX_EXTENTS	VARCHAR2(40)		Default value of MAXEXTENTS to be used when adding a partition. DEFAULT if MAXEXTENTS was not specified.
DEF_PCT_INCREASE	VARCHAR2(40)		Default value of PCTINCREASE to be used when adding a partition. DEFAULT if PCTINCREASE was not specified.
DEF_FREELISTS	NUMBER	NOT NULL	Default value of FREELISTS to be used when adding a partition
DEF_FREELIST_GROUPS	NUMBER	NOT NULL	Default value of FREELIST GROUPS to be used when adding a partition
DEF_LOGGING	VARCHAR2(7)		Default logging attribute to be used when adding a partition
DEF_BUFFER_POOL	VARCHAR2(7)		Default buffer pool to be used when adding a partition

ALL_PARTIAL_DROP_TABS

ALL_PARTIAL_DROP_TABS describes tables accessible to the current user that have partially completed DROP COLUMN operations. Such operations might have been interrupted by the user or by a system crash.

Related views:

- DBA_PARTIAL_DROP_TABS describes all tables in the database that have partially completed DROP COLUMN operations.
- USER_PARTIAL_DROP_TABS describes tables in the schema of the current user that have partially completed DROP COLUMN operations. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table

ALL_POLICIES

ALL_POLICIES describes all security policies on all tables and views accessible to the current user.

Related views:

- DBA_POLICIES describes all policies in the database.
- USER_POLICIES describes policies on all tables and views owned by the current user. This view does not display the OBJECT_OWNER column.

See Also:

- *Oracle8i Concepts* for information on security policies and fine-grained access control.
- The DBMS_RLS packages in *Oracle8i Supplied PL/SQL Packages Reference* for information on administering security policies.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table or view
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the table or view

Column	Datatype	NULL	Description
POLICY_NAME	VARCHAR2(30)	NOT NULL	Name of the policy
PF_OWNER	VARCHAR2(30)	NOT NULL	Owner of the policy function
PACKAGE	VARCHAR2(30)		Name of the package containing the policy function
FUNCTION	VARCHAR2(30)	NOT NULL	Name of the policy function
SEL	VARCHAR2(3)		If YES, policy is applied to query on the object
INS	VARCHAR2(3)		If YES, policy is applied to insert on the object
UPD	VARCHAR2(3)		If YES, policy is applied to update on the object
DEL	VARCHAR2(3)		If YES, policy is applied to delete on the object
CHK_OPTION	VARCHAR2(3)		Is check option enforced for this policy?
ENABLE	VARCHAR2(3)		Is this policy enabled?

ALL_QUEUE_TABLES

ALL_QUEUE_TABLES describes queues in all queue tables accessible to the current user.

Related views:

- DBA_QUEUE_TABLES describes queues in all queue tables in the database.
- USER_QUEUE_TABLES describes queues in queue tables created in the user's schema. This view does not display the OWNER column.

> **See Also:** *Oracle8i Application Developer's Guide - Advanced Queuing* for more information about these views and Advanced Queuing.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Schema of the queue table
QUEUE_TABLE	VARCHAR2(30)		Name of the queue table
TYPE	VARCHAR2(7)		Type of user data: ■ RAW: raw type ■ OBJECT: user-defined object type ■ VARIANT: variant type (internal use only)
OBJECT_TYPE	VARCHAR2(61)		Object type of the payload when TYPE is OBJECT
SORT_ORDER	VARCHAR2(22)		User specified sort order

Column	Datatype	NULL	Description
RECIPIENTS	VARCHAR2(8)		SINGLE or MULTIPLE recipients
MESSAGE_GROUPING	VARCHAR2(13)		NONE or TRANSACTIONAL
COMPATIBLE	VARCHAR2(5)		The lowest release level which this is compatible with (eg, 8.0.4)
PRIMARY_INSTANCE	NUMBER		Indicates the instance number of the instance which is the primary owner of the queue table. A value of 0 indicates that there is no primary owner
SECONDARY_INSTANCE	NUMBER		Indicates the instance number of the instance which is the secondary owner of the queue table. This instance becomes the owner of the queue table if the primary owner is not alive. A value of 0 indicates that there is no secondary owner
OWNER_INSTANCE	NUMBER		The instance number of the instance which currently owns the queue table
USER_COMMENT	VARCHAR2(50)		Comment supplied by the user

ALL_QUEUES

ALL_QUEUES describes all queues on which the current user has enqueue or dequeue privileges. If the user has any Advanced Queuing system privileges, like MANAGE ANY QUEUE, ENQUEUE ANY QUEUE or DEQUEUE ANY QUEUE, then this view describes all queues in the database.

Related views:

- DBA_QUEUES describes all queues in the database.
- USER_QUEUES describes the operational characteristics of every queue owned by the current user. This view does not display the OWNER column.

See Also: *Oracle8i Application Developer's Guide - Advanced Queuing* for more information about these views and Advanced Queuing.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue
NAME	VARCHAR2(30)	NOT NULL	Name of the queue
QUEUE_TABLE	VARCHAR2(30)	NOT NULL	Name of the table the queue data resides in
QID	NUMBER	NOT NULL	Object number of the queue

Column	Datatype	NULL	Description
QUEUE_TYPE	VARCHAR2(15)		Type of the queue
MAX_RETRIES	NUMBER		Maximum number of retries allowed when dequeuing from the queue
RETRY_DELAY	NUMBER		Time interval between retries
ENQUEUE_ENABLED	VARCHAR2(7)		Queue is enabled for enqueue
DEQUEUE_ENABLED	VARCHAR2(7)		Queue is enabled for dequeue
RETENTION	VARCHAR2(40)		Time interval processed messages retained in the queue
USER_COMMENT	VARCHAR2(50)		User specified comment

ALL_REFRESH

ALL_REFRESH describes all the refresh groups accessible to the current user.

Related views:

- DBA_REFRESH describes all refresh groups in the database.
- USER_REFRESH describes all refresh groups owned by the current user.

Column	Datatype	NULL	Description
ROWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the refresh group
RNAME	VARCHAR2(30)	NOT NULL	Name of the refresh group
REFGROUP	NUMBER		Internal identifier of refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		Y or N; if Y, then destroy the refresh group when its last item is subtracted
PUSH_DEFERRED_RPC	VARCHAR2(1)		Y or N; if Y then push changes from snapshot to master before refresh
REFRESH_AFTER _ERRORS	VARCHAR2(1)		If Y, proceed with refresh despite error when pushing deferred RPCs
ROLLBACK_SEG	VARCHAR2(30)		Name of the rollback segment to use while refreshing
JOB	NUMBER		Identifier of job used to refresh the group automatically
NEXT_DATE	DATE		Date that this job will next be refreshed automatically, if not broken
INTERVAL	VARCHAR2(200)		A date function used to compute the next NEXT_DATE
BROKEN	VARCHAR2(1)		Y or N; Y means the job is broken and will never be run
PURGE_OPTION	NUMBER(38)		The method for purging the transaction queue after each push. 1=quick purge option; 2=precise purge option

Column	Datatype	NULL	Description
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

ALL_REFRESH_CHILDREN

ALL_REFRESH_CHILDREN lists all the objects in refresh groups that are accessible to the current user.

Related views:

- DBA_REFRESH_CHILDREN describes the objects in all refresh groups in the database.
- USER_REFRESH_CHILDREN describes the objects in all refresh groups owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
TYPE	VARCHAR2(30)		Type of the object in the refresh group
ROWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the refresh group
RNAME	VARCHAR2(30)	NOT NULL	Name of the refresh group
REFGROUP	NUMBER		Internal identifier of refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		Y or N; if Y, then destroy the refresh group when its last item is subtracted
PUSH_DEFERRED_RPC	VARCHAR2(1)		Y or N; if Y then push changes from snapshot to master before refresh
REFRESH_AFTER _ERRORS	VARCHAR2(1)		If Y, proceed with refresh despite error when pushing deferred RPCs
ROLLBACK_SEG	VARCHAR2(30)		Name of the rollback segment to use while refreshing
JOB	NUMBER		Identifier of job used to refresh the group automatically
NEXT_DATE	DATE		Date that this job will next be refreshed automatically, if not broken
INTERVAL	VARCHAR2(200)		A date function used to compute the next NEXT_DATE
BROKEN	VARCHAR2(1)		Y or N; Y means the job is broken and will never be run

Column	Datatype	NULL	Description
PURGE_OPTION	NUMBER(38)		The method for purging the transaction queue after each push. 1=quick purge option; 2=precise purge option
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap.

ALL_REFRESH_DEPENDENCIES

ALL_REFRESH_DEPENDENCIES lists the names of the dependent detail or container tables of all the summaries or snapshots in the current schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name, unique within this schema
PARENT_OBJECT_TYPE	CHAR(7)		'SNAPSHOT' or 'SUMMARY'
OLDEST_REFRESH_SCN	NUMBER		The minimum SCN of any summary or snapshot that has TABLE_NAME as a detail table
OLDEST_REFRESH_DATE	DATE		SYSDATE when last refreshed

ALL_REFS

ALL_REFS describes the REF columns and REF attributes in object type columns accessible to the current user.

Related views:

- DBA_REFS describes all REF columns and REF attributes in the database.
- USER_REFS describes the REF columns and REF attributes in object type columns owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
COLUMN_NAME	VARCHAR2(4000)		Name of the REF column or attribute. If it is not a top-level attribute, the value of COLUMN_NAME should be a path name starting with the column name

Column	Datatype	NULL	Description
WITH_ROWID	VARCHAR2(3)		Whether the REF value is stored with ROWID (YES or NO)?
IS_SCOPED	VARCHAR2(3)		Whether the REF column is scoped (YES or NO)?
SCOPE_TABLE_OWNER	VARCHAR2(30)		Name of the owner of the scope table, if it exists and is accessible by the user
SCOPE_TABLE_NAME	VARCHAR2(30)		Name of the scope table, if it exists and is accessible by the user
OBJECT_ID_TYPE	VARCHAR2(16)		If user-defined OID, then USER-DEFINED, else if system generated OID, then SYSTEM GENERATED

ALL_REGISTERED_SNAPSHOTS

ALL_REGISTERED_SNAPSHOTS describes all registered snapshots accessible to the current user.

Related views:

- DBA_REGISTERED_SNAPSHOTS describes all registered snapshots in the database.
- USER_REGISTERED_SNAPSHOTS describes all registered snapshots owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot
NAME	VARCHAR2(30)	NOT NULL	Name of the snapshot
SNAPSHOT_SITE	VARCHAR2(128)	NOT NULL	Global name of the snapshot site.
CAN_USE_LOG	VARCHAR2(3)		YES if this snapshot can use a snapshot log, NO if this snapshot is too complex to use a log
UPDATABLE	VARCHAR2(3)		Specifies whether the snapshot is updatable. YES if it is, NO if it is not. If set to NO, the snapshot is read only
REFRESH_METHOD	VARCHAR2(11)		Whether the snapshot uses rowids or primary key for fast refresh
SNAPSHOT_ID	NUMBER(38)		Identifier for the snapshot used by the master for fast refresh
VERSION	VARCHAR2(17)		Version of snapshot
QUERY_TXT	LONG		Original query of which this snapshot is an instantiation

ALL_SEQUENCES

ALL_SEQUENCES describes all sequences accessible to the user.

Related views:

- DBA_SEQUENCES describes all sequences in the database.
- USER_SEQUENCES describes all sequences owned by the current user. This view does not display the SEQUENCE_OWNER column.

Column	Datatype	NULL	Description
SEQUENCE_OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the sequence
SEQUENCE_NAME	VARCHAR2(30)	NOT NULL	Sequence name
MIN_VALUE	NUMBER		Minimum value of the sequence
MAX_VALUE	NUMBER		Maximum value of the sequence
INCREMENT_BY	NUMBER	NOT NULL	Value by which sequence is incremented
CYCLE_FLAG	VARCHAR2(1)		Does sequence wrap around on reaching limit
ORDER_FLAG	VARCHAR2(1)		Are sequence numbers generated in order
CACHE_SIZE	NUMBER	NOT NULL	Number of sequence numbers to cache
LAST_NUMBER	NUMBER	NOT NULL	Last sequence number written to disk. If a sequence uses caching, the number written to disk is the last number placed in the sequence cache. This number is likely to be greater than the last sequence number that was used

ALL_SNAPSHOT_LOGS

ALL_SNAPSHOT_LOGS lists all snapshot logs accessible to the current user.

Related views:

- DBA_SNAPSHOT_LOGS describes all snapshot logs in the database.
- USER_SNAPSHOT_LOGS describes all snapshot logs owned by the current user.

Column	Datatype	NULL	Description
LOG_OWNER	VARCHAR2(30)	NOT NULL	Owner of the log
MASTER	VARCHAR2(30)	NOT NULL	Name of the master table whose changes are logged
LOG_TABLE	VARCHAR2(30)	NOT NULL	Name of the table where the changes to the master table are recorded

Column	Datatype	NULL	Description
LOG_TRIGGER	VARCHAR2(30)		Obsolete with the release of Oracle8*i* Server. Set to NULL. Formerly, this parameter was an after-row trigger on the master which inserted rows into the log
ROWIDS	VARCHAR2(3)		If YES, records ROWID information
PRIMARY_KEY	VARCHAR2(3)		If YES, records primary key information
FILTER_COLUMNS	VARCHAR2(3)		If YES, snapshot log records filter columns
CURRENT_SNAPSHOTS	DATE		One date per snapshot; the date the snapshot of the master was last refreshed
SNAPSHOT_ID	NUMBER(38)		Unique identifier of the snapshot

ALL_SNAPSHOT_REFRESH_TIMES

ALL_SNAPSHOT_REFRESH_TIMES describes refresh times of snapshots accessible to the current owner.

Related views:

- DBA_SNAPSHOT_REFRESH_TIMES describes refresh times of all snapshots in the database.
- USER_SNAPSHOT_REFRESH_TIMES describes refresh times of all snapshots owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot
NAME	VARCHAR2(30)	NOT NULL	Name of the snapshot view
MASTER_OWNER	VARCHAR2(30)		Owner of the master table
MASTER	VARCHAR2(30)		Name of the master table
LAST_REFRESH	DATE		The last refresh

ALL_SNAPSHOTS

ALL_SNAPSHOTS describes all snapshots accessible to the user.

Related views:

- DBA_SNAPSHOTS describes all snapshots in the database.
- USER_SNAPSHOTS describes all snapshots owned by the current user.

Note: All of the information in these views is also displayed in ALL_MVIEWS and its related views. Oracle Corporation recommends that you refer to ALL_MVIEWS for this information instead of these views.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the snapshot
NAME	VARCHAR2(30)	NOT NULL	Name of the view used by users and applications for viewing the snapshot
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table the snapshot is stored in. This table may have additional columns
MASTER_VIEW	VARCHAR2(30)		View of the master table, owned by the snapshot owner, used for refreshes. This is obsolete in Oracle8*i* and is set to NULL
MASTER_OWNER	VARCHAR2(30)		Owner of the master table
MASTER	VARCHAR2(30)		Name of the master table of which this snapshot is a copy
MASTER_LINK	VARCHAR2(128)		Database link name to the master site
CAN_USE_LOG	VARCHAR2(3)		YES if this snapshot can use a snapshot log, NO if this snapshot is too complex to use a log
UPDATABLE	VARCHAR2(3)		Specifies whether the snapshot is updatable. YES if it is, NO if it is not. If set to YES, the snapshot is read only
REFRESH_METHOD	VARCHAR2(11)		Values used to drive a refresh of the snapshot (PRIMARY KEY / ROWID / COMPLEX). If PRIMARY KEY, then the snapshot uses primary keys to drive a fast refresh. If ROWID, then it uses RowIDs to drive a fast refresh. If COMPLEX, then fast refreshes are not allowed and the snapshot can only perform complete refreshes
LAST_REFRESH	DATE		Date and time at the master site of the last refresh
ERROR	NUMBER		The number of failed automatic refreshes since last successful refresh
FR_OPERATIONS	VARCHAR2(10)		Status of generated fast refresh operations: (REGENERATE, VALID)
CR_OPERATIONS	VARCHAR2(10)		Status of generated complete refresh operations: (REGENERATE, VALID)
TYPE	VARCHAR2(8)		Type of refresh for all automatic refreshes: COMPLETE, FAST, FORCE

Column	Datatype	NULL	Description
NEXT	VARCHAR2(200)		Date function used to compute next refresh dates
START_WITH	DATE		Date function used to compute next refresh dates
REFRESH_GROUP	NUMBER		All snapshots in a given refresh group get refreshed in the same transaction
UPDATE_TRIG	VARCHAR2(30)		Obsolete. It is NULL for Oracle8*i* snapshots. Formerly, the name of the trigger that fills the UPDATE_LOG
UPDATE_LOG	VARCHAR2(30)		The table that logs changes made to an updatable snapshots
QUERY	LONG		Original query of which this snapshot is an instantiation
MASTER_ROLLBACK_SEG	VARCHAR2(30)		Rollback segment to use at the master site
STATUS	VARCHAR2(7)		The status of the contents of the snapshot
REFRESH_MODE	VARCHAR2(8)		This indicates how and when the snapshot will be refreshed
PREBUILT	VARCHAR2(3)		If YES, this snapshot uses a prebuilt table as the base table

ALL_SOURCE

ALL_SOURCE describes the text source of all stored objects accessible to the user.

Related views:

- DBA_SOURCE describes the text source of all stored objects in the database.
- USER_SOURCE describes the text source of all stored objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of object: PROCEDURE, PACKAGE, FUNCTION, PACKAGE BODY, TRIGGER, TYPE, TYPE BODY
LINE	NUMBER	NOT NULL	Line number of this line of source
TEXT	VARCHAR2(4000)		Text source of the stored object

ALL_SUBPART_COL_STATISTICS

ALL_SUBPART_COL_STATISTICS describes column statistics and histogram information for subpartitions of partitioned objects accessible to the current user.

Related views:

- DBA_SUBPART_COL_STATISTICS provides this information for all subpartitions in the database.
- USER_SUBPART_COL_STATISTICS provides this information for subpartitions of all partitioned objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner name
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
SUBPARTITION_NAME	VARCHAR2(30)		Table subpartition name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
NUM_DISTINCT	NUMBER		Number of distinct values in the column
LOW_VALUE	RAW(32)		Low value in the column
HIGH_VALUE	RAW(32)		High value in the column
DENSITY	NUMBER		Density of the column
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in histogram for the column
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
GLOBAL_STATS	VARCHAR2(3)		Indicates whether column statistics for the subpartition were collected by analyzing the table as a whole (YES) or estimated from statistics gathered on partitions and subpartitions (NO).
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

ALL_SUBPART_HISTOGRAMS

ALL_SUBPART_HISTOGRAMS lists the actual histogram data (end-points per histogram) for histograms on table subpartitions accessible to the current user.

Related views:

- DBA_SUBPART_HISTOGRAMS provides this information for all subpartitions in the database.
- USER_SUBPART_HISTOGRAMS provides this information for subpartitions of all partitioned objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner name
TABLE_NAME	VARCHAR2(30)		Table name
SUBPARTITION_NAME	VARCHAR2(30)		Table subpartition name
COLUMN_NAME	VARCHAR2(30)		Column name
BUCKET_NUMBER	NUMBER		Bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket
ENDPOINT_ACTUAL_VALUE	VARCHAR2(1000)		Actual (not normalized) string value of the endpoint for this bucket.

ALL_SUBPART_KEY_COLUMNS

ALL_SUBPART_KEY_COLUMNS lists subpartitioning key columns for composite-partitioned tables (and local indexes on composite-partitioned tables) accessible to the current user.

Related views:

- DBA_SUBPART_KEY_COLUMNS provides this information for all subpartitions in the database.
- USER_SUBPART_KEY_COLUMNS provides this information for subpartitions of all partitioned objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		The partitioned table or index owner
NAME	VARCHAR2(30)		The partitioned table or index name
OBJECT_TYPE	VARCHAR2(11)		The object type (TABLE or INDEX)
COLUMN_NAME	VARCHAR2(30)		The column name

Column	Datatype	NULL	Description
COLUMN_POSITION	NUMBER		The position of the column within the subpartitioning key

ALL_SUMDELTA

ALL_SUMDELTA lists direct path load entries accessible to the user.

Column	Datatype	NULL	Description
TABLEOBJ#	NUMBER	NOT NULL	Object number of the table
PARTITIONOBJ#	NUMBER	NOT NULL	Object number of table partitions (if the table is partitioned)
DMLOPERATION	VARCHAR2(1)		Type of DML operation applied to the table
SCN	NUMBER	NOT NULL	SCN when the bulk DML occurred
TIMESTAMP	DATE	NOT NULL	Timestamp of log entry
LOWROWID	ROWID	NOT NULL	The start ROWID in the loaded rowid range
HIGHROWID	ROWID	NOT NULL	The end ROWID in the loaded rowid range

ALL_SYNONYMS

ALL_SYNONYMS describes all synonyms accessible to the user.

Related views:

- DBA_SYNONYMS describes all synonyms in the database.
- USER_SYNONYMS describes all synonyms owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the synonym
SYNONYM_NAME	VARCHAR2(30)	NOT NULL	Name of the synonym
TABLE_OWNER	VARCHAR2(30)		Owner of the object referenced by the synonym
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object referenced by the synonym
DB_LINK	VARCHAR2(128)		Name of the database link referenced, if any

ALL_TAB_COL_STATISTICS

ALL_TAB_COL_STATISTICS provides column statistics and histogram information extracted from "ALL_TAB_COLUMNS" on page 2-75.

Related views:

- DBA_TAB_COL_STATISTICS provides such information extracted from "DBA_TAB_COLUMNS" on page 2-135.
- USER_TAB_COL_STATISTICS provides such information extracted from "USER_TAB_COLUMNS" on page 2-182.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
NUM_DISTINCT	NUMBER		Number of distinct values in the column
LOW_VALUE	RAW(32)		Low value in the column
HIGH_VALUE	RAW(32)		High value in the column
DENSITY	NUMBER		Density of the column
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in histogram for the column
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether column statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO).
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

ALL_TAB_COLUMNS

ALL_TAB_COLUMNS describes the columns of all tables, views, and clusters accessible to the user. To gather statistics for this view, use the SQL command ANALYZE.

Related views:

- DBA_TAB_COLUMNS describes the columns of all tables, views, and clusters in the database.
- USER_TAB_COLUMNS describes the columns of all tables, views, and clusters owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table, view or cluster
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table, view, or cluster name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
DATA_TYPE	VARCHAR2(30)		Datatype of the column
DATA_TYPE_MOD	VARCHAR2(3)		Datatype modifier of the column
DATA_TYPE_OWNER	VARCHAR2(30)		Owner of the datatype of the column
DATA_LENGTH	NUMBER	NOT NULL	Length of the column in bytes
DATA_PRECISION	NUMBER		Decimal precision for NUMBER datatype; binary precision for FLOAT datatype, null for all other datatypes
DATA_SCALE	NUMBER		Digits to right of decimal point in a number
NULLABLE	VARCHAR2(1)		Specifies whether a column allows NULLs. Value is N if there is a NOT NULL constraint on the column or if the column is part of a PRIMARY KEY.
COLUMN_ID	NUMBER	NOT NULL	Sequence number of the column as created
DEFAULT_LENGTH	NUMBER		Length of default value for the column
DATA_DEFAULT	LONG		Default value for the column
NUM_DISTINCT	NUMBER		These columns remain for backward compatibility with Oracle7. This information is now in the {TAB|PART}_COL_STATISTICS views. This view now picks up these values from HIST_HEAD$ rather than COL$.
LOW_VALUE	RAW(32)		
HIGH_VALUE	RAW(32)		
DENSITY	NUMBER		
NUM_NULLS	NUMBER		The number of nulls in the column
NUM_BUCKETS	NUMBER		The number of buckets in histogram for the column
LAST_ANALYZED	DATE		The date on which this column was most recently analyzed
SAMPLE_SIZE			The sample size used in analyzing this column
CHARACTER_SET_NAME	VARCHAR2(44)		The name of the character set: CHAR_CS or NCHAR_CS
CHAR_COL_DECL_LENGTH	NUMBER		The length

Column	Datatype	NULL	Description
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether column statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO).
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?
AVG_COL_LEN	NUMBER		The average length of the column, in bytes

ALL_TAB_COMMENTS

ALL_TAB_COMMENTS describes comments on tables and views accessible to the user.

Related views:

- DBA_TAB_COMMENTS describes comments on all tables and views in the database.
- USER_TAB_COMMENTS describes comments on all tables and views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
TABLE_TYPE	VARCHAR2(11)		Type of the object
COMMENTS	VARCHAR2(4000)		Comment on the object

ALL_TAB_HISTOGRAMS

ALL_TAB_HISTOGRAMS describes histograms on tables and views accessible to the user.

Related views:

- DBA_TAB_HISTOGRAMS describes histograms on all tables and views in the database.
- USER_TAB_HISTOGRAMS describes histograms on all tables and views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of table
TABLE_NAME	VARCHAR2(30)		Name of the table
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of the object type column
ENDPOINT_NUMBER	NUMBER		Histogram bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint value for this bucket
ENDPOINT_ACTUAL_VALUE	VARCHAR2(1000)		Actual (not normalized) string value of the endpoint for this bucket.

ALL_TAB_MODIFICATIONS

ALL_TAB_MODIFICATIONS describes tables accessible to the current user that have been modified since the last time statistics were gathered on the tables.

Related views:

- DBA_TAB_MODIFICATIONS provides such information for all tables in the database.
- USER_TAB_MODIFICATIONS provides such information for tables owned by the current user. This view does not display the TABLE_OWNER column.

> **Note:** These views are populated only for tables with the MONITORING attribute. They are intended for statistics collection over a long period of time. For performance reasons, Oracle may not populate these views until a few hours after the actual modifications occurred.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Owner of the modified table.
TABLE_NAME	VARCHAR2(30)		Name of the modified table
PARTITION_NAME	VARCHAR2(30)		Name of the modified partition
SUBPARTITION_NAME	VARCHAR2(30)		Name of the modified subpartition
INSERTS	NUMBER		Approximate number of inserts since the last time statistics were gathered
UPDATES	NUMBER		Approximate number of updates since the last time statistics were gathered

Column	Datatype	NULL	Description
DELETES	NUMBER		Approximate number of deletes since the last time statistics were gathered
TIMESTAMP	DATE		Indicates the last time the table was modified
TRUNCATED	VARCHAR2(3)		Whether or not the table was truncated since the last time statistics were gathered

ALL_TAB_PARTITIONS

ALL_TAB_PARTITIONS provides partition-level partitioning information, partition storage parameters, and partition statistics collected by ANALYZE statements for partitions accessible to the current user.

Related views:

- DBA_TAB_PARTITIONS provides such information for all partitions in the database.
- USER_TAB_PARTITIONS provides such information for partitions of all partitioned objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COMPOSITE	VARCHAR2(3)		YES if the partition belongs to a Local index on a table partitioned using Composite method; NO otherwise
PARTITION_NAME	VARCHAR2(30)		Partition name
SUBPARTITION_COUNT	NUMBER		If this is a Local index on a table partitioned using a Composite method, the number of subpartitions in the partition
HIGH_VALUE	LONG		Partition bound value expression
HIGH_VALUE_LENGTH	NUMBER	NOT NULL	Length of partition bound value expression
PARTITION_POSITION	NUMBER	NOT NULL	Position of the partition within the table
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the partition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
PCT_USED	NUMBER	NOT NULL	Minimum percentage of used space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions

Column	Datatype	NULL	Description
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of partition
NUM_ROWS	NUMBER		Number of rows in the partition
BLOCKS	NUMBER		Number of used blocks in the partition
EMPTY_BLOCKS	NUMBER		Number of empty (never used) blocks in the partition
AVG_SPACE	NUMBER		Average available free space in the partition
CHAIN_CNT	NUMBER		Number of chained rows in the partition
AVG_ROW_LEN	NUMBER		Average row length, including row overhead
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this partition
LAST_ANALYZED	DATE		Date on which this partition was most recently analyzed
BUFFER_POOL	VARCHAR2(7)		The default buffer pool to be used for the partition blocks
GLOBAL_STATS	VARCHAR2(3)		Indicates whether statistics were collected for the partition as a whole (YES) or were estimated from statistics on underlying subpartitions (NO).
USER_STATS	VARCHAR2(3)		User statistics

ALL_TAB_PRIVS

ALL_TAB_PRIVS lists the grants on objects where the user or PUBLIC is the grantee. This view does not display the OWNER column.

Related views:

- DBA_TAB_PRIVS lists all grants on all objects in the database. This view does not display the TABLE_SCHEMA column.
- USER_TAB_PRIVS lists grants on all objects where the current user is the grantee. This view does not display the TABLE_SCHEMA column.

Column	Datatype	NULL	Description
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access is granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Schema of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2(3)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

ALL_TAB_PRIVS_MADE

ALL_TAB_PRIVS_MADE lists the all object grants made by the current user or made on the objects owned by the current user.

Related view:

- USER_TAB_PRIVS_MADE lists grants on all objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2(3)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

ALL_TAB_PRIVS_RECD

ALL_TAB_PRIVS_RECD lists object grants for which the user or PUBLIC is the grantee.

Related view:

- USER_TAB_PRIVS_RECD lists object grants for which the current user is the grantee. This view does not display the OWNER column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2(3)		YES if the privilege was granted with ADMIN OPTION; otherwise NO

ALL_TAB_SUBPARTITIONS

ALL_TAB_SUBPARTITIONS describes, for each table subpartition accessible to the user, the subpartition name, name of the table and partition to which it belongs, and its storage attributes.

Related views:

- DBA_TAB_SUBPARTITIONS provides such information for all subpartitions in the database.
- USER_TAB_SUBPARTITIONS provides such information for subpartitions of all partitioned objects owned by the current user. This view does not display the TABLE_OWNER column.

Note: Statistics are not collected on a per-subpartition basis.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
PARTITION_NAME	VARCHAR2(30)		Partition name
SUBPARTITION_NAME	VARCHAR2(30)		Subpartition name
SUBPARTITION_POSITION	NUMBER	NOT NULL	Position of a subpartition within a partition
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the subpartition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
PCT_USED	NUMBER	NOT NULL	Minimum percentage of used space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions

Column	Datatype	NULL	Description
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of freelist groups allocated in this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of subpartition
NUM_ROWS	NUMBER		The number of rows
BLOCKS	NUMBER		The number of blocks
EMPTY_BLOCKS	NUMBER		The number of empty blocks
AVG_SPACE	NUMBER		The average space
CHAIN_CNT	NUMBER		The chain count
AVG_ROW_LEN	NUMBER		The average row length
SAMPLE_SIZE	NUMBER		The sample size
LAST_ANALYZED	DATE		The date on which this table was most recently analyzed
BUFFER_POOL	VARCHAR2(7)		The default buffer pool for this subpartition
GLOBAL_STATS	VARCHAR2(3)		Indicates whether column statistics for the subpartition were collected by analyzing the table as a whole (YES) or estimated from statistics collected for partitions and subpartitions (NO).
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?

ALL_TABLES

ALL_TABLES describes all relational tables accessible to the user. To gather statistics for this view, use the SQL command ANALYZE.

Related views:

- DBA_TABLES describes all relational tables in the database.
- USER_TABLES describes all relational tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the table; NULL for partitioned tables
CLUSTER_NAME	VARCHAR2(30)		Name of the cluster, if any, to which the table belongs
IOT_NAME	VARCHAR2(30)		Name of the index organized table, if any, to which the overflow entry belongs
PCT_FREE	NUMBER		Minimum percentage of free space in a block; NULL for partitioned tables
PCT_USED	NUMBER		Minimum percentage of used space in a block; NULL for partitioned tables
INI_TRANS	NUMBER		Initial number of transactions; NULL for partitioned tables
MAX_TRANS	NUMBER		Maximum number of transactions; NULL for partitioned tables
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes; NULL for partitioned tables
NEXT_EXTENT	NUMBER		Size of the secondary extension bytes; NULL for partitioned tables
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment; NULL for partitioned tables
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment; NULL for partitioned tables
PCT_INCREASE	NUMBER		Percentage increase in extent size; NULL for partitioned tables
FREELISTS	NUMBER		Number of process freelists allocated to this segment; NULL for partitioned tables
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment; NULL for partitioned tables
LOGGING	VARCHAR2(3)		Logging attribute; NULL for partitioned tables
BACKED_UP	VARCHAR2(1)		Has table been backed up since last change
NUM_ROWS	NUMBER		Number of rows in the table
BLOCKS	NUMBER		Number of used data blocks in the table
EMPTY_BLOCKS	NUMBER		Number of empty (never used) data blocks in the table
AVG_SPACE	NUMBER		Average amount of free space, in bytes, in a data block allocated to the table

Column	Datatype	NULL	Description
CHAIN_CNT	NUMBER		Number of rows in the table that are chained from one data block to another, or which have migrated to a new block, requiring a link to preserve the old ROWID
AVG_ROW_LEN	NUMBER		Average length of a row in the table in bytes
AVG_SPACE_FREELIST_BLOCKS	NUMBER		The average freespace of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		The number of blocks on the freelist
DEGREE	VARCHAR2(10)		The number of threads per instance for scanning the table
INSTANCES	VARCHAR2(10)		The number of instances across which the table is to be scanned
CACHE	VARCHAR2(5)		Whether the cluster is to be cached in the buffer cache (CACHE | NOCACHE)
TABLE_LOCK	VARCHAR2(8)		Whether table locking is enabled or disabled
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this table
LAST_ANALYZED	DATE		Date on which this table was most recently analyzed
PARTITIONED	VARCHAR2(3)		Indicates whether this table is partitioned. Set to YES if it is partitioned
IOT_TYPE	VARCHAR2(12)		If this is an index organized table, then IOT_TYPE is IOT or IOT_OVERFLOW. If this is not an index organized table, then IOT_TYPE is NULL
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
SECONDARY	VARCHAR2(1)		Whether the trigger is a secondary object created by the ODCIIndexCreate method of the Oracle8*i* Data Cartridge (Y or N)
NESTED	VARCHAR2(3)		Is the table a nested table?
BUFFER_POOL	VARCHAR2(7)		The default buffer pool for the object. NULL for partitioned tables
ROW_MOVEMENT	VARCHAR2(8)		Whether partitioned row movement is enabled or disabled
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO).
USER_STATS	VARCHAR2(3)		Were the statistics entered directly by the user?

Column	Datatype	NULL	Description
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: ■ SYS$SESSION: the rows are preserved for the duration of the session ■ SYS$TRANSACTION: the rows are deleted after COMMIT. Null for a permanent table.
SKIP_CORRUPT	VARCHAR2(8)		Whether Oracle ignores blocks marked corrupt during table and index scans (ENABLED) or raises an error (DISABLED). To enable this feature, run the DBMS_REPAIR.SKIP_CORRUPT_BLOCKS procedure.
MONITORING	VARCHAR2(3)		Whether the table has the MONITORING attribute set

ALL_TRIGGERS

ALL_TRIGGERS describes owned by the current user and triggers on tables owned by the current user. If the user has the CREATE ANY TRIGGER privilege, then this view describes all triggers in the database.

Related views:

- DBA_TRIGGERS describes all triggers in the database.
- USER_TRIGGERS describes all triggers owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the trigger
TRIGGER_NAME	VARCHAR2(30)	NOT NULL	Name of the trigger
TRIGGER_TYPE	VARCHAR2(16)		When the trigger fires: BEFORE STATEMENT, BEFORE EACH ROW, BEFORE EVENT, AFTER STATEMENT, AFTER EACH ROW, and AFTER EVENT
TRIGGERING_EVENT	VARCHAR2(216)		The DML, DDL, or database event that fires the trigger. For a listing of triggering events, see the CREATE TRIGGER statement in *Oracle8i SQL Reference*.
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table on which the trigger is defined
BASE_OBJECT_TYPE	VARCHAR2(16)		The base object on which the trigger is defined: TABLE, VIEW, SCHEMA, or DATABASE

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		If the base object type of the trigger is SCHEMA or DATABASE, then this column is NULL; if the base object type of the trigger is TABLE or VIEW, this columns indicates the table/view name on which the trigger is defined
COLUMN_NAME	VARCHAR2(30)		Name of the nested table column (if nested table trigger), else null
REFERENCING_NAMES	VARCHAR2(87)		Names used for referencing OLD and NEW column values from within the trigger
WHEN_CLAUSE	VARCHAR2(4000)		Must evaluate to TRUE for TRIGGER_BODY to execute
STATUS	VARCHAR2(8)		Whether the trigger is enabled: ENABLED or DISABLED
DESCRIPTION	VARCHAR2(4000)		Trigger description. Useful for re-creating a trigger creation statement.
ACTION_TYPE	VARCHAR2(11)		The action type of the trigger body: CALL or PL/SQL
TRIGGER_BODY	LONG		Statement(s) executed by the trigger when it fires

ALL_TRIGGER_COLS

ALL_TRIGGER_COLS describes the use of columns in triggers owned by user or in triggers on tables owned by user. If the user has the CREATE ANY TRIGGER privilege, this view describes the use of columns in all triggers.

Related views:

- DBA_TRIGGER_COLS provides such information for all triggers in the database.
- USER_TRIGGER_COLS provides such information for all triggers owned by the current user.

Column	Datatype	NULL	Description
TRIGGER_OWNER	VARCHAR2(30)	NOT NULL	Owner of the triggers
TRIGGER_NAME	VARCHAR2(30)	NOT NULL	Name of the trigger
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table on which the trigger is defined
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table on which the trigger is defined
COLUMN_NAME	VARCHAR2(4000)		Name of the column used in the trigger
COLUMN_LIST	VARCHAR2(3)		Column specified in UPDATE clause: Y | N

Column	Datatype	NULL	Description
COLUMN_USAGE	VARCHAR2(17)		How the column is used in the trigger. All applicable combinations of NEW, OLD, IN, OUT, and IN OUT.

ALL_TYPE_ATTRS

ALL_TYPE_ATTRS describes the attributes of object types accessible to the user.

Related views:

- DBA_TYPE_ATTRS describes the attributes of all object types in the database.
- USER_TYPE_ATTRS describes the attributes of all object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
ATTR_NAME	VARCHAR2(30)	NOT NULL	Name of the attribute
ATTR_TYPE_MOD	VARCHAR2(7)		Type modifier of the attribute
ATTR_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the attribute
ATTR_TYPE_NAME	VARCHAR2(30)		Name of the type of the attribute
LENGTH	NUMBER		Length of the CHAR attribute or maximum length of the VARCHAR or VARCHAR2 attribute
PRECISION	NUMBER		Decimal precision of the NUMBER or DECIMAL attribute or binary precision of the FLOAT attribute
SCALE	NUMBER		Scale of the NUMBER or DECIMAL attribute
CHARACTER_SET_NAME	VARCHAR2(44)		The name of the character set: CHAR_CS or NCHAR_CS
ATTR_NO	NUMBER	NOT NULL	Syntactical order number or position of the attribute as specified in the type specification or CREATE TYPE statement (not to be used as ID number)

ALL_TYPE_METHODS

ALL_TYPE_METHODS describes methods of types accessible to the user.

Related views:

- DBA_TYPE_METHODS describes the methods of all object types in the database.
- USER_TYPE_METHODS describes the methods of all object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
METHOD_NAME	VARCHAR2(30)	NOT NULL	Name of the method
METHOD_NO	NUMBER	NOT NULL	Method number for distinguishing overloaded method (not to be used as ID number)
METHOD_TYPE	VARCHAR2(6)		Type of the method
PARAMETERS	NUMBER	NOT NULL	Number of parameters to the method
RESULTS	NUMBER	NOT NULL	Number of results returned by the method

ALL_TYPES

ALL_TYPES describes all object types accessible to the user.

Related views:

- DBA_TYPES describes all object types in the database.
- USER_TYPES describes all object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
TYPE_OID	RAW(16)	NOT NULL	Object identifier (OID) of the type
TYPECODE	VARCHAR2(30)		Typecode of the type
ATTRIBUTES	NUMBER		Number of attributes in the type
METHODS	NUMBER		Number of methods in the type
PREDEFINED	VARCHAR2(3)		Indicates whether the type is a predefined type
INCOMPLETE	VARCHAR2(3)		Indicates whether the type is an incomplete type

ALL_UNUSED_COL_TABS

ALL_UNUSED_COL_TABS describes all tables accessible to the current user that contain unused columns.

Related views:

- DBA_UNUSED_COL_TABS describes all in the database that contain unused columns.
- USER_UNUSED_COL_TABS describes all tables owned by the current user that contain unused columns. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	The owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	The name of the table
COUNT	NUMBER		The number of unused columns

ALL_UPDATABLE_COLUMNS

ALL_UPDATABLE_COLUMNS describes all columns in a join view that are updatable by the current user.

Related views:

- DBA_UPDATABLE_COLUMNS describes all columns in a join view that are updatable by the database administrator.
- USER_UPDATABLE_COLUMNS describes all columns owned by the current user that are in a join view and are updatable by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
UPDATABLE	VARCHAR2(3)		Indicates whether the column is updatable
INSERTABLE	VARCHAR2(3)		Indicates whether the column is insertable
DELETABLE	VARCHAR2(3)		Indicates whether the column is deletable

ALL_USERS

ALL_USERS lists all users of the database visible to the current user. This view does not describe the users. See the related views.

Related views:

- DBA_USERS describes all users in the database, and contains more columns than ALL_USERS. See "DBA_USERS" on page 2-139.
- USER_USERS describes the current user, and contains more columns that ALL_USERS. See "USER_USERS" on page 2-186.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user
USER_ID	NUMBER	NOT NULL	ID number of the user
CREATED	DATE	NOT NULL	User creation date

ALL_USTATS

ALL_USTATS describes all user-defined statistics accessible to the current user.

Related views:

- DBA_USTATS describes all user-defined statistics in the database.
- USER_USTATS describes all user-defined statistics owned by the current user.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Owner of the table or index for which the statistics have been collected
OBJECT_NAME	VARCHAR2(30)		Name of the table or index for which the statistics have been collected
PROPERTY	VARCHAR2(6)		Property of the object - column or index
COLUMN_NAME	VARCHAR2(30)		Column name, if property is column for which statistics have been collected
STATSTYPE_SCHEMA	VARCHAR2(30)		Schema of statistics type which was used to collect the statistics
STATSTYPE_NAME	VARCHAR2(30)		Name of statistics type which was used to collect statistics
STATISTICS	RAW(2000)		User collected statistics for the object

ALL_VARRAYS

ALL_VARRAYS describes all varrays accessible to the user.

Related views:

- DBA_VARRAYS describes all varrays in the database.
- USER_VARRAYS describes all varrays owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table containing the varray
PARENT_TABLE_NAME	VARCHAR2(30)		Name of the containing table
PARENT_TABLE_COLUMN	VARCHAR2(4000)		Name of the varray column or attribute
TYPE_OWNER	VARCHAR2(30)		Owner of the varray type
TYPE_NAME	VARCHAR2(30)		Name of the varray type
LOB_NAME	VARCHAR2(30)		Name of the LOB if the varray is stored in a LOB
STORAGE_SPEC	VARCHAR2(30)		DEFAULT value indicates that the storage was defaulted. USER_SPECIFIED value indicates that the storage was user-specified
RETURN_TYPE	VARCHAR2(20)		Return type of the column

ALL_VIEWS

ALL_VIEWS describes all views accessible to the current user.

Related views:

- DBA_VIEWS describes all views in the database.
- USER_VIEWS describes all views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the view
VIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the view
TEXT_LENGTH	NUMBER		Length of the view text
TEXT	LONG		View text
TYPE_TEXT_LENGTH	NUMBER		Length of the type clause of the typed view

Column	Datatype	NULL	Description
TYPE_TEXT	VARCHAR2(4000)		Type clause of the typed view
OID_TEXT_LENGTH	NUMBER		Length of the WITH OID clause of the typed view
OID_TEXT	VARCHAR2(4000)		WITH OID clause of the typed view
VIEW_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the view if the view is a typed view
VIEW_TYPE	VARCHAR2(30)		Type of the view if the view is a typed view

AUDIT_ACTIONS

AUDIT_ACTIONS describes audit trail action type codes.

Column	Datatype	NULL	Description
ACTION	NUMBER	NOT NULL	Numeric audit trail action type code. These values also appear in the COMMAND column of the V$SESSION dynamic performance view, and are listed with that view. **See Also:** Table 3–3 on page 3-93 for a listing of the action type codes.
NAME	VARCHAR2(27)	NOT NULL	Name of the type of audit trail action

CATALOG

CATALOG is included for compatibility with Oracle version 5. Oracle Corporation recommends that you not use this view.

CAT

CAT is a synonym for "USER_CATALOG" on page 2-168.

CHAINED_ROWS

CHAINED_ROWS stores the output for the ANALYZE command with the LIST CHAINED ROWS clause. You must run the UTLCHAIN.SQL or UTLCHN1.SQL script to create this table.

Column	Description
OWNER_NAME	Table owner

Column	Description
TABLE_NAME	Table name
CLUSTER_NAME	Cluster the table is in, if any
PARTITION_NAME	The name of the partition
SUBPARTITION_NAME	The name of the subpartition
HEAD_ROWID	ROWID the chained row is accessed by
ANALYZE_TIMESTAMP	Date/time that the ANALYZE command was issued

CLU

CLU is a synonym for "USER_CLUSTERS" on page 2-168.

CODE_PIECES

Oracle accesses this view to create the DBA_OBJECT_SIZE and USER_OBJECT_SIZE views.

> **See Also:** "DBA_OBJECT_SIZE" on page 2-116 and "USER_OBJECT_SIZE" on page 2-176.

CODE_SIZE

Oracle accesses this view to create the DBA_OBJECT_SIZE and USER_OBJECT_SIZE views.

> **See Also:** "DBA_OBJECT_SIZE" on page 2-116 and "USER_OBJECT_SIZE" on page 2-176.

COL

COL is included for compatibility with Oracle version 5. Oracle Corporation recommends that you not use this view.

COLS

COLS is a synonym for USER_TAB_COLUMNS.

> **See Also:** "USER_TAB_COLUMNS" on page 2-182.

COLUMN_PRIVILEGES

COLUMN_PRIVILEGES is included for compatibility with Oracle version 6. Oracle Corporation does not recommend use of this view.

DBA_2PC_NEIGHBORS

DBA_2PC_NEIGHBORS describes incoming and outgoing connections for pending transactions.

Column	Datatype	NULL	Description
LOCAL_TRAN_ID	VARCHAR2(22)		Local identifier of a transaction
IN_OUT	VARCHAR2(3)		IN for incoming connections, OUT for outgoing
DATABASE	VARCHAR2(128)		IN: client database name; OUT: outgoing database link
DBUSER_OWNER	VARCHAR2(30)		IN: name of local user; OUT: owner of database link
INTERFACE	VARCHAR2(1)		C for request commit, otherwise N for prepare or request readonly commit
DBID	VARCHAR2(16)		The database ID at the other end of the connection
SESS#	NUMBER		Session number of the connection at this database
BRANCH	VARCHAR2(128)		Transaction branch ID of the connection at this database

DBA_2PC_PENDING

DBA_2PC_PENDING describes distributed transactions awaiting recovery.

Column	Datatype	NULL	Description
LOCAL_TRAN_ID	VARCHAR2(22)	NOT NULL	String of form: n.n.n; n is a number
GLOBAL_TRAN_ID	VARCHAR2(169)		Globally unique transaction ID
STATE	VARCHAR2(16)	NOT NULL	Collecting, prepared, committed, forced commit, or forced rollback
MIXED	VARCHAR2(3)		YES = part of the transaction committed and part rolled back
ADVICE	VARCHAR2(1)		C for commit, R for rollback, else NULL
TRAN_COMMENT	VARCHAR2(2000)		Text for commit work comment text
FAIL_TIME	DATE	NOT NULL	Value of SYSDATE when the row was inserted (tx or system recovery)

Column	Datatype	NULL	Description
FORCE_TIME	DATE		Time of manual force decision (null if not forced locally)
RETRY_TIME	DATE	NOT NULL	Time automatic recovery (RECO) last tried to recover the transaction
OS_USER	VARCHAR2(2000)		Time automatic recovery (RECO) last tried to recover the transaction
OS_TERMINAL	VARCHAR2(2000)		Time automatic recovery (RECO) last tried to recover the transaction
HOST	VARCHAR2(2000)		Name of the host machine for the end-user
DB_USER	VARCHAR2(30)		Name of the host machine for the end-user
COMMIT#	VARCHAR2(16)		Name of the host machine for the end-user

DBA_ALL_TABLES

DBA_ALL_TABLES displays descriptions of all object tables and relational tables in the database. Its columns are the same as those in "ALL_ALL_TABLES" on page 2-3.

DBA_ANALYZE_OBJECTS

DBA_ANALYZE_OBJECTS lists objects that have been analyzed.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	The owner
OBJECT_NAME	VARCHAR2(30)	NOT NULL	The object name
OBJECT_TYPE	VARCHAR2(7)		The object type

DBA_ASSOCIATIONS

DBA_ASSOCIATIONS describes all user-defined statistics in the database. Its columns are the same as those for "ALL_ASSOCIATIONS" on page 2-6.

DBA_AUDIT_EXISTS

DBA_AUDIT_EXISTS lists audit trail entries produced by AUDIT NOT EXISTS.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier of the user's terminal
TIMESTAMP	DATE	NOT NULL	Timestamp for the creation of the audit trail entry
OWNER	VARCHAR2(30)		Intended creator of the non-existent object
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action
ACTION_NAME	VARCHAR2(27)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of an object after a RENAME or the name of the underlying object
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a GRANT or REVOKE statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted or revoked by a GRANT or REVOKE statement
GRANTEE	VARCHAR2(30)		Name of grantee specified in a GRANT or REVOKE statement
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle Server message code generated by the action. Some useful values: ■ zero: the action succeeded ■ 2004: security violation

DBA_AUDIT_OBJECT

DBA_AUDIT_OBJECT contains audit trail records for all objects in the system.

Related view:

- USER_AUDIT_OBJECT contains audit trail records for all objects accessible to the current user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier of the user's terminal
TIMESTAMP	DATE	NOT NULL	Timestamp for the creation of the audit trail entry or login time for the CONNECT statement
OWNER	VARCHAR2(30)		Creator of the object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action
ACTION_NAME	VARCHAR2(27)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of an object after a RENAME or the name of the underlying object
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 16 characters, one for each action type in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE, REFERENCES, and EXECUTE. Positions 14, 15, and 16 are reserved for future use. The characters are: - for none, S for success, F for failure, and B for both)
COMMENT_TEXT	VARCHAR2(4000		Text comment on the audit trail
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle Server message code generated by the action. Some useful values: ■ zero: the action succeeded ■ 2004: security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action

DBA_AUDIT_SESSION

DBA_AUDIT_SESSION lists all audit trail records concerning CONNECT and DISCONNECT.

Related view:

- USER_AUDIT_SESSION contains audit trail records concerning connections and disconnections of the current user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier of the user's terminal
TIMESTAMP	DATE	NOT NULL	Timestamp for the creation of the audit trail entry or login time for the CONNECT statement
ACTION_NAME	VARCHAR2(27)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL
LOGOFF_TIME	DATE		Timestamp for user log off
LOGOFF_LREAD	NUMBER		Logical reads for the session
LOGOFF_PREAD	NUMBER		Physical reads for the session
LOGOFF_LWRITE	NUMBER		Logical writes for the session
LOGOFF_DLOCK	VARCHAR2(40)		Deadlocks detected during the session
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
RETURNCODE	NUMBER	NOT NULL	Oracle Server message code generated by the action. Some useful values: ■ zero: the action succeeded ■ 2004: security violation

DBA_AUDIT_STATEMENT

DBA_AUDIT_STATEMENT lists audit trail records concerning GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYSTEM statements throughout the database.

Related view:

- USER_AUDIT_STATEMENT contains audit trail records for the same statements issued by the current user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier of the user's terminal
TIMESTAMP	DATE	NOT NULL	Timestamp for the creation of the audit trail entry or login time for the CONNECT statement
OWNER	VARCHAR2(30)		Creator of the object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of object affected by the action
ACTION_NAME	VARCHAR2(27)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL
NEW_NAME	VARCHAR2(128)		New name of an object after a RENAME or the name of the underlying object
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a GRANT or REVOKE statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted or revoked by a GRANT or REVOKE statement
ADMIN_OPTION	VARCHAR2(1)		Signifies the role or system privilege was granted with ADMIN option
GRANTEE	VARCHAR2(30)		Name of grantee specified in a GRANT or REVOKE statement
AUDIT_OPTION	VARCHAR2(40)		Auditing option set with the AUDIT statement
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 16 characters, one for each action type in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE, REFERENCES, and EXECUTE. Positions 14, 15, and 16 are reserved for future use. The characters are: - for none, S for success, F for failure, and B for both)
COMMENT_TEXT	VARCHAR2(4000)		Text comment on the audit trail, inserted by the application
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session

Column	Datatype	NULL	Description
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle Server message code generated by the action. Some useful values: ■ zero: the action succeeded ■ 2004: security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action

DBA_AUDIT_TRAIL

DBA_AUDIT_TRAIL lists all audit trail entries.

Related view:

- USER_AUDIT_TRAIL contains all audit trail entries related to the current user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(2000)		Numeric instance ID for the Oracle instance from which the user is accessing the database
TERMINAL	VARCHAR2(2000)		Identifier of the user's terminal
TIMESTAMP	DATE	NOT NULL	Timestamp for the creation of the audit trail entry or login time for the CONNECT statement
OWNER	VARCHAR2(30)		Creator of the object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action
ACTION	NUMBER	NOT NULL	Numeric type code corresponding to the action
ACTION_NAME	VARCHAR2(27)		Name of the action type corresponding to the numeric code in the ACTION column
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of an object after a RENAME or the name of the underlying object
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a GRANT or REVOKE statement

Column	Datatype	NULL	Description
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted or revoked by a GRANT or REVOKE statement
ADMIN_OPTION	VARCHAR2(1)		Signifies the role or system privilege was granted with ADMIN option
GRANTEE	VARCHAR2(30)		Name of grantee specified in a GRANT or REVOKE statement
AUDIT_OPTION	VARCHAR2(40)		Auditing option set with the AUDIT statement
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 16 characters, one for each action type in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE, REFERENCES, and EXECUTE. Positions 14, 15, and 16 are reserved for future use. The characters are: - for none, S for success, F for failure, and B for both)
LOGOFF_TIME	DATE		Timestamp for user log off
LOGOFF_LREAD	NUMBER		Logical reads for the session
LOGOFF_PREAD	NUMBER		Physical reads for the session
LOGOFF_LWRITE	NUMBER		Logical writes for the session
LOGOFF_DLOCK	VARCHAR2(40)		Deadlocks detected during the session
COMMENT_TEXT	VARCHAR2(4000)		Text comment on the audit trail entry, providing more information about the statement audited Also indicates how the user was authenticated. The method can be one of the following: ■ DATABASE - authentication was done by password ■ NETWORK - authentication was done by Net8 or the Advanced Security option ■ PROXY - the client was authenticated by another user. The name of the proxy user follows the method type
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle Server message code generated by the action. Some useful values: ■ zero: the action succeeded ■ 2004: security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action

DBA_BLOCKERS

DBA_BLOCKERS displays a session if it is not waiting for a locked object but is holding a lock on an object for which another session is waiting.

Column	Datatype	NULL	Description
HOLDING_SESSION	NUMBER		Session holding a lock

DBA_CATALOG

DBA_CATALOG lists all tables, views, synonyms, and sequences in the database. Its columns are the same as those in "ALL_CATALOG" on page 2-7.

DBA_CLU_COLUMNS

DBA_CLU_COLUMNS maps all table columns to related cluster columns.

Related view:

- USER_CLU_COLUMNS maps all table columns owned by the current user to related cluster columns. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Cluster name
CLU_COLUMN_NAME	VARCHAR2(30)	NOT NULL	Key column in the cluster
TABLE_NAME	VARCHAR2(30)	NOT NULL	Clustered table name
TAB_COLUMN_NAME	VARCHAR2(4000)		Key column or attribute of the object type column

DBA_CLUSTERS

DBA_CLUSTERS describes all clusters in the database. Its columns are the same as those in "ALL_CLUSTERS" on page 2-8.

DBA_CLUSTER_HASH_EXPRESSIONS

DBA_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters in the database. Its columns are the same as those in "ALL_CLUSTER_HASH_EXPRESSIONS" on page 2-9.

DBA_COL_COMMENTS

DBA_COL_COMMENTS describes comments on columns of all tables and views in the database. Its columns are the same as those in "ALL_COL_COMMENTS" on page 2-9.

DBA_COL_PRIVS

DBA_COL_PRIVS describes all object column grants in the database. Its columns are the same as those in "ALL_COL_PRIVS" on page 2-10.

DBA_COLL_TYPES

DBA_COLL_TYPES describes all named collection types (arrays, nested tables, object tables, and so on) in the database. Its columns are the same as those in "ALL_COLL_TYPES" on page 2-12.

DBA_CONS_COLUMNS

DBA_CONS_COLUMNS describes all columns in the database that are specified in constraint definitions. Its columns are the same as those in "ALL_CONS_COLUMNS" on page 2-13.

DBA_CONSTRAINTS

DBA_CONSTRAINTS describes all constraint definitions on all tables in the database. Its columns are the same as those in "ALL_CONSTRAINTS" on page 2-13.

DBA_CONTEXT

DBA_CONTEXT provides all context namespace information in the database. Its columns are the same as those in "ALL_CONTEXT" on page 2-15.

DBA_DATA_FILES

DBA_DATA_FILES describes database files.

Column	Datatype	NULL	Description
FILE_NAME	VARCHAR2(513)		Name of the database file

Column	Datatype	NULL	Description
FILE_ID	NUMBER	NOT NULL	ID of the database file
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace to which the file belongs
BYTES	NUMBER		Size of the file in bytes
BLOCKS	NUMBER	NOT NULL	Size of the file in Oracle blocks
STATUS	VARCHAR2(9)		File status: AVAILABLE or INVALID (INVALID means that the file number is not in use, for example, a file in a tablespace that was dropped)
RELATIVE_FNO	NUMBER		Relative file number
AUTOEXTENSIBLE	VARCHAR2(3)		Autoextensible indicator
MAXBYTES	NUMBER		Maximum file size in bytes
MAXBLOCKS	NUMBER		Maximum file size in blocks
INCREMENT_BY	NUMBER		Autoextension increment
USER_BYTES	NUMBER		Corresponding number of bytes
USER_BLOCKS	NUMBER		Number of blocks which can be used by the data

DBA_DB_LINKS

DBA_DB_LINKS describes all database links in the database. Its columns are the same as those in "ALL_DB_LINKS" on page 2-15.

DBA_DDL_LOCKS

DBA_DDL_LOCKS lists all DDL locks held in the database and all outstanding requests for a DDL lock.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session identifier
OWNER	VARCHAR2(30)		Owner of the lock
NAME	VARCHAR2(30)		Name of the lock
TYPE	VARCHAR2(40)		Lock type: CURSOR, TABLE/PROCEDURE/TYPE, BODY, TRIGGER, INDEX, CLUSTER
MODE_HELD	VARCHAR2(9)		Lock mode: NONE, NULL, SHARE, EXCLUSIVE
MODE_REQUESTED	VARCHAR2(9)		Lock request type: NONE, NULL, SHARE, EXCLUSIVE

DBA_DEPENDENCIES

DBA_DEPENDENCIES describes all dependencies in the database between procedures, packages, functions, package bodies, and triggers, including dependencies on views created without any database links. Its columns are the same as those in "ALL_DEPENDENCIES" on page 2-16.

DBA_DIMENSIONS

DBA_DIMENSIONS represents dimension objects. Its columns are the same as those in "ALL_DIMENSIONS" on page 2-17.

DBA_DIM_ATTRIBUTES

DBA_DIM_ATTRIBUTES describes the relationships between dimension levels and functionally dependent columns in the database. The level columns and the dependent column must be in the same table. This view's columns are the same as those in "ALL_DIM_ATTRIBUTES" on page 2-18.

DBA_DIM_CHILD_OF

DBA_DIM_CHILD_OF describes a hierarchical relationship of 1 to *n* between all the pairs of levels in the dimensions in the database. Its columns are the same as those in "ALL_DIM_CHILD_OF" on page 2-18.

DBA_DIM_HIERARCHIES

DBA_DIM_HIERARCHIES describes all the dimension hierarchies in the database. Its columns are the same as those in "ALL_DIM_HIERARCHIES" on page 2-19.

DBA_DIM_JOIN_KEY

DBA_DIM_JOIN_KEY describes all joins in the database between two dimension tables. The join is always specified between a parent dimension level column and a child column. This view's columns are the same as those in "ALL_DIM_JOIN_KEY" on page 2-19.

DBA_DIM_LEVELS

DBA_DIM_LEVELS describes all dimension levels in the database. All columns of a dimension level must come from the same relation. This view's columns are the same as those in "ALL_DIM_LEVELS" on page 2-20.

DBA_DIM_LEVEL_KEY

DBA_DIM_LEVEL_KEY describes the columns of all dimension levels in the database. This view's columns are the same as those in "ALL_DIM_LEVEL_KEY" on page 2-20.

DBA_DIRECTORIES

DBA_DIRECTORIES describes all directory objects in the database. Its columns are the same as those in "ALL_DIRECTORIES" on page 2-21.

DBA_DML_LOCKS

DBA_DML_LOCKS lists all DML locks held in the database and all outstanding requests for a DML lock.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session holding or acquiring the lock
OWNER	VARCHAR2(30)	NOT NULL	Owner of the lock
NAME	VARCHAR2(30)	NOT NULL	Name of the lock
MODE_HELD	VARCHAR2(13)		The type of lock held. The values are: ■ ROWS_S (SS): row share lock ■ ROW-X (SX): row exclusive lock ■ SHARE (S): share lock ■ S/ROW-X (SSX): exclusive lock ■ NONE: lock requested but not yet obtained

Column	Datatype	NULL	Description
MODE_REQUESTED	VARCHAR2(13)		Lock request type. The value are: ■ ROWS_S (SS): row share lock ■ ROW-X (SX): row exclusive lock ■ SHARE (S): share lock ■ S/ROW-X (SSX): exclusive lock ■ NONE: Lock identifier obtained; lock not held or requested
LAST_CONVERT	NUMBER		The last convert
BLOCKING_OTHERS	VARCHAR2(40)		Blocking others

DBA_ERRORS

DBA_ERRORS describes current errors on all stored objects (views, procedures, functions, packages, and package bodies) in the database. Its columns are the same as those in "ALL_ERRORS" on page 2-21.

DBA_EXP_FILES

DBA_EXP_FILES describes export files.

Column	Datatype	NULL	Description
EXP_VERSION	NUMBER(3)	NOT NULL	Version number of the export session
EXP_TYPE	VARCHAR2(11)		Type of export file: complete, cumulative, or incremental
FILE_NAME	VARCHAR2(100)	NOT NULL	Name of the export file
USER_NAME	VARCHAR2(30)	NOT NULL	Name of user who executed export
TIMESTAMP	DATE	NOT NULL	Timestamp of the export session

DBA_EXP_OBJECTS

DBA_EXP_OBJECTS describes objects that have been incrementally exported.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of exported object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of exported object

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(12)		Type of exported object
CUMULATIVE	DATE		Timestamp of last cumulative export
INCREMENTAL	DATE	NOT NULL	Timestamp of last incremental export
EXPORT_VERSION	NUMBER(3)	NOT NULL	The ID of the export session

DBA_EXP_VERSION

DBA_EXP_VERSION displays the version number of the last export session.

Column	Datatypes	NULL	Description
EXP_VERSION	NUMBER(3)	NOT NULL	Version number of the last export session

DBA_EXTENTS

DBA_EXTENTS describes the extents comprising all segments in the database.

Related view:

- USER_EXTENTS describes extents in segments associated with objects owned by the current user. This view does not display the OWNER, FILE_ID, BLOCK_ID, or RELATIVE_FNO columns.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the segment associated with the extent
SEGMENT_NAME	VARCHAR2(81)		Name of the segment associated with the extent
SEGMENT_TYPE	VARCHAR2(17)		Type of the segment: INDEX PARTITION, TABLE PARTITION
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the extent
EXTENT_ID	NUMBER	NOT NULL	Extent number in the segment
FILE_ID	NUMBER	NOT NULL	Name of the file containing the extent
BLOCK_ID	NUMBER	NOT NULL	Starting block number of the extent
BYTES	NUMBER		Size of the extent in bytes
BLOCKS	NUMBER	NOT NULL	Size of the extent in Oracle blocks
RELATIVE_FNO	NUMBER	NOT NULL	Relative file number of the first extent block

Column	Datatype	NULL	Description
PARTITION_NAME	VARCHAR2(30)		Object Partition Name (Set to NULL for non-partitioned objects)

DBA_FREE_SPACE

DBA_FREE_SPACE lists the free extents in all tablespaces.

Related view:

- USER_FREE_SPACE lists the free extents in tablespaces accessible to the current user.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the extent
FILE_ID	NUMBER	NOT NULL	ID number of the file containing the extent
BLOCK_ID	NUMBER	NOT NULL	Starting block number of the extent
BYTES	NUMBER		Size of the extent in bytes
BLOCKS	NUMBER	NOT NULL	Size of the extent in Oracle blocks
RELATIVE_FNO	NUMBER	NOT NULL	Relative file number of the first extent block

DBA_FREE_SPACE_COALESCED

DBA_FREE_SPACE_COALESCED contains statistics on coalesced space in tablespaces.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of tablespace
TOTAL_EXTENTS	NUMBER		Total number of free extents in tablespace
EXTENTS_COALESCED	NUMBER		Total number of coalesced free extents in tablespace
PERCENT_EXTENTS_COALESCED	NUMBER		Percentage of coalesced free extents in tablespace
TOTAL_BYTES	NUMBER		Total number of free bytes in tablespace
BYTES_COALESCED	NUMBER		Total number of coalesced free bytes in tablespace
TOTAL_BLOCKS	NUMBER		Total number of free Oracle blocks in tablespace
BLOCKS_COALESCED	NUMBER		Total number of coalesced free Oracle blocks in tablespace

Column	Datatype	NULL	Description
PERCENT_BLOCKS_COALESCED	NUMBER		Percentage of coalesced free Oracle blocks in tablespace

DBA_IND_COLUMNS

DBA_IND_COLUMNS describes the columns of all the indexes on all tables and clusters in the database. Its columns are the same as those in "ALL_IND_COLUMNS" on page 2-22.

DBA_IND_EXPRESSIONS

DBA_IND_EXPRESSIONS lists expressions of function-based indexes on all tables and clusters in the database. Its columns are the same as those in "ALL_IND_EXPRESSIONS" on page 2-23.

DBA_IND_PARTITIONS

DBA_IND_PARTITIONS describes, for each index partition in the database, the partition-level partitioning information, the storage parameters for the partition, and various partition statistics collected by ANALYZE statements. Its columns are the same as those in "ALL_IND_PARTITIONS" on page 2-23.

DBA_IND_SUBPARTITIONS

DBA_IND_SUBPARTITIONS describes, for each index subpartition in the database, the partition-level partitioning information, the storage parameters for the subpartition, and various partition statistics collected by ANALYZE statements. Its columns are the same as those in "ALL_IND_SUBPARTITIONS" on page 2-25.

DBA_INDEXES

DBA_INDEXES describes all indexes in the database. To gather statistics for this view, use the SQL command ANALYZE. This view supports parallel partitioned index scans. Its columns are the same as those in "ALL_INDEXES" on page 2-27.

DBA_INDEXTYPES

DBA_INDEXTYPES describes all indextypes in the database. Its columns are the same as those in "ALL_INDEXTYPES" on page 2-30.

DBA_INDEXTYPE_OPERATORS

DBA_INDEXTYPE_OPERATORS lists all the operators supported by indextypes in the database. Its columns are the same as those in "ALL_INDEXTYPE_ OPERATORS" on page 2-31.

DBA_INTERNAL_TRIGGERS

DBA_INTERNAL_TRIGGERS describes internal triggers on all tables in the database. Its columns are the same as those in "ALL_INTERNAL_TRIGGERS" on page 2-31.

DBA_JOBS

DBA_JOBS describes all jobs in the database. Its columns are the same as those in "ALL_JOBS" on page 2-32.

> **See Also:** *Oracle8i Administrator's Guide* for more information on jobs.

DBA_JOBS_RUNNING

DBA_JOBS_RUNNING lists all jobs in the database that are currently running.

Column	Datatype	NULL	Description
SID	NUMBER		Identifier of process that is executing the job. See "V$LOCK".
JOB	NUMBER		Identifier of job. This job is currently executing.
FAILURES	NUMBER		Number of times this job started and failed since its last success
LAST_DATE	DATE		Date that this job last successfully executed
LAST_SEC	VARCHAR2(8)		Same as LAST_DATE. This is when the last successful execution started.
THIS_DATE	DATE		Date that this job started executing
THIS_SEC	VARCHAR2(8)		Same as THIS_DATE. This is when the last successful execution started.

Column	Datatype	NULL	Description
INSTANCE	NUMBER		Indicates which instance can execute or is executing the job; the default is 0

DBA_KGLLOCK

DBA_KGLLOCK lists all the locks and pins held on KGL objects (objects in the Kernel Generic Library cache).

Column	Datatype	NULL	Description
kgllkuse	RAW(4)		Address of the user session that holds the lock or pin.
kgllkhdl	RAW(4)		Address of the handle for the KGL object
kgllkmod	NUMBER		Current mode of the lock or pin
kgllkreq	NUMBER		Mode in which the lock or pin was requested
kgllktype	VARCHAR2(4)		Whether this is a lock or a pin

DBA_LIBRARIES

DBA_LIBRARIES describes all the libraries in the database. Its columns are the same as those in "ALL_LIBRARIES" on page 2-33.

DBA_LOBS

DBA_LOBS displays the BLOBs and CLOBs contained in all tables in the database. BFILEs are stored outside the database, so they are not described by this view. This view's columns are the same as those in "ALL_LOBS" on page 2-33.

DBA_LOB_PARTITIONS

DBA_LOB_PARTITIONS displays all LOB partitions in the database. Its columns are the same as those in "ALL_LOB_PARTITIONS" on page 2-34.

DBA_LOB_SUBPARTITIONS

DBA_LOB_SUBPARTITIONS displays partition-level attributes of all LOB data subpartitions in the database. Its columns are the same as those in "ALL_LOB_SUBPARTITIONS" on page 2-35.

DBA_LOCK_INTERNAL

DBA_LOCK_INTERNAL displays a row for each lock or latch that is being held, and one row for each outstanding request for a lock or latch.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session holding or acquiring the lock
LOCK_TYPE	VARCHAR2(56)		Lock type. For a listing of lock types, see Appendix B, "Oracle Enqueue Names".
MODE HELD	VARCHAR2(40)		Lock mode
MODE REQUESTED	VARCHAR2(40)		Lock mode requested
LOCK_ID1	VARCHAR2(1130)		Type-specific lock identifier, part 1
LOCK_ID2	VARCHAR2(40)		Type-specific lock identifier, part 2

DBA_LOCKS

DBA_LOCKS lists all locks or latches held in the database, and all outstanding requests for a lock or latch.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session holding or acquiring the lock
LOCK_TYPE	VARCHAR2(26)		Lock type. For a listing of lock types, see Appendix B, "Oracle Enqueue Names".
MODE HELD	VARCHAR2(40)		Lock mode
MODE REQUESTED	VARCHAR2(40)		Lock mode requested
LOCK_ID1	VARCHAR2(40)		Type-specific lock identifier, part 1
LOCK_ID2	VARCHAR2(40)		Type-specific lock identifier, part 2
LAST_CONVERT	NUMBER		The last convert
BLOCKING_OTHERS	VARCHAR2(40)		Whether the lock is currently blocking others

DBA_METHOD_PARAMS

DBA_METHOD_PARAMS describes all method parameters of all types in the database. Its columns are the same as those in "ALL_METHOD_PARAMS" on page 2-37.

DBA_METHOD_RESULTS

DBA_METHOD_RESULTS describes all method results of all types in the database. Its columns are the same as those in "ALL_METHOD_RESULTS" on page 2-37.

DBA_MVIEW_AGGREGATES

DBA_MVIEW_AGGREGATES describes the grouping functions (aggregated measures) that appear in the SELECT list of all aggregated materialized view in the database. Its columns are the same as those in "ALL_MVIEW_AGGREGATES" on page 2-38.

DBA_MVIEW_ANALYSIS

DBA_MVIEW_ANALYSIS describes all materialized views in the database that potentially support query rewrite and that provide additional information for analysis by applications. Its columns are the same as those in "ALL_MVIEW_ANALYSIS" on page 2-39.

Note: This view excludes materialized views that reference remote tables or that include references to non-static values such as SYSDATE or USER. This view also excludes materialized views that were created as snapshots prior to Oracle8*i* and that were never altered to enable query rewrite.

DBA_MVIEW_DETAIL_RELATIONS

DBA_MVIEW_DETAIL_RELATIONS represents the named detail relations that are either in the FROM list of a materialized view, or that are indirectly referenced through views in the FROM list. Its columns are the same as those in "ALL_MVIEW_DETAIL_RELATIONS" on page 2-41.

DBA_MVIEW_JOINS

DBA_MVIEW_JOINS describes a join between two columns in the WHERE clause of a subquery that defines a materialized view. Its columns are the same as those in "ALL_MVIEW_JOINS" on page 2-41.

DBA_MVIEW_KEYS

DBA_MVIEW_KEYS describes the columns or expressions in the SELECT list upon which materialized views in the database are based. Its columns are the same as those in "ALL_MVIEW_KEYS" on page 2-42.

DBA_MVIEWS

DBA_MVIEWS describes all materialized views in the database. Its columns are the same as those in "ALL_MVIEWS" on page 2-43.

DBA_NESTED_TABLES

DBA_NESTED_TABLES describes all nested tables contained in all tables in the database. Its columns are the same as those in "ALL_NESTED_TABLES" on page 2-46.

DBA_OBJECT_SIZE

DBA_OBJECT_SIZE lists the sizes, in bytes, of various PL/SQL objects.

Related view:

USER_OBJECT_SIZE lists the size of PL/SQL objects owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of the object: TYPE, TYPE BODY, TABLE, VIEW, SYNONYM, SEQUENCE, PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, JAVA SOURCE, JAVA CLASS or JAVA RESOURCE
SOURCE_SIZE	NUMBER		Size of the source in bytes. Must be in memory during compilation, or dynamic recompilation
PARSED_SIZE	NUMBER		Size of the parsed form of the object, in bytes. Must be in memory when an object is being compiled that references this object
CODE_SIZE	NUMBER		Code size, in bytes. Must be in memory when this object is executing
ERROR_SIZE	NUMBER		Size of error messages, in bytes. In memory during the compilation of the object when there are compilation errors

DBA_OBJECT_TABLES

DBA_OBJECT_TABLES describes all object tables in the database. Its columns are the same as those in "ALL_OBJECT_TABLES" on page 2-47.

DBA_OBJECTS

DBA_OBJECTS describes all objects in the database. Its columns are the same as those in "ALL_OBJECTS" on page 2-49.

DBA_OBJ_AUDIT_OPTS

DBA_OBJ_AUDIT_OPTS describes auditing options on all objects.

Related view:

- USER_OBJ_AUDIT_OPTS describes auditing options on all objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the object
OBJECT_TYPE	VARCHAR2(9)		Type of the object
ALT	VARCHAR2(3)		Auditing ALTER WHENEVER SUCCESSFUL/UNSUCCESSFUL
AUD	VARCHAR2(3)		Auditing AUDIT WHENEVER SUCCESSFUL/UNSUCCESSFUL
COM	VARCHAR2(3)		Auditing COMMENT WHENEVER SUCCESSFUL/UNSUCCESSFUL
DEL	VARCHAR2(3)		Auditing DELETE WHENEVER SUCCESSFUL/UNSUCCESSFUL
GRA	VARCHAR2(3)		Auditing GRANT WHENEVER SUCCESSFUL/UNSUCCESSFUL
IND	VARCHAR2(3)		Auditing INDEX WHENEVER SUCCESSFUL/UNSUCCESSFUL
INS	VARCHAR2(3)		Auditing INSERT WHENEVER SUCCESSFUL/UNSUCCESSFUL
LOC	VARCHAR2(3)		Auditing LOCK WHENEVER SUCCESSFUL/UNSUCCESSFUL
REN	VARCHAR2(3)		Auditing RENAME WHENEVER SUCCESSFUL/UNSUCCESSFUL
SEL	VARCHAR2(3)		Auditing SELECT WHENEVER SUCCESSFUL/UNSUCCESSFUL
UPD	VARCHAR2(3)		Auditing UPDATE WHENEVER SUCCESSFUL/UNSUCCESSFUL
REF	VARCHAR2(3)		Auditing REFERENCE WHENEVER SUCCESSFUL/UNSUCCESSFUL (not used)
EXE	VARCHAR2(3)		Auditing EXE WHENEVER SUCCESSFUL/UNSUCCESSFUL

Column	Datatype	NULL	Description
CRE	VARCHAR2(3)		Auditing CRE WHENEVER SUCCESSFUL/UNSUCCESSFUL
REA	VARCHAR2(3)		Auditing REA WHENEVER SUCCESSFUL/UNSUCCESSFUL
WRI	VARCHAR2(3)		Auditing WRI WHENEVER SUCCESSFUL/UNSUCCESSFUL

DBA_OPANCILLARY

DBA_OPANCILLARY provides ancillary information for all operator bindings in the database. Its columns are the same as those in "ALL_OPANCILLARY" on page 2-50.

DBA_OPARGUMENTS

DBA_OPARGUMENTS provides argument information for all operator bindings in the database. Its columns are the same as those in "ALL_OPARGUMENTS" on page 2-51.

DBA_OPBINDINGS

DBA_OPBINDINGS describes bindings of all operators in the database. Its columns are the same as those in "ALL_OPBINDINGS" on page 2-51.

DBA_OPERATORS

DBA_OPERATORS describes all operators in the database. Its columns are the same as those in "ALL_OPERATORS" on page 2-52.

DBA_ORPHAN_KEY_TABLE

DBA_ORPHAN_KEY_TABLE reports key values from indexes where the underlying base table has block corruptions. To create the view, run the DBMS_REPAIR.ADMIN_TABLES procedure. To populate the orphan key table for an index, run the DBMS_REPAIR.DUMP_ORPHAN_KEYS procedure on the index. For each key in the index that points to a corrupt data block, Oracle inserts a row into the orphan key table.

Column	Datatype	Null	Description
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema name of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index

Column	Datatype	Null	Description
IPART_NAME	VARCHAR2(30)	NULL	Name of the index partition or subpartition
INDEX_ID	NUMBER	NOT NULL	Dictionary object ID of the index
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the base table of the index
PART_NAME	VARCHAR2(30)	NULL	Name of the base table partition or subpartition
TABLE_ID	NUMBER	NOT NULL	Dictionary object ID of the base table
KEYDATA	UROWID	NOT NULL	Physical rowid of the corrupt data row
KEY	UROWID	NOT NULL	Key values for the index entry
DUMP_TIMESTAMP	DATE	NOT NULL	Timestamp when the entry was made into the orphan key table

DBA_OUTLINE_HINTS

DBA_OUTLINE_HINTS describes the set of hints that make up the all the outlines in the database. Its columns are the same as those in "ALL_OUTLINE_HINTS" on page 2-53.

DBA_OUTLINES

DBA_OUTLINES describes all outlines in the database. Its columns are the same as those in "ALL_OUTLINES" on page 2-53.

DBA_PART_COL_STATISTICS

DBA_PART_COL_STATISTICS provides column statistics and histogram information for partitions of all tables in the database. Its columns are the same as those in "ALL_PART_COL_STATISTICS" on page 2-54.

DBA_PART_HISTOGRAMS

DBA_PART_HISTOGRAMS provides the histogram data (end-points per histogram) for histograms on all table partitions in the database. Its columns are the same as those in "ALL_PART_HISTOGRAMS" on page 2-55.

DBA_PART_INDEXES

DBA_PART_INDEXES provides object-level partitioning information for all partitioned indexes in the database. Its columns are the same as those in "ALL_PART_INDEXES" on page 2-56.

DBA_PART_KEY_COLUMNS

DBA_PART_KEY_COLUMNS describes the partitioning key columns for all partitioned objects in the database. Its columns are the same as those in "ALL_PART_KEY_COLUMNS" on page 2-57.

DBA_PART_LOBS

DBA_PART_LOBS provides table-level information for all partitioned LOBs in the database, including default attributes for LOB data partitions. Its columns are the same as those in "ALL_PART_LOBS" on page 2-58.

DBA_PART_TABLES

DBA_PART_TABLES provides object-level partitioning information for all partitioned tables in the database. Its columns are the same as those in "ALL_PART_TABLES" on page 2-59.

DBA_PARTIAL_DROP_TABS

DBA_PARTIAL_DROP_TABS describes all tables in the database that have partially completed DROP COLUMN operations. Its columns are the same as those in "ALL_PARTIAL_DROP_TABS" on page 2-61.

DBA_PENDING_TRANSACTIONS

DBA_PENDING_TRANSACTIONS describes unresolved transactions (either due to failure or if the co-ordinator has not sent a commit/rollback).

Column	Datatype	NULL	Description
FORMATID	NUMBER		The format identifier of the transaction identifier
GLOBALID	RAW(64)		The global part (gtrid) of the transaction identifier
BRANCHID	RAW(64)		The branch qualifier (bqual) of the transaction identifier

DBA_POLICIES

DBA_POLICIES describes all security policies in the database. Its columns are the same as those in "ALL_POLICIES" on page 2-61.

DBA_PRIV_AUDIT_OPTS

DBA_PRIV_AUDIT_OPTS describes current system privileges being audited across the system and by user.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2(30)		User name if by user auditing. ANY CLIENT if access by a proxy on behalf of a client is being audited NULL for system-wide auditing
PROXY_NAME	VARCHAR2(30)		The name of the proxy user which is performing an operation for the client. NULL if the client is performing the operation directly
PRIVILEGE	VARCHAR2(40)	NOT NULL	Name of the system privilege being audited
SUCCESS	VARCHAR2(10)		Mode for WHENEVER SUCCESSFUL system auditing
FAILURE	VARCHAR2(10)		Mode for WHENEVER NOT SUCCESSFUL system auditing

DBA_PROFILES

DBA_PROFILES displays all profiles and their limits.

Column	Datatype	NULL	Description
PROFILE	VARCHAR2(30)	NOT NULL	Profile name
RESOURCE_NAME	VARCHAR2(32)	NOT NULL	Resource name
RESOURCE_TYPE	VARCHAR2(8)		Indicates whether the resource profile is a KERNEL or a PASSWORD parameter
LIMIT	VARCHAR2(40)		Limit placed on this resource for this profile

DBA_QUEUE_SCHEDULES

DBA_QUEUE_SCHEDULES describes the current schedules for propagating messages.

Related view:

- USER_QUEUE_SCHEDULES describes current schedules related to the current user. This view does not display the SCHEMA column.

Column	Datatype	NULL	Description
SCHEMA	VARCHAR2(30)	NOT NULL	The schema
QNAME	VARCHAR2(30)	NOT NULL	Source queue name
DESTINATION	VARCHAR2(128)	NOT NULL	Destination name, currently limited to be a DBLINK name
START_DATE	DATE		Date to start propagation in the default date format
START_TIME	VARCHAR2(8)		Time of day at which to start propagation in HH:MI:SS format
PROPAGATION_WINDOW	NUMBER		Duration in seconds for the propagation window
NEXT_TIME	VARCHAR2(200)		Function to compute the start of the next propagation window
LATENCY	NUMBER		Maximum wait time to propagate a message during the propagation window
SCHEDULE_DISABLED	VARCHAR2(1)		N if enabled; Y if disabled and schedule will not be executed
PROCESS_NAME	VARCHAR2(8)		The name of the process executing the schedule. NULL if not currently executing
SESSION_ID	NUMBER		The session ID (SID, SERIAL#) of the job executing this schedule. NULL if not currently executing
INSTANCE	NUMBER		The OPS instance number executing the schedule
LAST_RUN_DATE	DATE		The date on the last successful execution
LAST_RUN_TIME	VARCHAR2(8)		The time of the last successful execution in HH:MI:SS format
CURRENT_START_DATE	DATE		The date at which the current window of this schedule was started
CURRENT_START_TIME	VARCHAR2(8)		The time of day at which the current window of this schedule was started in HH:MI:SS format
NEXT_RUN_DATE	DATE		The date at which the next window of this schedule will be started
NEXT_RUN_TIME	VARCHAR2(8)		The time of day at which the next window of this schedule will be started in HH:MI:SS format
TOTAL_TIME	NUMBER		The total time, in seconds, spent by the system in executing this schedule
TOTAL_NUMBER	NUMBER		The total number of messages propagated in this schedule
TOTAL_BYTES	NUMBER		The total number of bytes propagated in this schedule

Column	Datatype	NULL	Description
MAX_NUMBER	NUMBER		The maximum number of messages propagated in a propagation window
MAX_BYTES	NUMBER		The maximum number of bytes propagated in a propagation window
AVG_NUMBER	NUMBER		The average number of messages propagated in a propagation window
AVG_SIZE	NUMBER		The average size of a propagated message in bytes
AVG_TIME	NUMBER		The average time, in seconds, to propagate a message
FAILURES	NUMBER		The number of times the execution failed. If 16, the schedule will be disabled
LAST_ERROR_DATE	DATE		The date of the last unsuccessful execution
LAST_ERROR_TIME	VARCHAR2(8)		The time of the last unsuccessful execution
LAST_ERROR_MSG	VARCHAR2(4000)		The error number and error message text of the last unsuccessful execution

DBA_QUEUE_TABLES

DBA_QUEUE_TABLES describes the queues in all of the queue tables created in the database. Its columns are the same as those in "ALL_QUEUE_TABLES" on page 2-62.

> **See Also:** *Oracle8i Application Developer's Guide - Advanced Queuing* for more information about this view and Advanced Queuing.

DBA_QUEUES

DBA_QUEUES describes the operational characteristics of every queue in a database. Its columns are the same as those in "ALL_QUEUES" on page 2-63.

> **See Also:** *Oracle8i Application Developer's Guide - Advanced Queuing* for more information about this view and Advanced Queuing.

DBA_RCHILD

DBA_RCHILD lists all the children in any refresh group.

Column	Datatype	NULL	Description
REFGROUP	NUMBER		Internal identifier of refresh group
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
TYPE#	VARCHAR2(30)		Type of the object in the refresh group

DBA_REFRESH

DBA_REFRESH describes all refresh groups in the database. Its columns are the same as those in "ALL_REFRESH" on page 2-64.

DBA_REFRESH_CHILDREN

DBA_REFRESH_CHILDREN lists all of the objects in all refresh groups in the database. Its columns are the same as those in "ALL_REFRESH_CHILDREN" on page 2-65.

DBA_REFS

DBA_REFS describes the REF columns and REF attributes in object type columns of all the objects in the database. Its columns are the same as those in "ALL_REFS" on page 2-66.

DBA_REGISTERED_SNAPSHOT_GROUPS

DBA_REGISTERED_SNAPSHOT_GROUPS lists all the snapshot repgroups at this site. This view is available only if you have installed the Oracle replication packages.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the snapshot replication group
SNAPSHOT_SITE	VARCHAR2(128)		Site of the master of the snapshot repgroup
GROUP_COMMENT	VARCHAR2(80)		Description of the snapshot repgroup
VERSION	VARCHAR2(8)		Version of the snapshot repgroup
FNAME	VARCHAR2(30)		Name of the flavor of the snapshot object group

DBA_REGISTERED_SNAPSHOTS

DBA_REGISTERED_SNAPSHOTS describes all registered snapshots in the database. Its columns are the same as those in "ALL_REGISTERED_SNAPSHOTS" on page 2-67.

DBA_REPAIR_TABLE

DBA_REPAIR_TABLE describes any corruptions found by the DBMS_REPAIR.CHECK_OBJECT procedure. This information is used by the DBMS_REPAIR.FIX_CORRUPT_BLOCKS procedure on execution. To create this view, first run the DBMS_REPAIR.ADMIN_TABLES procedure. To populate the resulting repair table for an object, run the DBMS_REPAIR.CHECK_OBJECT procedure on the object.

Note: The table created by the DBMS_REPAIR.ADMIN_TABLES procedure is called REPAIR TABLE by default. If you specify a different name, this view will have the name you specify, preceded by "DBA_REPAIR_".

Column	Datatype	NULL	Description
OBJECT_ID	NUMBER	NOT NULL	Dictionary object number of the object with the corruption
TABLESPACE_ID	NUMBER	NOT NULL	Tablespace number of the corrupt object
RELATIVE_FILE_ID	NUMBER)	NOT NULL	Relative file number of the corrupt object
BLOCK_ID	NUMBER	NOT NULL	Block number of the corruption
CORRUPT_TYPE	NUMBER	NOT NULL	Type of corruption encountered
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema of the corrupt object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the corrupt object
BASEOBJECT_NAME	VARCHAR2(30)	NULL	If the object is an index, the name of its base table
PARTITION_NAME	VARCHAR2(30)	NULL	Partition or subpartition name, if applicable
CORRUPT_DESCRIPTION	VARCHAR2(200)	NULL	Description of corruption
REPAIR_DESCRIPTION	VARCHAR2(200)	NULL	Description of repair action
MARKED_CORRUPT	VARCHAR2(10)	NOT NULL	Whether the block is marked corrupt (TRUE | FALSE)

Column	Datatype	NULL	Description
CHECK_TIMESTAMP	DATE	NOT NULL	Date on which this row was insert into the repair table
FIX_TIMESTAMP	DATE	NULL	Date on which the block was modified by the FIX_CORRUPT_BLOCKS procedure, if applicable
REFORMAT_TIMESTAMP	DATE	NULL	Reserved for future use.

DBA_RGROUP

DBA_RGROUP lists all refresh groups.

Column	Datatype	NULL	Description
REFGROUP	NUMBER		Internal identifier of refresh group
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		Y or N; if Y, then destroy the refresh group when its last item is removed
PUSH_DEFERRED_RPC	VARCHAR2(1)		Y or N; if Y, then push changes from snapshot to master before refresh
REFRESH_AFTER_ERRORS	VARCHAR2(1)		Y or N; if Y, proceed with refresh despite error when pushing deferred RPC's
ROLLBACK_SEG	VARCHAR2(30)		Name of the rollback segment to use while refreshing
JOB	NUMBER	NOT NULL	Identifier of job used to refresh the group automatically
PURGE_OPTION	NUMBER(38)		The method for purging the transaction queue after each push. 1=quick purge option; 2=precise purge option
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

DBA_ROLES

DBA_ROLES lists all roles that exist in the database.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Role name
PASSWORD_REQUIRED	VARCHAR2(8)		Indicates if the role requires a password to be enabled

DBA_ROLE_PRIVS

DBA_ROLE_PRIVS lists roles granted to users and roles.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)		Grantee name, user or role receiving the grant
GRANTED_ROLE	VARCHAR2(30)	NOT NULL	Granted role name
ADMIN_OPTION	VARCHAR2(3)		Whether the grant was with the ADMIN option: YES | NO
DEFAULT_ROLE	VARCHAR2(3)		Whether the role is designated as a DEFAULT ROLE for the user: YES | NO

DBA_ROLLBACK_SEGS

DBA_ROLLBACK_SEGS describes rollback segments.

Column	Datatype	NULL	Description
SEGMENT_NAME	VARCHAR2(30)	NOT NULL	Name of the rollback segment
OWNER	VARCHAR2(6)		Owner of the rollback segment
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the rollback segment
SEGMENT_ID	NUMBER	NOT NULL	ID number of the rollback segment
FILE_ID	NUMBER	NOT NULL	ID number of the file containing the segment head
BLOCK_ID	NUMBER	NOT NULL	ID number of the block containing the segment header
INITIAL_EXTENT	NUMBER		Initial extent size in bytes
NEXT_EXTENT	NUMBER		Secondary extent size in bytes
MIN_EXTENTS	NUMBER	NOT NULL	Minimum number of extents
MAX_EXTENTS	NUMBER	NOT NULL	Maximum number of extent
PCT_INCREASE	NUMBER	NOT NULL	Percent increase for extent size
STATUS	VARCHAR2(16)		Rollback segment status
INSTANCE_NUM	VARCHAR2(40)		Rollback segment owning parallel server instance number
RELATIVE_FNO	NUMBER	NOT NULL	Relative file number of the segment header

DBA_RSRC_CONSUMER_GROUP_PRIVS

DBA_RSRC_CONSUMER_GROUP_PRIVS lists all resource consumer groups and the users and roles assigned to them. The grant referred to in this view and the related

view is the grant of the SWITCH_CONSUMER_GROUP object privilege, which is granted using the DBMS_RESOURCE_MANAGER_PRIVS package. (This privilege is not granted through the SQL GRANT statement.)

> **See Also:** *Oracle8i Supplied PL/SQL Packages Reference* for more information on consumer group privileges.

Related view:

- USER_RSRC_CONSUMER_GROUP_PRIVS lists all resource consumer groups to which the current user is assigned. This view does not display the GRANTEE column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	User or role receiving the grant
GRANTED_GROUP	VARCHAR2(30)		Granted consumer group name
GRANT_OPTION	VARCHAR2(3)		Whether grant was with the GRANT option
INITIAL_GROUP	VARCHAR2(3)		Whether consumer group is designated as the default for this user or role

DBA_RSRC_CONSUMER_GROUPS

DBA_RSRC_CONSUMER_GROUPS lists all resource consumer groups in the database.

Column	Datatype	NULL	Description
CONSUMER_GROUP	VARCHAR2(30)		Name of the consumer group
CPU_METHOD	VARCHAR2(30)		CPU resource allocation method for the consumer group
COMMENTS	VARCHAR2(2000)		Text comment on the consumer group
STATUS	VARCHAR2(30)		Status of the consumer group: PENDING if the group is part of the pending area, ACTIVE otherwise.
MANDATORY	VARCHAR2(3)		Whether the consumer group is mandatory: YES | NO

DBA_RSRC_MANAGER_SYSTEM_PRIVS

DBA_RSRC_MANAGER_SYSTEM_PRIVS lists all the users and roles that have been granted the ADMINISTER_RESOURCE_MANAGER system privilege, which is granted using the DBMS_RESOURCE_MANAGER_PRIVS package. (This privilege is not granted through the SQL GRANT statement.)

See Also: *Oracle8i Supplied PL/SQL Packages Reference* for more information on consumer group privileges.

Related view:

- USER_RSRC_MANAGER_SYSTEM_PRIVS lists users who are granted system privilege for the DBMS_RESOURCE_MANAGER package. This view does not display the GRANTEE column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	User or role receiving the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Name of the system privilege
ADMIN_OPTION	VARCHAR2(3)		Whether the grant was with the ADMIN option

DBA_RSRC_PLANS

DBA_RSRC_PLANS lists all resource plans in the database. For a list of currently active plans, see "V$RSRC_PLAN" on page 3-89.

See Also:

- *Oracle8i Administrator's Guide* for information on resource plans in general.
- *Oracle8i Supplied PL/SQL Packages Reference* for more information on creating resource plans with the DBMS_RESOURCE_MANAGER package.

Column	Datatype	NULL	Description
PLAN	VARCHAR2(30)		Name of the resource plan
NUM_PLAN_DIRECTIVES	NUMBER		Number of plan directives for the plan
CPU_METHOD	VARCHAR2(30)		CPU resource allocation method for the plan
MAX_ACTIVE_SESS_TARGET_MTH	VARCHAR2(30)		Reserved for future use.
PARALLEL_DEGREE_LIMIT_MTH	VARCHAR2(30)		Parallel degree limit resource allocation method for the plan
COMMENTS	VARCHAR2(2000)		Text comment on the plan
STATUS	VARCHAR2(30)		Status of the plan: PENDING if it is part of the pending area, ACTIVE otherwise

Column	Datatype	NULL	Description
MANDATORY	VARCHAR2(3)		Whether the plan is mandatory

DBA_RSRC_PLAN_DIRECTIVES

DBA_RSRC_PLAN_DIRECTIVES lists all resource plan directives in the database.

See Also:

- *Oracle8i Administrator's Guide* for information on resource plans in general.
- *Oracle8i Supplied PL/SQL Packages Reference* for more information on creating resource plans with the DBMS_RESOURCE_MANAGER package.

Column	Datatype	NULL	Description
PLAN	VARCHAR2(30)		Name of the plan to which this directive belongs
GROUP_OR_SUBPLAN	VARCHAR2(30)		Name of the consumer group or subplan referred to
TYPE	VARCHAR2(14)		Whether GROUP_OR_SUBPLAN refers to a consumer group or plan
CPU_P1	NUMBER		First parameter for the CPU resource allocation method
CPU_P2	NUMBER		Second parameter for the CPU resource allocation method
CPU_P3	NUMBER		Third parameter for the CPU resource allocation method
CPU_P4	NUMBER		Fourth parameter for the CPU resource allocation method
CPU_P5	NUMBER		Fifth parameter for the CPU resource allocation method
CPU_P6	NUMBER		Sixth parameter for the CPU resource allocation method
CPU_P7	NUMBER		Seventh parameter for the CPU resource allocation method
CPU_P8	NUMBER		Eighth parameter for the CPU resource allocation method
MAX_ACTIVE_SESS_TARGET_P1	NUMBER		Reserved for future use.
PARALLEL_DEGREE_LIMIT_P1	NUMBER		First parameter for the parallel degree limit resource allocation method
COMMENTS	VARCHAR2(2000)		Text comment on the plan directive
STATUS	VARCHAR2(30)		Status of the directive: PENDING if it is part of the pending area, ACTIVE otherwise
MANDATORY	VARCHAR2(3)		Whether the plan is mandatory: YES | NO

DBA_RULESETS

DBA_RULESETS provides information about rulesets.

Related view:

- USER_RULESETS provides information about rulesets. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	The user owning the ruleset
RULESET_NAME	VARCHAR2(30)	NOT NULL	The name of the ruleset
RULESET_STORAGE_TABLE	VARCHAR2(61)		The table in which the ruleset is stored (will be schema.table)
BASE_TABLE	VARCHAR2(61)		The table on which the rules are defined (will be schema.table)

DBA_SEGMENTS

DBA_SEGMENTS describes storage allocated for all database segments.

Related view:

- USER_SEGMENTS describes storage allocated for database segments owned by the current user's objects. This view does not display the OWNER, HEADER_FILE, HEADER_BLOCK, or RELATIVE_FNO columns.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Username of the segment owner
SEGMENT_NAME	VARCHAR2(81)		Name, if any, of the segment
PARTITION_NAME	VARCHAR2(30)		Object Partition Name (Set to NULL for non-partitioned objects)
SEGMENT_TYPE	VARCHAR2(17)		Type of segment: INDEX PARTITION, TABLE PARTITION, TABLE, CLUSTER, INDEX, ROLLBACK, DEFERRED ROLLBACK, TEMPORARY, CACHE, LOBSEGMENT and LOBINDEX
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the segment
HEADER_FILE	NUMBER		ID of the file containing the segment header
HEADER_BLOCK	NUMBER		ID of the block containing the segment header
BYTES	NUMBER		Size in bytes, of the segment

Column	Datatype	NULL	Description
BLOCKS	NUMBER		Size, in Oracle blocks, of the segment
EXTENTS	NUMBER		Number of extents allocated to the segment
INITIAL_EXTENT	NUMBER		Size in bytes requested for the initial extent of the segment at create time. (Oracle rounds the extent size to multiples of 5 blocks if the requested size is greater than 5 blocks.)
NEXT_EXTENT	NUMBER		Size in bytes of the next extent to be allocated to the segment
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percent by which to increase the size of the next extent to be allocated
FREELISTS	NUMBER		Number of process freelists allocated to this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment
RELATIVE_FNO	NUMBER		Relative file number of the segment header
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the object

DBA_SEQUENCES

DBA_SEQUENCES describes all sequences in the database. Its columns are the same as those in "ALL_SEQUENCES" on page 2-68.

DBA_SNAPSHOT_LOGS

DBA_SNAPSHOT_LOGS describes all snapshot logs in the database. Its columns are the same as those in "ALL_SNAPSHOT_LOGS" on page 2-68.

DBA_SNAPSHOT_LOG_FILTER_COLS

DBA_SNAPSHOT_LOG_FILTER_COLS lists all filter columns (excluding primary key columns) being logged in the snapshot logs.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the master table being logged
NAME	VARCHAR2(30)	NOT NULL	Name of the master table being logged
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Filter column being logged

DBA_SNAPSHOT_REFRESH_TIMES

DBA_SNAPSHOT_REFRESH_TIMES lists refresh times of all snapshots in the database. Its columns are the same as those in "ALL_SNAPSHOT_REFRESH_TIMES" on page 2-69.

DBA_SNAPSHOTS

DBA_SNAPSHOTS describes all snapshots in the database. Its columns are the same as those in "ALL_SNAPSHOTS" on page 2-69.

DBA_SOURCE

DBA_SOURCE describes the text source of all stored objects in the database. Its columns are the same as those in "ALL_SOURCE" on page 2-71.

DBA_STMT_AUDIT_OPTS

DBA_STMT_AUDIT_OPTS describes current system auditing options across the system and by user.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2(30)		User name if by user auditing. ANY CLIENT if access by a proxy on behalf of a client is being audited NULL for system-wide auditing
PROXY_NAME	VARCHAR2(30)		The name of the proxy user which is performing an operation for the client. NULL if the client is performing the operation directly
AUDIT_OPTION	VARCHAR2(40)	NOT NULL	Name of the system auditing option
SUCCESS	VARCHAR2(10)		Mode for WHENEVER SUCCESSFUL system auditing
FAILURE	VARCHAR2(10)		Mode for WHENEVER NOT SUCCESSFUL system auditing

DBA_SUBPART_COL_STATISTICS

DBA_SUBPART_COL_STATISTICS provides column statistics and histogram information for all subpartitions in the database. Its columns are the same as those in "ALL_SUBPART_COL_STATISTICS" on page 2-72.

DBA_SUBPART_HISTOGRAMS

DBA_SUBPART_HISTOGRAMS lists actual histogram data (end-points per histogram) for histograms on all table subpartitions in the database. Its columns are the same as those in "ALL_SUBPART_HISTOGRAMS" on page 2-72.

DBA_SUBPART_KEY_COLUMNS

DBA_SUBPART_KEY_COLUMNS lists subpartitioning key columns for all composite-partitioned tables (and local indexes on composite-partitioned tables) in the database. Its columns are the same as those in "ALL_SUBPART_KEY_COLUMNS" on page 2-73.

DBA_SYNONYMS

DBA_SYNONYMS describes all synonyms in the database. Its columns are the same as those in "ALL_SYNONYMS" on page 2-74.

DBA_SYS_PRIVS

DBA_SYS_PRIVS lists system privileges granted to users and roles. This view does not display the USERNAME column.

Related view:

- USER_SYS_PRIVS lists system privileges granted to the current user. This view does not display the GRANTEE column, but instead displays the USERNAME column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Grantee name, user, or role receiving the grant
USERNAME	VARCHAR2(30)		Name of the current user.
PRIVILEGE	VARCHAR2(40)	NOT NULL	System privilege
ADMIN_OPTION	VARCHAR2(3)		Grant was with the ADMIN option

DBA_TAB_COL_STATISTICS

DBA_TAB_COL_STATISTICS contains column statistics and histogram information extracted from "DBA_TAB_COLUMNS" on page 2-135. Its columns are the same as those in "ALL_TAB_COL_STATISTICS" on page 2-75.

DBA_TAB_COLUMNS

DBA_TAB_COLUMNS describes columns of all tables, views, and clusters in the database. Its columns are the same as those in "ALL_TAB_COLUMNS" on page 2-75. To gather statistics for this view, use the SQL command ANALYZE.

DBA_TAB_COMMENTS

DBA_TAB_COMMENTS describes comments on all tables and views in the database. Its columns are the same as those in "ALL_TAB_COMMENTS" on page 2-77.

DBA_TAB_HISTOGRAMS

DBA_TAB_HISTOGRAMS describes histograms on columns of all tables in the database. Its columns are the same as those in "ALL_TAB_HISTOGRAMS" on page 2-77.

DBA_TAB_MODIFICATIONS

DBA_TAB_MODIFICATIONS describes modifications to all tables in the database that have been modified since the last time statistics were gathered on the tables. Its columns are the same as those in "ALL_TAB_MODIFICATIONS" on page 2-78.

Note: This view is populated only for tables with the MONITORING attribute. It is intended for statistics collection over a long period of time and may not be populated until a few hours after the actual modifications occurred.

DBA_TAB_PARTITIONS

DBA_TAB_PARTITIONS provides the partition-level partitioning information, partition storage parameters, and partition statistics determined by ANALYZE

statements for all partitions in the database. Its columns are the same as those in "ALL_TAB_PARTITIONS" on page 2-79.

DBA_TAB_PRIVS

DBA_TAB_PRIVS lists all grants on all objects in the database. Its columns are the same as those in "ALL_TAB_PRIVS" on page 2-80.

DBA_TAB_SUBPARTITIONS

DBA_TAB_SUBPARTITIONS describes, for each table subpartition, the subpartition name, name of the table and partition to which it belongs, and its storage attributes. Its columns are the same as those in "ALL_TAB_SUBPARTITIONS" on page 2-82.

Note: Statistics are not collected on a per-subpartition basis.

DBA_TABLES

DBA_TABLES describes all relational tables in the database. Its columns are the same as those in "ALL_TABLES" on page 2-83. To gather statistics for this view, use the SQL command ANALYZE.

DBA_TABLESPACES

DBA_TABLESPACES describes all tablespaces in the database.

Related view:

- USER_TABLESPACES describes all tablespaces accessible to the current user. This view does not display the PLUGGED_IN column.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Tablespace name
INITIAL_EXTENT	NUMBER		Default initial extent size
NEXT_EXTENT	NUMBER		Default incremental extent size
MIN_EXTENTS	NUMBER	NOT NULL	Default minimum number of extents
MAX_EXTENTS	NUMBER	NOT NULL	Default maximum number of extents
PCT_INCREASE	NUMBER	NOT NULL	Default percent increase for extent size

Column	Datatype	NULL	Description
MIN_EXTLEN	NUMBER		Minimum extent size for the tablespace
STATUS	VARCHAR2(9)		Tablespace status: ONLINE, OFFLINE, or READ ONLY
CONTENTS	VARCHAR2(9)		Tablespace contents: PERMANENT, or TEMPORARY
LOGGING	VARCHAR2(9)		Default logging attribute
EXTENT_MANAGEMENT	VARCHAR2(10)		Extent management tracking: DICTIONARY or LOCAL
ALLOCATION_TYPE	VARCHAR2(9)		Type of extent allocation in effect for this tablespace
PLUGGED_IN	VARCHAR2(3)		YES - the tablespace is plugged in; NO - it is not plugged in

DBA_TEMP_FILES

DBA_TEMP_FILES describes all temporary files (tempfiles) in the database.

Column	Datatype	NULL	Description
FILE_NAME	VARCHAR2(513)		Name of the database temp file
FILE_ID	NUMBER		ID of the database temp file
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace to which the file belongs
BYTES	NUMBER		Size of the file in bytes
BLOCKS	NUMBER		Size of the file in Oracle blocks
STATUS	CHAR(9)		File status: AVAILABLE
RELATIVE_FNO	NUMBER		Tablespace-relative file number
AUTOEXTENSIBLE	VARCHAR2(3)		Autoextensible indicator: YES or NO
MAXBYTES	NUMBER		maximum size of the file in bytes
MAXBLOCKS	NUMBER		Maximum size of the file in Oracle database blocks
INCREMENT_BY	NUMBER		Default increment for autoextension
USER_BYTES	NUMBER		Size of the useful portion of file in bytes
USER_BLOCKS	NUMBER		Size of the useful portion of file in Oracle database blocks

DBA_TRIGGERS

DBA_TRIGGERS describes all triggers in the database. Its columns are the same as those in "ALL_TRIGGERS" on page 2-86.

DBA_TRIGGER_COLS

DBA_TRIGGER_COLS lists the use of columns in all triggers in the database. Its columns are the same as those in "ALL_TRIGGER_COLS" on page 2-87.

DBA_TS_QUOTAS

DBA_TS_QUOTAS describes tablespace quotas for all users.

Related view:

- USER_TS_QUOTAS describes tablespace quotas for the current user. This view does not display the USERNAME column.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Tablespace name
USERNAME	VARCHAR2(30)	NOT NULL	User with resource rights on the tablespace
BYTES	NUMBER		Number of bytes charged to the user
MAX_BYTES	NUMBER		User's quota in bytes, or -1 if no limit.
BLOCKS	NUMBER	NOT NULL	Number of Oracle blocks charged to the user
MAX_BLOCKS	NUMBER		User's quota in Oracle blocks, or -1 if no limit.

DBA_TYPE_ATTRS

DBA_TYPE_ATTRS describes the attributes of all object types in the database. Its columns are the same as those in "ALL_TYPE_ATTRS" on page 2-88.

DBA_TYPE_METHODS

DBA_TYPE_METHODS describes methods of all object types in the database. Its columns are the same as those in "ALL_TYPE_METHODS" on page 2-88.

DBA_TYPES

DBA_TYPES describes all object types in the database. Its columns are the same as those in "ALL_TYPES" on page 2-89.

DBA_UNUSED_COL_TABS

DBA_UNUSED_COL_TABS describes all tables in the database containing unused columns. Its columns are the same as those in "ALL_UNUSED_COL_TABS" on page 2-90.

DBA_UPDATABLE_COLUMNS

DBA_UPDATABLE_COLUMNS describes all columns in a join view that can be updated by the database administrator. Its columns are the same as those in "ALL_UPDATABLE_COLUMNS" on page 2-90.

> **See Also:** *Oracle8i Concepts* for information on updatable join views.

DBA_USERS

DBA_USERS describes all users of the database. This view does not display the PROFILE column.

Related View:

- USER_USERS describes the current user. This view does not display the PASSWORD column.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user
USER_ID	NUMBER	NOT NULL	ID number of the user
PASSWORD	VARCHAR2(30)		Encrypted password
ACCOUNT_STATUS	VARCHAR2(30)	NOT NULL	Indicates if the account is locked, expired, or unlocked
LOCK_DATE	DATE		Date the account was locked if account status was locked
EXPIRY_DATE	DATE		Date of expiration of the account
DEFAULT_TABLESPACE	VARCHAR2(30)	NOT NULL	Default tablespace for data
TEMPORARY_TABLESPACE	VARCHAR2(30)	NOT NULL	Default tablespace for temporary table
CREATED	DATE	NOT NULL	User creation date
PROFILE	VARCHAR2(30)	NOT NULL	User resource profile name

Column	Datatype	NULL	Description
INITIAL_RSRC_ CONSUMER_GROUP	VARCHAR2(30)		The initial resource consumer group for the user
EXTERNAL_NAME	VARCHAR2(4000)		User external name

DBA_USTATS

DBA_USTATS describes all user-defined statistics collected on all objects in the database. Its columns are the same as those in "ALL_USTATS" on page 2-91.

DBA_VARRAYS

DBA_VARRAYS describes all the varrays in the database. Its columns are the same as those in "ALL_VARRAYS" on page 2-92.

DBA_VIEWS

DBA_VIEWS displays the text of all views in the database. Its columns are the same as those in "ALL_VIEWS" on page 2-92.

DBA_WAITERS

DBA_WAITERS shows all the sessions that are waiting for a lock, but do not hold locks for which another session is waiting.

Column	Datatype	NULL	Description
WAITING_SESSION	NUMBER		The waiting session
HOLDING_SESSION	NUMBER		The holding session
LOCK_TYPE	VARCHAR2(26)		The lock type
MODE_HELD	VARCHAR2(40)		The mode held
MODE_REQUESTED	VARCHAR2(40)		The mode requested
LOCK_ID1	VARCHAR2(40)		Lock ID 1
LOCK_ID2	VARCHAR2(40)		Lock ID 2

DBMS_ALERT_INFO

DBMS_ALERT_INFO describes registered alerts.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the alert
SID	VARCHAR2(30)	NOT NULL	Session ID of a session waiting for this alert
CHANGED	VARCHAR2(1)		Boolean flag to indicate that an alert has been signaled. Y: Alert signaled N: No alert.
MESSAGE	VARCHAR2(1800)		Optional message passed by signaler

DBMS_LOCK_ALLOCATED

DBMS_LOCK_ALLOCATED describes user-allocated locks.

Column	Datatype	NULL	Description
NAME	VARCHAR2(128)	NOT NULL	Name of the lock
LOCKID	NUMBER(38)		Lock identifier number
EXPIRATION	DATE		Planned lock expiration date (updates whenever the allocation procedure is run)

DEPTREE

This view, created by UTLDTREE.SQL, contains information on the object dependency tree. For user SYS, this view displays shared cursors (and only shared cursors) that depend on the object. For all other users, it displays objects other than shared cursors. Other users can access SYS.DEPTREE for information on shared cursors.

Column	Datatype	NULL	Description
NESTED_LEVEL	NUMBER		Nesting level in the dependency tree
TYPE	VARCHAR2(15)		Object type
OWNER	VARCHAR2(30)		Object owner
NAME	VARCHAR2(1002)		Object name
SEQ#	NUMBER		Sequence number in the dependency tree. Used for ordering queries. (See also: "IDEPTREE" on page 2-152)

DICT

DICT is a synonym for DICTIONARY.

See Also: "DICTIONARY" on page 2-142.

DICTIONARY

DICTIONARY contains descriptions of data dictionary tables and views.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Name of the object
COMMENTS	VARCHAR2(4000)		Text comment on the object

DICT_COLUMNS

DICT_COLUMNS contains descriptions of columns in data dictionary tables and views.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Name of the object that contains the column
COLUMN_NAME	VARCHAR2(30)		Name of the column
COMMENTS	VARCHAR2(4000)		Text comment on the column

ERROR_SIZE

ERROR_SIZE is accessed to create "DBA_OBJECT_SIZE" on page 2-116 and "USER_OBJECT_SIZE" on page 2-176.

EXCEPTIONS

EXCEPTIONS contains information on violations of integrity constraints. This view is created by the UTLEXCPT.SQL script.

Column	Datatype	NULL	Description
ROW_ID	ROWID		Row that caused the violation
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
CONSTRAINT	VARCHAR2(30)		Integrity constraint that was violated

FILE_LOCK

This Parallel Server view displays the mapping of PCM locks to datafiles as specified in initialization parameter GC_FILES_TO_LOCKS.

> **See Also:** "GC_FILES_TO_LOCKS" on page 1-40 and *Oracle8i Parallel Server Concepts.*

Column	Datatype	NULL	Description
FILE_ID	NUMBER	NOT NULL	Datafile identifier number (to find file name, query DBA_DATA_FILES or V$DBFILES)
FILE_NAME	VARCHAR2(513)		The datafile name
TS_NAME	VARCHAR2(30)	NOT NULL	The tablespace name for the datafile
START_LK	NUMBER		The first lock corresponding to the datafile
NLOCKS	NUMBER		The number of PCM locks allocated to the datafile
BLOCKING	NUMBER		The number of contiguous blocks protected by each PCM lock on the datafile

FILE_PING

This Parallel Server view displays the number of blocks pinged per datafile. You can use this information to determine access usage of existing datafiles for better settings of GC_FILES_TO_LOCKS.

> **See Also:** "GC_FILES_TO_LOCKS" on page 1-40 and *Oracle8i Parallel Server Concepts.*

Column	Datatype	NULL	Description
FILE_ID	NUMBER	NOT NULL	Datafile identifier number (to find file name, query DBA_DATA_FILES or V$DBFILES)
FILE_NAME	NUMBER		The file name
TS_NAME	VARCHAR2(30)	NOT NULL	Datafile identifier number (to find file name, query DBA_DATA_FILES or V$DBFILES)
FREQUENCY	NUMBER		The ping count
X_2_NULL	NUMBER		Number of lock conversions from Exclusive-to-NULL for all blocks in the file

Column	Datatype	NULL	Description
X_2_NULL_FORCED_WRITE	NUMBER		Number of forced writes that occur for blocks of the specified file due to Exclusive-to-NULL conversions
X_2_NULL_FORCED_STALE	NUMBER		Number of times a block in the file was made STALE due to Exclusive-to-NULL conversions
X_2_S	NUMBER		Number of lock conversions from Exclusive-to-Shared for all blocks in the file
X_2_S_FORCED_WRITES	NUMBER		Number of forced writes that occur for blocks of the specified file due to Exclusive-to-Shared conversions
X_2_SSX	NUMBER		Number of lock conversions from Exclusive-to-Sub Shared Exclusive for all blocks in the file
X_2_SSX_FORCED_WRITES	NUMBER		Number of forced writes that occur for blocks of the specified file due to Exclusive-to-Sub Shared Exclusive conversions
S_2_NULL	NUMBER		Number of lock conversions from Shared-to-NULL for all blocks in the file
S_2_NULL_FORCED_STALE	NUMBER		Number of times a block in the file was made STALE due to Shared-to-NULL conversions
SS_2_NULL	NUMBER		Number of lock conversions from Sub Shared-to-NULL for all blocks in the file
WRB	NUMBER		Number of times the instance received a write single buffer cross instance call for this file
WRB_FORCED_WRITE	NUMBER		Number of blocks written due to write single buffer cross instance calls for this file
RBR	NUMBER		Number of times the instance received a reuse block range cross instance call for this file
RBR_FORCED_WRITE	NUMBER		Number of blocks written due to resuse block range cross instance calls for this file
RBR_FORCED_STALE	NUMBER		Number of times a block in this file was made STALE due to resuse block range cross instance calls
CBR	NUMBER		Number of times the instance received a checkpoint block range cross instance call for this file
CBR_FORCED_WRITE	NUMBER		Number of blocks in this file which were written due to checkpoint cross range cross instance calls
NULL_2_X	NUMBER		Number of lock conversions from NULL-to-Exclusive for all blocks of the specified file
S_2_X	NUMBER		Number of lock conversions from Shared-to-Exclusive for all blocks of the specified file

Column	Datatype	NULL	Description
SSX_2_X	NUMBER		Number of lock conversions from Sub Shared Exclusive-to-Exclusive for all blocks of the specified file
N_2_S	NUMBER		Number of lock conversions from NULL-to-Shared for all blocks of the specified file
N_2_SS	NUMBER		Number of lock conversions from NULL-to-Sub Shared for all blocks of the specified file

FILEXT$

FILEXT$ is the equivalent of DBA_DATA_FILES. Oracle Corporation recommends you use DBA_DATA_FILES instead of FILEXT$.

See Also: "DBA_DATA_FILES".

Column	Datatype	NULL	Description
FILE#	NUMBER	NOT NULL	Absolute file number
MAXEXTEND	NUMBER		Maximum file size
INC	NUMBER		Increment amount

GLOBAL_NAME

GLOBAL_NAME contains one row that displays the global name of the current database.

Column	Datatype	NULL	Description
GLOBAL_NAME	VARCHAR2(4000)		Global name of the database

HS_ALL_CAPS

HS_ALL_CAPS contains information about all of the capabilities (that is, features) associated with non-Oracle (FDS) data stores.

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER		Capability number
CONTEXT	NUMBER		Context in which this capability is applicable

Column	Datatype	NULL	Description
TRANSLATION	VARCHAR2(255)		Valid for functions; contains translation to FDS dialect
ADDITIONAL_INFO	NUMBER		Flag for internal use
FDS_CLASS_NAME	VARCHAR2(30)		Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)		Name of the FDS instance

HS_ALL_DD

HS_ALL_DD contains data dictionary information about non-Oracle (FDS) data stores.

Column	Datatype	NULL	Description
DD_TABLE_NAME	VARCHAR2(30)		Data dictionary table name
TRANSLATION_TYPE	CHAR(1)		T = Translation, M = Mimic
TRANSLATION_TEXT	VARCHAR2(4000)		SQL statement containing the mapping
FDS_CLASS_NAME	VARCHAR2(30)		Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)		Name of the FDS instance
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table

HS_ALL_INITS

HS_ALL_INITS contains initialization parameter information about non-Oracle (FDS) data stores.

Column	Datatype	NULL	Description
INIT_VALUE_NAME	VARCHAR2(64)		Name of the initialization parameter
INIT_VALUE	VARCHAR2(255)		Value of the initialization parameter
INIT_VALUE_TYPE	VARCHAR2(1)		Environment variable (T or F). T= this is an environment variable, F= do not set as an environment variable
FDS_CLASS_NAME	VARCHAR2(30)		Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)		Name of the FDS instance

HS_BASE_CAPS

HS_BASE_CAPS contains information about base capability (that is, base features) of the non-Oracle (FDS) data store.

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER	NOT NULL	Capability number
CAP_DESCRIPTION	VARCHAR2(255)		Description of the capability

HS_BASE_DD

HS_BASE_DD displays information from the base data dictionary translation table.

Column	Datatype	NULL	Description
DD_TABLE_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
DD_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the Oracle data dictionary table
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table

HS_CLASS_CAPS

HS_CLASS_CAPS contains information about the class-specific (driver) capabilities belonging to the non-Oracle (FDS) data store.

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER	NOT NULL	Capability number
CAP_DESCRIPTION	VARCHAR2(255)		Capability description
CONTEXT	NUMBER		Flag indicating the context in which the capability is enabled
TRANSLATION	VARCHAR2(255)		Valid for functions; contains translation to FDS dialect
ADDITIONAL_INFO	NUMBER		Additional flags for internal use
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_CLASS_DD

HS_CLASS_DD displays information from the non-Oracle data store (FDS) class-specific data dictionary translations.

Column	Datatype	NULL	Description
DD_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the Oracle data dictionary table
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table
TRANSLATION_TYPE	CHAR(1)	NOT NULL	T = Translation, M = Mimic
TRANSLATION_TEXT	VARCHAR2(4000)		SQL statement containing the mapping
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
DD_TABLE_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_CLASS_INIT

HS_CLASS_INIT displays information about the non-Oracle (FDS) class-specific initialization parameters.

Column	Datatype	NULL	Description
INIT_VALUE_NAME	VARCHAR2(64)	NOT NULL	Name of the initialization parameter
INIT_VALUE	VARCHAR2(255)	NOT NULL	Value of the initialization parameter
INIT_VALUE_TYPE	VARCHAR2(1)	NOT NULL	Environment variable (T or F). T= this is an environment variable, F= do not set as an environment variable
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_CLASS_INIT_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_EXTERNAL_OBJECT_PRIVILEGES

HS_EXTERNAL_OBJECT_PRIVILEGES contains information about the privileges on objects that are granted to users.

Column	Datatype	NULL	Description
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object. Name is unique for each instance.
PRIVILEGE_NAME	VARCHAR2(30)	NOT NULL	Name of the privilege that was granted
GRANTEE	VARCHAR2(30)	NOT NULL	ID of the user that was granted the privilege
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance

HS_EXTERNAL_OBJECTS

HS_EXTERNAL_OBJECTS contains information about all of the distributed external objects accessible from the Oracle server.

Column	Datatype	NULL	Description
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
OWNER	VARCHAR(30)	NOT NULL	Name of the user who created the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object. Name is unique for each instance
OBJECT_TYPE	VARCHAR2(13)	NOT NULL	Type of object: FUNCTION, PROCEDURE, PACKAGE, or LIBRARY
OBJECT_TEXT	LONG	NOT NULL	SQL text used to create the object

HS_EXTERNAL_USER_PRIVILEGES

HS_EXTERNAL_USER_PRIVILEGES contains information about all of the granted privileges that are not tied to any particular object.

Column	Datatype	NULL	Description
PRIVILEGE_NAME	VARCHAR2(30)	NOT NULL	Name of the privilege that was granted
GRANTEE	VARCHAR2(30)	NOT NULL	ID of the user that was granted the privilege
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance

HS_FDS_CLASS

HS_FDS_CLASS contains information about legal non-Oracle (FDS) classes.

Column	Datatype	NULL	Description
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (eg, ODBC, DB2)
FDS_CLASS _COMMENTS	VARCHAR2(255)		Text description of the non-Oracle class
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_FDS_INST

HS_FDS_INST contains information about non-Oracle (FDS) instances.

Column	Datatype	NULL	Description
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
FDS_INST_COMMENTS	VARCHAR2(255)		Text description of the non-Oracle instance
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class
FDS_INST_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_INST_CAPS

HS_INST_CAPS contains information about instance-specific capabilities (that is, features).

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER	NOT NULL	Capability number
CAP_DESCRIPTION	VARCHAR2(255)		Capability description
CONTEXT	NUMBER		Context in which this capability is applicable
TRANSLATION	VARCHAR2(255)		Valid for functions; contains translation to FDS dialect
ADDITIONAL_INFO	NUMBER		Additional flags for internal use
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (eg, ODBC, DB2)
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_INST_DD

HS_INST_DD displays information from the non-Oracle (FDS) instance-specific data dictionary translations.

Column	Datatype	NULL	Description
DD_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the Oracle data dictionary table
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table
TRANSLATION_TYPE	CHAR(1)	NOT NULL	T = Translation, M = Mimic
TRANSLATION_TEXT	VARCHAR2(4000)		SQL statement containing the mapping
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (eg, ODBC, DB2)
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
DD_TABLE_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

HS_INST_INIT

HS_INST_INIT contains information about the non-Oracle (FDS) instance-specific initialization parameters.

Column	Datatype	NULL	Description
INIT_VALUE_NAME	VARCHAR2(64)	NOT NULL	Name of the initialization parameter
INIT_VALUE	VARCHAR2(255)	NOT NULL	Value of the initialization parameter
INIT_VALUE_TYPE	VARCHAR2(1)	NOT NULL	Environment variable (T or F). T= this is an environment variable, F= do not set as an environment variable
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (for example: ODBC, DB2)
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
FDS_INST_INIT_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence - a counter which is incremented for every row inserted (used internally)

IDEPTREE

This view, created by UTLDTREE.SQL, lists the indented dependency tree. It is a pre-sorted, pretty-print version of DEPTREE.

Column	Datatype	NULL	Description
NESTED_LEVEL	NUMBER		Nesting level in the dependency tree
TYPE	VARCHAR2(15)		Object type
OWNER	VARCHAR2(30)		Object owner
NAME	VARCHAR2(1002)		Object name

IND

IND is a synonym for "USER_INDEXES" on page 2-172.

INDEX_HISTOGRAM

INDEX_HISTOGRAM contains information from the ANALYZE INDEX ... VALIDATE STRUCTURE command.

Column	Datatype	NULL	Description
REPEAT_COUNT	NUMBER		Number of times that one or more index keys is repeated in the table
KEYS_WITH_REPEAT_COUNT	NUMBER		Number of index keys that are repeated that many times

INDEX_STATS

INDEX_STATS stores information from the last ANALYZE INDEX ... VALIDATE STRUCTURE command.

Column	Datatype	NULL	Description
HEIGHT	NUMBER		Height of the B-Tree
BLOCKS	NUMBER	NOT NULL	Blocks allocated to the segment
NAME	VARCHAR2(30)	NOT NULL	Name of the index
PARTITION_NAME	VARCHAR2(30)		Name of the partition of the index which was analyzed. If the index is not partitioned, a NULL is returned
LF_ROWS	NUMBER		Number of leaf rows (values in the index)

Column	Datatype	NULL	Description
LF_BLKS	NUMBER		Number of leaf blocks in the B-Tree
LF_ROWS_LEN	NUMBER		Sum of the lengths of all the leaf rows
LF_BLK_LEN	NUMBER		Usable space in a leaf block
BR_ROWS	NUMBER		Number of branch rows in the B-Tree
BR_BLKS	NUMBER		Number of branch blocks in the B-Tree
BR_ROWS_LEN	NUMBER		Sum of the lengths of all the branch blocks in the B-Tree
BR_BLK_LEN	NUMBER		Usable space in a branch block
DEL_LF_ROWS	NUMBER		Number of deleted leaf rows in the index
DEL_LF_ROWS_LEN	NUMBER		Total length of all deleted rows in the index
DISTINCT_KEYS	NUMBER		Number of distinct keys in the index (may include rows that have been deleted)
MOST_REPEATED_KEY	NUMBER		How many times the most repeated key is repeated (may include rows that have been deleted)
BTREE_SPACE	NUMBER		Total space currently allocated in the B-Tree
USED_SPACE	NUMBER		Total space that is currently being used in the B-Tree
PCT_USED	NUMBER		Percent of space allocated in the B-Tree that is being used
ROWS_PER_KEY	NUMBER		Average number of rows per distinct key (this figure is calculated without consideration of deleted rows)
BLKS_GETS_PER_ACCESS	NUMBER		Expected number of consistent mode block reads per row, assuming that a randomly chosen row is accessed using the index. Used to calculate the number of consistent reads that will occur during an index scan
PRE_ROWS	NUMBER		Number of prefix rows (values in the index)
PRE_ROWS_LEN	NUMBER		Sum of lengths of all prefix rows

NLS_DATABASE_PARAMETERS

NLS_DATABASE_PARAMETERS lists permanent NLS parameters of the database.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)	NOT NULL	Parameter name
VALUE	VARCHAR2(30)		Parameter value

NLS_INSTANCE_PARAMETERS

NLS_INSTANCE_PARAMETERS lists NLS parameters of the instance.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)		Parameter name
VALUE	VARCHAR2(30)		Parameter value

NLS_SESSION_PARAMETERS

NLS_SESSION_PARAMETERS lists NLS parameters of the user session.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)		Parameter name
VALUE	VARCHAR2(30)		Parameter value

OBJ

OBJ is a synonym for "USER_OBJECTS" on page 2-175.

PARSED_PIECES

Oracle accesses this view to create views about object size.

> **See Also:** "DBA_OBJECT_SIZE" on page 2-116 and "USER_OBJECT_SIZE" on page 2-176.

PARSED_SIZE

Oracle accesses this view to create views about object size.

> **See Also:** "DBA_OBJECT_SIZE" on page 2-116 and "USER_OBJECT_SIZE" on page 2-176.

PLAN_TABLE

PLAN_TABLE is the default table for results of the EXPLAIN PLAN statement. It is created by UTLXPLAN.SQL, and it contains one row for each step in the execution plan.

Column	Datatype	NULL	Description
STATEMENT_ID	VARCHAR2(30)		Optional statement identifier specified in the EXPLAIN PLAN statement
TIMESTAMP	DATE		Date and time that the EXPLAIN PLAN statement was issued
REMARKS	VARCHAR2(80)		Place for comments that can be added to the steps of the execution plan
OPERATION	VARCHAR2(30)		Name of the operation performed at this step
OPTIONS	VARCHAR2(30)		Options used for the operation performed at this step
OBJECT_NODE	VARCHAR2(128)		Name of the database link used to reference the object
OBJECT_OWNER	VARCHAR2(30)		Owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the object
OBJECT_INSTANCE	NUMBER(38)		Numbered position of the object name in the original SQL statement
OBJECT_TYPE	VARCHAR2(30)		Descriptive modifier that further describes the type of object
OPTIMIZER	VARCHAR2(255)		The current mode of the optimizer
SEARCH_COLUMNS	NUMBER		Not currently used
ID	NUMBER(38)		Identification number for this step in the execution plan
PARENT_ID	NUMBER(38)		ID of the next step that operates on the results of this step
POSITION	NUMBER(38)		Order of processing for steps with the same parent ID. For cost-based optimization, the value in the first row of the plan is the statement's execution cost. For rule-based optimization, the value is null in the first row
COST	NUMBER(38)		The cost of the current operation estimated by the cost-based optimizer (CBO)
CARDINALITY	NUMBER(38)		The number of rows returned by the current operation (estimated by the CBO)
BYTES	NUMBER(38)		The number of bytes returned by the current operation

Column	Datatype	NULL	Description
OTHER_TAG	VARCHAR2(255)		OTHER_TAG describes the function of the SQL text in the OTHER column. Values for OTHER_TAG are: ■ SERIAL - the SQL is the text of a locally-executed, serial query plan. Currently, SQL is not loaded in OTHER for this case. ■ SERIAL_FROM_REMOTE - the SQL text shown in the OTHER column will be executed at a remote site. ■ PARALLEL_COMBINED_WITH_PARENT - the parent of this operation is a DFO that performs both operations in the parallel execution plan. ■ PARALLEL_COMBINED_WITH_CHILD - the child of this operation is a DFO that performs both operations in the parallel execution plan. ■ PARALLEL_TO_SERIAL - the SQL text shown in the OTHER column is the top-level of the parallel plan. ■ PARALLEL_TO_PARALLEL - the SQL text shown in the OTHER column is executed and output in parallel. ■ PARALLEL_FROM_SERIAL - this operation consumes data from a serial operation and outputs it in parallel.
PARTITION_START	VARCHAR2(255)		The start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2(255)		The stop partition of a range of accessed partitions
PARTITION_ID	NUMBER(38)		The step that has computed the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	LONG		Holds SQL text for remote cursors and parallel execution slaves
DISTRIBUTION	VARCHAR2(30)		The distribution method

PLUGGABLE_SET_CHECK

PLUGGABLE_SET_CHECK contains pluggable set checks.

Column	Datatype	NULL	Description
OBJ1_OWNER	VARCHAR2(30)		Owner of object
OBJ1_NAME	VARCHAR2(30)		Object 1
OBJ1_SUBNAME	VARCHAR2(30)		SubObject1Name
OBJ1_TYPE	VARCHAR2(15)		Object Type
TS1_NAME	VARCHAR2(30)		Tablespace containing Object 1
OBJ2_NAME	VARCHAR2(30)		Object Name

Column	Datatype	NULL	Description
OBJ2_SUBNAME	VARCHAR2(30)		SubObject2Name
OBJ2_TYPE	VARCHAR2(15)		Object Type
OBJ2_OWNER	VARCHAR2(30)		Object owner of second object
TS2_NAME	VARCHAR2(30)		Tablespace containing Object 1
CONSTRAINT_NAME	VARCHAR2(30)		Name of dependent constraint
REASON	VARCHAR2(79)		Reason for Pluggable check violation
MESG_ID	NUMBER		The message ID

PRODUCT_COMPONENT_VERSION

PRODUCT_COMPONENT_VERSION contains version and status information for component products.

Column	Datatype	NULL	Description
PRODUCT	VARCHAR2(64)		Product name
VERSION	VARCHAR2(64)		Version number
STATUS	VARCHAR2(64)		Status of release

PROXY_USERS

PROXY_USERS describes users who can assume the identity of other users.

Column	Datatype	NULL	Description
PROXY	VARCHAR2(30)	NOT NULL	Name of the client whose identity a proxy may assume
CLIENT	VARCHAR2(30)	NOT NULL	Name of a user which assume the identity of a client
ROLE	VARCHAR2(30)		Name of a role that a proxy user may or may not activate while acting as a client
FLAGS	VARCHAR2(35)		Indicates whether or not a proxy can activate all client roles, no client roles, or a specific client role

PSTUBTBL

This table contains information on stubs generated by the PSTUB utility so that an Oracle Forms 3.0 client can call stored procedures in an Oracle database.

Note: The contents of this table are intended only for use by the PSTUB utility.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)		Schema part of the identifier of a stored procedure
DBNAME	VARCHAR2(128)		Database link part of the identifier of a stored procedure
LUN	VARCHAR2(30)		Library unit name part of the identifier of a stored procedure
LUTYPE	VARCHAR2(3)		Type of the stored procedure
LINENO	NUMBER		Line number of the stub
LINE	VARCHAR2(1800)		Text of the stub

PUBLICSYN

PUBLICSYN contains information on public synonyms.

Column	Datatype	NULL	Description
SNAME	VARCHAR2(30)		Name of the synonym
CREATOR	VARCHAR2(30)		Owner of the synonym
TNAME	VARCHAR2(30)		Table of which this is a synonym
DATABASE	VARCHAR2(128)		Database in which the table resides
TABTYPE	VARCHAR2(9)		Type of table

PUBLIC_DEPENDENCY

PUBLIC_DEPENDENCY lists dependencies to and from objects, by object number.

Column	Datatype	NULL	Description
OBJECT_ID	NUMBER	NOT NULL	Object number
REFERENCED_OBJECT_ID	NUMBER	NOT NULL	Referenced object (the parent object)

QUEUE_PRIVILEGES

QUEUE_PRIVILEGES shows all Advanced Queuing object privileges granted to the session.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
ENQUEUE_PRIVILEGE	NUMBER		Permission to ENQUEUE to the queue
DEQUEUE_PRIVILEGE	NUMBER		Permission to DEQUEUE from the queue

RC

RC_ views are not created by default in the database, they are only available after you create an optional recovery catalog (which contains schemas containing information about backups) for use with Recovery Manager. Please see the *Oracle8i Recovery Manager User's Guide and Reference* for details about these views.

RESOURCE_COST

RESOURCE_COST lists the cost for each resource.

Column	Datatype	NULL	Description
RESOURCE_NAME	VARCHAR2(32)	NOT NULL	Name of the resource
UNIT_COST	NUMBER	NOT NULL	Cost of the resource

RESOURCE_MAP

RESOURCE_MAP describes resources and maps the resource name to the resource number.

Column	Datatype	NULL	Description
RESOURCE#	NUMBER	NOT NULL	Numeric resource code
TYPE#	NUMBER	NOT NULL	Name of type
NAME	VARCHAR2(32)	NOT NULL	Name of resource

ROLE_ROLE_PRIVS

ROLE_ROLE_PRIVS describes roles granted to other roles. Information is provided only about roles to which the user has access.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the role
GRANTED_ROLE	VARCHAR2(30)	NOT NULL	Role that was granted
ADMIN_OPTION	VARCHAR2(3)		Signifies that the role was granted with ADMIN option

ROLE_SYS_PRIVS

ROLE_SYS_PRIVS contains information about system privileges granted to roles. Information is provided only about roles to which the user has access.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the role
PRIVILEGE	VARCHAR2(40)	NOT NULL	System privilege granted to the role
ADMIN_OPTION	VARCHAR2(3)		Signifies the grant was with the ADMIN option

ROLE_TAB_PRIVS

ROLE_TAB_PRIVS contains information about table privileges granted to roles. Information is provided only about roles to which the user has access.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the role
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)		Name of the column, if applicable
PRIVILEGE	VARCHAR2(40)	NOT NULL	Object privilege granted to the role
GRANTABLE	VARCHAR2(3)		YES if the role was granted with ADMIN OPTION; otherwise NO

SEQ

SEQ is a synonym for "USER_SEQUENCES" on page 2-181.

SESSION_CONTEXT

SESSION_CONTEXT lists attributes and the values set for the current session.

See Also: *Oracle8i SQL Reference* for information on the predefined attributes of the current session.

Column	Datatype	NULL	Description
NAMESPACE	VARCHAR2(30)	NOT NULL	The namespace that the active attribute is in
ATTRIBUTE	VARCHAR2(30)	NOT NULL	The name of the active attribute
VALUE	VARCHAR2(4000)	NOT NULL	The value of the active attribute

SESSION_PRIVS

SESSION_PRIVS lists the privileges that are currently available to the user.

Column	Datatype	NULL	Description
PRIVILEGE	VARCHAR2(40)	NOT NULL	Name of the privilege

SESSION_ROLES

SESSION_ROLES lists the roles that are currently enabled to the user.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the role

SOURCE_SIZE

Oracle accesses this view to create views about object size.

See Also: "DBA_OBJECT_SIZE" on page 2-116 and "USER_OBJECT_SIZE" on page 2-176.

STMT_AUDIT_OPTION_MAP

STMT_AUDIT_OPTION_MAP lists auditing option type codes.

Column	Datatype	NULL	Description
OPTION#	NUMBER	NOT NULL	Numeric auditing option type code
NAME	VARCHAR2(40)	NOT NULL	Name of the auditing option
PROPERTY	NUMBER	NOT NULL	

SYN

SYN is a synonym for "USER_SYNONYMS" on page 2-182.

SYNONYMS

SYNONYMS is included for compatibility with Oracle version 5. Use of this view is not recommended.

SYSCATALOG

SYSCATALOG is included for compatibility with Oracle version 5. Use of this view is not recommended.

SYSFILES

SYSFILES is included for compatibility with Oracle version 5. Use of this view is not recommended.

SYSSEGOBJ

SYSSEGOBJ is included for compatibility with Oracle version 5. Use of this view is not recommended.

SYSTEM_PRIVILEGE_MAP

SYSTEM_PRIVILEGE_MAP contains information about system privilege codes.

Column	Datatype	NULL	Description
PRIVILEGE	NUMBER	NOT NULL	Numeric privilege type code
NAME	VARCHAR2(40)	NOT NULL	Name of the type of privilege
PROPERTY	NUMBER	NOT NULL	

SYS_OBJECTS

SYS_OBJECTS maps object IDs to object types and segment data block addresses.

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(15)		Type of the object
OBJECT_TYPE_ID	NUMBER		Type ID of the object
SEGMENT_TYPE_ID	NUMBER		Type of segment: TABLE, CLUSTER, INDEX, ROLLBACK, DEFERRED ROLLBACK, TEMPORARY, CACHE
OBJECT_ID	NUMBER		Object identifier
HEADER_FILE	NUMBER		ID of the file containing the segment header
HEADER_BLOCK	NUMBER		ID of the block containing the segment header
TS_NUMBER	NUMBER		The tablespace number

TAB

TAB is included for compatibility with Oracle version 5. Use of this view is not recommended.

TABLE_PRIVILEGES

TABLE_PRIVILEGES contains information on grants on objects for which the user is the grantor, grantee, or owner, or PUBLIC is the grantee. This view is included for compatibility with Oracle version 6. Use of this view is not recommended.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access is granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant

Column	Datatype	NULL	Description
SELECT_PRIV	VARCHAR2(1)		Permission to select from the object
INSERT_PRIV	VARCHAR2(1)		Permission to insert into the object
DELETE_PRIV	VARCHAR2(1)		Permission to delete from the object
UPDATE_PRIV	VARCHAR2(1)		Permission to update the object
REFERENCES_PRIV	VARCHAR2(1)		Permission to reference the object
ALTER_PRIV	VARCHAR2(1)		Permission to alter the object
INDEX_PRIV	VARCHAR2(1)		Permission to create or drop an index on the object
CREATED	VARCHAR2(0)		Timestamp for the grant

TABLE_PRIVILEGE_MAP

TABLE_PRIVILEGE_MAP contains information about access privilege codes.

Column	Datatype	NULL	Description
PRIVILEGE	NUMBER	NOT NULL	Numeric privilege (auditing option) type code
NAME	VARCHAR2(40)	NOT NULL	Name of the type of privilege (auditing option)

TABS

TABS is a synonym for "USER_TABLES" on page 2-184.

TABQUOTAS

TABQUOTAS is included for compatibility with Oracle version 5. Use of this view is not recommended.

TRUSTED_SERVERS

TRUSTED_SERVERS displays whether a server is trusted or untrusted.

Column	Datatype	NULL	Description
TRUST	VARCHAR2(9)		Trustedness of the server listed. Values can be TRUSTED or UNTRUSTED Servers which are not listed in the NAME column have opposite trustedness. See the examples below
NAME	VARCHAR2(128)		Server name. Can be a specific server name or ALL for all servers

Table 2–2 shows examples of the values returned depending on the status of the servers.

Table 2–2 TRUSTED_SERVERS Values

Condition (If . . .)	TRUSTED column	NAME column
... all servers are trusted	Trusted	ALL
... no servers are trusted	Untrusted	ALL
... all servers except DB1 are trusted	Untrusted	DB1
... all servers except DB1 are untrusted	Trusted	DB1

See Also: *Oracle8i Distributed Database Systems.*

TS_PITR_CHECK

This view, created by CATPITR.SQL provides information on any dependencies or restrictions which might prevent tablespace point-in-time recovery from proceeding. This view applies only to the tablespace point-in-time recovery feature.

See Also: *Oracle8i Backup and Recovery Guide.*

Column	Datatype	NULL	Description
OBJ1_OWNER	VARCHAR2(30)	NOT NULL	The owner of the object preventing tablespace point-in-time recovery. See the REASON column for details.
OBJ1_NAME	VARCHAR2(30)	NOT NULL	The name of the object preventing tablespace point-in-time recovery
OBJ1_TYPE	VARCHAR2(15)		The object type for the object preventing tablespace point-in-time recovery
OBJ1_SUBNAME	VARCHAR2(30)		Subordinate to OBJ1_NAME
TS1_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the object preventing tablespace point-in-time recovery
OBJ2_NAME	VARCHAR2(30)		The name of a second object which may be preventing tablespace point-in-time recovery. If NULL, object 1 is the only object preventing recovery
OBJ2_TYPE	VARCHAR2(15)		The object type for the second object (will be NULL if OBJ2_NAME is NULL)
OBJ2_OWNER	VARCHAR2(30)		The owner of the second object (will be NULL if OBJ2_NAME is NULL)
OBJ2_SUBNAME	VARCHAR2(30)		Subordinate to OBJ2_NAME

Column	Datatype	NULL	Description
TS2_NAME	VARCHAR2(30)		Name of the tablespace containing second object which may be preventing tablespace point-in-time recovery (-1 indicates not applicable)
CONSTRAINT_NAME	VARCHAR2(30)		Name of the constraint
REASON	VARCHAR2(78)		Reason why tablespace point-in-time recovery cannot proceed

TS_PITR_OBJECTS_TO_BE_DROPPED

TS_PITR_OBJECTS_TO_BE_DROPPED lists all objects lost as a result of performing tablespace point-in-time recovery. This view applies only to the tablespace point-in-time recovery feature.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	The owner of the object
NAME	VARCHAR2(30)	NOT NULL	The name of the object that will be lost as a result of undergoing tablespace point-in-time recovery
CREATION_TIME	DATE	NOT NULL	Creation timestamp of the object
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the object

UNI_PLUGGABLE_SET_CHECK

UNI_PLUGGABLE_SET_CHECK contains pluggable check information.

Column	Datatype	NULL	Description
OBJ1_OWNER	VARCHAR2(30)		Owner of object
OBJ1_NAME	VARCHAR2(30)		Object 1
OBJ1_SUBNAME	VARCHAR2(30)		SubObject1Name
OBJ1_TYPE	VARCHAR2(15)		Object Type
TS1_NAME	VARCHAR2(30)		Tablespace containing Object 1
OBJ2_NAME	VARCHAR2(30)		Object Name
OBJ2_SUBNAME	VARCHAR2(30)		SubObject2Name
OBJ2_TYPE	VARCHAR2(15)		Object Type
OBJ2_OWNER	VARCHAR2(30)		Object owner of second object
TS2_NAME	VARCHAR2(30)		Tablespace containing Object 1

Column	Datatype	NULL	Description
CONSTRAINT_NAME	VARCHAR2(30)		Name of dependent constraint
REASON	VARCHAR2(79)		Reason for Pluggable check violation
MESG_ID	NUMBER		The message ID

USER_ALL_TABLES

USER_ALL_TABLES contains descriptions of the object tables and relational tables owned by the current user. Its columns are the same as those in "ALL_ALL_TABLES" on page 2-3.

USER_ARGUMENTS

USER_ARGUMENTS lists the arguments in all procedures and functions that are owned by the current user. Its columns are the same as those in "ALL_ARGUMENTS" on page 2-5.

USER_ASSOCIATIONS

USER_ASSOCIATIONS describes user-defined statistics associated with objects owned by the current user. Its columns are the same as those in "ALL_ASSOCIATIONS" on page 2-6.

USER_AUDIT_OBJECT

This view, created by CATAUDIT.SQL, lists audit trail records for statements concerning objects that are accessible to the current user. Its columns are the same as those in "DBA_AUDIT_OBJECT" on page 2-97.

USER_AUDIT_SESSION

This view, created by CATAUDIT.SQL, lists all audit trail records concerning connections and disconnections for the user. Its columns are the same as those in "DBA_AUDIT_SESSION" on page 2-99.

USER_AUDIT_STATEMENT

This view, created by CATAUDIT.SQL, lists audit trail entries for the following statements issued by the user: GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYSTEM. Its columns are the same as those in "DBA_AUDIT_STATEMENT" on page 2-99.

USER_AUDIT_TRAIL

This view, created by CATAUDIT.SQL, lists audit trail entries relevant to the user. Its columns are the same as those in "DBA_AUDIT_TRAIL" on page 2-101.

USER_CATALOG

USER_CATALOG lists tables, views, synonyms, and sequences owned by the current user. Its columns are the same as those in "ALL_CATALOG" on page 2-7.

USER_CLU_COLUMNS

USER_CLU_COLUMNS maps columns in the current user's tables to cluster columns. Its columns are the same as those in "DBA_CLU_COLUMNS" on page 2-103.

USER_CLUSTERS

USER_CLUSTERS describes all the clusters owned by the current user. Its columns are the same as those in "ALL_CLUSTERS" on page 2-8.

USER_CLUSTER_HASH_EXPRESSIONS

USER_CLUSTER_HASH_EXPRESSIONS lists hash functions for the hash clusters owned by the current user. Its columns are the same as those in "ALL_CLUSTER_HASH_EXPRESSIONS" on page 2-9.

USER_COL_COMMENTS

USER_COL_COMMENTS describes comments on columns of tables and views in the current user's schema. Its columns are the same as those in "ALL_COL_COMMENTS" on page 2-9.

USER_COL_PRIVS

USER_COL_PRIVS describes all column object grants for which the current user is the owner, grantor, or grantee. Its columns are the same as those in "ALL_COL_PRIVS" on page 2-10.

USER_COL_PRIVS_MADE

USER_COL_PRIVS_MADE describes all column object grants for which the current user is the grantor. Its columns are the same as those in "ALL_COL_PRIVS_MADE" on page 2-11.

USER_COL_PRIVS_RECD

USER_COL_PRIVS_RECD describes column object grants for which the current user is the grantee. Its columns are the same as those in "ALL_COL_PRIVS_RECD" on page 2-11.

USER_COLL_TYPES

USER_COLL_TYPES describes named collection types (VARRAYs, nested tables, object tables, and so on) in the current user's schema. Its columns are the same as those in "ALL_COLL_TYPES" on page 2-12.

USER_CONS_COLUMNS

USER_CONS_COLUMNS describes columns that are owned by the current user and that are specified in constraint definitions. Its columns are the same as those in "ALL_CONS_COLUMNS" on page 2-13.

USER_CONSTRAINTS

USER_CONSTRAINTS describes all constraint definitions on tables owned by the current user. Its columns are the same as those in "ALL_CONSTRAINTS" on page 2-13.

USER_DB_LINKS

USER_DB_LINKS describes database links owned by the current user. Its columns are the same as those in "ALL_DB_LINKS" on page 2-15.

USER_DEPENDENCIES

USER_DEPENDENCIES describes dependencies between procedures, packages, functions, package bodies, and triggers owned by the current user, including dependencies on views created without any database links. Its columns are the same as those in "ALL_DEPENDENCIES" on page 2-16.

USER_DIMENSIONS

USER_DIMENSIONS describes dimension objects in the user's schema. Its columns are the same as those in "ALL_DIMENSIONS" on page 2-17.

USER_DIM_ATTRIBUTES

USER_DIM_ATTRIBUTES describes the relationship between dimension levels and functionally dependent columns in the current user's schema. The level columns and the dependent column must be in the same table. This view's columns are the same as those in "ALL_DIM_ATTRIBUTES" on page 2-18.

USER_DIM_CHILD_OF

USER_DIM_CHILD_OF describes a hierarchical relationship of 1 to *n* between pairs of levels in dimensions owned by the current user. Its columns are the same as those in "ALL_DIM_CHILD_OF" on page 2-18.

USER_DIM_HIERARCHIES

USER_DIM_HIERARCHIES describes the dimension hierarchies owned by the current user. Its columns are the same as those in "ALL_DIM_HIERARCHIES" on page 2-19.

USER_DIM_JOIN_KEY

USER_DIM_JOIN_KEY describes the join between two dimension tables owned by the current user. The join is always specified between a parent dimension level column and a child column. This view's columns are the same as those in "ALL_DIM_JOIN_KEY" on page 2-19.

USER_DIM_LEVELS

USER_DIM_LEVELS describes the levels of dimensions owned by the current user. All columns of a dimension level must come from the same relation. This view's columns are the same as those in "ALL_DIM_LEVELS" on page 2-20.

USER_DIM_LEVEL_KEY

USER_DIM_LEVEL_KEY describes columns of dimension levels owned by the current user. This view's columns are the same as those in "ALL_DIM_LEVEL_KEY" on page 2-20.

USER_ERRORS

USER_ERRORS describes current errors on all stored objects (views, procedures, functions, packages, and package bodies) owned by the current user. Its columns are the same as those in "ALL_ERRORS" on page 2-21.

USER_EXTENTS

USER_EXTENTS describes extents of the segments belonging to objects owned by the current user. Its columns are a subset of those in "DBA_EXTENTS" on page 2-109.

USER_FREE_SPACE

USER_FREE_SPACE lists the free extents in tablespaces accessible to the user. Its columns are a subset of those in "DBA_FREE_SPACE" on page 2-110.

USER_IND_COLUMNS

USER_IND_COLUMNS describes the columns of the indexes owned by the current user and columns of indexes on tables owned by the current user. Its columns are the same as those in "ALL_IND_COLUMNS" on page 2-22.

USER_IND_EXPRESSIONS

USER_IND_EXPRESSIONS describes expressions of function-based indexes on tables owned by the current user. Its columns are the same as those in "ALL_IND_EXPRESSIONS" on page 2-23.

USER_IND_PARTITIONS

USER_IND_PARTITIONS describes, for each index partition owned by the current user, the partition-level partitioning information, the storage parameters for the partition, and various partition statistics analyzed by ANALYZE statements. Its columns are the same as those in "ALL_IND_PARTITIONS" on page 2-23.

USER_IND_SUBPARTITIONS

USER_IND_SUBPARTITIONS describes, for each index subpartition owned by the current user, the partition-level partitioning information, the storage parameters for the subpartition, and various partition statistics collected by ANALYZE statements. Its columns are the same as those in "ALL_IND_SUBPARTITIONS" on page 2-25.

USER_INDEXES

USER_INDEXES describes indexes owned by the current user. To gather statistics for this view, use the SQL command ANALYZE. This view supports parallel partitioned index scans. Its columns are the same as those in "ALL_INDEXES" on page 2-27.

USER_INDEXTYPES

USER_INDEXTYPES describes all indextypes owned by the current user. Its columns are the same as those in "ALL_INDEXTYPES" on page 2-30.

USER_INDEXTYPE_OPERATORS

USER_INDEXTYPE_OPERATORS lists all the operators supported by indextypes owned by the current user. Its columns are the same as those in "ALL_INDEXTYPE_OPERATORS" on page 2-31.

USER_INTERNAL_TRIGGERS

USER_INTERNAL_TRIGGERS describes the internal triggers on all tables owned by the current user. Its columns are the same as those in "ALL_INTERNAL_TRIGGERS" on page 2-31.

USER_JOBS

USER_JOBS describes all jobs owned by the user. Its columns are the same as those in "ALL_JOBS" on page 2-32.

> **See Also:** *Oracle8i Administrator's Guide* for more information on jobs.

USER_LIBRARIES

USER_LIBRARIES describes all libraries owned by the current user. Its columns are the same as those in "ALL_LIBRARIES" on page 2-33.

USER_LOBS

USER_LOBS displays the user's CLOBs and BLOBs contained in the user's tables. BFILEs are stored outside the database, so they are not described by this view. this view's columns are the same as those in "ALL_LOBS" on page 2-33.

USER_LOB_PARTITIONS

USER_LOB_PARTITIONS displays the LOB partitions contained in tables owned by the current user. Its columns are the same as those in "ALL_LOB_PARTITIONS" on page 2-34.

USER_LOB_SUBPARTITIONS

USER_LOB_SUBPARTITIONS describes partition-level attributes of LOB data subpartitions owned by the current user. Its columns are the same as those in "ALL_LOB_SUBPARTITIONS" on page 2-35.

USER_METHOD_PARAMS

USER_METHOD_PARAMS describes method parameters of types owned by the current user. Its columns are the same as those in "ALL_METHOD_PARAMS" on page 2-37.

USER_METHOD_RESULTS

USER_METHOD_RESULTS describes the method results of types owned by the current user. Its columns are the same as those in "ALL_METHOD_RESULTS" on page 2-37.

USER_MVIEW_AGGREGATES

USER_MVIEW_AGGREGATES describes the grouping functions (aggregated measures) that appear in the SELECT list of aggregated materialized views owned by the current user. Its columns are the same as those in "ALL_MVIEW_AGGREGATES" on page 2-38.

USER_MVIEW_ANALYSIS

USER_MVIEW_ANALYSIS describes all materialized views owned by the current user that potentially support query rewrite and that provide additional information for analysis by applications. Its columns are the same as those in "ALL_MVIEW_ANALYSIS" on page 2-39.

Note: This view excludes materialized views that reference remote tables or that include references to non-static values such as SYSDATE or USER. This view also excludes materialized views that were created as snapshots prior to Oracle8*i* and that were never altered to enable query rewrite.

USER_MVIEW_DETAIL_RELATIONS

USER_MVIEW_DETAIL_RELATIONS represents the named detail relations that are either in the FROM list of a materialized view, or that are indirectly referenced through views in the FROM list. Its columns are the same as those in "ALL_MVIEW_DETAIL_RELATIONS" on page 2-41.

USER_MVIEW_JOINS

USER_MVIEW_JOINS describes a join between two columns in the WHERE clause of a subquery that defines a materialized view. Its columns are the same as those in "ALL_MVIEW_JOINS" on page 2-41.

USER_MVIEW_KEYS

USER_MVIEW_KEYS describes the columns or expressions in the SELECT list upon which materialized views in the current user's schema are based. Its columns are the same as those in "ALL_MVIEW_KEYS" on page 2-42.

USER_MVIEWS

USER_MVIEWS describes the materialized views owned by the current user. Its columns are the same as those in "ALL_MVIEWS" on page 2-43.

USER_NESTED_TABLES

USER_NESTED_TABLES describes the nested tables in tables owned by the current user. Its columns are the same as those in "ALL_NESTED_TABLES" on page 2-46.

USER_OBJECT_TABLES

USER_OBJECT_TABLES describes the object tables owned by the current user. Its columns are the same as those in "ALL_OBJECT_TABLES" on page 2-47.

USER_OBJECTS

USER_OBJECTS describes all objects owned by the current user. Its columns are the same as those in "ALL_OBJECTS" on page 2-49.

USER_OBJECT_SIZE

USER_OBJECT_SIZE lists the sizes, in bytes, of various PL/SQL objects. Its columns are the same as those in "DBA_OBJECT_SIZE" on page 2-116.

USER_OBJ_AUDIT_OPTS

This view, created by CATAUDIT.SQL, lists auditing characteristics for all objects owned by the current user. Its columns are the same as those in "DBA_OBJ_AUDIT_OPTS" on page 2-117.

USER_OPANCILLARY

USER_OPANCILLARY provides ancillary information for operators owned by the current user. Its columns are the same as those in "ALL_OPANCILLARY" on page 2-50.

USER_OPARGUMENTS

USER_OPARGUMENTS provides argument information for operator bindings owned by the current user. Its columns are the same as those in "ALL_OPARGUMENTS" on page 2-51.

USER_OPBINDINGS

USER_OPBINDINGS describes bindings of operators owned by the current user. Its columns are the same as those in "ALL_OPBINDINGS" on page 2-51.

USER_OPERATORS

USER_OPERATORS describes all operators owned by the current user. Its columns are the same as those in "ALL_OPERATORS" on page 2-52.

USER_OUTLINE_HINTS

USER_OUTLINE_HINTS describes the set of hints that make up the outlines owned by the current user. Its columns are the same as those in "ALL_OUTLINE_HINTS" on page 2-53.

USER_OUTLINES

USER_OUTLINES describes all outlines owned by the current user. Its columns are the same as those in "ALL_OUTLINES" on page 2-53.

USER_PART_COL_STATISTICS

USER_PART_COL_STATISTICS provides column statistics and histogram information for table partitions owned by the current user. Its columns are the same as those in "ALL_PART_COL_STATISTICS" on page 2-54.

USER_PART_HISTOGRAMS

USER_PART_HISTOGRAMS contains the histogram data (end-points per histogram) for histograms on table partitions that the current user can access. Its columns are the same as those in "ALL_PART_HISTOGRAMS" on page 2-55.

USER_PART_KEY_COLUMNS

USER_PART_KEY_COLUMNS describes the partitioning key columns for partitioned objects owned by the current user. Its columns are the same as those in "ALL_PART_KEY_COLUMNS" on page 2-57.

USER_PART_INDEXES

USER_PART_INDEXES describes object-level partitioning information for all partitioned indexes owned by the current user. Its columns are the same as those in "ALL_PART_INDEXES" on page 2-56.

USER_PART_LOBS

USER_PART_LOBS provides table-level information for partitioned LOBs owned by the current user, including default attributes for LOB data partitions. Its columns are the same as those in "ALL_PART_LOBS" on page 2-58.

USER_PART_TABLES

USER_PART_TABLES provides object-level partitioning information for partitioned tables owned by the current user. Its columns are the same as those in "ALL_PART_TABLES" on page 2-59.

USER_PARTIAL_DROP_TABS

USER_PARTIAL_DROP_TABS describes all tables in the schema of the current user that have partially completed DROP COLUMN operations. Its columns are the same as those in "ALL_PARTIAL_DROP_TABS" on page 2-61.

USER_PASSWORD_LIMITS

USER_PASSWORD_LIMITS describes the password profile parameters that are assigned to the user.

Column	Datatype	NULL	Description
RESOURCE_NAME	VARCHAR2(32)	NOT NULL	Name of the password resource
LIMIT	VARCHAR2(40)		Value of the resource limit

USER_POLICIES

USER_POLICIES describes security policies on all objects owned by the current user. Its columns are the same as those in "ALL_POLICIES" on page 2-61.

USER_QUEUE_SCHEDULES

USER_QUEUE_SCHEDULES lists information about queue schedules. Its columns are the same as those in "DBA_QUEUE_SCHEDULES" on page 2-121.

USER_QUEUE_TABLES

USER_QUEUE_TABLES describes the queues in the queue tables created in the user's schema. Its columns are the same as those in "ALL_QUEUE_TABLES" on page 2-62.

See Also: *Oracle8i Application Developer's Guide - Advanced Queuing* for more information about these views and Advanced Queuing.

USER_QUEUES

USER_QUEUES describes the operational characteristics of every queue in the user's schema. Its columns are the same as those in "ALL_QUEUES" on page 2-63.

See Also: *Oracle8i Application Developer's Guide - Advanced Queuing* for more information about these views and Advanced Queuing.

USER_REFRESH

USER_REFRESH describes all refresh groups owned by the current user. Its columns are the same as those in "ALL_REFRESH" on page 2-64.

USER_REFRESH_CHILDREN

USER_REFRESH_CHILDREN lists all the objects in refresh groups owned by the current user. Its columns are the same as those in "ALL_REFRESH_CHILDREN" on page 2-65.

USER_REFS

USER_REFS describes the REF columns and REF attributes in the object type columns of tables owned by the current user. Its columns are the same as those in "ALL_REFS" on page 2-66.

USER_REGISTERED_SNAPSHOTS

USER_REGISTERED_SNAPSHOTS describes all registered snapshots owned by the current user. Its columns are the same as those in "ALL_REGISTERED_SNAPSHOTS" on page 2-67.

USER_RESOURCE_LIMITS

USER_RESOURCE_LIMITS displays the resource limits for the current user.

Column	Datatype	NULL	Description
RESOURCE_NAME	VARCHAR2(32)	NOT NULL	Name of the resource
LIMIT	VARCHAR2(40)		Limit placed on this resource

USER_ROLE_PRIVS

USER_ROLE_PRIVS lists roles granted to the user.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)		Name of the user, or PUBLIC
GRANTED_ROLE	VARCHAR2(30)		Name of the role granted to the user
ADMIN_OPTION	VARCHAR2(3)		Granted with ADMIN option: YES | NO
DEFAULT_ROLE	VARCHAR2(3)		Role is designated as the user's default role: YES | NO
OS_GRANTED	VARCHAR2(3)		Granted by the operating system: Y / N (occurs if configuration parameter OS_ROLES = TRUE)

USER_RSRC_CONSUMER_GROUP_PRIVS

USER_RSRC_CONSUMER_GROUP_PRIVS lists all resource consumer groups granted to the user. Its columns are the same as those in "DBA_RSRC_CONSUMER_GROUP_PRIVS" on page 2-127.

USER_RSRC_MANAGER_SYSTEM_PRIVS

USER_RSRC_MANAGER_SYSTEM_PRIVS lists all the users who are granted system privileges for the DBMS_RESOURCE_MANAGER package. Its columns are the same as those in "DBA_RSRC_MANAGER_SYSTEM_PRIVS" on page 2-128.

USER_RULESETS

USER_RULESETS lists information about rulesets. Its columns are the same as those in "DBA_RULESETS" on page 2-131.

USER_SEGMENTS

USER_SEGMENTS lists information about storage allocation for database segments belonging to a user's objects. Its columns are the same as those in "DBA_SEGMENTS" on page 2-131.

USER_SEQUENCES

USER_SEQUENCES describes all sequences owned by the current user. Its columns are the same as those in "ALL_SEQUENCES" on page 2-68.

USER_SNAPSHOTS

USER_SNAPSHOTS describes all snapshots owned by the current user. Its columns are the same as those in "ALL_SNAPSHOTS" on page 2-69.

USER_SNAPSHOT_LOGS

USER_SNAPSHOT_LOGS lists all snapshot logs owned by the current user. Its columns are the same as those in "ALL_SNAPSHOT_LOGS" on page 2-68.

USER_SNAPSHOT_REFRESH_TIMES

USER_SNAPSHOT_REFRESH_TIMES describes refresh times of snapshots owned by the current user. Its columns are the same as those in "ALL_SNAPSHOT_REFRESH_TIMES" on page 2-69.

USER_SOURCE

USER_SOURCE contains text source of all stored objects belonging to the user. Its columns are the same as those in "ALL_SOURCE" on page 2-71.

USER_SUBPART_COL_STATISTICS

USER_SUBPART_COL_STATISTICS provides column statistics and histogram information for subpartitions of subpartitioned objects owned by the current user. Its columns are the same as those in "ALL_SUBPART_COL_STATISTICS" on page 2-72.

USER_SUBPART_HISTOGRAMS

USER_SUBPART_HISTOGRAMS lists actual histogram data (end-points per histogram) for histograms on table subpartitions owned by the current user. Its columns are the same as those in "ALL_SUBPART_HISTOGRAMS" on page 2-72.

USER_SUBPART_KEY_COLUMNS

USER_SUBPART_KEY_COLUMNS lists subpartitioning key columns for composite-partitioned tables (and local indexes on composite-partitioned tables) owned by the current user. Its columns are the same as those in "ALL_SUBPART_KEY_COLUMNS" on page 2-73.

USER_SYNONYMS

USER_SYNONYMS describes private synonyms (synonyms owned by the current user). Its columns are the same as those in "ALL_SYNONYMS" on page 2-74.

USER_SYS_PRIVS

USER_SYS_PRIVS lists system privileges granted to the current user. Its columns are the same as those in "DBA_SYS_PRIVS" on page 2-134.

USER_TAB_COL_STATISTICS

USER_TAB_COL_STATISTICS contains column statistics and histogram information extracted from "USER_TAB_COLUMNS" on page 2-182. Its columns are the same as those in "ALL_TAB_COL_STATISTICS" on page 2-75.

USER_TAB_COLUMNS

USER_TAB_COLUMNS describes the columns of tables, views, and clusters owned by the current user. Its columns are the same as those in "ALL_TAB_COLUMNS" on page 2-75. To gather statistics for this view, use the SQL command ANALYZE.

USER_TAB_COMMENTS

USER_TAB_COMMENTS describes comments on the tables and views owned by the current user. Its columns are the same as those in "ALL_TAB_COMMENTS" on page 2-77.

USER_TAB_HISTOGRAMS

USER_TAB_HISTOGRAMS describes histograms on columns of tables owned by the current user. Its columns are the same as those in "ALL_TAB_HISTOGRAMS" on page 2-77.

USER_TAB_MODIFICATIONS

USER_TAB_MODIFICATIONS describes modifications to all tables owned by the current user that have been modified since the last time statistics were gathered on the tables. Its columns are the same as those in "ALL_TAB_MODIFICATIONS" on page 2-78.

Note: This view is populated only for tables with the MONITORING attribute. It is intended for statistics collection over a long period of time and may not be populated until a few hours after the actual modifications occurred.

USER_TAB_PARTITIONS

USER_TAB_PARTITIONS describes partition-level partitioning information, partition storage parameters, and partition statistics determined by ANALYZE statements for all partitions owned by the current user. Its columns are the same as those in "ALL_TAB_PARTITIONS" on page 2-79.

USER_TAB_PRIVS

USER_TAB_PRIVS describes grants on objects where the current user is the object owner, grantor, or grantee. Its columns are the same as those in "ALL_TAB_PRIVS" on page 2-80.

USER_TAB_PRIVS_MADE

USER_TAB_PRIVS_MADE lists all grants on objects owned by the user. Its columns are the same as those in "ALL_TAB_PRIVS_MADE" on page 2-81.

USER_TAB_PRIVS_RECD

USER_TAB_PRIVS_RECD lists object grants for which the current user is the grantee. Its columns are the same as those in "ALL_TAB_PRIVS_RECD" on page 2-81.

USER_TAB_SUBPARTITIONS

USER_TAB_SUBPARTITIONS describes, for each table subpartition, the subpartition name, name of the table and partition to which it belongs, and its storage attributes. Its columns are the same as those in "ALL_TAB_SUBPARTITIONS" on page 2-82.

Note: Statistics are not collected on a per-subpartition basis.

USER_TABLES

USER_TABLES describes all relational tables owned by the current. Its columns are the same as those in "ALL_TABLES" on page 2-83. To gather statistics for this view, use the SQL command ANALYZE.

USER_TABLESPACES

USER_TABLESPACES contains descriptions of tablespaces accessible to the current user. Its columns are the same as those in "DBA_TABLESPACES" on page 2-136.

USER_TRIGGERS

USER_TRIGGERS describes all triggers owned by the current. Its columns are the same as those in "ALL_TRIGGERS" on page 2-86.

USER_TRIGGER_COLS

USER_TRIGGER_COLS describes the use of columns in triggers owned by the current user and triggers on tables owned by the current user. Its columns are the same as those in "ALL_TRIGGER_COLS" on page 2-87.

USER_TS_QUOTAS

USER_TS_QUOTAS contains information about tablespace quotas for the current user. Its columns are the same as those in "DBA_TS_QUOTAS" on page 2-138.

USER_TYPES

USER_TYPES describes all object types owned by the current user. Its columns are the same as those in "ALL_TYPES" on page 2-89.

USER_TYPE_ATTRS

USER_TYPE_ATTRS describes the attributes of object types owned by the current user. Its columns are the same as those in "ALL_TYPE_ATTRS" on page 2-88.

USER_TYPE_METHODS

USER_TYPE_METHODS describes methods of object types owned by the current user. Its columns are the same as those in "ALL_TYPE_METHODS" on page 2-88.

USER_UNUSED_COL_TABS

USER_UNUSED_COL_TABS contains a description of all tables containing unused columns. Its columns are the same as those in "ALL_UNUSED_COL_TABS" on page 2-90.

USER_UPDATABLE_COLUMNS

USER_UPDATABLE_COLUMNS describes columns in a join view that can be updated by the current user. Its columns are the same as those in "ALL_UPDATABLE_COLUMNS" on page 2-90.

See Also: *Oracle8i Concepts* for information on updatable join views.

USER_USERS

USER_USERS describes the current user. Its columns are the same as those in "DBA_USERS" on page 2-139.

USER_USTATS

USER_USTATS describes all user-defined statistics owned by the current user. Its columns are the same as those in "ALL_USTATS" on page 2-91.

USER_VARRAYS

USER_VARRAYS describes the varrays owned by the current user. Its columns are the same as those in "ALL_VARRAYS" on page 2-92.

USER_VIEWS

USER_VIEWS displays the text of views owned by the current user. Its columns are the same as those in "ALL_VIEWS" on page 2-92.

3

Dynamic Performance (V$) Views

This chapter describes the dynamic performance views, which are often referred to as V$ views.

About Dynamic Performance Views

Oracle contains a set of underlying views that are maintained by the server and accessible to the database administrator user SYS. These views are called **dynamic performance views** because they are continuously updated while a database is open and in use, and their contents relate primarily to performance.

Although these views appear to be regular database tables, they are not. These views provide data on internal disk structures and memory structures. You can select from these views, but you can never update or alter them.

Notes:

- You can query the dynamic performance views to extract information from them. However, only simple queries are supported. If sorts, joins, GROUP BY clauses and the like are needed, you should copy the information from each V$ view into a table (for example, using a CREATE TABLE ... AS SELECT statement), and then query from those tables.
- Because the information in the V$ views is dynamic, read consistency is not guaranteed for SELECT operations on these views.

The CATALOG.SQL script contains definitions of the views and public synonyms for the dynamic performance views. You must run CATALOG.SQL to create these views

and synonyms. After installation, only username SYS or anyone with SYSDBA role has access to the dynamic performance tables.

V$ Views

The actual dynamic performance views are identified by the prefix V_$. Public synonyms for these views have the prefix V$. Database administrators and other users should access only the V$ objects, not the V_$ objects.

The dynamic performance views are used by Enterprise Manager and Oracle Trace, which is the primary interface for accessing information about system performance. Once the instance is started, the V$ views that read from memory are accessible. Views that read data from disk require that the database be mounted, and some require that the database be open.

GV$ Views

For almost every V$ view described in this chapter, Oracle has a corresponding GV$ (global V$) view. In a parallel server environment, querying a GV$ view retrieves the V$ view information from all qualified instances. In addition to the V$ `information`, each GV$ view contains an extra column named `INST_ID` of datatype `INTEGER`. The `INST_ID` column displays the instance number from which the associated V$ view information was obtained. The INST_ID column can be used as a filter to retrieve V$ information from a subset of available instances. For example, the query:

```
SELECT * FROM GV$LOCK WHERE INST_ID = 2 OR INST_ID = 5
```

retrieves the information from the V$ views on instances 2 and 5.

In order to query the GV$ views, the value of the PARALLEL_MAX_SERVERS parameter must be greater than zero on all instances mounting the database.

See Also: *Oracle8i Parallel Server Concepts*.

The sections that follow list the columns and public synonyms for the dynamic performance views.

V$ACCESS

This view displays objects in the database that are currently locked and the sessions that are accessing them.

Column	Datatype	Description
SID	NUMBER	Session number that is accessing an object
OWNER	VARCHAR2(64)	Owner of the object
OBJECT	VARCHAR2(1000)	Name of the object
TYPE	VARCHAR2(12)	Type identifier for the object

V$ACTIVE_INSTANCES

This view maps instance names to instance numbers for all instances that have the database currently mounted.

Column	Datatype	Description
INST_NUMBER	NUMBER	The instance number
INST_NAME	VARCHAR2(60)	The instance name

V$AQ

This view describes statistics for the queues in the database.

Column	Datatype	Description
QID	NUMBER	The unique queue identifier
WAITING	NUMBER	Number of messages in the queue in the state 'WAITING'
READY	NUMBER	Number of messages in the queue in the state 'READY'
EXPIRED	NUMBER	Number of messages in the queue the state 'EXPIRED'
TOTAL_WAIT	NUMBER	Total wait time of all 'READY' messages in the queue
AVERAGE_WAIT	NUMBER	Average wait time of 'READY' messages in the queue

V$ARCHIVE

This view contains information on redo log files in need of archiving. Each row provides information for one thread. This information is also available in V$LOG. Oracle recommends that you use V$LOG.

See Also: "V$LOG" on page 3-61

Column	Datatype	Description
GROUP#	NUMBER	Log file group number
THREAD#	NUMBER	Log file thread number
SEQUENCE#	NUMBER	Log file sequence number
CURRENT	VARCHAR2(3)	Archive log currently in use
FIRST_CHANGE#	NUMBER	First SCN stored in the current log

V$ARCHIVE_DEST

This view describes, for the current instance, all the archive log destinations, their current value, mode, and status.

Column	Datatype	Description
DEST_ID	NUMBER	ID (1-5)
STATUS	VARCHAR2(9)	Status: ■ VALID: Initialized and available ■ INACTIVE: No destination information ■ DEFERRED: Manually disabled by the user ■ ERROR: Error during open or copy ■ DISABLED: Disabled after error ■ BAD PARAM: Parameter has errors
BINDING	VARCHAR2(9)	Requirement for success: ■ MANDATORY- must succeed ■ OPTIONAL - need not succeed (depends on LOG_ARCHIVE_MIS_SUCCEED_DEST)
NAME_SPACE	VARCHAR2(7)	Definition scope: ■ SYSTEM-System definition ■ SESSION-Session definition
TARGET	VARCHAR2(7)	Target: ■ PRIMARY-Copy to primary ■ STANDBY-Copy to standby
REOPEN_SECS	NUMBER	Retry time in seconds (after error)
DESTINATION	VARCHAR2(256)	Destination text string (translated primary location or standby service name)
FAIL_DATE	DATE	Date and time of any last error
FAIL_SEQUENCE	NUMBER	Any log sequence number at last error

Column	Datatype	Description
FAIL_BLOCK	NUMBER	Any block number at last error
ERROR	VARCHAR2(256)	Text of any last error

See Also:

- "LOG_ARCHIVE_DEST" on page 1-55 and "LOG_ARCHIVE_DEST_n" on page 1-56
- "LOG_ARCHIVE_DUPLEX_DEST" on page 1-58 and "LOG_ARCHIVE_DEST_STATE_n" on page 1-57
- "STANDBY_ARCHIVE_DEST" on page 1-115
- "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-60

V$ARCHIVED_LOG

This view displays archived log information from the control file including archive log names. An archive log record is inserted after the online redo log is successfully archived or cleared (name column is NULL if the log was cleared). If the log is archived twice, there will be two archived log records with the same THREAD#, SEQUENCE#, and FIRST_CHANGE#, but with a different name. An archive log record is also inserted when an archive log is restored from a backup set or a copy and whenever a copy of a log is made with the RMAN **copy** command.

Column	Datatype	Description
RECID	NUMBER	Archived log record ID
STAMP	NUMBER	Archived log record stamp
NAME	VARCHAR2(513)	Archived log file name. If set to NULL, the log file was cleared before it was archived
THREAD#	NUMBER	Redo thread number
SEQUENCE#	NUMBER	Redo log sequence number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change# of the database when this log was written
RESETLOGS_TIME	DATE	Resetlogs time of the database when this log was written
FIRST_CHANGE#	NUMBER	First change# in the archived log
FIRST_TIME	DATE	Timestamp of the first change
NEXT_CHANGE#	NUMBER	First change in the next log

Column	Datatype	Description
NEXT_TIME	DATE	Timestamp of the next change
BLOCKS	NUMBER	Size of the archived log in blocks
BLOCK_SIZE	NUMBER	Redo log block size
ARCHIVED	VARCHAR2(3)	indicates that the online redo log was archived (YES) or that RMAN only inspected the log and created a record for future application of redo logs during recovery. For additional information see *Oracle8i Recovery Manager User's Guide and Reference.*
DELETED	VARCHAR2(3)	specifies (with YES or NO) whether an RMAN DELETE command has physically deleted the archived log file from disk, as well as logically removing it from the control file of the target database and from the recovery catalog.
COMPLETION_TIME	DATE	Time when the archiving completed

V$ARCHIVE_PROCESSES

This view provides information about the state of the various ARCH processes for the instance.

Column	Datatype	Description
PROCESS	NUMBER	The identifier for the ARCH process for the instance, numbered from 0-9
STATUS	VARCHAR2(10)	The status of the ARCH process, displayed as a keyword. Possible values are: STOPPED, SCHEDULED, STARTING, ACTIVE, STOPPING, and TERMINATED.
LOG_SEQUENCE	NUMBER	This is the online redo log sequence number currently being archived, if STATE="BUSY"
STATE	VARCHAR2(4)	This is the current state of the ARCH process, displayed as a keyword. Possible keywords are: IDLE or BUSY

V$BACKUP

This view displays the backup status of all online datafiles.

Column	Datatype	Description
FILE#	NUMBER	File identifier
STATUS	VARCHAR2(18)	File status: NOT ACTIVE, ACTIVE (backup in progress), OFFLINE NORMAL, or description of an error
CHANGE#	NUMBER	System change number when backup started
TIME	DATE	Time the backup started

V$BACKUP_ASYNC_IO

This view displays performance information about ongoing and recently completed RMAN backups and restores. For each backup, it contains one row for each input datafile, one row for the aggregate total performance of all datafiles, and one row for the output backup piece. This data is not stored persistently, and is not preserved when the instance is re-started.

See Also: *Oracle8i Designing and Tuning for Performance* for information on how to use this table to tune backup performance.

Column	Datatype	Description
SID	NUMBER	The Oracle SID of the session doing the backup or restore
SERIAL	NUMBER	The use count for the SID doing the backup or restore
USE_COUNT	NUMBER	A counter that can be used to identify rows from different backup sets
DEVICE_TYPE	VARCHAR2(17)	The device type where the file is located
TYPE	VARCHAR2(9)	INPUT; OUTPUT; or AGGREGATE
STATUS	VARCHAR2(11)	NOT STARTED; IN PROGRESS; or FINISHED
FILENAME	VARCHAR2(513)	The name of the backup file being read or written
SET_COUNT	NUMBER	The set_count of the backup set being read or written
SET_STAMP	NUMBER	The set_stamp of the backup set being read or written
BUFFER_SIZE	NUMBER	The size of the buffers being used to read / write this file, in bytes
BUFFER_COUNT	NUMBER	The number of buffers being used to read / write this file
TOTAL_BYTES	NUMBER	The total number of bytes that will be read or written for this file, if known. If not known, this column will be null
OPEN_TIME	DATE	The time this file was opened. If TYPE='AGGREGATE', then this is the time that the first file in the aggregate was opened
CLOSE_TIME	DATE	The time this file was closed. If TYPE='AGGREGATE', then this is the time that the last file in the aggregate was closed
ELAPSED_TIME	NUMBER	The time, in 100ths of seconds, that the file was open
MAXOPENFILES	NUMBER	The number of concurrently open DISK files. This value is only present in rows where TYPE='AGGREGATE'
BYTES	NUMBER	The number of bytes read or written so far
EFFECTIVE_BYTES_PER_SECOND	NUMBER	The I/O rate that was achieved with this device during this backup
IO_COUNT	NUMBER	The number of I/Os that were performed to this file

Column	Datatype	Description
READY	NUMBER	The number of asynchronous requests for which a buffer was immediately ready for use
SHORT_WAITS	NUMBER	The number of times that a buffer was not immediately available, but a buffer became available after doing a non-blocking poll for I/O completion
SHORT_WAIT_TIME_TOTAL	NUMBER	The total time, in 100ths of seconds, taken by non-blocking polls for I/O completion
SHORT_WAIT_TIME_MAX	NUMBER	The maximum time taken for a non-blocking poll for I/O completion, in 100ths of seconds
LONG_WAITS	NUMBER	The number of times that a buffer was not immediately available, and only became available after a blocking wait was issued
LONG_WAIT_TIME_TOTAL	NUMBER	The total time, in 100ths of seconds, taken by blocking waits for I/O completion
LONG_WAIT_TIME_MAX	NUMBER	The maximum time taken for a blocking wait for I/O completion, in 100ths of seconds

V$BACKUP_CORRUPTION

This view displays information about corruptions in datafile backups from the control file. Note that corruptions are not tolerated in the control file and archived log backups.

Column	Datatype	Description
RECID	NUMBER	Backup corruption record ID
STAMP	NUMBER	Backup corruption record stamp
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup set count
PIECE#	NUMBER	Backup piece number
FILE#	NUMBER	Datafile number
BLOCK#	NUMBER	First block of the corrupted range
BLOCKS	NUMBER	Number of contiguous blocks in the corrupted range
CORRUPTION_CHANGE#	NUMBER	Change# at which the logical corruption was detected. Set to 0 to indicate media corruption
MARKED_CORRUPT	VARCHAR2(3)	YES | NO. If set to YES the blocks were not marked corrupted in the datafile, but were detected and marked as corrupted while making the datafile backup

V$BACKUP_DATAFILE

This view displays backup datafile and backup control file information from the control file.

Column	Datatype	Description
RECID	NUMBER	Backup datafile record ID
STAMP	NUMBER	Backup datafile record stamp
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup set count
FILE#	NUMBER	Datafile number. Set to 0 for control file
CREATION_CHANGE#	NUMBER	Creation change of the datafile
CREATION_TIME	DATE	Creation timestamp of the datafile
RESETLOGS_CHANGE#	NUMBER	Resetlogs change# of the datafile when it was backed up
RESETLOGS_TIME	DATE	Resetlogs timestamp of the datafile when it was backed up
INCREMENTAL_LEVEL	NUMBER	(0-4) incremental backup level
INCREMENTAL_CHANGE#	NUMBER	All blocks changed after incremental change# is included in this backup. Set to 0 for a full backup
CHECKPOINT_CHANGE#	NUMBER	All changes up to checkpoint change# are included in this backup
CHECKPOINT_TIME	DATE	Timestamp of the checkpoint
ABSOLUTE_FUZZY_CHANGE#	NUMBER	Highest change# in this backup
MARKED_CORRUPT	NUMBER	Number of blocks marked corrupt
MEDIA_CORRUPT	NUMBER	Number of blocks media corrupt
LOGICALLY_CORRUPT	NUMBER	Number of blocks logically corrupt
DATAFILE_BLOCKS	NUMBER	Size of the datafile in blocks at backup time. This value is also the number of blocks taken by the datafile restarted from this backup
BLOCKS	NUMBER	Size of the backup datafile in blocks. Unused blocks are not copied to the backup
BLOCK_SIZE	NUMBER	Block size
OLDEST_OFFLINE_RANGE	NUMBER	The RECID of the oldest offline range record in this backup control file. 0 for datafile backups
COMPLETION_TIME	DATE	The time completed

V$BACKUP_DEVICE

This view displays information about supported backup devices. If a device type does not support named devices, then one row with the device type and a null device name is returned for that device type. If a device type supports named devices then one row is returned for each available device of that type. The special device type DISK is not returned by this view because it is always available.

Column	Datatype	Description
DEVICE_TYPE	VARCHAR2(17)	Type of the backup device
DEVICE_NAME	VARCHAR2(512)	Name of the backup device

V$BACKUP_PIECE

This view displays information about backup pieces from the control file. Each backup set consist of one or more backup pieces.

Column	Datatype	Description
RECID	NUMBER	Backup piece record ID
STAMP	NUMBER	Backup piece record stamp
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup set count
PIECE#	NUMBER	Backup piece number (1-N)
COPY#	NUMBER	Indicates the copy number for backup pieces created with duplex enabled. 1 if the backup piece is not duplexed
DEVICE_TYPE	VARCHAR2(17)	Type of the device on which the backup piece resides. Set to DISK for backup sets on disk. See V$BACKUP_DEVICE
HANDLE	VARCHAR2(513)	Backup piece handle identifies the backup piece on restore
COMMENTS	VARCHAR2(81)	Comment returned by the operating system or storage subsystem. Set to NULL for backup pieces on disk. This value is informational only; not needed for restore
MEDIA	VARCHAR2(65)	Name of the media on which the backup piece resides. This value is informational only; not needed for restore
MEDIA_POOL	NUMBER	The media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager BACKUP command
CONCUR	VARCHAR2(3)	YES | NO, Indicates whether the piece on a media that can be accessed concurrently
TAG	VARCHAR2(32)	Backup piece tag. The tag is specified at backup set level, but stored at piece level

Column	Datatype	Description
STATUS	VARCHAR2(1)	Indicates the status of the piece: A (available), D (deleted), or X (expired)
START_TIME	DATE	The starting time
COMPLETION_TIME	DATE	The completion time
ELAPSED_SECONDS	NUMBER	The number of elapsed seconds

V$BACKUP_REDOLOG

This view displays information about archived logs in backup sets from the control file. Note that online redo logs cannot be backed up directly; they must be archived first to disk and then backed up. An archive log backup set can contain one or more archived logs.

Column	Datatype	Description
RECID	NUMBER	Record ID for this row. It is an integer that identifies this row
STAMP	NUMBER	Timestamp used with RECID to uniquely identify this row
SET_STAMP	NUMBER	One of the foreign keys for the row of the V$BACKUP_SET table that identifies this backup set
SET_COUNT	NUMBER	One of the foreign keys for the row of the V$BACKUP_SET table that identifies this backup set
THREAD#	NUMBER	Thread number for the log
SEQUENCE#	NUMBER	Log sequence number
RESETLOGS_CHANGE#	NUMBER	Change number of the last resetlogs before the log was written
RESETLOGS_TIME	DATE	Change time of the last resetlogs before the log was written. These will be the same for all logs in a backup set
FIRST_CHANGE#	NUMBER	SCN when the log was switched into. The redo in the log is at this SCN and greater
FIRST_TIME	DATE	Time allocated when the log was switched into
NEXT_CHANGE#	NUMBER	SCN when the next log in this thread was switched into. The redo in the log is below this SCN.
NEXT_TIME	DATE	Time when the next log in this thread was switched into
BLOCKS	NUMBER	Size of the log in logical blocks including the header block
BLOCK_SIZE	NUMBER	Size of the log blocks in bytes

V$BACKUP_SET

This view displays backup set information from the control file. A backup set record is inserted after the backup set is successfully completed.

Column	Datatype	Description
RECID	NUMBER	Backup set record ID
STAMP	NUMBER	Backup set record timestamp
SET_STAMP	NUMBER	Backup set stamp. The backup set stamp and count uniquely identify the backup set Primary key for the V$BACKUP_SET table, and the foreign key for the following tables: V$BACKUP_PIECE; V$BACKUP_DATAFILE, V$BACKUP_REDOLOG; V$BACKUP_CORRUPTION
SET_COUNT	NUMBER	Backup set count. The backup set count is incremented by one every time a new backup set is started (if the backup set is never completed the number is "lost"). If the control file is recreated then the count is reset to 1. Therefore the count must be used with the stamp to uniquely identify a backup set Primary key for the V$BACKUP_SET table, and the foreign key for the following tables: V$BACKUP_PIECE; V$BACKUP_DATAFILE; V$BACKUP_REDOLOG; V$BACKUP_CORRUPTION
BACKUP_TYPE	VARCHAR2(1)	Type of files that are in this backup. If the backup contains archived redo logs, the value is `L'. If this is a datafile full backup, the value is `D'. If this is an incremental backup, the value is `I'
CONTROLFILE_INCLUDED	VARCHAR2(3)	Set to YES if there is a control file included in this backup set, otherwise set to NO
INCREMENTAL_LEVEL	NUMBER	Location where this backup set fits into the database's backup strategy. Set to zero for full datafile backups, non-zero for incremental datafile backups, and NULL for archivelog backups
PIECES	NUMBER	Number of distinct backup pieces in the backup set
START_TIME	DATE	The starting time
COMPLETION_TIME	DATE	The time that this backup set completed
ELAPSED_SECONDS	NUMBER	The number of elapsed seconds
BLOCK_SIZE	NUMBER	Block size of the backup set

V$BACKUP_SYNC_IO

This view displays performance information about ongoing and recently completed RMAN backups and restores. For each backup, it contains one row for each input datafile, one row for the aggregate total performance of all datafiles, and one row for the output backup piece. This data is not stored persistently, and is not preserved when the instance is re-started.

See Also: *Oracle8i Designing and Tuning for Performance* for information on how to use this table to tune backup performance.

Column	Datatype	Description
SID	NUMBER	The Oracle SID of the session doing the backup or restore
SERIAL	NUMBER	The use count for the SID doing the backup or restore
USE_COUNT	NUMBER	A counter that can be used to identify rows from different backup sets
DEVICE_TYPE	VARCHAR2(17)	The device type where the file is located
TYPE	VARCHAR2(9)	INPUT; OUTPUT; or AGGREGATE
STATUS	VARCHAR2(11)	NOT STARTED; IN PROGRESS; or FINISHED
FILENAME	VARCHAR2(513)	The name of the backup file being read or written
SET_COUNT	NUMBER	The set_count of the backup set being read or written
SET_STAMP	NUMBER	The set_stamp of the backup set being read or written
BUFFER_SIZE	NUMBER	The size of the buffers being used to read / write this file, in bytes
BUFFER_COUNT	NUMBER	The number of buffers being used to read / write this file
TOTAL_BYTES	NUMBER	The total number of bytes that will be read or written for this file, if known. If not known, this column will be null
OPEN_TIME	DATE	The time this file was opened. If TYPE='AGGREGATE', then this is the time that the first file in the aggregate was opened
CLOSE_TIME	DATE	The time this file was closed. If TYPE='AGGREGATE', then this is the time that the last file in the aggregate was closed
ELAPSED_TIME	NUMBER	The time, in 100ths of seconds, that the file was open
MAXOPENFILES	NUMBER	The number of concurrently open DISK files. This value is only present in rows where TYPE='AGGREGATE'
BYTES	NUMBER	The number of bytes read or written so far
EFFECTIVE_BYTES_PER_SECOND	NUMBER	The I/O rate that was achieved with this device during this backup
IO_COUNT	NUMBER	The number of I/Os that were performed to this file
IO_TIME_TOTAL	NUMBER	The total time, in 100ths of seconds, taken to do I/O for this file
IO_TIME_MAX	NUMBER	The maximum time taken for a single I/O request
DISCRETE_BYTES_PER_SECOND	NUMBER	The average transfer rate for this file

V$BGPROCESS

This view describes the background processes.

Column	Datatype	Description
PADDR	RAW(4)	Address of the process state object
NAME	VARCHAR2	Name of this background process
DESCRIPTION	VARCHAR2	Description of the background process
ERROR	NUMBER	Error encountered

V$BH

This is an Oracle Parallel Server view. This view gives the status and number of pings for every buffer in the SGA.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query DBA_DATA_FILES or V$DBFILES)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	The class number
STATUS	VARCHAR2(1)	Status of the buffer: ■ FREE= not currently in use ■ XCUR= exclusive ■ SCUR= shared current ■ CR= consistent read ■ READ= being read from disk ■ MREC= in media recovery mode ■ IREC= in instance recovery mode
XNC	NUMBER	Number of PCM x to null lock conversions due to contention with another instance. This column is obsolete but is retained for historical compatibility.
LOCK_ELEMENT_ADDR	RAW(4)	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_NAME	NUMBER	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_CLASS	NUMBER	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
FORCED_READS	NUMBER	Number of times the block had to be made re-read from disk because another instance had forced it out of this instance's cache by requesting the PCM lock on this block in lock mode.

Column	Datatype	Description
FORCED_WRITES	NUMBER	Number of times DBWR had to write this block to disk because this instance had dirtied the block and another instance had requested the PCM lock on the block in conflicting mode.
DIRTY	VARCHAR2(1)	Y = block modified.
TEMP	VARCHAR2(1)	Y = temporary block
PING	VARCHAR2(1)	Y = block pinged
STALE	VARCHAR2(1)	Y = block is stale
DIRECT	VARCHAR2(1)	Y = direct block
NEW	VARCHAR2(1)	Always set to N. This column is obsolete but is retained for historical compatibility
OBJD	NUMBER	Database object number of the block that the buffer represents
TS#	NUMBER	Tablespace number of block

See Also: *Oracle8i Parallel Server Concepts*.

V$BSP

This view displays statistics on the block server background processes (BSP*n*) used in cache fusion.

Note: This view contains internal diagnostic information for use by Oracle support personnel. It is subject to change without notice.

See Also: *Oracle8i Parallel Server Concepts* for more information on these processes and on cache fusion.

Column	Datatype	Description
CR_REQUESTS	NUMBER	Number of requests received for a version of a block at a specific SCN. ("CR" refers to "consistent-read".)
CURRENT_REQUESTS	NUMBER	Number of requests for the most recent (current) version of a block. CR_REQUESTS + CURRENT_REQUESTS = total number of requests handled by the BSP*n* processes.
DATA_REQUESTS	NUMBER	Number of current or CR requests for data blocks.
UNDO_REQUESTS	NUMBER	Number of CR requests for undo blocks.

Column	Datatype	Description
TX_REQUESTS	NUMBER	Number of CR requests for undo segment header blocks. DATA_REQUESTS + UNDO_REQUESTS + TX_REQUESTS = total number of requests handled by the BSP*n* processes.
CURRENT_RESULTS	NUMBER	Number of requests for which no changes were rolled out of the block returned to the requesting instance.
PRIVATE_RESULTS	NUMBER	Number of requests for which changes were rolled out of the block returned to the requesting instance, and only the requesting transaction can use the resulting CR block.
ZERO_RESULTS	NUMBER	Number of requests for which changes were rolled out of the block returned to the requesting instance. Only zero-XID transactions can use the block.
DISK_READ_RESULTS	NUMBER	Number of requests for which the requesting instance had to read the requested block from disk.
FAIL_RESULTS	NUMBER	Number of requests that failed; the requesting transaction must reissue the request.
FAIRNESS_DOWN_CONVERTS	NUMBER	Number of times an instance receiving a request has down-converted an X lock on a block because it was not modifying the block
FAIRNESS_CLEARS	NUMBER	Number of times the "fairness counter" was cleared. This counter tracks the number of times a block was modified after it was served.
FREE_LOCK_ELEMENTS	NUMBER	Number of times a request was received from another instance and the X lock had no buffers.
FLUSHES	NUMBER	Number of times the log has been flushed by a BSP*n* process.
LIGHT_WORKS	NUMBER	Number of times the light-work rule was evoked. This rule prevents the BSP back ground process from going to disk while responding to CR requests for data, undo, or undo segment header blocks. This rule can prevent the BSP process from completing its response to the CR request.

V$BUFFER_POOL

This view displays information about all buffer pools available for the instance. The "sets" pertain to the number of LRU latch sets.

See Also: "DB_BLOCK_LRU_LATCHES" on page 1-26.

Column	Datatype	Description
ID	NUMBER	Buffer pool ID number
NAME	VARCHAR2	Buffer pool name
LO_SETID	NUMBER	Low set ID number

Column	Datatype	Description
HI_SETID	NUMBER	High set ID number
SET_COUNT	NUMBER	Number of sets in this buffer pool. This is HI_SETID - LO_SETID + 1
BUFFERS	NUMBER	Number of buffers allocated to the buffer pool
LO_BNUM	NUMBER	Low buffer number for this pool
HI_BNUM	NUMBER	High buffer number for this pool

V$BUFFER_POOL_STATISTICS

V$BUFFER_POOL_STATISTICS displays information about all buffer pools available for the instance. The "sets" pertain to the number of LRU latch sets.

See Also: "DB_BLOCK_LRU_LATCHES" on page 1-26.

Column	Datatype	Description
ID	NUMBER	Buffer pool ID number
NAME	VARCHAR2(20)	Buffer pool name
SET_MSIZE	NUMBER	Buffer pool maximum set size
CNUM_REPL	NUMBER	Number of buffers on replacement list
CNUM_WRITE	NUMBER	Number of buffers on write list
CNUM_SET	NUMBER	Number of buffers in set
BUF_GOT	NUMBER	Number of buffers gotten by the set
SUM_WRITE	NUMBER	Number of buffers written by the set
SUM_SCAN	NUMBER	Number of buffers scanned in the set
FREE_BUFFER_WAIT	NUMBER	Free buffer wait statistic
WRITE_COMPLETE_WAIT	NUMBER	Write complete wait statistic
BUFFER_BUSY_WAIT	NUMBER	Buffer busy wait statistic
FREE_BUFFER_INSPECTED	NUMBER	Free buffer inspected statistic
DIRTY_BUFFERS_INSPECTED	NUMBER	Dirty buffers inspected statistic
DB_BLOCK_CHANGE	NUMBER	Database blocks changed statistic
DB_BLOCK_GETS	NUMBER	Database blocks gotten statistic
CONSISTENT_GETS	NUMBER	Consistent gets statistic
PHYSICAL_READS	NUMBER	Physical reads statistic
PHYSICAL_WRITES	NUMBER	Physical writes statistic

V$CACHE

This is an Oracle Parallel Server view. This view contains information from the block header of each block in the SGA of the current instance as related to particular database objects.

See Also: *Oracle8i Parallel Server Concepts*.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query DBA_DATA_FILES or V$DBFILES)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	Class number
STATUS	VARCHAR2(1)	Status of block: ■ FREE = not currently in use ■ XCUR = exclusive ■ SCUR = shared current ■ CR = consistent read ■ READ = being read from disk ■ MREC = in media recovery mode ■ IREC = in instance recovery mode
XNC	NUMBER	Number of PCM x to null lock conversions due to contention with another instance. This column is obsolete but is retained for historical compatibility
FORCED_READS	NUMBER	The forced reads
FORCED_WRITES	NUMBER	The forced writes
NAME	VARCHAR2(30)	Name of the database object containing the block
PARTITION_NAME	VARCHAR2(30)	The name of the partition; NULL for non-partitioned objects
KIND	VARCHAR2(12)	Type of database object:

Column	Datatype	Description
		■ 1: index ■ 2: table ■ 3: cluster ■ 4: view ■ 5: synonym ■ 6: sequence ■ 7: procedure ■ 8: function ■ 9: package ■ 10: nonexistent ■ 11: package body ■ 12: trigger ■ 13: type ■ 14: type body ■ 19: table partition ■ 20: index partition ■ 21: LOB ■ 22: library ■ Null: Unknown
OWNER#	NUMBER	Owner number
LOCK_ELEMENT_ADDR	RAW(4)	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock
LOCK_ELEMENT_NAME	NUMBER	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock

V$CACHE_LOCK

This is an Oracle Parallel Server view. V$CACHE_LOCK is similar to V$CACHE, except for the platform-specific lock manager identifiers. This information may be useful if the platform- specific lock manager provides tools for monitoring the PCM lock operations that are occurring. For example, first query to find the lock element address using INDX and CLASS, then query V$BH to find the buffers that are covered by the lock. See also "V$CACHE" on page 3-18.

See Also: *Oracle8i Parallel Server Concepts*.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query DBA_DATA_FILES or V$DBFILES)
BLOCK#	NUMBER	Block number

Column	Datatype	Description
STATUS	VARCHAR2(4)	Status of block: ■ FREE: not currently in use ■ XCUR: exclusive ■ SCUR: shared current ■ CR: consistent read ■ READ: being read from disk ■ MREC: in media recovery mode ■ IREC: in instance recovery mode
XNC	NUMBER	Number of parallel cache management (PCM) lock conversions due to contention with another instance
NAME	VARCHAR2(30)	Name of the database object containing the block
KIND	VARCHAR2(12)	Type of database object: ■ 1: index ■ 2: table ■ 3: cluster ■ 4: view ■ 5: synonym ■ 6: sequence ■ 7: procedure ■ 8: function ■ 9: package ■ 10: nonexistent ■ 11: package body ■ 12: trigger ■ 13: type ■ 14: type body ■ 19: table partition ■ 20: index partition ■ 21: LOB ■ 22: library ■ Null: Unknown
OWNER#	NUMBER	Owner number
LOCK_ELEMENT_ADDR	RAW(4)	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock
LOCK_ELEMENT_NAME	NUMBER	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock
FORCED_READS	NUMBER	Number of times the block had to be made re-read from disk because another instance had forced it out of this instance's cache by requesting the PCM lock on this block in lock mode
FORCED_WRITES	NUMBER	Number of times DBWR had to write this block to disk because this instance had dirtied the block and another instance had requested the PCM lock on the block in conflicting mode
INDX	NUMBER	Platform-specific lock manager identifier
CLASS	NUMBER	Platform-specific lock manager identifier

V$CIRCUIT

This view contains information about virtual circuits, which are user connections to the database through dispatchers and servers.

Column	Datatype	Description
CIRCUIT	RAW(4)	Circuit address
DISPATCHER	RAW(4)	Current dispatcher process address
SERVER	RAW(4)	Current server process address
WAITER	RAW(4)	Address of server process that is waiting for the (currently busy) circuit to become available
SADDR	RAW(4)	Address of session bound to the circuit
STATUS	VARCHAR2(16)	Status of the circuit: ■ BREAK (currently interrupted) ■ EOF (about to be removed) ■ OUTBOUND (an outward link to a remote database) ■ NORMAL (normal circuit into the local database)
QUEUE	VARCHAR2(16)	Queue the circuit is currently on: ■ COMMON (on the common queue, waiting to be picked up by a server process) ■ DISPATCHER (waiting for the dispatcher) ■ SERVER (currently being serviced) ■ NONE (idle circuit)
MESSAGE0	NUMBER	Size in bytes of the messages in the first message buffer
MESSAGE1	NUMBER	Size in bytes of the messages in the second message buffer
MESSAGE2	NUMBER	Size in bytes of the messages in the third message buffer
MESSAGE3	NUMBER	Size in bytes of the messages in the fourth message buffer
MESSAGES	NUMBER	Total number of messages that have gone through this circuit
BYTES	NUMBER	Total number of bytes that have gone through this circuit
BREAKS	NUMBER	Total number of breaks (interruptions) for this circuit
PRESENTATION	VARCHAR2(16)	The presentation protocol used by the client and server

V$CLASS_PING

V$CLASS_PING displays the number of blocks pinged per block class. Use this view to compare contentions for blocks in different classes.

Column	Datatype	Description
CLASS	NUMBER	Number that represents the block class
X_2_NULL	NUMBER	Number of lock conversions from Exclusive-to-NULL for all blocks of the specified CLASS
X_2_NULL_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified CLASS due to Exclusive-to-NULL conversions
X_2_NULL_FORCED_STALE	NUMBER	Number of times a block in the CLASS was made STALE due to Exclusive-to-NULL conversions
X_2_S	NUMBER	Number of lock conversions from Exclusive-to-Shared for all blocks of the specified CLASS
X_2_S_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified CLASS due to Exclusive-to-Shared conversions
X_2_SSX	NUMBER	Number of lock conversions from Exclusive-to-Sub Shared Exclusive for all blocks of the specified CLASS
X_2_SSX_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified CLASS due to Exclusive-to-Sub Shared Exclusive conversions
S_2_NULL	NUMBER	Number of lock conversions from Shared-to-NULL for all blocks of the specified CLASS
S_2_NULL_FORCED_STALE	NUMBER	Number of times a block in the CLASS was made STALE due to Shared-to-NULL conversions
SS_2_NULL	NUMBER	Number of lock conversions from Sub Shared-to-NULL for all blocks of the specified CLASS
NULL_2_X	NUMBER	Number of lock conversions from NULL-to-Exclusive for all blocks of the specified CLASS
S_2_X	NUMBER	Number of lock conversions from Shared-to-Exclusive for all blocks of the specified CLASS
SSX_2_X	NUMBER	Number of lock conversions from Sub Shared Exclusive-to-Exclusive for all blocks of the specified CLASS
NULL_2_S	NUMBER	Number of lock conversions from NULL-to-Shared for all blocks of the specified CLASS
NULL_2_SS	NUMBER	Number of lock conversions from NULL-to-Sub Shared for all blocks of the specified CLASS

V$COMPATIBILITY

This view displays features in use by the database instance that may prevent downgrading to a previous release. This is the dynamic (SGA) version of this information, and may not reflect features that other instances have used, and may include temporary incompatibilities (like UNDO segments) that will not exist after the database is shut down cleanly.

Column	Datatype	Description
TYPE_ID	VARCHAR2(8	Internal feature identifier
RELEASE	VARCHAR2(60)	Release in which that feature appeared
DESCRIPTION	VARCHAR2(64)	Description of the feature

V$COMPATSEG

This view lists the permanent features in use by the database that will prevent moving back to an earlier release.

Column	Datatype	Description
TYPE_ID	VARCHAR2(8)	Internal feature identifier
RELEASE	VARCHAR2(60)	Release in which that feature appeared. The software must be able to interpret data formats added in that release
UPDATED	VARCHAR2(60)	Release that first used the feature

V$CONTEXT

This view lists set attributes in the current session.

Column	Datatype	Description
NAMESPACE	VARCHAR2(30)	The namespace that the attribute is in
ATTRIBUTE	VARCHAR2(30)	The name of attribute
VALUE	VARCHAR2(4000)	The value of attribute

V$CONTROLFILE

This view lists the names of the control files.

Column	Datatype	Description
STATUS	VARCHAR2(7)	INVALID if the name cannot be determined, which should not occur. NULL if the name can be determined
NAME	VARCHAR2(257)	The name of the control file

V$CONTROLFILE_RECORD_SECTION

This view displays information about the control file record sections.

Column	Datatype	Description
TYPE	VARCHAR2(17)	Identifies the type of record section: DATABASE, CKPT PROGRESS, REDO THREAD, REDO LOG, DATAFILE, FILENAME, TABLESPACE, LOG HISTORY, OFFLINE RANGE, ARCHIVED LOG, BACKUP SET, BACKUP PIECE, BACKUP DATAFILE, BACKUP REDOLOG, DATAFILE COPY, BACKUP CORRUPTION, COPY CORRUPTION, DELETED OBJECT, or PROXY COPY
RECORD_SIZE	NUMBER	Record size in bytes
RECORDS_TOTAL	NUMBER	Number of records allocated for the section
RECORDS_USED	NUMBER	Number of records used in the section
FIRST_INDEX	NUMBER	Index (position) of the first record
LAST_INDEX	NUMBER	Index of the last record
LAST_RECID	NUMBER	Record ID of the last record

V$COPY_CORRUPTION

This view displays information about datafile copy corruptions from the control file.

Column	Datatype	Description
RECID	NUMBER	Copy corruption record ID
STAMP	NUMBER	Copy corruption record stamp
COPY_RECID	NUMBER	Datafile copy record ID
COPY_STAMP	NUMBER	Datafile copy record stamp
FILE#	NUMBER	Datafile number
BLOCK#	NUMBER	First block of the corrupted range
BLOCKS	NUMBER	Number of contiguous blocks in the corrupted range
CORRUPTION_CHANGE#	NUMBER	Change# at which the logical corruption was detected. Set to 0 to indicate media corruption
MARKED_CORRUPT	VARCHAR2(3)	YES | NO. If set to YES the blocks were not marked corrupted in the datafile, but were detected and marked as corrupted while making the datafile copy

V$DATABASE

This view contains database information from the control file.

Column	Datatype	Description
DBID	NUMBER	The database ID
NAME	VARCHAR2	Name of the database
CREATED	DATE	Creation date
LOG_MODE	VARCHAR2	Archive log mode: NOARCHIVELOG or ARCHIVELOG
CHECKPOINT_ CHANGE#	NUMBER	Last SCN checkpointed
ARCHIVE_CHANGE#	NUMBER	Last SCN archived
DBID	NUMBER	Database ID calculated when database is created and stored in all file headers
RESETLOGS_CHANGE#	NUMBER	Change# at open resetlogs
RESETLOGS_TIME	DATE	Timestamp of open resetlogs
PRIOR_RESETLOGS_CHANGE#	NUMBER	Change# at prior resetlogs
PRIOR_RESETLOGS_TIME	DATE	Timestamp of prior resetlogs
CONTROLFILE_TYPE	VARCHAR2(7)	The type of control file: ■ STANDBY indicates database is in standby mode. ■ CLONE indicates a clone database. ■ BACKUP | CREATED indicates database is being recovered using a backup or created control file. ■ CURRENT: The control file changes to this type following a standby database activate or database open after recovery.
CONTROLFILE_CREATED	DATE	Control file creation timestamp
CONTROLFILE_SEQUENCE#	NUMBER	Control file sequence number incremented by control file transactions
CONTROLFILE_CHANGE#	NUMBER	Last change# in backup control file. Set to NULL if the control file is not a backup
CONTROLFILE_TIME	DATE	Last timestamp in backup control file. Set to NULL if the control file is not a backup
OPEN_RESETLOGS	VARCHAR2(11)	NOT ALLOWED | ALLOWED | REQUIRED. Indicates whether next database open allows or requires the resetlogs option
VERSION_TIME	DATE	The version time
OPEN_MODE	VARCHAR2(10)	Open mode information

V$DATAFILE

This view contains datafile information from the control file.

See Also: "V$DATAFILE_HEADER" on page 3-28, which displays information from datafile headers.

Column	Datatype	Description
FILE#	NUMBER	File identification number
STATUS	VARCHAR2	Type of file (system or user) and its status. Values: OFFLINE, ONLINE, SYSTEM, RECOVER, SYSOFF (an offline file from the SYSTEM tablespace)
ENABLED	VARCHAR2(10)	Describes how accessible the file is from SQL: ■ DISABLED: No SQL access allowed ■ READ ONLY: No SQL updates allowed ■ READ WRITE: Full access allowed ■ UNKNOWN: Should not occur unless the control file is corrupted.
CHECKPOINT_CHANGE#	NUMBER	SCN at last checkpoint
CHECKPOINT_TIME	DATE	Time stamp of the checkpoint#
UNRECOVERABLE _ CHANGE#	NUMBER	Last unrecoverable change# made to this datafile. This column is always updated when an unrecoverable operation completes
UNRECOVERABLE_TIME	DATE	Time stamp of the last unrecoverable change
BYTES	NUMBER	Current size in bytes; 0 if inaccessible
CREATE_BYTES	NUMBER	Size when created, in bytes
NAME	VARCHAR2	Name of the file
CREATION_CHANGE#	NUMBER	Change number at which the datafile was created
CREATION_TIME	DATE	Timestamp of the datafile creation
TS#	NUMBER	Tablespace number
RFILE#	NUMBER	Tablespace relative datafile number
LAST_CHANGE#	NUMBER	Last change# made to this datafile. Set to NULL if the datafile is being changed
LAST_TIME	DATE	Timestamp of the last change
OFFLINE_CHANGE#	NUMBER	Offline change# of the last offline range. This column is updated only when the datafile is brought online
ONLINE_CHANGE#	NUMBER	Online change# of the last offline range
ONLINE_TIME	DATE	Online timestamp of the last offline range
BLOCKS	NUMBER	Current datafile size in blocks; 0 if inaccessible
BLOCK_SIZE	NUMBER	Block size of the datafile
NAME	VARCHAR2(512)	Datafile name

Column	Datatype	Description
PLUGGED_IN	NUMBER	Describes whether the tablespace is plugged in. The value is 1 if the tablespace is plugged in and has not been made read-write, 0 if not.

V$DATAFILE_COPY

This view displays datafile copy information from the control file.

Column	Datatype	Description
RECID	NUMBER	Datafile copy record ID
STAMP	NUMBER	Datafile copy record stamp
NAME	VARCHAR2(512)	Filename of the datafile copy. The maximum length of the name is OS dependent
TAG	VARCHAR2(32)	Datafile copy tag
FILE#	NUMBER	Absolute datafile number
RFILE#	NUMBER	Tablespace relative datafile number
CREATION_CHANGE#	NUMBER	Datafile creation change#
CREATION_TIME	DATE	Datafile creation timestamp
RESETLOGS_CHANGE#	NUMBER	Resetlogs change# of the datafile when the copy was made
RESETLOGS_TIME	DATE	Resetlogs timestamp of the datafile when the copy was made
INCREMENTAL_LEVEL	NUMBER	The incremental level
CHECKPOINT_CHANGE#	NUMBER	Checkpoint change# of the datafile when the copy was made
CHECKPOINT_TIME	DATE	Checkpoint timestamp of the datafile when the copy was made
ABSOLUTE_FUZZY_ CHANGE#	NUMBER	Highest change seen when the datafile was copied
RECOVERY_FUZZY_ CHANGE#	NUMBER	Highest change written to the file by media recovery
RECOVERY_FUZZY_TIME	DATE	Timestamp of the highest change written to the file by media recovery
ONLINE_FUZZY	VARCHAR2(3)	YES | NO. If set to YES, this is a copy taken using an operating system utility after a crash or offline immediate (or an invalid copy taken while datafile was online and the database open). Recovery will need to apply all redo up to the next crash recovery marker to make the file consistent
BACKUP_FUZZY	VARCHAR2(3)	YES | NO. If set to YES, this is a copy taken using the BEGIN BACKUP/END BACKUP technique. Recovery will need to apply all redo up to the end backup marker to make this copy consistent
MARKED_CORRUPT	NUMBER	Number of blocks marked corrupt by this copy operation. That is, blocks that were not marked corrupted in the source datafile, but were detected and marked as corrupted during the copy operation

Column	Datatype	Description
MEDIA_CORRUPT	NUMBER	Total number of media corrupt blocks. For example, blocks with checksum errors are marked media corrupt
LOGICALLY_CORRUPT	NUMBER	Total number of logically corrupt blocks. For example, applying redo for unrecoverable operations will mark affected blocks logically corrupt
BLOCKS	NUMBER	Size of the datafile copy in blocks (also the size of the datafile when the copy was made)
BLOCK_SIZE	NUMBER	Block size of the datafile
OLDEST_OFFLINE _RANGE	NUMBER	The RECID of the oldest offline range record in this control file copy. 0 for datafile copies
COMPLETION_TIME	DATE	Time when the copy was completed
DELETED	VARCHAR2(3)	YES | NO. If set to YES the datafile copy has been deleted or overwritten

V$DATAFILE_HEADER

This view displays datafile information from the datafile headers.

Column	Datatype	Description
FILE#	NUMBER	Datafile number (from control file)
STATUS	VARCHAR2(7)	ONLINE | OFFLINE (from control file)
ERROR	VARCHAR2(18)	NULL if the datafile header read and validation were successful. If the read failed then the rest of the columns are NULL. If the validation failed then the rest of columns may display invalid data. If there is an error then usually the datafile must be restored from a backup before it can be recovered or used.
FORMAT	NUMBER	Indicates the format for the header block. The possible values are 6, 7, 8, or 0. 6 - indicates Oracle Version 6 7 - indicates Oracle Version 7 8 - indicates Oracle Version 8 0 - indicates the format could not be determined (for example, the header could not be read)
RECOVER	VARCHAR2(3)	File needs media recovery YES | NO
FUZZY	VARCHAR2(3)	File is fuzzy YES | NO
CREATION_CHANGE#	NUMBER	Datafile creation change#
CREATION_TIME	DATE	Datafile creation timestamp
TABLESPACE_NAME	VARCHAR2(30)	Tablespace name
TS#	NUMBER	Tablespace number

Column	Datatype	Description
RFILE#	NUMBER	Tablespace relative datafile number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change#
RESETLOGS_TIME	DATE	Resetlogs timestamp
CHECKPOINT_CHANGE#	NUMBER	Datafile checkpoint change#
CHECKPOINT_TIME	DATE	Datafile checkpoint timestamp
CHECKPOINT_COUNT	NUMBER	Datafile checkpoint count
BYTES	NUMBER	Current datafile size in bytes
BLOCKS	NUMBER	Current datafile size in blocks
NAME	VARCHAR2(512)	Datafile name

V$DBFILE

This view lists all datafiles making up the database. This view is retained for historical compatibility. Use of V$DATAFILE is recommended instead.

See Also: "V$DATAFILE" on page 3-25.

Column	Datatype	Description
FILE#	NUMBER	File identifier
NAME	VARCHAR2	Name of file

V$DBLINK

This view describes all database links (links with IN_TRANSACTION = YES) opened by the session issuing the query on V$DBLINK. These database links must be committed or rolled back before being closed.

Column	Datatype	Description
DB_LINK	VARCHAR2(128)	Name of the database link
OWNER_ID	NUMBER	Owner of the database link UID
LOGGED_ON	VARCHAR2(3)	Whether the database link is currently logged on
HETEROGENEOUS	VARCHAR2(3)	Whether the database link is heterogeneous
PROTOCOL	VARCHAR2(6)	Communication protocol for the database link
OPEN_CURSORS	NUMBER	Whether there are open cursors for the database link

Column	Datatype	Description
IN_TRANSACTION	VARCHAR2(3)	Whether the database link is currently in a transaction
UPDATE_SENT	VARCHAR2(3)	Whether there has been an update on the database link
COMMIT_POINT _ STRENGTH	NUMBER	Commit point strength of the transactions on the database link

V$DB_OBJECT_CACHE

This view displays database objects that are cached in the library cache. Objects include tables, indexes, clusters, synonym definitions, PL/SQL procedures and packages, and triggers.

Column	Datatype	Description
OWNER	VARCHAR2	Owner of the object
NAME	VARCHAR2	Name of the object
DB_LINK	VARCHAR2	Database link name, if any
NAMESPACE	VARCHAR2	Library cache namespace of the object: TABLE/PROCEDURE, BODY, TRIGGER, INDEX, CLUSTER, OBJECT
TYPE	VARCHAR2	Type of the object: INDEX, TABLE, CLUSTER, VIEW, SET, SYNONYM, SEQUENCE, PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, TRIGGER, CLASS, OBJECT, USER, DBLINK
SHARABLE_MEM	NUMBER	Amount of sharable memory in the shared pool consumed by the object
LOADS	NUMBER	Number of times the object has been loaded. This count also increases when an object has been invalidated
EXECUTIONS	NUMBER	Not used. **See:** "V$SQLAREA" on page 3-107 to see actual execution counts.
LOCKS	NUMBER	Number of users currently locking this object
PINS	NUMBER	Number of users currently pinning this object
KEPT	VARCHAR2(3)	YES or NO, depending on whether this object has been "kept" (permanently pinned in memory) with the PL/SQL procedure DBMS_SHARED_POOL.KEEP

V$DB_PIPES

This view displays the pipes that are currently represented in the shared pool for this instance.

Column	Datatype	Description
OWNERID	NUMBER	The owner ID of the owner if this is a private pipe. NULL otherwise.
NAME	VARCHAR2(1000)	The name of the pipe (for example, SCOTT.PIPE)
TYPE	VARCHAR2(7)	PUBLIC or PRIVATE
PIPE_SIZE	NUMBER	The amount of memory the pipe uses

V$DELETED_OBJECT

This view displays information about deleted archived logs, datafile copies and backup pieces from the control file. The only purpose of this view is to optimize the recovery catalog resync operation. When an archived log, datafile copy, or backup piece is deleted, the corresponding record is marked deleted.

Column	Datatype	Description
RECID	NUMBER	Deleted object record ID
STAMP	NUMBER	Deleted object record stamp
TYPE	VARCHAR2(13)	Identifies the type of deleted object: ARCHIVED LOG, DATAFILE COPY, BACKUP PIECE, PROXY COPY, BACKUP PIECE AVAILABLE, BACKUP PIECE EXPIRED, PROXY COPY AVAILABLE, or PROXY COPY EXPIRED.
OBJECT_RECID	NUMBER	Record ID of the deleted object
OBJECT_STAMP	NUMBER	Record timestamp of the deleted object

V$DISPATCHER

This view provides information on the dispatcher processes.

Column	Datatype	Description
NAME	VARCHAR2	Name of the dispatcher process
NETWORK	VARCHAR2	Network address of this dispatcher.
PADDR	RAW(4)	Process address

Column	Datatype	Description
STATUS	VARCHAR2	The status of the dispatcher (one of the following): ■ WAIT: idle ■ SEND: sending a message ■ RECEIVE: receiving a message ■ CONNECT: establishing a connection ■ DISCONNECT: handling a disconnect request ■ BREAK: handling a break ■ TERMINATE: in the process of terminating ■ ACCEPT: accepting connections (no further information available) ■ REFUSE: rejecting connections (no further information available)
ACCEPT	VARCHAR2	Whether this dispatcher is accepting new connections: YES | NO
MESSAGES	NUMBER	Number of messages processed by this dispatcher
BYTES	NUMBER	Size in bytes of messages processed by this dispatcher
BREAKS	NUMBER	Number of breaks occurring in this connection
OWNED	NUMBER	Number of circuits owned by this dispatcher
CREATED	NUMBER	Number of circuits created by this dispatcher
IDLE	NUMBER	Total idle time for this dispatcher in hundredths of a second
BUSY	NUMBER	Total busy time for this dispatcher in hundredths of a second
LISTENER	NUMBER	The most recent Oracle error number the dispatcher received from the listener
CONF_INDX	NUMBER	Zero-based index of the MTS_DISPATCHERS configuration used by this dispatcher

V$DISPATCHER_RATE

This view provides rate statistics for a number of activities performed by the dispatcher processes. Collected samples have an activity-specific "time-to-live" (TTL_* columns). Statistics are reported over two types of time intervals:

- current statistics (CUR_ columns): use samples collected over the most recent time-to-live interval
- historical statistics (AVG_ and most of the MAX_ columns): make use of all samples that are not current anymore

At the time of collection, a sample is current. After the time-to-live has elapsed, the sample becomes historical. Each type of activity has a specific scale (represented by the SCALE_* columns) at which the statistics are reported.

Column	Datatype	Description
NAME	CHAR	Name of the dispatcher process
PADDR	RAW	Address of the dispatcher process
CUR_LOOP_RATE	NUMBER	Rate at which the dispatcher has been iterating through its dispatching loop, reported over the past TTL_LOOPS, in iterations / SCALE_LOOPS
CUR_EVENT_RATE	NUMBER	Rate at which the dispatcher has been processing dispatcher events, reported over the past TTL_LOOPS, in events / SCALE_LOOPS. Such dispatcher events include network events and shared server requests.
CUR_EVENTS_PER_LOOP	NUMBER	Average number of events the dispatcher has been processing in each iteration through its dispatching loop, reported over the past TTL_LOOPS, in events / iteration
CUR_MSG_RATE	NUMBER	Rate at which the dispatcher has been relaying messages between clients and shared servers, reported over the past TTL_MSG, in messages / SCALE_MSG
CUR_SVR_BUF_RATE	NUMBER	Rate at which the dispatcher has been relaying buffers to shared servers, reported over the past TTL_SVR_BUF, in messages / SCALE_SVR_BUF
CUR_SVR_BYTE_RATE	NUMBER	Rate at which the dispatcher has been relaying data to shared servers, reported over the past TTL_SVR_BUF, in bytes / SCALE_SVR_BUF
CUR_SVR_BYTE_PER_BUF	NUMBER	Average number of data types in each buffer relayed to shared servers, reported over the past TTL_SVR_BUF, in bytes / buffer
CUR_CLT_BUF_RATE	NUMBER	Rate at which the dispatcher has been relaying buffers to clients, reported over the past TTL_CLT_BUF, in buffers / SCALE_CLT_BUF
CUR_CLT_BYTE_RATE	NUMBER	Rate at which the dispatcher has been relaying data to clients, reported over the past TTL_CLT_BUF, in bytes / SCALE_CLT_BUF
CUR_CLT_BYTE_PER_BUF	NUMBER	Average number of data bytes in each buffer relayed to clients, reported over the past TTL_CLT_BUF, in bytes / buffer
CUR_BUF_RATE	NUMBER	Rate at which the dispatcher has been relaying buffers to either clients or shared servers, reported over the past TTL_BUF, in bytes / SCALE_BUF
CUR_BYTE_RATE	NUMBER	Rate at which the dispatcher has been relaying data to either clients or shared servers, reported over the past TTL_BUF, in bytes / SCALE_BUF
CUR_BYTE_PER_BUF	NUMBER	Average number of data bytes in each buffer relayed to either clients or shared servers, reported over the past TTL_BUF, in bytes / buffer
CUR_IN_CONNECT_RATE	NUMBER	Rate at which the dispatcher has been accepting incoming client connections, reported over the past TTL_IN_CONNECT, in connections / SCALE_IN_CONNECT
CUR_OUT_CONNECT_RATE	NUMBER	Rate at which the dispatcher has been establishing outbound connections, reported over the past TTL_OUT_CONNECT, in connections / SCALE_OUT_CONNECT

Column	Datatype	Description
CUR_RECONNECT_RATE	NUMBER	In a connection pooling setup, the rate at which clients have been reconnecting to the dispatcher, reported over the past TTL_RECONNECT, in reconnections / SCALE_RECONNECT
MAX_LOOP_RATE	NUMBER	Maximum rate at which the dispatcher has ever iterated through its dispatching loop, reported in iterations / SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS
MAX_EVENT_RATE	NUMBER	Maximum rate at which the dispatcher has ever processed dispatcher events, reported in events / SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS
MAX_EVENTS_PER_LOOP	NUMBER	Maximum number of events the dispatcher has ever processed in one iteration through its dispatching loop, reported in events / iteration, over the dispatcher's lifetime
MAX_MSG_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed messages between clients and shared servers, reported in messages / SCALE_MSG, over the dispatcher's lifetime excluding the past TTL_MSG
MAX_SVR_BUF_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to shared servers, reported in buffers / SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
MAX_SVR_BYTE_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed data to shared servers, reported in bytes / SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
MAX_SVR_BYTE_PER_BUF	NUMBER	Maximum number of data bytes the dispatcher has ever relayed in one buffer to a client, reported in bytes / buffer, over the dispatcher's lifetime
MAX_CLT_BUF_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to either clients or shared servers, reported in buffers / SCALE_CLT_BUF, over the dispatcher's life time excluding the past TTL_CLT_BUF
MAX_CLT_BYTE_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to clients, reported in bytes / SCALE_CLT_BUF, over the dispatcher's lifetime excluding the last TTL_CLT_BUF
MAX_CLT_BYTE_PER_BUF	NUMBER	Maximum number of data bytes the dispatcher has ever relayed in one buffer to a client, reported in bytes / buffer, over the dispatcher's lifetime
MAX_BUF_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to either clients or shared servers, reported in buffers / SCALE_BUF, over the dispatcher's lifetime, excluding the past TTL_BUF
MAX_BYTE_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed data to either clients or shared servers, reported in bytes / SCALE_BUF, over the dispatcher's lifetime excluding the past TTL_BUF
MAX_BYTE_PER_BUF	NUMBER	Maximum number of data bytes the dispatcher has ever relayed in one buffer to either a client or a shared server, reported in bytes / buffer, over the dispatcher's lifetime
MAX_IN_CONNECT_RATE	NUMBER	Maximum rate at which the dispatcher has ever accepted incoming client connections, reported in connections / SCALE_IN_CONNECT, over the dispatcher's lifetime excluding the past TTL_IN_CONNECT

Column	Datatype	Description
MAX_OUT_CONNECT_RATE	NUMBER	Maximum rate at which the dispatcher has ever established outbound connections, reported in connections / SCALE_OUT_CONNECT, over the dispatcher's lifetime excluding the past TTL_OUT_CONNECT
MAX_RECONNECT_RATE	NUMBER	In a connection pooling setup, the maximum rate at which clients have ever reconnected to this dispatcher, reported in reconnections / SCALE_RECONNECT, over the dispatcher's lifetime excluding the past TTL_RECONNECT
AVG_LOOP_RATE	NUMBER	Historical average rate at which the dispatcher has iterated through its dispatching loop, reported in iterations / SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS
AVG_EVENT_RATE	NUMBER	Historical average rate at which the dispatcher has processed dispatcher events, reported in events / SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS
AVG_EVENTS_PER_LOOP	NUMBER	Historical average number of events the dispatcher has processed in one iteration through its dispatching loop, reported in events / iteration, over the dispatcher's lifetime excluding the past TTL_LOOPS
AVG_MSG_RATE	NUMBER	Historical average rate at which the dispatcher has relayed messages between clients and shared servers, reported in messages / SCALE_MSG, over the dispatcher's lifetime excluding the past TTL_MSG
AVG_SVR_BUF_RATE	NUMBER	Historical average rate at which the dispatcher has relayed buffers to shared servers, reported in buffers / SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
AVG_SVR_BYTE_RATE	NUMBER	Historical average rate at which the dispatcher has relayed data to shared servers, reported in bytes / SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
AVG_SVR_BYTE_PER_BUF	NUMBER	Historical average number of data bytes per buffer the dispatcher has relayed to shared servers, reported in bytes / buffer, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
AVG_CLT_BUF_RATE	NUMBER	Historical average rate at which the dispatcher has relayed buffers to clients, reported in buffers / SCALE_CLT_BUF, over the dispatcher's lifetime excluding the past TTL_CLT_BUF
AVG_CLT_BYTE_RATE	NUMBER	Historical average rate at which the dispatcher has relayed data to clients, reported in bytes / SCALE_CLT_BUF, over the dispatcher's lifetime excluding the past TTL_CLT_BUF
AVG_CLT_BYTE_PER_BUF	NUMBER	Historical average number of data bytes per buffer the dispatcher has relayed to clients, reported in bytes / buffer, over the dispatcher's lifetime excluding the past TTL_CLT_BUF
AVG_BUF_RATE	NUMBER	Historical average rate at which the dispatcher has relayed buffers to either clients or shared servers, reported in buffers / SCALE_BUF, over the dispatcher's lifetime excluding the past TTL_BUF
AVG_BYTE_RATE	NUMBER	Historical average rate at which the dispatcher has relayed data to either clients or shared servers, reported in bytes / SCALE_BUF, over the dispatcher's lifetime excluding the past TTL_BUF

Column	Datatype	Description
AVG_BYTE_PER_BUF	NUMBER	Historical average number of data bytes per buffer the dispatcher has relayed to either clients or shared servers, reported in bytes/buffer, over the dispatcher's lifetime excluding the past TTL_BUF
AVG_IN_CONNECT_RATE	NUMBER	Historical average rate at which the dispatcher has accepted incoming client connections, reported in connections/SCALE_IN_CONNECT, over the dispatcher's lifetime excluding the past TTL_IN_CONNECT
AVG_OUT_CONNECT_RATE	NUMBER	Historical average rate at which the dispatcher has established outbound connections, reported in connections/SCALE_OUT_CONNECT, over the dispatcher's lifetime excluding the past TTL_OUT_CONNECT
AVG_RECONNECT_RATE	NUMBER	In a connection pooling setup, the historical average rate at which clients have reconnected to this dispatcher, reported in reconnections/SCALE_RECONNECT, over the dispatcher's lifetime excluding the past TTL_RECONNECT
TTL_LOOPS	NUMBER	Time-to-live for "loops" samples, reported in hundredths of a second. Default is 10 minutes.
TTL_MSG	NUMBER	Time-to-live for "messages" samples, reported in hundredths of a second. Default is 10 seconds.
TTL_SVR_BUF	NUMBER	Time-to-live for "buffers to servers" samples, reported in hundredths of a second. Default is 1 second.
TTL_CLT_BUF	NUMBER	Time-to-live for "buffers to clients" samples, reported in hundredths of a second. Default is 1 second.
TTL_BUF	NUMBER	Time-to-live for "buffers to clients/servers" samples, reported in hundredths of a second. Default is 1 second.
TTL_IN_CONNECT	NUMBER	Time-to-live for "inbound connections" samples, reported in hundredths of a second. Default is 10 minutes.
TTL_OUT_CONNECT	NUMBER	Time-to-live for "outbound connections" samples, reported in hundredths of a second. Default is 10 minutes.
TTL_RECONNECT	NUMBER	Time-to-live for "reconnections" samples, reported in hundredths of a second. Default is 10 minutes.
SCALE_LOOPS	NUMBER	Scale for "loops" statistics, reported in hundredths of a second. Default is 1 minute.
SCALE_MSG	NUMBER	Scale for "messages" statistics, reported in hundredths of a second. Default is 1 second.
SCALE_SVR_BUF	NUMBER	Scale for "buffers to servers" statistics, reported in hundredths of a second. Default is 1/10 second.
SCALE_CLT_BUF	NUMBER	Scale for "buffers to clients" statistics, reported in hundredths of a second. Default is 1/10 second.
SCALE_BUF	NUMBER	Scale for "buffers to clients/servers" statistics, reported in hundredths of a second. Default is 1/10 second.
SCALE_IN_CONNECT	NUMBER	Scale for "inbound connections" statistics, reported in hundredths of a second. Default is 1 minute.

Column	Datatype	Description
SCALE_OUT_CONNECT	NUMBER	Scale for "outbound connections" statistics, reported in hundredths of a second. Default is 1 minute.
SCALE_RECONNECT	NUMBER	Scale for "reconnections" statistics, reported in hundredths of a second. Default is 1 minute.

V$DLM_ALL_LOCKS

This Oracle Parallel Server view describes all locks currently known to lock manager.

See Also: "V$DLM_LOCKS" on page 3-39 for a description of all such locks that are currently blocking or being blocked.

Column	Datatype	Description
LOCKP	RAW(4)	Lock Pointer
GRANT_LEVEL	VARCHAR2(9)	Granted level of the lock
REQUEST_LEVEL	VARCHAR2(9)	Requested level of the lock
RESOURCE_NAME1	VARCHAR2(30)	Resource name for the lock
RESOURCE_NAME2	VARCHAR2(30)	Resource name for the lock
PID	NUMBER	Process identifier which holds the lock
TRANSACTION_ID0	NUMBER	Lower 4 bytes of the transaction identifier where the lock belongs to
TRANSACTION_ID1	NUMBER	Upper 4 bytes of the transaction identifier where the lock belongs to
GROUP_ID	NUMBER	Group identifier for the lock
OPEN_OPT_DEADLOCK	NUMBER	1 if DEADLOCK open option is set, otherwise 0
OPEN_OPT_PERSISTENT	NUMBER	1 if PERSISTENT open option is set, otherwise 0
OPEN_OPT_PROCESS _OWNED	NUMBER	1 if PROCESS_OWNED open option is set, otherwise 0
OPEN_OPT_NO_XID	NUMBER	1 if NO_XID open option is set, otherwise 0
CONVERT_OPT _GETVALUE	NUMBER	1 if GETVALUE convert option is set, otherwise 0
CONVERT_OPT _PUTVALUE	NUMBER	1 if PUTVALUE convert option is set, otherwise 0
CONVERT_OPT _NOVALUE	NUMBER	1 if NOVALUE convert option is set, otherwise 0
CONVERT_OPT _DUBVALUE	NUMBER	1 if DUBVALUE convert option is set, otherwise 0
CONVERT_OPT _NOQUEUE	NUMBER	1 if NOQUEUE convert option is set, otherwise 0

Column	Datatype	Description
CONVERT_OPT_EXPRESS	NUMBER	1 if EXPRESS convert option is set, otherwise 0
CONVERT_OPT_ NODEADLOCKWAIT	NUMBER	1 if NODEADLOCKWAIT convert option is set, otherwise 0
CONVERT_OPT_ NODEADLOCKBLOCK	NUMBER	1 if NODEADLOCKBLOCK convert option is set, otherwise 0
WHICH_QUEUE	NUMBER	Which queue the lock is currently located. 0 for NULL queue; 1 for GRANTED queue; 2 for CONVERT queue
LOCKSTATE	VARCHAR2(64)	State of the lock as the owner sees it
AST_EVENT0	NUMBER	Last AST event
OWNER_NODE	NUMBER	Node identifier
BLOCKED	NUMBER	1 if this lock request is blocked by others, otherwise 0
BLOCKER	NUMBER	1 if this lock is blocking others, otherwise 0

V$DLM_CONVERT_LOCAL

V$DLM_CONVERT_LOCAL displays the elapsed time for the local lock conversion operation.

Column	Datatype	Description
INST_ID	NUMBER	ID of the instance
CONVERT_TYPE	VARCHAR2(64)	Conversion types are listed in Table 3-3
AVERAGE _CONVERT_ TIME	NUMBER	Average conversion time for each type of lock operation (in 100th of a second).
CONVERT_COUNT	NUMBER	The number of operations

V$DLM_CONVERT_REMOTE

V$DLM_CONVERT_REMOTE displays the elapsed time for the remote lock conversion operation.

Column	Datatype	Description
INST_ID	NUMBER	ID of the instance
CONVERT_TYPE	VARCHAR2(64)	Conversion types are listed in Table 3-3

Column	Datatype	Description
		■ NULL -> SS: NULL mode to subshared mode ■ NULL -> SX: NULL mode to shared exclusive mode ■ NULL -> S: NULL mode to shared mode ■ NULL -> SSX: NULL mode to subshared exclusive mode ■ NULL -> X: NULL mode to exclusive mode ■ SS -> SX: subshared mode to shared exclusive mode ■ SS -> S: subshared mode to shared mode ■ SS -> SSX: subshared mode to subshared exclusive mode ■ SS -> X: subshared mode to exclusive mode ■ SX -> S: shared exclusive mode to shared mode ■ SX -> SSX: shared exclusive mode to subshared exclusive mode ■ SX -> X: shared exclusive mode to exclusive mode ■ S -> SX: shared mode to shared exclusive mode ■ S -> SSX: shared mode to subshared exclusive mode ■ S -> X: shared mode to exclusive mode ■ SSX -> X: sub-shared exclusive mode to exclusive mode
AVERAGE_CONVERT_TIME	NUMBER	Average conversion time for each type of lock operation (in 100th of a second)
CONVERT_COUNT	NUMBER	The number of operations

V$DLM_LATCH

V$DLM_LATCH is obsolete.

See Also: "V$LATCH" on page 3-53 for statistics about DLM latch performance.

V$DLM_LOCKS

This Oracle Parallel Server view describes all locks currently known to lock manager that are being blocked or blocking others. The output of this view is a subset of the output from V$DLM_ALL_LOCKS.

See Also: "V$DLM_ALL_LOCKS" on page 3-37 for a description of all locks known to the lock manager.

Column	Datatype	Description
LOCKP	RAW(4)	Lock Pointer
GRANT_LEVEL	VARCHAR2(9)	Granted level of the lock
REQUEST_LEVEL	VARCHAR2(9)	Requested level of the lock
RESOURCE_NAME1	VARCHAR2(30)	Resource name for the lock
RESOURCE_NAME2	VARCHAR2(30)	Resource name for the lock
PID	NUMBER	Process identifier which holds the lock
TRANSACTION_ID0	NUMBER	Lower 4 bytes of the transaction identifier where the lock belongs to
TRANSACTION_ID1	NUMBER	Upper 4 bytes of the transaction identifier where the lock belongs to
GROUP_ID	NUMBER	Group identifier for the lock
OPEN_OPT_DEADLOCK	NUMBER	1 if DEADLOCK open option is set, otherwise 0
OPEN_OPT_PERSISTENT	NUMBER	1 if PERSISTENT open option is set, otherwise 0
OPEN_OPT_PROCESS _ OWNED	NUMBER	1 if PROCESS_OWNED open option is set, otherwise 0
OPEN_OPT_NO_XID	NUMBER	1 if NO_XID open option is set, otherwise 0
CONVERT_OPT _GETVALUE	NUMBER	1 if GETVALUE convert option is set, otherwise 0
CONVERT_OPT _PUTVALUE	NUMBER	1 if PUTVALUE convert option is set, otherwise 0
CONVERT_OPT _NOVALUE	NUMBER	1 if NOVALUE convert option is set, otherwise 0
CONVERT_OPT _DUBVALUE	NUMBER	1 if DUBVALUE convert option is set, otherwise 0
CONVERT_OPT _NOQUEUE	NUMBER	1 if NOQUEUE convert option is set, otherwise 0
CONVERT_OPT_EXPRESS	NUMBER	1 if EXPRESS convert option is set, otherwise 0
CONVERT_OPT_ NODEADLOCKWAIT	NUMBER	1 if NODEADLOCKWAIT convert option is set, otherwise 0
CONVERT_OPT_ NODEADLOCKBLOCK	NUMBER	1 if NODEADLOCKBLOCK convert option is set, otherwise 0
WHICH_QUEUE	NUMBER	Which queue the lock is currently located. 0 for NULL queue; 1 for GRANTED queue; 2 for CONVERT queue
LOCKSTATE	VARCHAR2(64)	State of lock as owner sees it
AST_EVENT0	NUMBER	Last AST event
OWNER_NODE	NUMBER	Node identifier
BLOCKED	NUMBER	1 if this lock request is blocked by others, otherwise 0
BLOCKER	NUMBER	1 if this lock is blocking others, otherwise 0

V$DLM_MISC

V$DLM_MISC displays miscellaneous DLM statistics.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number
NAME	VARCHAR2(64)	Name of the statistic
VALUE	NUMBER	Value associated with the statistic

V$DLM_RESS

V$DLM_RESS is an Oracle Parallel Server view. It displays information of all resources currently known to the lock manager.

Column	Datatype	Description
RESP	RAW(4)	Resource pointer
RESOURCE_NAME	VARCHAR2(30)	Resource name in hexadecimal for the lock
ON_CONVERT_Q	NUMBER	1 if on convert queue, 0 otherwise
ON_GRANT_Q	NUMBER	1 if on granted queue, 0 otherwise
PERSISTENT_RES	NUMBER	1 if it is a persistent resource, 0 otherwise
RDOMAIN_NAME	VARCHAR2(25)	Recovery domain name
RDOMAINP	RAW(4)	Recovery domain pointer
MASTER_NODE	NUMBER	Master node ID
NEXT_CVT_LEVEL	VARCHAR2(9)	Next lock level to convert on global convert queue
VALUE_BLK_STATE	VARCHAR2(32)	State of the value block
VALUE_BLK	VARCHAR2(64)	First 64 bytes of the value block

V$ENABLEDPRIVS

This view displays which privileges are enabled. These privileges can be found in the table SYS.SYSTEM_PRIVILEGES_MAP.

Column	Datatype	Description
PRIV_NUMBER	NUMBER	Numeric identifier of enabled privileges

V$ENQUEUE_LOCK

This view displays all locks owned by enqueue state objects. The columns in this view are identical to the columns in V$LOCK.

See Also: "V$LOCK" on page 3-57.

Column	Datatype	Description
ADDR	RAW(4)	Address of lock state object
KADDR	RAW(4)	Address of lock
SID	NUMBER	Identifier for session holding or acquiring the lock
TYPE	VARCHAR2(2)	Type of lock. For a list of user and system types that can have locks.
ID1	NUMBER	Lock identifier #1 (depends on type)
ID2	NUMBER	Lock identifier #2 (depends on type)
LMODE	NUMBER	Lock mode in which the session holds the lock: ■ 0, None ■ 1, Null (NULL) ■ 2, Row-S (SS) ■ 3, Row-X (SX) ■ 4, Share (S) ■ 5, S/Row-X (SSX) ■ 6, Exclusive (X)
REQUEST	NUMBER	Lock mode in which the process requests the lock: ■ 0, None ■ 1, Null (NULL) ■ 2, Row-S (SS) ■ 3, Row-X (SX) ■ 4, Share (S) ■ 5, S/Row-X (SSX) ■ 6, Exclusive (X)
CTIME	NUMBER	Time since current mode was granted
BLOCK	NUMBER	The lock is blocking another lock

V$EVENT_NAME

This view contains information about wait events.

Column	Datatype	Description
EVENT#	NUMBER	The number of the wait event
NAME	VARCHAR2(64)	The name of the wait event
PARAMETER1	VARCHAR2(64)	The description of the first parameter for the wait event
PARAMETER2	VARCHAR2(64)	The description of the second parameter for the wait event
PARAMETER3	VARCHAR2(64)	The description of the third parameter for the wait event

V$EXECUTION

This view displays information on parallel execution.

Column	Datatype	Description
PID	NUMBER	Session ID
DEPTH	NUMBER	The depth
FUNCTION	VARCHAR2(10)	Session serial number
TYPE	VARCHAR2(7)	Name of the OBJECT_NODE in plan table
NVALS	NUMBER	Elapsed time for OBJECT_NODE
VAL1	NUMBER	The value for number 1
VAL2	NUMBER	The value for number 2
SEQH	NUMBER	A sequence
SEQL	NUMBER	A sequence

V$FALSE_PING

V$FALSE_PING is an Oracle Parallel Server view. This view displays buffers that may be getting false pings. That is, buffers pinged more than 10 times that are protected by the same lock as another buffer that pinged more than 10 times. Buffers identified as getting false pings can be remapped in "GC_FILES_TO_LOCKS"to reduce lock collisions.

> **See Also:** "GC_FILES_TO_LOCKS" on page 1-40 and *Oracle8i Parallel Server Concepts*.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query DBA_DATA_FILES or V$DBFILES)
BLOCK#	NUMBER	Block number
STATUS	VARCHAR2(1)	Status of block: ■ FREE = not currently in use ■ XCUR = exclusive ■ SCUR = shared current ■ CR = consistent read ■ READ = being read from disk ■ MREC = in media recovery mode ■ IREC = in instance recovery mode
XNC	NUMBER	Number of PCM lock conversions from Exclusive mode due to contention with another instance. This column is obsolete but is retained for historical compatibility
FORCED_READS	NUMBER	Number of times the block had to be reread from disk because another instance had forced it out of this instance's cache by requesting the PCM lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times DBWR had to write this block to disk because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
NAME	VARCHAR2(30)	Name of the database object containing the block
PARTITION_NAME	VARCHAR2	NULL for non-partitioned objects
KIND	VARCHAR2(12)	Type of database object.
OWNER#	NUMBER	Owner number
LOCK_ELEMENT_ADDR	RAW(4)	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock
LOCK_ELEMENT_NAME	NUMBER	The name of the lock that contains the PCM lock that is covering the buffer
LOCK_ELEMENT_CLASS	NUMBER	The lock element class

V$FAST_START_SERVERS

V$FAST_START_SERVERS provides information about all the recovery slaves performing parallel transaction recovery.

See Also: *Oracle8i Backup and Recovery Guide.*

Column	Datatype	Description
STATE	VARCHAR2(11)	State of the server; IDLE or RECOVERING
UNDOBLOCKSDONE	NUMBER	The percentage of the assigned work done so far
PID	NUMBER	The process ID

V$FAST_START_TRANSACTIONS

V$FAST_START_TRANSACTIONS contains information about the progress of the transactions that Oracle is recovering.

See Also: *Oracle8i Backup and Recovery Guide.*

Column	Datatype	Description
USN	NUMBER	The undo segment number of the transaction
SLT	NUMBER	The slot within the rollback segment
SEQ	NUMBER	The incarnation number of the slot
STATE	VARCHAR2(16)	The state of the transaction may be TO BE RECOVERED, RECOVERED, or RECOVERING
UNDOBLOCKSDONE	NUMBER	The number of undo blocks completed on this transaction
UNDOBLOCKSTOTAL	NUMBER	The total number of undo blocks that need recovery
PID	NUMBER	The ID of the current server it has been assigned to
CPUTIME	NUMBER	The time for which recovery has progressed, in seconds
PARENTUSN	NUMBER	The undo segment number of the parent transaction in PDML
PARENTSLT	NUMBER	The slot of the parent transaction in PDML
PARENTSEQ	NUMBER	The sequence number of the parent transaction in PDML

V$FILE_PING

The view V$FILE_PING displays the number of blocks pinged per datafile. This information in turn can be used to determine access patterns to existing datafiles and deciding new mappings from datafile blocks to PCM locks.

Column	Datatype	Description
FILE_NUMBER	NUMBER	Number of the datafile
FREQUENCY	NUMBER	The frequency

Column	Datatype	Description
X_2_NULL	NUMBER	Number of lock conversions from Exclusive-to-NULL for all blocks in the file
X_2_NULL_FORCED _ WRITE	NUMBER	Number of forced writes that occur for blocks of the specified file due to Exclusive-to-NULL conversions
X_2_NULL_FORCED _ STALE	NUMBER	Number of times a block in the file was made STALE due to Exclusive-to-NULL conversions
X_2_S	NUMBER	Number of lock conversions from Exclusive-to-Shared for all blocks in the file
X_2_S_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified file due to Exclusive-to-Shared conversions
X_2_SSX	NUMBER	Number of lock conversions from Exclusive-to-subshared Exclusive for all blocks in the file
X_2_SSX_FORCED _WRITE	NUMBER	Number of forced writes that occur for blocks of the specified file due to Exclusive-to-Sub Shared Exclusive conversions
S_2_NULL	NUMBER	Number of lock conversions from Shared-to-NULL for all blocks in the file
S_2_NULL_FORCED _ STALE	NUMBER	Number of times a block in the file was made STALE due to Shared-to-NULL conversions
SS_2_NULL	NUMBER	Number of lock conversions from Sub Shared-to-NULL for all blocks in the file
SS_2_RLS	NUMBER	Number of pcm locks sslocks released. 0 in Oracle 8.1
WRB	NUMBER	Number of times the instance received a write single buffer cross instance call for this file
WRB_FORCED_WRITE	NUMBER	Number of blocks written due to write single buffer cross instance calls for this file
RBR	NUMBER	Number of times the instance received a reuse block range cross instance call for this file
RBR_FORCED_WRITE	NUMBER	Number of blocks written due to reuse block range cross instance calls for this file
RBR_FORCED_STALE	NUMBER	Number of times a block in this file was made STALE due to reuse block range cross instance calls
CBR	NUMBER	Number of times the instance received a checkpoint block range cross instance call for this file
CBR_FORCED_WRITE	NUMBER	Number of blocks in this file which were written due to checkpoint cross range cross instance calls
NULL_2_X	NUMBER	Number of lock conversions from NULL-to-Exclusive for all blocks of the specified file
S_2_X	NUMBER	Number of lock conversions from Shared-to-Exclusive for all blocks of the specified file
SSX_2_X	NUMBER	Number of lock conversions from Sub Shared Exclusive-to-Exclusive for all blocks of the specified file

Column	Datatype	Description
NULL_2_S	NUMBER	Number of lock conversions from NULL-to-Shared for all blocks of the specified file
NULL_2_SS	NUMBER	Number of lock conversions from NULL-to-Sub Shared for all blocks of the specified file
OP_2_SS	NUMBER	Number of pcm locks ss locks opened. 0 in Oracle 8.1

V$FILESTAT

This view contains information about file read / write statistics.

Column	Datatype	Description
FILE#	NUMBER	Number of the file
PHYRDS	NUMBER	Number of physical reads done
PHYWRTS	NUMBER	Number of times DBWR is required to write
PHYBLKRD	NUMBER	Number of physical blocks read
PHYBLKWRT	NUMBER	Number of blocks written to disk; which may be the same as PHYWRTS if all writes are single blocks
READTIM	NUMBER	Time (in hundredths of a second) spent doing reads if the TIMED_ STATISTICS parameter is TRUE; 0 if FALSE
WRITETIM	NUMBER	Time (in hundredths of a second) spent doing writes if the TIMED_ STATISTICS parameter is TRUE; 0 if FALSE
AVGIOTIM	NUMBER	The average time (in hundredths of a second) spent on I / O, if the TIMED_ STATISTICS parameter is TRUE; 0 if FALSE
LSTIOTIM	NUMBER	The time (in hundredths of a second) spent doing the last I / O, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
MINIOTIM	NUMBER	The minimum time (in hundredths of a second) spent on a single I / O, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
MAXIOWTM	NUMBER	The maximum time (in hundredths of a second) spent doing a single write, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
MAXIORTM	NUMBER	The maximum time (in hundredths of a second) spent doing a single read, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE

V$FIXED_TABLE

This view displays all dynamic performance tables, views, and derived tables in the database. Some V$ tables (for example, V$ROLLNAME) refer to real tables and are therefore not listed.

Column	Datatype	Description
NAME	VARCHAR2(30)	Name of the object
OBJECT_ID	NUMBER	Identifier of the fixed object
TYPE	VARCHAR2(5)	Object type: TABLE, VIEW
TABLE_NUM	NUMBER	Number that identifies the dynamic performance table if it is of type TABLE

V$FIXED_VIEW_DEFINITION

This view contains the definitions of all the fixed views (views beginning with V$). Use this table with caution. Oracle tries to keep the behavior of fixed views the same from release to release, but the definitions of the fixed views can change without notice. Use these definitions to optimize your queries by using indexed columns of the dynamic performance tables.

Column	Datatype	Description
VIEW_NAME	VARCHAR2(30)	The name of the fixed view
VIEW_DEFINITION	VARCHAR2(2000)	The definition of the fixed view

V$GLOBAL_BLOCKED_LOCKS

This view displays global blocked locks.

Column	Datatype	Description
ADDR	RAW(4)	Address of lock state object (raw)
KADDR	RAW(4)	Address of lock (raw)
SID	NUMBER	Identifier of session holding the lock (number)
TYPE	VARCHAR2(2)	Resource type (char)
ID1	NUMBER	Resource identifier #1 (number)
ID2	NUMBER	Resource identifier #2 (number)
LMODE	NUMBER	Lock mode held (number)
REQUEST	NUMBER	Lock mode requested (number)
CTIME	NUMBER	Time since current mode was granted

V$GLOBAL_TRANSACTION

This view displays information on the currently active global transactions.

Column	Datatype	Description
FORMATID	NUMBER	Format identifier of the global transaction
GLOBALID	RAW(64)	Global transaction identifier of the global transaction
BRANCHID	RAW(64)	Branch qualifier of the global transaction
BRANCHES	NUMBER	Total number of branches in the global transaction
REFCOUNT	NUMBER	Number of siblings for this global transaction, must be the same as branches.
PREPARECOUNT	NUMBER	Number of branches of the global transaction that have prepared
STATE	VARCHAR2(18)	State of the branch of the global transaction
FLAGS	NUMBER	The numerical representation of the state
COUPLING	VARCHAR2(15)	Whether the branches are loosely coupled or tightly coupled

V$HS_AGENT

This view identifies the set of HS agents currently running on a given host, using one row per agent process.

Column	Datatype	Description
AGENT_ID	NUMBER	Net8 session identifier used for connections to agent (the identifier used in the LISTENER.ORA file). Maps to the AGENT_ID column of V$HS_AGENT.
MACHINE	VARCHAR2(64)	Operating system machine name
PROCESS	VARCHAR2(9)	Operating system process identifier of agent
PROGRAM	VARCHAR2(48)	Program name of agent
OSUSER	VARCHAR2(30)	Operating system user
STARTTIME	DATE	The starting time
AGENT_TYPE	NUMBER	Type of agent
FDS_CLASS_ID	NUMBER	The ID of the Foreign Data Store class
FDS_INST_ID	NUMBER	The instance name of the Foreign Data Store

V$HS_PARAMETER

This view describes the initialization parameters in use by the server and agent.

Column	Datatype	Description
HS_SESSION_ID	NUMBER	Unique HS session identifier. this column maps to the HS_SESSION_ID column of V$HS_SESSION.

Column	Datatype	Description
PARAMETER	VARCHAR2(64)	Name of the parameter
VALUE	VARCHAR2(64)	Value of the parameter
SOURCE	CHAR(1)	Whether the parameter was defined in the agent (A) or server (S)
ENV	CHAR(1)	Whether the parameter was also set as an environment variable in the agent or elsewhere (T | F)

V$HS_SESSION

This view describes the current HS session.

Column	Datatype	Description
HS_SESSION_ID	NUMBER	Unique HS session identifier
AGENT_ID	NUMBER	Net8 session identifier used for connections to the agent. Maps to the AGENT_ID column of V$HS_AGENT.
SID	NUMBER	User session identifier. Maps to the SID column of V$SESSION.
DB_LINK	VARCHAR2(128)	Server database link name used to access the agent. Blank if no database link is used (for example, when using external procedures).
DB_LINK_OWNER	NUMBER	Owner of the database link in DB_LINK
STARTTIME	DATE	Time the connection was initiated

V$INDEXED_FIXED_COLUMN

This view displays the columns in dynamic performance tables that are indexed (X$ tables). The X$ tables can change without notice. Use this view only to write queries against fixed views (V$ views) more efficiently.

Column	Datatype	Description
TABLE_NAME	VARCHAR2(30)	The name of the dynamic performance table that is indexed
INDEX_NUMBER	NUMBER	Number that distinguishes to which index a column belongs
COLUMN_NAME	VARCHAR2(30)	Name of the column that is being indexed
COLUMN_POSITION	NUMBER	Position of the column in the index key (this is mostly relevant for multicolumn indexes)

V$INSTANCE

This view displays the state of the current instance. This version of V$INSTANCE is not compatible with earlier versions of V$INSTANCE.

Column	Datatype	Description
INSTANCE_NUMBER	NUMBER	Instance number used for instance registration. Corresponds to INSTANCE_NUMBER initialization parameter See "INSTANCE_NUMBER"
INSTANCE_NAME	VARCHAR2(16)	Instance name
HOST_NAME	VARCHAR2(64)	Name of the host machine
VERSION	VARCHAR2(17)	RDBMS version
STARTUP_TIME	DATE	Time when instance was started up
STATUS	VARCHAR2(7)	STARTED / MOUNTED / OPEN STARTED after startup nomount MOUNTED after startup mount or alter database close OPEN after startup or after database open
PARALLEL	VARCHAR2(3)	YES / NO in parallel server mode
THREAD#	NUMBER	Redo thread opened by the instance
ARCHIVER	VARCHAR2(7)	STOPPED | STARTED | FAILED. FAILED means that the archiver failed to archive a log last time, but will try again within 5 minutes
LOG_SWITCH_WAIT	VARCHAR2(11)	ARCHIVE LOG / CLEAR LOG / CHECKPOINT event log switching is waiting for. Note that if ALTER SYSTEM SWITCH LOGFILE is hung, but there is room in the current online redo log, then value is NULL
LOGINS	VARCHAR2(10)	ALLOWED / RESTRICTED
SHUTDOWN_PENDING	VARCHAR2(3)	YES / NO
DATABASE_STATUS	VARCHAR2(17)	The status of the database
INSTANCE_ROLE	VARCHAR2(16)	Describes whether the instance is an active instance (PRIMARY_INSTANCE) or a inactive secondary instance (SECONDARY_INSTANCE), or UNKNOWN if the instance has been started but not mounted. **Note:** SECONDARY_INSTANCE is displayed only if your cluster has just two instances **and** you have set the ACTIVE_INSTANCE_COUNT initialization parameter to 1.

V$INSTANCE_RECOVERY

This view monitors the mechanisms available to users to limit recovery I / O. Those mechanisms are:

- Set the initialization parameter LOG_CHECKPOINT_TIMEOUT.
- Set the initialization parameter LOG_CHECKPOINT_INTERVAL.
- Set the initialization parameter FAST_START_IO_TARGET.
- Set the size of the smallest redo log.

See Also:

- *Oracle8i Designing and Tuning for Performance* for more information on limiting recovery I/O.
- LOG_CHECKPOINT_INTERVAL on page 1-63.
- FAST_START_IO_TARGET on page 1-38.

Column	Datatype	Description
RECOVERY_EXTIMATED_IOS	NUMBER	Estimated number of blocks that would be processed during recovery. This estimate is based on the FAST_START_IO_TARGET initialization parameter. The value is not meaningful unless that parameter is driving checkpointing behavior (see the TARGET_REDO_BLOCKS column below).
ACTUAL_REDO_BLOCKS	NUMBER	The current actual number of redo blocks required for recovery.
TARGET_REDO_BLOCKS	NUMBER	The current target number of redo blocks that must be processed for recovery. This value is the minimum of the following four columns, and indicates which of those four user-defined limits dominates checkpointing.
LOG_FILE_SIZE_REDO_BLKS	NUMBER	Maximum number of redo blocks required to guarantee that a log switch does not occur before the checkpoint completes.
LOG_CHKPT_TIMEOUT_REDO_BLKS	NUMBER	Number of redo blocks that need to be processed during recovery to satisfy the LOG_CHECKPOINT_TIMEOUT parameter. The value displayed is not meaningful unless that parameter has been set.
LOG_CHKPT_INTERVAL_REDO_BLKS	NUMBER	Number of redo blocks that need to be processed during recovery to satisfy the LOG_CHECKPOINT_INTERVAL parameter. The value displayed is not meaningful unless that parameter has been set.
FAST_START_IO_TARGET_REDO_BLKS	NUMBER	Number of redo blocks that need to be processed during recovery to satisfy the FAST_START_IO_TARGET parameter. The value displayed is not meaningful unless that parameter has been set.

V$LATCH

This view lists statistics for non-parent latches and summary statistics for parent latches. That is, the statistics for a parent latch include counts from each of its children.

Note: Columns SLEEP5, SLEEP6,... SLEEP11 are present for compatibility with previous versions of Oracle. No data are accumulated for these columns.

Column	Datatype	Description
ADDR	RAW(4)	Address of latch object
LATCH#	NUMBER	Latch number
LEVEL#	NUMBER	Latch level
NAME	VARCHAR2(64)	Latch name
GETS	NUMBER	Number of times obtained a wait
MISSES	NUMBER	Number of times obtained a wait but failed on the first try
SLEEPS	NUMBER	Number of times slept when wanted a wait
IMMEDIATE_GETS	NUMBER	Number of times obtained without a wait
IMMEDIATE_MISSES	NUMBER	Number of times failed to get without a wait
WAITERS_WOKEN	NUMBER	How many times a wait was awakened
WAITS_HOLDING _ LATCH	NUMBER	Number of waits while holding a different latch
SPIN_GETS	NUMBER	Gets that missed first try but succeeded on spin
SLEEP1 ... SLEEP11	NUMBER	Waits that slept 1 time through 11 times, respectively

V$LATCHHOLDER

This view contains information about the current latch holders.

Column	Datatype	Description
PID	NUMBER	Identifier of process holding the latch
SID	NUMBER	Identifier of the session that owns the latch
LADDR	RAW(4)	Latch address
NAME	VARCHAR2	Name of latch being held

V$LATCHNAME

This view contains information about decoded latch names for the latches shown in V$LATCH. The rows of V$LATCHNAME have a one-to-one correspondence to the rows of V$LATCH.

See Also: "V$LATCH" on page 3-53.

Column	Datatype	Description
LATCH#	NUMBER	Latch number
NAME	VARCHAR2(64)	Latch name

V$LATCH_CHILDREN

This view contains statistics about child latches. This view includes all columns of V$LATCH plus the CHILD# column. Note that child latches have the same parent if their LATCH# columns match each other.

See Also: "V$LATCH" on page 3-53.

Column	Datatype	Description
ADDR	RAW(4)	Address of latch object
LATCH#	NUMBER	Latch number for a parent latch
CHILD#	NUMBER	Child number of a parent latch shown in LATCH#
LEVEL#	NUMBER	Latch level
NAME	VARCHAR2(64)	Latch name
GETS	NUMBER	Number of times obtained a wait
MISSES	NUMBER	Number of times obtained a wait but failed on the first try
SLEEPS	NUMBER	Number of times slept when wanted a wait
IMMEDIATE_GETS	NUMBER	Number of times obtained without a wait
IMMEDIATE_MISSES	NUMBER	Number of time failed to get without a wait
WAITERS_WOKEN	NUMBER	How many times a wait was awakened
WAITS_HOLDING_LATCH	NUMBER	Number of waits while holding a different latch
SPIN_GETS	NUMBER	Gets that missed first try but succeeded on spin
SLEEPn	NUMBER	Waits that slept *n* times

V$LATCH_MISSES

This view contains statistics about missed attempts to acquire a latch.

Column	Datatype	Description
PARENT_NAME	VARCHAR2	Latch name of a parent latch
WHERE	VARCHAR2	Location that attempted to acquire the latch
NWFAIL_COUNT	NUMBER	Number of times that no-wait acquisition of the latch failed
SLEEP_COUNT	NUMBER	Number of times that acquisition attempts caused sleeps
WTR_SLP_COUNT	NUMBER	???
LONGHOLD_COUNT	NUMBER	???

V$LATCH_PARENT

This view contains statistics about the parent latch. The columns of V$LATCH_PARENT are identical to those in V$LATCH.

See Also: "V$LATCH" on page 3-53.

V$LIBRARYCACHE

This view contains statistics about library cache performance and activity.

Column	Datatype	Description
NAMESPACE	VARCHAR2(15)	The library cache namespace
GETS	NUMBER	The number of times a lock was requested for objects of this namespace
GETHITS	NUMBER	The number of times an object's handle was found in memory
GETHITRATIO	NUMBER	The ratio of GETHITS to GETS
PINS	NUMBER	The number of times a PIN was requested for objects of this namespace
PINHITS	NUMBER	The number of times all of the metadata pieces of the library object were found in memory
PINHITRATIO	NUMBER	The ratio of PINHITS to PINS
RELOADS	NUMBER	Any PIN of an object that is not the first PIN performed since the object handle was created, and which requires loading the object from disk
INVALIDATIONS	NUMBER	The total number of times objects in this namespace were marked invalid because a dependent object was modified
DLM_LOCK_REQUESTS	NUMBER	The number of GET requests lock instance locks

Column	Datatype	Description
DLM_PIN_REQUESTS	NUMBER	The number of PIN requests lock instance locks
DLM_PIN_RELEASES	NUMBER	The number of release requests PIN instance locks
DLM_INVALIDATION _ REQUESTS	NUMBER	The number of GET requests for invalidation instance locks
DLM_INVALIDATIONS	NUMBER	The number of invalidation pings received from other instances

V$LICENSE

This view contains information about license limits.

Column	Datatype	Description
SESSIONS_MAX	NUMBER	Maximum number of concurrent user sessions allowed for the instance
SESSIONS_WARNING	NUMBER	Warning limit for concurrent user sessions for the instance
SESSIONS_CURRENT	NUMBER	Current number of concurrent user sessions
SESSIONS _HIGHWATER	NUMBER	Highest number of concurrent user sessions since the instance started
USERS_MAX	NUMBER	Maximum number of named users allowed for the database

V$LOADCSTAT

This view contains SQL*Loader statistics compiled during the execution of a direct load. These statistics apply to the whole load. Any SELECT against this table results in "no rows returned" since you cannot load data and do a query at the same time.

Column	Datatype	Description
READ	NUMBER	Number of records read
REJECTED	NUMBER	Number of records rejected
TDISCARD	NUMBER	Total number of discards during the load
NDISCARD	NUMBER	Number of discards from the current file

V$LOADTSTAT

SQL*Loader statistics compiled during the execution of a direct load. These statistics apply to the current table. Any SELECT against this table results in "no rows returned" since you cannot load data and do a query at the same time.

Column	Datatype	Description
LOADED	NUMBER	Number of records loaded
REJECTED	NUMBER	Number of records rejected
FAILWHEN	NUMBER	Number of records that failed to meet any WHEN clause
ALLNULL	NUMBER	Number of records that were completely null and were therefore not loaded
LEFT2SKIP	NUMBER	Number of records yet to skip during a continued load
PTNLOADED	NUMBER	Number of records loaded PTN

V$LOCK

This view lists the locks currently held by the Oracle server and outstanding requests for a lock or latch.

Column	Datatype	Description
ADDR	RAW(4)	Address of lock state object
KADDR	RAW(4)	Address of lock
SID	NUMBER	Identifier for session holding or acquiring the lock
TYPE	VARCHAR2(2)	Type of user or system lock. The locks on the user types are obtained by user applications. Any process that is blocking others is likely to be holding one of these locks. The user type locks are: TM: DML enqueue TX: Transaction enqueue UL: User supplied The locks on the system types are held for extremely short periods of time. The system type locks are listed in Table 3–1.
ID1	NUMBER	Lock identifier #1 (depends on type)
ID2	NUMBER	Lock identifier #2 (depends on type)
LMODE	NUMBER	Lock mode in which the session holds the lock: ■ 0, None ■ 1, Null (NULL) ■ 2, Row-S (SS) ■ 3, Row-X (SX) ■ 4, Share (S) ■ 5, S / Row-X (SSX) ■ 6, Exclusive (X)

Column	Datatype	Description
REQUEST	NUMBER	Lock mode in which the process requests the lock: ■ 0, None ■ 1, Null (NULL) ■ 2, Row-S (SS) ■ 3, Row-X (SX) ■ 4, Share (S) ■ 5, S / Row-X (SSX) ■ 6, Exclusive (X)
CTIME	NUMBER	Time since current mode was granted
BLOCK	NUMBER	The lock is blocking another lock

Table 3–1 Values for the TYPE column: System Types

System Type	Description	System Type	Description
BL	Buffer hash table instance	NA..NZ	Library cache pin instance (A..Z = namespace)
CF	Control file schema global enqueue	PF	Password File
CI	Cross-instance function invocation instance	PI, PS	Parallel operation
CU	Cursor bind	PR	Process startup
DF	Data file instance	QA..QZ	Row cache instance (A..Z = cache)
DL	Direct loader parallel index create	RT	Redo thread global enqueue
DM	Mount / startup db primary / secondary instance	SC	System commit number instance
DR	Distributed recovery process	SM	SMON
DX	Distributed transaction entry	SN	Sequence number instance
FS	File set	SQ	Sequence number enqueue
HW	Space management operations on a specific segment	SS	Sort segment
IN	Instance number	ST	Space transaction enqueue
IR	Instance recovery serialization global enqueue	SV	Sequence number value
IS	Instance state	TA	Generic enqueue
IV	Library cache invalidation instance	TS	Temporary segment enqueue (ID2=0)
JQ	Job queue	TS	New block allocation enqueue (ID2=1)
KK	Thread kick	TT	Temporary table enqueue

Table 3–1 Values for the TYPE column: System Types

System Type	Description	System Type	Description
LA .. LP	Library cache lock instance lock (A..P = namespace)	UN	User name
MM	Mount definition global enqueue	US	Undo segment DDL
MR	Media recovery	WL	Being-written redo log instance

V$LOCK_ACTIVITY

This is an Oracle Parallel Server view. V$LOCK_ACTIVITY displays the DLM lock operation activity of the current instance. Each row corresponds to a type of lock operation.

See Also: *Oracle8i Parallel Server Concepts*.

Column	Datatype	Description
FROM_VAL	VARCHAR2(4)	PCM lock initial state: NULL; S; X; SSX
TO_VAL	VARCHAR2(4)	PCM lock initial state: NULL; S; X; SSX
ACTION_VAL	VARCHAR2(51)	Description of lock conversions ■ Lock buffers for read ■ Lock buffers for write ■ Make buffers CR (no write) ■ Upgrade read lock to write ■ Make buffers CR (write dirty buffers) ■ Downgrade write lock to read (write dirty buffers) ■ Write transaction table / undo blocks ■ Transaction table / undo blocks (write dirty buffers) ■ Make transaction table / undo blocks available share ■ Rearm transaction table write mechanism
COUNTER	NUMBER	Number of times the lock operation executed

V$LOCK_ELEMENT

This is an Oracle Parallel Server view. There is one entry in V$LOCK_ELEMENT for each PCM lock that is used by the buffer cache. The name of the PCM lock that corresponds to a lock element is {'BL', indx, class}.

See Also: *Oracle8i Parallel Server Concepts*.

Column	Datatype	Description
LOCK_ELEMENT_ ADDR	RAW(4)	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_NAME	NUMBER	The name of the lock that contains the PCM lock that is covering the buffer.
INDX	NUMBER	Platform specific lock manager identifier
CLASS	NUMBER	Platform specific lock manager identifier
MODE_HELD	NUMBER	Platform dependent value for lock mode held; often: 3 = share; 5 = exclusive
BLOCK_COUNT	NUMBER	Number of blocks covered by PCM lock
RELEASING	NUMBER	Non-zero if PCM lock is being downgraded
ACQUIRING	NUMBER	Non-zero if PCM lock is being upgraded
INVALID	NUMBER	Non-zero if PCM lock is invalid. (A lock may become invalid after a system failure.)
FLAGS	NUMBER	Process level flags for the LE

V$LOCKED_OBJECT

This view lists all locks acquired by every transaction on the system.

Column	Datatype	Description
XIDUSN	NUMBER	Undo segment number
XIDSLOT	NUMBER	Slot number
XIDSQN	NUMBER	Sequence number
OBJECT_ID	NUMBER	Object ID being locked
SESSION_ID	NUMBER	Session ID
ORACLE_USERNAME	VARCHAR2(30)	Oracle user name
OS_USER_NAME	VARCHAR2(15)	OS user name
PROCESS	VARCHAR2(9)	OS process ID
LOCKED_MODE	NUMBER	Lock mode

V$LOCKS_WITH_COLLISIONS

This is an Oracle Parallel Server view. Use this view to find the locks that protect multiple buffers, each of which has been either force-written or force-read at least 10

times. It is very likely that those buffers are experiencing false pings due to being mapped to the same lock.

See Also: *Oracle8i Parallel Server Concepts*.

Column	Datatype	Description
LOCK_ELEMENT_ ADDR	RAW(4)	The address of the lock element that contains the PCM lock covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock

V$LOG

This view contains log file information from the control files.

Column	Datatype	Description
GROUP#	NUMBER	Log group number
THREAD#	NUMBER	Log thread number
SEQUENCE#	NUMBER	Log sequence number
BYTES	NUMBER	Size of the log in bytes
MEMBERS	NUMBER	Number of members in the log group
ARCHIVED	VARCHAR2	Archive status: YES, NO
STATUS	VARCHAR2(16)	Log status: ■ UNUSED: The online redo log has never been written to. This is the state of a redo log that was just added, or just after a RESETLOGS, when it is not the current redo log. ■ CURRENT: This is the current redo log. This implies that the redo log is active. The redo log could be open or closed. ■ ACTIVE: The log is active but is not the current log. It is needed for crash recovery. It may be in use for block recovery. It might or might not be archived. ■ CLEARING: The log is being re-created as an empty log after an ALTER DATABASE CLEAR LOGFILE command. After the log is cleared, the status changes to UNUSED. ■ CLEARING_CURRENT: The current log is being cleared of a closed thread. The log can stay in this status if there is some failure in the switch such as an I/O error writing the new log header. ■ INACTIVE: The log is no longer needed for instance recovery. It may be in use for media recovery. It might or might not be archived.
FIRST_CHANGE#	NUMBER	Lowest SCN in the log
FIRST_TIME	DATE	Time of first SCN in the log

V$LOGFILE

This view contains information about redo log files.

Column	Datatype	Description
GROUP#	NUMBER	Redo log group identifier number
STATUS	VARCHAR2	Status of this log member: INVALID (file is inaccessible), STALE (file's contents are incomplete), DELETED (file is no longer used), or blank (file is in use)
MEMBER	VARCHAR2	Redo log member name

V$LOGHIST

This view contains log history information from the control file. This view is retained for historical compatibility. Use of V$LOG_HISTORY is recommended instead.

See Also: "V$LOG_HISTORY" on page 3-65.

Column	Datatype	Description
THREAD#	NUMBER	Log thread number
SEQUENCE#	NUMBER	Log sequence number
FIRST_CHANGE#	NUMBER	Lowest SCN in the log
FIRST_TIME	DATE	Time of first SCN in the log
SWITCH_CHANGE#	NUMBER	SCN at which the log switch occurred; one more than highest SCN in the log

V$LOGMNR_CONTENTS

This view contains log history information.

Column	Datatype	Description
SCN	NUMBER(15)	The system change number
TIMESTAMP	DATE	The timestamp
THREAD#	NUMBER	The thread number
LOG_ID	NUMBER	The log ID
XIDUSN	NUMBER	The transaction ID undo segment number
XIDSLOT	NUMBER	The transaction ID slot number

Column	Datatype	Description
XIDSQN	NUMBER	The transaction ID log sequence number
RBASQN	NUMBER	The RBA log sequence number
RBABLK	NUMBER	The RBA block number
RBABYTE	NUMBER	The RBA byte offset
UBAFIL	NUMBER	The UBA file number
UBABLK	NUMBER	The UBA block number
UBAREC	NUMBER	The UBA record index
UBASQN	NUMBER	The UBA undo block sequence number
ABS_FILE#	NUMBER	The data block absolute file number
REL_FILE#	NUMBER	The data block relative file number
DATA_BLK#	NUMBER	The data block number
DATA_OBJ#	NUMBER	The data block object number
DATA_DOBJ#	NUMBER	The data block data object number
SEG_OWNER	VARCHAR2(30)	The owner name of the segment
SEG_NAME	VARCHAR2(81)	The segment name
SEG_TYPE	NUMBER	The segment type
TABLE_SPACE_NAME	VARCHAR2(30)	The tablespace name of segment
ROW_ID	VARCHAR2(18)	The row ID
SESSION#	NUMBER	The session number
SERIAL#	NUMBER	The serial number
USER_NAME	VARCHAR2(30)	The user name
SESSION_INFO	VARCHAR2(4000)	Session information
ROLLBACK	NUMBER	The rollback request
OPERATION	VARCHAR2(30)	The operation
SQL_REDO	VARCHAR2(4000)	SQL redo
SQL_UNDO	VARCHAR2(4000)	SQL undo
RS_ID	VARCHAR2(30)	Record set ID
SSN	NUMBER	SQL sequence number
CSF	NUMBER	Continuation SQL flag
INFO	VARCHAR2(32)	Informational message
STATUS	VARCHAR2 (16)	The status

V$LOGMNR_DICTIONARY

This view contains log history information.

Column	Datatype	Description
TIMESTAMP	DATE	The date the dictionary was created
DB_ID	NUMBER	The database ID
DB_NAME	VARCHAR2(8)	The name of the database
FILENAME	VARCHAR2(513)	The dictionary filename
DICTIONARY_SCN	NUMBER	The system change number when the dictionary was created
RESET_SCN	NUMBER	The reset log SCN when the dictionary was created
RESET_SCN_TIME	NUMBER	The time when the reset log SCN was obtained to create the dictionary
ENABLED_THREAD_MAP	RAW(16)	Bit map of currently enabled threads when the dictionary was created
INFO	VARCHAR2(32)	Informational / Status message BAD_DATE indicates that the SCN of the dictionary file does not match the SCN range of the log files
STATUS	NUMBER	A NULL indicates a valid dictionary file for the list of log files. A non-NULL value indicates further information is contained in the INFO column as a text string

V$LOGMNR_LOGS

This view contains log information.

Column	Datatype	Description
LOG_ID	NUMBER	Identifies the log file. The value of this field is also reported in the LOG_ID column of the V$LOG
FILENAME	VARCHAR2(513)	The filename
LOW_TIME	DATE	The oldest date of any records in the file
HIGH_TIME	DATE	The most recent date of any records in the file
DB_ID	NUMBER	The database ID
DB_NAME	VARCHAR2(8)	The name of the database
RESET_SCN	NUMBER	The reset log SCN when the log was created
RESET_SCN_TIME	NUMBER	The time when the reset log SCN was obtained to create the log
THREAD_ID	NUMBER	The thread number
THREAD_SQN	NUMBER	The thread sequence number
LOW_SCN	NUMBER	SCN allocated when log switched into

Column	Datatype	Description
NEXT_SCN	NUMBER	SCN after this log. Low SCN of the next log
INFO	VARCHAR2(32)	Informational message. A value of MISSING_LOGFILE will be assigned to a row entry where a needed logfile is missing from the list of log files
STATUS	NUMBER	Indicates the status of a logfile. A NULL value indicates a valid logfile; a non-NULL value indicates further information is contained in the INFO column as a text string. All logfiles successfully added to the file list will have a status value of NULL

V$LOGMNR_PARAMETERS

This view contains log information.

Column	Datatype	Description
START_DATE	DATE	The date to start search at
END_DATE	DATE	The date to end search at
START_SCN	NUMBER	The system change number to start search
END_SCN	NUMBER	The system change number to end search
INFO	VARCHAR2(32)	An informational message
STATUS	NUMBER	The status. A NULL value indicates parameters are valid. A non-NULL value indicates further information is contained in the INFO column as a text string

V$LOG_HISTORY

This view contains log history information from the control file.

Column	Datatype	Description
THREAD#	NUMBER	Thread number of the archived log
SEQUENCE#	NUMBER	Sequence number of the archived log
FIRST_TIME	DATE	Time of first entry (lowest SCN) in the log. This column was previously named TIME
FIRST_CHANGE#	NUMBER	Lowest SCN in the log. This column was previously named LOW_CHANGE#
NEXT_CHANGE#	NUMBER	Highest SCN in the log. This column was previously named HIGH_CHANGE#
RECID	NUMBER	Control file record ID
STAMP	NUMBER	Control file record stamp

V$MTS

This view contains information for tuning the multi-threaded server.

Column	Datatype	Description
MAXIMUM__ CONNECTIONS	NUMBER	The highest number of virtual circuits in use at one time since the instance started. If this value reaches the value set for the MTS_CIRCUITS initialization parameter, consider raising the value of MTS_CIRCUITS. **See Also:** "MTS_CIRCUITS" on page 1-68.
MAXIMUM_ SESSIONS	NUMBER	The highest number of multi-threaded server sessions in use at one time since the instance started. If this reaches the value set for the MTS_SESSIONS initialization parameter, consider raising the value of MTS_SESSIONS. **See Also:** "MTS_SESSIONS" on page 1-73.
SERVERS _ STARTED	NUMBER	The total number of multi-threaded servers started since the instance started (but not including those started during startup)
SERVERS _ TERMINATED	NUMBER	The total number of multi-threaded servers stopped by Oracle since the instance started
SERVERS _ HIGHWATER	NUMBER	The highest number of servers running at one time since the instance started. If this value reaches the value set for the MTS_MAX_SERVERS initialization parameter, consider raising the value of MTS_SERVERS. **See Also:** "MTS_SERVERS" on page 1-72.

V$MYSTAT

This view contains statistics on the current session.

Column	Datatype	Description
SID	NUMBER	The ID of the current session
STATISTIC#	NUMBER	The number of the statistic
VALUE	NUMBER	The value of the statistic

V$NLS_PARAMETERS

This view contains current values of NLS parameters.

Column	Datatype	Description
PARAMETER	VARCHAR2	Parameter name: NLS_CALENDAR; NLS_CHARACTERSET; NLS_CURRENCY, NLS_DATE_FORMAT; NLS_DATE_LANGUAGE; NLS_ISO_CURRENCY, NLS_ LANGUAGE; NLS_NUMERIC_CHARACTERS; NLS_SORT, NLS_TERRITORY; NLS_UNION_CURRENCY; NLS_NCHAR_CHARACTERSET, NLS_COMP

Column	Datatype	Description
VALUE	VARCHAR2	NLS parameter value

V$NLS_VALID_VALUES

This view lists all valid values for NLS parameters.

Column	Datatype	Description
PARAMETER	VARCHAR2(64)	Parameter name: LANGUAGE; SORT; TERRITORY; CHARACTERSET
VALUE	VARCHAR2(64)	NLS parameter value

V$OBJECT_DEPENDENCY

This view can be used to determine what objects are depended on by a package, procedure, or cursor that is currently loaded in the shared pool. For example, together with V$SESSION and V$SQL, it can be used to determine which tables are used in the SQL statement that a user is currently executing.

See Also: "V$SESSION" on page 3-90 and "V$SQL" on page 3-103.

Column	Datatype	Description
FROM_ADDRESS	RAW(4)	The address of a procedure, package, or cursor that is currently loaded in the shared pool
FROM_HASH	NUMBER	The hash value of a procedure, package, or cursor that is currently loaded in the shared pool
TO_OWNER	VARCHAR2(64)	The owner of the object that is depended on
TO_NAME	VARCHAR2(1000)	The name of the object that is depended on
TO_ADDRESS	RAW(4)	The address of the object that is depended on. These can be used to look up more information on the object in V$DB_OBJECT_CACHE
TO_HASH	NUMBER	The hash value of the object that is depended on. These can be used to look up more information on the object in V$DB_OBJECT_CACHE
TO_TYPE	NUMBER	The type of the object that is depended on

V$OBSOLETE_PARAMETER

This view lists obsolete parameters. If any value is true, you should examine why.

Column	Datatype	Description
NAME	VARCHAR2(64)	The name of the parameter
ISSPECIFIED	VARCHAR2(5)	Whether the parameter was specified in the config file

V$OFFLINE_RANGE

This view displays datafile offline information from the control file. Note that the last offline range of each datafile is kept in the DATAFILE record.

An offline range is created for a datafile when its tablespace is first altered to be OFFLINE NORMAL or READ ONLY, and then subsequently altered to be ONLINE or read-write. Note that no offline range is created if the datafile itself is altered to be OFFLINE or if the tablespace is altered to be OFFLINE IMMEDIATE.

See Also: "V$DATAFILE" on page 3-25.

Column	Datatype	Description
RECID	NUMBER	Record ID
STAMP	NUMBER	Record stamp
FILE#	NUMBER	Datafile number
OFFLINE_CHANGE#	NUMBER	SCN at which offlined
ONLINE_CHANGE#	NUMBER	SCN at which onlined
ONLINE_TIME	DATE	Time of offline SCN

V$OPEN_CURSOR

This view lists cursors that each user session currently has opened and parsed.

Column	Datatype	Description
SADDR	RAW	Session address
SID	NUMBER	Session identifier
USER_NAME	VARCHAR2(30)	User that is logged in to the session
ADDRESS	RAW	Used with HASH_VALUE to identify uniquely the SQL statement being executed in the session
HASH_VALUE	NUMBER	Used with ADDRESS to identify uniquely the SQL statement being executed in the session
SQL_TEXT	VARCHAR2(60)	First 60 characters of the SQL statement that is parsed into the open cursor

V$OPTION

This view lists options that are installed with the Oracle Server.

Column	Datatype	Description
PARAMETER	VARCHAR2(64)	The name of the option
VALUE	VARCHAR2(64)	TRUE if the option is installed

V$PARALLEL_DEGREE_LIMIT_MTH

This view displays all available parallel degree limit resource allocation methods.

Column	Datatype	Description
NAME	VARCHAR2(40)	The name of the parallel degree limit resource allocation method

V$PARAMETER

This view lists information about initialization parameters.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(64)	Parameter name
TYPE	NUMBER	Parameter type; 1 = Boolean; 2 = string; 3 = integer
VALUE	VARCHAR2(512)	Parameter value
ISDEFAULT	VARCHAR2(9)	Whether the parameter value is the default
ISSES_MODIFIABLE	VARCHAR2(5)	TRUE = the parameter can be changed with ALTER SESSION FALSE= the parameter cannot be changed with ALTER SESSION
ISSYS_MODIFIABLE	VARCHAR2(9)	IMMEDIATE = the parameter can be changed with ALTER SYSTEM DEFERRED=the parameter cannot be changed until the next session FALSE= the parameter cannot be changed with ALTER SYSTEM
ISMODIFIED	VARCHAR2(10)	Indicates how the parameter was modified. If an ALTER SESSION was performed, the value will be MODIFIED. If an ALTER SYSTEM (which will cause all the currently logged in sessions' values to be modified) was performed the value will be SYS_MODIFIED
ISADJUSTED	VARCHAR2(5)	Indicates that the rdbms adjusted the input value to a more suitable value (e.g., the parameter value should be prime, but the user input a non-prime number, so the rdbms adjusted the value to the next prime number)
DESCRIPTION	VARCHAR2(64)	A descriptive comment about the parameter

V$PING

This is an Oracle Parallel Server view. The V$PING view is identical to the V$CACHE view but only displays blocks that have been pinged at least once. This view contains information from the block header of each block in the SGA of the current instance as related to particular database objects.

See Also: "V$CACHE" on page 3-18 and *Oracle8i Parallel Server Concepts*.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query "DBA_DATA_FILES" on page 2-104 or "V$DBFILE" on page 3-29)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	Class number
STATUS	VARCHAR2(4)	Status of block: ■ FREE= not currently in use ■ XCUR= exclusive ■ SCUR= shared current ■ CR= consistent read ■ READ= being read from disk ■ MREC= in media recovery mode ■ IREC= in instance recovery mode
XNC	NUMBER	Number of PCM lock conversions due to contention with another instance. This column is obsolete but is retained for historical compatibility
FORCED_READS	NUMBER	Number of times the block had to be reread from disk because another instance had forced it out of this instance's cache by requesting the PCM lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times DBWR had to write this block to disk because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
NAME	VARCHAR2(30)	Name of the database object containing the block
PARTITION_NAME	VARCHAR2(30)	NULL for non-partitioned objects
KIND	VARCHAR2(15)	Type of database object. See Table 3–1 on page 3-58.
OWNER#	NUMBER	Owner number
LOCK_ELEMENT _ ADDR	RAW(4)	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock

Column	Datatype	Description
LOCK_ELEMENT _ NAME	NUMBER	The name of the lock that contains the PCM lock that is covering the buffer

V$PQ_SESSTAT

This view lists session statistics for parallel queries.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	The following statistics (fixed rows) have been defined for this view. After you have run a query or DML operation, you can use the information derived from V$PQ_ SESSTAT to view the number of slave processes used, and other information for the session and system. Name of the statistic: ■ Queries Parallelized: Number of queries run in parallel ■ DML Parallelized: Number of DML operations run in parallel ■ DFO Trees: Number of executed DFO trees ■ Server Threads: Total number of parallel servers used ■ Allocation Height: Requested number of servers per instance ■ Allocation Width: Requested number of instances ■ Local Msgs Sent: Number of local (intra-instance) messages sent ■ Distr Msgs Sent: Number of remote (inter-instance) messages sent ■ Local Msgs Recv'd: Number of local (intra-instance) messages received ■ Distr Msgs Recv'd: Number of remote (inter-instance) messages received
LAST_QUERY	NUMBER	The value of the statistic for the last operation
SESSION_TOTAL	NUMBER	The value of the statistic for the entire session to this point in time

V$PQ_SLAVE

This view lists statistics for each of the active parallel execution servers on an instance.

Column	Datatype	Description
SLAVE_NAME	VARCHAR2(4)	Name of the parallel execution server
STATUS	VARCHAR2(4)	The current status of the parallel execution server (BUSY or IDLE)
SESSIONS	NUMBER	The number of sessions that have used this parallel execution server

Column	Datatype	Description
IDLE_TIME_CUR	NUMBER	The amount of time spent idle while processing statements in the current session
BUSY_TIME_CUR	NUMBER	The amount of time spent busy while processing statements in the current session
CPU_SECS_CUR	NUMBER	The amount of CPU time spent on the current session
MSGS_SENT_CUR	NUMBER	The number of messages sent while processing statements for the current session
MSGS_RCVD_CUR	NUMBER	The number of messages received while processing statements for the current session
IDLE_TIME_TOTAL	NUMBER	The total amount of time this query server has been idle
BUSY_TIME_TOTAL	NUMBER	The total amount of time this query server has been active
CPU_SECS_TOTAL	NUMBER	The total amount of CPU time this query server has used to process statements
MSGS_SENT_TOTAL	NUMBER	The total number of messages this query server has sent
MSGS_RCVD_TOTAL	NUMBER	The total number of messages this query server has received

V$PQ_SYSSTAT

This view lists system statistics for parallel queries.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	The following statistics (fixed rows) have been defined for this view. After you have run a query or DML operation, you can use the information derived from V$PQ_SYSSTAT to view the number of slave processes used, and other information for the system. Name of the statistic: ■ **Servers Busy**: Number of currently busy servers on this instance ■ **Servers Idle**: Number of currently idle servers on this instance ■ **Servers Highwater**: Number of active servers on this instance that have partaken in >= 1 operation so far ■ **Server Sessions**: Total number of operations executed in all servers on this instance ■ **Servers Started**: Total number of servers started on this instance ■ **Servers Shutdown**: Total number of servers shutdown on this instance ■ **Servers Cleaned Up**: Total number of servers on this instance cleaned up due to process death

Column	Datatype	Description
		■ **Queries Initiated**: Total number of parallel queries initiated on this instance ■ **DML Initiated**: Total number of parallel DML operations that were initiated ■ **DFO Trees**: Total number of DFO trees executed on this instance ■ **Local Msgs Sent**: Total number of local (intra-instance) messages sent on this instance ■ **Distr Msgs Sent**: Total number of remote (inter-instance) messages sent on this instance ■ **Local Msgs Recv'd**: Total number of remote (inter-instance) messages received on this instance ■ **Distr Msgs Recv'd**: Total number of remote (inter-instance) messages received on this instance
VALUE	NUMBER	The value of the statistic

V$PQ_TQSTAT

This view contains statistics on parallel execution operations. The statistics are compiled after the query completes and only remain for the duration of the session. It displays the number of rows processed through each parallel execution server at each stage of the execution tree. This view can help determine skew problems in a query's execution.

Column	Datatype	Description
DFO_NUMBER	NUMBER	The data flow operator (DFO) tree number to differentiate queries
TQ_ID	NUMBER	The table queue ID within the query, which represents the connection between two DFO nodes in the query execution tree
SERVER_TYPE	VARCHAR2(10)	The role in table queue - producer/consumer/ranger
NUM_ROWS	NUMBER	The number of rows produced/consumed
BYTES	NUMBER	The number of bytes produced/consumed
OPEN_TIME	NUMBER	Time (secs) the table queue remained open
AVG_LATENCY	NUMBER	Time (ms) for a message to be dequeued after it enters the queue
WAITS	NUMBER	The number of waits encountered during dequeue
TIMEOUTS	NUMBER	The number of timeouts when waiting for a message
PROCESS	VARCHAR2(10)	Process ID
INSTANCE	NUMBER	Instance ID

V$PROCESS

This view contains information about the currently active processes. While the LATCHWAIT column indicates what latch a process is waiting for, the LATCHSPIN column indicates what latch a process is spinning on. On multi-processor machines, Oracle processes will spin on a latch before waiting on it.

Column	Datatype	Description
ADDR	RAW(4)	Address of process state object
PID	NUMBER	Oracle process identifier
SPID	VARCHAR2	Operating system process identifier
USERNAME	VARCHAR2	Operating system process username. Any Two-Task user coming across the network has "-T" appended to the username.
SERIAL#	NUMBER	Process serial number
TERMINAL	VARCHAR2	Operating system terminal identifier
PROGRAM	VARCHAR2	Program in progress
BACKGROUND	VARCHAR2	1 for a background process; NULL for a normal process
LATCHWAIT	VARCHAR2	Address of latch the process is waiting for; NULL if none
LATCHSPIN	VARCHAR2	Address of latch the process is being spun on; NULL if none

V$PROXY_ARCHIVEDLOG

This view contains descriptions of archived log backups which are taken with a new feature called Proxy Copy. Each row represents a backup of one archived log.

Column	Datatype	Description
RECID	NUMBER	Proxy copy record ID
STAMP	NUMBER	Proxy copy record stamp
DEVICE_TYPE	VARCHAR2(17)	Type of the device on which the copy resides
HANDLE	VARCHAR2(513)	Proxy copy handle identifies the copy for restore
COMMENTS	VARCHAR2(81)	Comment returned by the operating system or storage subsystem. This value is informational only; not needed for restore
MEDIA	VARCHAR2(65)	Name of the media on which the copy resides. This value is informational only. It is not needed for restore
MEDIA_POOL	NUMBER	The media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager BACKUP command

Column	Datatype	Description
STATUS	VARCHAR2(1)	Indicates the status of the copy: ■ A: The object is available. ■ D: The object is deleted. ■ X: The object has been "cross-checked" and found not to exist. A subsequent "delete expired" command will change the status to D. If, for some reason, the object really does still exist, then a subsequent "cross-check" command will change the status back to A.
THREAD#	NUMBER	Redo thread number
SEQUENCE#	NUMBER	Redo log sequence number
RESETLOGS _ CHANGE#	NUMBER	Resetlogs change number of the database when this log was written
RESETLOGS_TIME	DATE	Resetlogs time of the database when this log was written
FIRST_CHANGE#	NUMBER	First change number in the archived log
FIRST_TIME	DATE	Timestamp of the first change
NEXT_CHANGE#	NUMBER	First change number in the next log
NEXT_TIME	DATE	Timestamp of the next change
BLOCKS	NUMBER	Size of the archived log in blocks
BLOCK_SIZE	NUMBER	Redo log block size
START_TIME	DATE	The starting time
COMPLETION_TIME	DATE	The completion time
ELAPSED_SECONDS	NUMBER	The number of elapsed seconds

V$PROXY_DATAFILE

This view contains descriptions of datafile and control file backups which are taken with a new feature called Proxy Copy. Each row represents a backup of one database file.

Column	Datatype	Description
RECID	NUMBER	Proxy copy record ID
STAMP	NUMBER	Proxy copy record stamp
DEVICE_TYPE	VARCHAR2(17)	Type of the device on which the copy resides
HANDLE	VARCHAR2(513)	Proxy copy handle identifies the copy for restore
COMMENTS	VARCHAR2(81)	Comment returned by the operating system or storage subsystem. This value is informational only; not needed for restore

Column	Datatype	Description
MEDIA	VARCHAR2(65)	Name of the media on which the copy resides. This value is informational only; not needed for restore
MEDIA_POOL	NUMBER	The media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager BACKUP command
TAG	VARCHAR2(32)	Proxy copy tag
STATUS	VARCHAR2(1)	Indicates the status of the copy: ■ A: The object is available. ■ D: The object is deleted. ■ X: The object has been "cross-checked" and found not to exist. A subsequent "delete expired" command will change the status to D. If, for some reason, the object really does still exist, then a subsequent "cross-check" command will change the status back to A.
FILE#	NUMBER	Absolute datafile number, or 0 if this is a control file backup
CREATION_CHANGE#	NUMBER	Datafile creation change number
CREATION_TIME	DATE	Datafile creation Timestamp
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the datafile when the copy was made
RESETLOGS_TIME	DATE	Resetlogs timestamp of the datafile when the copy was made
CHECKPOINT _CHANGE#	NUMBER	Checkpoint change number of the datafile when the copy was made
CHECKPOINT_TIME	DATE	Checkpoint timestamp of the datafile when the copy was made
ABSOLUTE_FUZZY _ CHANGE#	NUMBER	The highest change in any block of the file, if known
RECOVERY_FUZZY _ CHANGE#	NUMBER	Highest change written to the file by media recovery
RECOVERY_FUZZY _TIME	DATE	Timestamp of the highest change written to the file by media recovery
INCREMENTAL_LEVEL	NUMBER	0 if this backup is part of an incremental backup strategy, otherwise NULL
ONLINE_FUZZY	VARCHAR2(3)	YES / NO. If set to YES, this copy was made after a crash or offline immediate (or is a copy of a copy which was taken improperly while the database was open). Recovery will need to apply all redo up to the next crash recovery marker to make the file consistent
BACKUP_FUZZY	VARCHAR2(3)	YES / NO. If set to YES, this is a copy taken using the BEGIN BACKUP / END BACKUP technique. Note that the BEGIN BACKUP / END BACKUP technique is used internally when proxy copies of open files are created. Recovery will need to apply all redo up to the end backup marker to make this copy consistent
BLOCKS	NUMBER	Size of the copy in blocks (also the size of the datafile when the copy was made)
BLOCK_SIZE	NUMBER	Block size of the datafile

Column	Datatype	Description
OLDEST_OFFLINE _ RANGE	NUMBER	If file# is 0 (ie, this is a control file backup), the RECID of the oldest offline range record in this control file copy. 0 for datafile copies
START_TIME	DATE	The starting time
COMPLETION_TIME	DATE	The completion time
ELAPSED_SECONDS	NUMBER	The number of elapsed seconds

V$PWFILE_USERS

This view lists users who have been granted SYSDBA and SYSOPER privileges as derived from the password file.

Column	Datatype	Description
USERNAME	VARCHAR2(30)	The name of the user that is contained in the password file
SYSDBA	VARCHAR2(5)	If the value of this column is TRUE, the user can connect with SYSDBA privileges
SYSOPER	VARCHAR2(5)	If the value of this column is TRUE, the user can connect with SYSOPER privileges

V$PX_PROCESS

This view contains information about the sessions running parallel execution.

Column	Datatype	Description
SERVER_NAME	VARCHAR2(4)	The name of the parallel server (P000, P001, etc)
STATUS	VARCHAR2(9)	The state of the parallel server. Either In Use or Available
PID	NUMBER	The process identifier
SPID	VARCHAR2(9)	The OS process ID
SID	NUMBER	The session ID of slave, if in use
SERIAL#	NUMBER	The session serial number of slave, if in use

V$PX_PROCESS_SYSSTAT

This view contains information about the sessions running parallel execution.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	The name of the statistic: ■ **Servers In Use:** The number of PX servers currently performing parallel operations ■ **Servers Available:** The number of PX servers available to perform parallel operations ■ **Servers Started:** The number of times the system has had to create a PX server process ■ **Server Shutdown:** The number of times a PX server process has been shutdown. A PX server process will be shutdown if it has not been used recently. The length of time it may remain "Available" is controlled by the initialization parameter PARALLEL_SERVER_IDLE_TIME If this value is large, consider increasing the parameter. This will increase performance due to avoiding the latency of PX server process creation ■ **Servers HWM:** The maximum number of concurrent PX server processes If this number is equal to the initialization parameter PARALLEL_MAX_SERVERS, consider increasing the parameter. This could allow you to increase your throughput, especially if your system is under-utilized and the V$SYSSTAT statistic "Parallel operations downgraded to serial" is large ■ **Servers Cleaned Up:** The number of times PMON had to clean up a PX server. This should only happen during abnormal termination of a parallel operation If this number is large, it is recommended that the cause be determined ■ **Sessions:** The total number of sessions created by all PX servers ■ **Memory Chunks Allocs:** The number of large memory chunks allocated by PX servers ■ **Memory Chunks Freed:** The number of large memory chunks freed ■ **Memory Chunks Current:** The number of large memory chunks currently being used ■ **Memory Chunks HWM:** The maximum number of concurrently allocated chunks ■ **Buffers allocated:** The number of times a message buffer has been allocated ■ **Buffers freed:** The number of times a message buffer has been freed ■ **Buffers Current:** The number of message buffers currently being used ■ **Buffers HWM:** The maximum number of concurrently allocated message buffers
VALUE	NUMBER	The value of the statistic

V$PX_SESSION

This view contains information about the sessions running parallel execution.

Column	Datatype	Description
SADDR	RAW(4)	Session address
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number
QCSID	NUMBER	Session identifier of the parallel coordinator
QCSERIAL#	NUMBER	Session serial number of the parallel coordinator
QCINST_ID	NUMBER	Instance number on which the parallel coordinator is running
SERVER_GROUP	NUMBER	The logical group of servers to which this parallel server process belongs
SERVER_SET	NUMBER	The logical set of servers that this parallel server process belongs to. A single server group will have at most two server sets
SERVER#	NUMBER	The logical number of a parallel server process within a server set
DEGREE	NUMBER	The degree of parallelism being used by the server set
REQ_DEGREE	NUMBER	The degree of parallelism that was requested by the user when the statement was issued and prior to any resource, multi-user, or load balancing reductions

V$PX_SESSTAT

This view contains information about the sessions running parallel execution.

Column	Datatype	Description
SADDR	RAW(4)	Session address
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number
QCSID	NUMBER	Session identifier of the parallel coordinator
QCSERIAL#	NUMBER	Session serial number of the parallel coordinator
QCINST_ID	NUMBER	Instance number on which the parallel coordinator is running
SERVER_GROUP	NUMBER	The logical group of servers to which this parallel server process belongs
SERVER_SET	NUMBER	The logical set of servers that this parallel server process belongs to. A single server group will have at most two server sets
SERVER#	NUMBER	The logical number of a parallel server process within a server set
DEGREE	NUMBER	The degree of parallelism being used by the server set

Column	Datatype	Description
REQ_DEGREE	NUMBER	The degree of parallelism that was requested by the user when the statement was issued and prior to any resource, multi-user, or load balancing reductions
STATISTIC#	NUMBER	Statistic number (identifier)
VALUE	NUMBER	Statistic value

V$QUEUE

This view contains information on the multi-thread message queues.

Column	Datatype	Description
PADDR	RAW(4)	Address of the process that owns the queue
TYPE	VARCHAR2	Type of queue: COMMON (processed by servers), DISPATCHER
QUEUED	NUMBER	Number of items in the queue
WAIT	NUMBER	Total time that all items in this queue have waited. Divide by TOTALQ for average wait per item
TOTALQ	NUMBER	Total number of items that have ever been in the queue

V$RECOVER_FILE

This view displays the status of files needing media recovery.

Column	Datatype	Description
FILE#	NUMBER	File identifier number
ONLINE	VARCHAR2	Online status: ONLINE, OFFLINE
ERROR	VARCHAR2	Why the file needs to be recovered: NULL if reason unknown, or OFFLINE NORMAL if recovery not needed
CHANGE#	NUMBER	SCN where recovery must start
TIME	DATE	Time of SCN when recovery must start

V$RECOVERY_FILE_STATUS

V$RECOVERY_FILE_STATUS contains one row for each datafile for each RECOVER command. This view contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V$RECOVERY_FILE_STATUS will be empty to all other Oracle users.

See Also: *Oracle8i Recovery Manager User's Guide and Reference*.

Column	Datatype	Description
FILENUM	NUMBER	The number of the file being recovered
FILENAME	VARCHAR2(257)	The filename of the datafile being recovered
STATUS	VARCHAR2(13)	The status of the recovery. Contains one of the following values: IN RECOVERY; CURRENT; NOT RECOVERED

V$RECOVERY_LOG

This view lists information about archived logs that are needed to complete media recovery. This information is derived from the log history view, V$LOG_HISTORY.

V$RECOVERY_LOG contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V$RECOVERY_LOG will be empty to all other Oracle users.

See Also: "V$LOG_HISTORY" on page 3-65 and *Oracle8i Recovery Manager User's Guide and Reference*.

For more information, see .

Column	Datatype	Description
THREAD#	NUMBER	Thread number of the archived log
SEQUENCE#	NUMBER	Sequence number of the archived log
TIME	VARCHAR2	Time of first entry (lowest SCN) in the log
ARCHIVE_NAME	VARCHAR2	Name of the file when archived, using the naming convention specified by "LOG_ARCHIVE_FORMAT"

V$RECOVERY_PROGRESS

V$RECOVERY_PROGRESS can be used to track database recovery operations to ensure that they are not stalled, and also to estimate the time required to complete the operation in progress.

V$RECOVERY_PROGRESS is a subview of V$SESSION_LONGOPS.

See Also: *Oracle8i Backup and Recovery Guide*.

Column	Datatype	Description
TYPE	VARCHAR2(64)	The type of recovery operation being performed
ITEM	VARCHAR2(32)	The item being measured
SOFAR	NUMBER	The amount of work done so far
TOTAL	NUMBER	The total amount of work expected

V$RECOVERY_STATUS

V$RECOVERY_STATUS contains statistics of the current recovery process. This view contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V$RECOVERY_STATUS will be empty to all other Oracle users.

See Also: *Oracle8i Recovery Manager User's Guide and Reference.*

Column	Datatype	Description
RECOVERY_CHECKPOINT	DATE	The point in time to which the recovery has occurred. If no logs have been applied, this is the point in time the recovery starts
THREAD	NUMBER	The number of the redo thread currently being processed
SEQUENCE_NEEDED	NUMBER	Log sequence number of the log needed by the recovery process. The value is 0 if no log is needed
SCN_NEEDED	VARCHAR2(16)	The low SCN of the log needed by recovery. The value is 0 if unknown or no log is needed
TIME_NEEDED	DATE	Time when the log was created. The value is midnight on 1/1/88 if the time is unknown or if no log is needed
PREVIOUS_LOG_NAME	VARCHAR2(257)	The filename of the log
PREVIOUS_LOG_STATUS	VARCHAR2(13)	The status of the previous log. Contains one of the following values: RELEASE; WRONG NAME; MISSING NAME; UNNEEDED NAME; NONE
REASON	VARCHAR2(13)	The reason recovery is returning control to the user. Contains one of the following values: NEED LOG; LOG REUSED; THREAD DISABLED

V$REQDIST

This view lists statistics for the histogram of MTS dispatcher request times, divided into 12 buckets, or ranges of time. The time ranges grow exponentially as a function of the bucket number.

Column	Datatype	Description
BUCKET	NUMBER	Bucket number: 0 - 11; the maximum time for each bucket is (4 * 2^N)/100 seconds
COUNT	NUMBER	Count of requests whose total time to complete (excluding wait time) falls in this range

V$RESERVED_WORDS

This view gives a list of all the keywords that are used by the PL/SQL compiler. This view helps developers to determine whether a word is already being used as a keyword in the language.

Column	Datatype	Description
KEYWORD	VARCHAR2(64)	The name of the keyword
LENGTH	NUMBER	The length of the keyword

V$RESOURCE

This view contains resource name and address information.

Column	Datatype	Description
ADDR	RAW(4)	Address of resource object
TYPE	VARCHAR2	Resource type. The resource types are listed in Table 3–1 on page 3-58.
ID1	NUMBER	Resource identifier #1
ID2	NUMBER	Resource identifier #2

V$RESOURCE_LIMIT

This view displays information about global resource use for some of the system resources. Use this view to monitor the consumption of resources so that you can take corrective action, if necessary. Many of the resources correspond to initialization parameters listed in Table 3–2.

Some resources, those used by DLM for example, have an initial allocation (soft limit), and the hard limit, which is theoretically infinite (although in practice it is limited by SGA size). During SGA reservation/initialization, a place is reserved in SGA for the INITIAL_ALLOCATION of resources, but if this allocation is exceeded, additional resources are allocated up to the value indicated by LIMIT_VALUE. The CURRENT_UTILIZATION column indicates whether the initial allocation has been exceeded. When the initial allocation value is exceeded, the additional required

resources are allocated from the shared pool, where they must compete for space with other resources.

A good choice for the value of INITIAL_ALLOCATION will avoid the contention for space. For most resources, the value for INITIAL_ALLOCATION is the same as the LIMIT_VALUE. Exceeding LIMIT_VALUE results in an error.

Column	Datatype	Description
RESOURCE_NAME	VARCHAR2(30)	Name of the resource (see Table 3–2)
CURRENT_UTILIZATION	NUMBER	Number of (resources, locks, or processes) currently being used
MAX_UTILIZATION	NUMBER	Maximum consumption of this resource since the last instance start-up
INITIAL_ALLOCATION	VARCHAR2(10)	Initial allocation. This will be equal to the value specified for the resource in the initialization parameter file. (UNLIMITED for infinite allocation)
LIMIT_VALUE	VARCHAR2(10)	Unlimited for resources and locks. This can be greater than the initial allocation value. (UNLIMITED for infinite limit)

Table 3–2 Values for RESOURCE_NAME column

Resource Name	Corresponds to this Initialization Parameter
DISTRIBUTED_TRANSACTIONS	See "DISTRIBUTED_TRANSACTIONS" on page 1-35.
DML_LOCKS	See "DML_LOCKS" on page 1-36.
ENQUEUE_LOCKS	This value is computed by Oracle. Use the V$ENQUEUE_LOCK view (described on on page 3-42) to obtain more information about the enqueue locks.
ENQUEUE_RESOURCES	See "ENQUEUE_RESOURCES" on page 1-37.
LM_PROCESSES	Lock manager processes
LM_RESOURCES	See "LM_RESS" on page 1-53.
LM_LOCKS	See "LM_LOCKS" on page 1-52.
MTS_MAX_SERVERS	See "MTS_MAX_SERVERS" on page 1-72.
PARALLEL_SLAVES	See "PARALLEL_MAX_SERVERS" on page 1-94.
PROCESSES	See "PROCESSES" on page 1-99.
ROLLBACK_SEGMENTS	See "MAX_ROLLBACK_SEGMENTS" on page 1-68.
SESSIONS	See "SESSIONS" on page 1-109
SORT_SEGMENT_LOCKS	This value is computed by Oracle
TEMPORARY_LOCKS	This value is computed by Oracle
TRANSACTIONS	See "TRANSACTIONS" on page 1-119

V$ROLLNAME

This view lists the names of all online rollback segments. It can only be accessed when the database is open.

Column	Datatype	Description
USN	NUMBER	Rollback (undo) segment number
NAME	VARCHAR2	Rollback segment name

V$ROLLSTAT

This view contains rollback segment statistics.

Column	Datatype	Description
USN	NUMBER	Rollback segment number
EXTENTS	NUMBER	Number of extents in rollback segment
RSSIZE	NUMBER	Size in bytes of rollback segment. This values differs by the number of bytes in one database block from the value of the BYTES column of the ALL / DBA / USER_SEGMENTS views. **See Also:** *Oracle8i Administrator's Guide.*
WRITES	NUMBER	Number of bytes written to rollback segment
XACTS	NUMBER	Number of active transactions
GETS	NUMBER	Number of header gets
WAITS	NUMBER	Number of header waits
OPTSIZE	NUMBER	Optimal size of rollback segment
HWMSIZE	NUMBER	High water mark of rollback segment size
SHRINKS	NUMBER	Number of times the size of a rollback segment decreases
WRAPS	NUMBER	Number of times rollback segment is wrapped
EXTENDS	NUMBER	Number of times rollback segment size is extended
AVESHRINK	NUMBER	Average shrink size
AVEACTIVE	NUMBER	Current size of active extents, averaged over time.
STATUS	VARCHAR2(15)	Rollback segment status
CUREXT	NUMBER	Current extent
CURBLK	NUMBER	Current block

V$ROWCACHE

This view displays statistics for data dictionary activity. Each row contains statistics for one data dictionary cache.

Column	Datatype	Description
CACHE#	NUMBER	Row cache ID number
TYPE	VARCHAR2	Parent or subordinate row cache type
SUBORDINATE#	NUMBER	Subordinate set number
PARAMETER	VARCHAR2	Name of the initialization parameter that determines the number of entries in the data dictionary cache
COUNT	NUMBER	Total number of entries in the cache
USAGE	NUMBER	Number of cache entries that contain valid data
FIXED	NUMBER	Number of fixed entries in the cache
GETS	NUMBER	Total number of requests for information on the data object
GETMISSES	NUMBER	Number of data requests resulting in cache misses
SCANS	NUMBER	Number of scan requests
SCANMISSES	NUMBER	Number of times a scan failed to find the data in the cache
SCANCOMPLETES	NUMBER	For a list of subordinate entries, the number of times the list was scanned completely
MODIFICATIONS	NUMBER	Number of inserts, updates, and deletions
FLUSHES	NUMBER	Number of times flushed to disk
DLM_REQUESTS	NUMBER	The number of DLM requests
DLM_CONFLICTS	NUMBER	The number of DLM conflicts
DLM_RELEASES	NUMBER	The number of DLM releases

V$ROWCACHE_PARENT

This view displays information for parent objects in the data dictionary. There is one row per lock owner, and one waiter for each object. This row shows the mode held or requested. For objects with no owners or waiters, a single row is displayed.

Column	Datatype	Description
INDX	NUMBER	Index of the row
HASH	NUMBER	The hash value
ADDRESS	RAW(4)	The address of the parent object

Column	Datatype	Description
CACHE#	NUMBER	The parent cache ID
CACHE_NAME	VARCHAR2(64)	The parent cache name
EXISTENT	VARCHAR2(1)	Whether the object is an existing object
LOCK_MODE	NUMBER	The mode the lock is held in
LOCK_REQUEST	NUMBER	The mode the lock is requested in
TXN	RAW(4)	The transaction currently locking the object
SADDR	RAW(4)	The address of the session
INST_LOCK_REQUEST	NUMBER	Relevant only for Parallel Server. The mode instance lock is being requested in
INST_LOCK_RELEASE	NUMBER	Relevant only for Parallel Server. Whether the instance lock needs to be released
INST_LOCK_TYPE	VARCHAR2(2)	Relevant only for Parallel Server. The type of instance lock
INST_LOCK_ID1	RAW(4)	Relevant only for Parallel Server. The ID associated with the instance lock
INST_LOCK_ID2	RAW(4)	Relevant only for Parallel Server. The ID associated with the instance lock
KEY	RAW(100)	Relevant only for Parallel Server. The contents of the key

V$ROWCACHE_SUBORDINATE

This view displays information for subordinate objects in the data dictionary.

Column	Datatype	Description
INDX	NUMBER	The index
HASH	NUMBER	The hash value
ADDRESS	RAW(4)	The address of the subordinate object
CACHE#	NUMBER	The parent cache ID
SUBCACHE#	NUMBER	The subcache ID
SUBCACHE_NAME	VARCHAR2(64)	The subcache name
EXISTENT	VARCHAR2(1)	Whether the object is an existing object
PARENT	RAW(4)	The address of the parent object
KEY	RAW(100)	The contents of the key

V$RSRC_CONSUMER_GROUP

This view displays data related to currently active resource consumer groups.

See Also:

- *Oracle8i Administrator's Guide* for information on resource groups.
- *Oracle8i Supplied PL/SQL Packages Reference* for information on creating resource groups with the DBMS_RESOURCE_MANAGER package.

Column	Datatype	Description
NAME	VARCHAR2(32)	Name of the consumer group
ACTIVE_SESSIONS	NUMBER	Number of currently active sessions in this consumer group
EXECUTION_WAITERS	NUMBER	Number of currently active sessions waiting for an execution time slice in which they will be able to use CPU.
REQUESTS	NUMBER	Cumulative number of requests that were executed in this consumer group
CPU_WAIT_TIME	NUMBER	Cumulative amount of time that sessions waited for CPU
CPU_WAITS	NUMBER	Cumulative number of times all sessions in this consumer group had to wait for CPU
CONSUMED_CPU_TIME	NUMBER	Cumulative amount of CPU time consumed by all sessions in this consumer group
YIELDS	NUMBER	Cumulative number of times sessions in this consumer group had to yield the CPU
SESSIONS_QUEUED	NUMBER	Reserved for future use.

V$RSRC_CONSUMER_GROUP_CPU_MTH

This view lists all resource allocation methods defined for resource consumer groups.

See Also:

- *Oracle8i Administrator's Guide* for information on resource allocation methods.
- *Oracle8i Supplied PL/SQL Packages Reference* on defining resource allocation methods for consumer groups with the DBMS_RESOURCE_MANAGER package.
- "V$RSRC_PLAN_CPU_MTH" on page 3-89 for a listing of all resource allocation methods defined for resource plans.

Column	Datatype	Description
NAME	VARCHAR2(40)	The name of the CPU resource allocation method

V$RSRC_PLAN

This view displays the names of all currently active resource plans.

See Also:

- "DBA_RSRC_PLANS" on page 2-129 for a listing of all plans in the database.
- *Oracle8i Administrator's Guide* for information on resource plane.
- *Oracle8i Supplied PL/SQL Packages Reference* on defining resource allocation methods for consumer groups with the DBMS_RESOURCE_MANAGER package.

Column	Datatype	Description
NAME	VARCHAR2(32)	Name of the resource plan

V$RSRC_PLAN_CPU_MTH

This view lists all available CPU resource allocation methods defined for resource plans.

See Also:

- "V$RSRC_CONSUMER_GROUP_CPU_MTH" on page 3-88 for a listing of resource allocation methods defined for consumer groups.
- *Oracle8i Administrator's Guide* for information on resource plane.
- *Oracle8i Supplied PL/SQL Packages Reference* on defining resource allocation methods for consumer plans with the DBMS_RESOURCE_MANAGER package.

Column	Datatype	Description
NAME	VARCHAR2(32)	Name of the resource allocation method

V$SESSION

This view lists session information for each current session.

Column	Datatype	Description
SADDR	RAW(4)	Session address
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number. Used to identify uniquely a session's objects. Guarantees that session-level commands are applied to the correct session objects if the session ends and another session begins with the same session ID
AUDSID	NUMBER	Auditing session ID
PADDR	RAW(4)	Address of the process that owns this session
USER#	NUMBER	Oracle user identifier
USERNAME	VARCHAR2(30)	Oracle username
COMMAND	NUMBER	Command in progress (last statement parsed); for a list of values, see Table 3–3. These values also appear in the data dictionary view AUDIT_ACTIONS.
OWNERID	NUMBER	The column contents are invalid if the value is 2147483644. Otherwise, this column contains the identifier of the user who owns the migratable session For operations using Parallel Slaves, interpret this value as a 4Byte value. The low-order 2Bytes of which represent the session number, and the high-order bytes the instance ID of the query coordinator
TADDR	VARCHAR2(8)	Address of transaction state object
LOCKWAIT	VARCHAR2(8)	Address of lock waiting for; NULL if none
STATUS	VARCHAR2(8)	Status of the session: ACTIVE (currently executing SQL), INACTIVE, KILLED (marked to be killed), CACHED (temporarily cached for use by Oracle*XA), SNIPED (session inactive, waiting on the client)
SERVER	VARCHAR2(9)	Server type: DEDICATED, SHARED, PSEUDO, NONE
SCHEMA#	NUMBER	Schema user identifier
SCHEMANAME	VARCHAR2(30)	Schema user name
OSUSER	VARCHAR2(15)	Operating system client user name
PROCESS	VARCHAR2(9)	Operating system client process ID
MACHINE	VARCHAR2(64)	Operating system machine name
TERMINAL	VARCHAR2(10)	Operating system terminal name

Column	Datatype	Description
PROGRAM	VARCHAR2(48)	Operating system program name
TYPE	VARCHAR2(10)	Session type
PREV_SQL_ADDR	RAW(4)	Used with PREV_HASH_VALUE to identify the last SQL statement executed
PREV_HASH_VALUE	NUMBER	Used with SQL_HASH_VALUE to identify the last SQL statement executed.
SQL_ADDRESS	RAW(4)	Used with SQL_HASH_VALUE to identify the SQL statement that is currently being executed
SQL_HASH_VALUE	NUMBER	Used with SQL_ADDRESS to identify the SQL statement that is currently being executed
MODULE	VARCHAR2(48)	Contains the name of the currently executing module as set by calling the DBMS_APPLICATION_INFO.SET_MODULE procedure
MODULE_HASH	NUMBER	The hash value of the above MODULE
ACTION	VARCHAR2(32)	Contains the name of the currently executing action as set by calling the DBMS_APPLICATION_INFO.SET_ACTION procedure
ACTION_HASH	NUMBER	The hash value of the above action name
CLIENT_INFO	VARCHAR2(64)	Information set by the DBMS_APPLICATION_INFO.SET_CLIENT_INFO procedure
FIXED_TABLE_SEQUENCE	NUMBER	This contains a number that increases every time the session completes a call to the database and there has been an intervening select from a dynamic performance table. This column can be used by performance monitors to monitor statistics in the database. Each time the performance monitor looks at the database, it only needs to look at sessions that are currently active or have a higher value in this column than the highest value that the performance monitor saw the last time. All the other sessions have been idle since the last time the performance monitor looked at the database
ROW_WAIT_OBJ#	NUMBER	Object ID for the table containing the ROWID specified in ROW_WAIT_ROW#
ROW_WAIT_FILE#	NUMBER	Identifier for the datafile containing the ROWID specified in ROW_WAIT_ROW#. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1
ROW_WAIT_BLOCK#	NUMBER	Identifier for the block containing the ROWID specified in ROW_WAIT_ROW#. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1
ROW_WAIT_ROW#	NUMBER	The current ROWID being locked. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1
LOGON_TIME	DATE	Time of logon
LAST_CALL_ET	NUMBER	The last call
PDML_ENABLED	VARCHAR2(3)	This column has been replaced by PDML_STATUS. See below.

Column	Datatype	Description
FAILOVER_TYPE	VARCHAR2(13)	Indicates whether and to what extent transparent application failover (TAF) is enabled for the session: ■ NONE: failover is disabled for this session ■ SESSION: the client is able to fail over its session following a disconnect ■ SELECT: the client is able to fail over queries in progress as well **See Also:** ■ *Oracle8i Concepts* for more information on TAF ■ *Net8 Administrator's Guide* for information on configuring TAF.
FAILOVER_METHOD	VARCHAR2(10)	Indicates the transparent application failover method for the session: ■ NONE: failover is disabled for this session ■ BASIC: the client itself reconnects following a disconnect ■ PRECONNECT: the backup instance can support all connections from every instance for which it is backup
FAILED_OVER	VARCHAR2(3)	Indicates (YES or NO) whether the session is running in failover mode and failover has occurred
RESOURCE_ CONSUMER_GROUP	VARCHAR2(32)	Name of the session's current resource consumer group
PDML_STATUS	VARCHAR2(8)	If ENABLED, the session is in a PARALLEL DML enabled mode. If DISABLED, PARALLEL DML enabled mode is not supported for the session. If FORCED, the session has been altered to force PARALLEL DML.
PDDL_STATUS	VARCHAR2(8)	If ENABLED, the session is in a PARALLEL DDL enabled mode. If DISABLED, PARALLEL DDL enabled mode is not supported for the session. If FORCED, the session has been altered to force PARALLEL DDL.
PQ_STATUS	VARCHAR(8)	If ENABLED, the session is in a PARALLEL QUERY enabled mode. If DISABLED, PARALLEL QUERY enabled mode is not supported for the session. If FORCED, the session has been altered to force PARALLEL QUERY.

Table 3–3 Numbers Displayed in COMMAND Column of V$SESSION and Corresponding Commands

1: CREATE TABLE	31: NOAUDIT	61: DROP TRIGGER
2: INSERT	32: CREATE DATABASE LINK	62: ANALYZE TABLE
3: SELECT	33: DROP DATABASE LINK	63: ANALYZE INDEX
4: CREATE CLUSTER	34: CREATE DATABASE	64: ANALYZE CLUSTER
5: ALTER CLUSTER	35: ALTER DATABASE	65: CREATE PROFILE
6: UPDATE	36: CREATE ROLLBACK SEGMENT	66: DROP PROFILE
7: DELETE	37: ALTER ROLLBACK SEGMENT	67: ALTER PROFILE
8: DROP CLUSTER	38: DROP ROLLBACK SEGMENT	68: DROP PROCEDURE
9: CREATE INDEX	39: CREATE TABLESPACE	69: DROP PROCEDURE
10: DROP INDEX	40: ALTER TABLESPACE	70: ALTER RESOURCE COST
11: ALTER INDEX	41: DROP TABLESPACE	71: CREATE SNAPSHOT LOG
12: DROP TABLE	42: ALTER SESSION	72: ALTER SNAPSHOT LOG
13: CREATE SEQUENCE	43: ALTER USE	73: DROP SNAPSHOT LOG
14: ALTER SEQUENCE	44: COMMIT	74: CREATE SNAPSHOT
15: ALTER TABLE	45: ROLLBACK	75: ALTER SNAPSHOT
16: DROP SEQUENCE	46: SAVEPOINT	76: DROP SNAPSHOT
17: GRANT	47: PL/SQL EXECUTE	79: ALTER ROLE
18: REVOKE	48: SET TRANSACTION	85: TRUNCATE TABLE
19: CREATE SYNONYM	49: ALTER SYSTEM SWITCH LOG	86: TRUNCATE COUSTER
20: DROP SYNONYM	50: EXPLAIN	88: ALTER VIEW
21: CREATE VIEW	51: CREATE USER	91: CREATE FUNCTION
22: DROP VIEW	25: CREATE ROLE	92: ALTER FUNCTION
23: VALIDATE INDEX	53: DROP USER	93: DROP FUNCTION
24: CREATE PROCEDURE	54: DROP ROLE	94: CREATE PACKAGE
25: ALTER PROCEDURE	55: SET ROLE	95: ALTER PACKAGE
26: LOCK TABLE	56: CREATE SCHEMA	96: DROP PACKAGE
27: NO OPERATION	57: CREATE CONTROL FILE	97: CREATE PACKAGE BODY
28: RENAME	58: ALTER TRACING	98: ALTER PACKAGE BODY
29: COMMENT	59: CREATE TRIGGER	99: DROP PACKAGE BODY
30: AUDIT	60: ALTER TRIGGER	

V$SESSION_CONNECT_INFO

This view displays information about network connections for the current session.

Column	Datatype	Description
SID	NUMBER	Session identifier (can be used to join this view with V$SESSION)
AUTHENTICATION_TYPE	VARCHAR2(15)	How the user was authenticated: DATABASE: username/password authentication OS: operating system external user authentication NETWORK: network protocol or ANO authentication PROXY: OCI proxy connection authentication
OSUSER	VARCHAR2(30)	The external username for this database user
NETWORK _SERVICE_ BANNER	VARCHAR2(2000)	Product banners for each Net8 service used for this connection (one row per banner)

V$SESSION_CURSOR_CACHE

This view displays information on cursor usage for the current session.

Note: The V$SESSION_CURSOR_CACHE view is not a measure of the effectiveness of the SESSION_CACHED_CURSORS initialization parameter

Column	Datatype	Description
MAXIMUM	NUMBER	Maximum number of cursors to cache. Once you hit this number, some cursors will need to be closed in order to open more. The value in this column is derived from the initialization parameter OPEN_CURSORS
COUNT	NUMBER	The current number of cursors (whether they are in use or not)
OPENED_ONCE	NUMBER	Number of cursors opened at least once
OPEN	NUMBER	Current number of open cursors
OPENS	NUMBER	Cumulative total of cursor opens minus one. This is because the cursor that is currently open and being used for this query is not counted in the OPENS statistic
HITS	NUMBER	Cumulative total of cursor open hits
HIT_RATIO	NUMBER	Ratio of the number of times an open cursor was found divided by the number of times a cursor was sought

V$SESSION_EVENT

This view lists information on waits for an event by a session. Note that the TIME_ WAITED and AVERAGE_WAIT columns will contain a value of zero on those

platforms that do not support a fast timing mechanism. If you are running on one of these platforms and you want this column to reflect true wait times, you must set TIMED_STATISTICS to TRUE in the parameter file. Please remember that doing this will have a small negative effect on system performance.

See Also: "TIMED_STATISTICS" on page 1-118.

Column	Datatype	Description
SID	NUMBER	The ID of the session
EVENT	VARCHAR2(64)	The name of the wait event. **See Also:** Appendix A, "Oracle Wait Events"
TOTAL_WAITS	NUMBER	The total number of waits for this event by this session
TOTAL_TIMEOUTS	NUMBER	The total number of timeouts for this event by this session
TIME_WAITED	NUMBER	The total amount of time waited for this event by this session, in hundredths of a second
AVERAGE_WAIT	NUMBER	The average amount of time waited for this event by this session, in hundredths of a second
MAX_WAIT	NUMBER	The maximum time (in hundredths of a second) waited for this event by this session

V$SESSION_LONGOPS

This view displays the status of various operations that run for longer than 6 seconds (in absolute time). These operations currently include many backup and recovery functions, statistics gathering, and query execution, and more operations are added for every Oracle release.

To monitor query execution progress, you must be using the cost-based optimizer and you must:

- Set the TIMED_STATISTICS or SQL_TRACE parameter to TRUE and
- Gather statistics for your objects with the ANALYZE statement or the DBMS_STATS package.

You can add information to this view about application-specific long-running operations by using the DBMS_APPLICATION_INFO.SET_SESSION_LONGOPS procedure.

See Also: *Oracle8i Supplied PL/SQL Packages Reference* for more information on DBMS_APPLICATION_INFO.SET_SESSION_LONGOPS.

Column	Datatype	Description
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number
OPNAME	VARCHAR2(64)	Brief description of the operation
TARGET	VARCHAR2(64)	The object on which the operation is carried out
TARGET_DESC	VARCHAR2(32)	Description of the target
SOFAR	NUMBER	The units of work done so far
TOTALWORK	NUMBER	The total units of work
UNITS	VARCHAR2(32)	The units of measurement
START_TIME	DATE	The starting time of operation
LAST_UPDATE_TIME	DATE	Time when statistics last updated
TIME_REMAINING	NUMBER	Estimate (in seconds) of time remaining for the operation to complete.
ELAPSED_SECONDS	NUMBER	The number of elapsed seconds from the start of operations
CONTEXT	NUMBER	Context
MESSAGE	VARCHAR2(512)	Statistics summary message
USERNAME	VARCHAR2(30)	User ID of the user performing the operation.
SQL_ADDRESS	RAW(4)	Used with the value of the SQL_HASH_VALUE column to identify the SQL statement associated with the operation.
SQL_HASH_VALUE	NUMER	Used with the value of the SQL_ADDRESS column to identify the SQL statement associated with the operation.
QCSID	NUMBER	Session identifier of the parallel coordinator.

V$SESSION_OBJECT_CACHE

This view displays object cache statistics for the current user session on the local server (instance).

Column	Datatype	Description
PINS	NUMBER	Number of object pins or look-ups in the cache
HITS	NUMBER	Number of object pins that found the object already in the cache

Column	Datatype	Description
TRUE_HITS	NUMBER	Number of object pins that found the object already in the cache and in the desired state (thus, not requiring refresh from the database)
HIT_RATIO	NUMBER	The ratio of HITS / PINS
TRUE_HIT_RATIO	NUMBER	The ratio of TRUE_HITS / PINS
OBJECT_REFRESHES	NUMBER	Number of objects in the cache that were refreshed with a new value from the database
CACHE_REFRESHES	NUMBER	Number of times the whole cache (all objects) were refreshed
OBJECT_FLUSHES	NUMBER	Number of objects in the cache that were flushed to the database
CACHE_FLUSHES	NUMBER	Number of times the whole cache (all objects) were flushed to the database
CACHE_SHRINKS	NUMBER	Number of times the cache was shrunk to the optimal size
CACHED_OBJECTS	NUMBER	Number of objects currently cached
PINNED_OBJECTS	NUMBER	Number of objects currently pinned
CACHE_SIZE	NUMBER	Current size of the cache in bytes
OPTIMAL_SIZE	NUMBER	Optimal size of the cache in bytes
MAXIMUM_SIZE	NUMBER	Maximum size of the cache in bytes

V$SESSION_WAIT

This view lists the resources or events for which active sessions are waiting.

The following are tuning considerations:

- P1RAW, P2RAW, and P3RAW display the same values as the P1, P2, and P3 columns, except that the numbers are displayed in hexadecimal.
- The WAIT_TIME column contains a value of -2 on platforms that do not support a fast timing mechanism. If you are running on one of these platforms and you want this column to reflect true wait times, you must set the TIMED_STATISTICS parameter to TRUE. Remember that doing this has a small negative effect on system performance.

  In previous releases the WAIT_TIME column contained an arbitrarily large value instead of a negative value to indicate the platform did not have a fast timing mechanism.
- The STATE column interprets the value of WAIT_TIME and describes the state of the current or most recent wait.

See Also: "TIMED_STATISTICS" on page 1-118 and Appendix A, "Oracle Wait Events".

Column	Datatype	Description
SID	NUMBER	Session identifier
SEQ#	NUMBER	Sequence number that uniquely identifies this wait. Incremented for each wait.
EVENT	VARCHAR2(64)	Resource or event for which the session is waiting. **See Also:** Appendix A, "Oracle Wait Events"
P1TEXT	VARCHAR2	Description of first additional parameter
P1	NUMBER	First additional parameter
P1RAW	RAW(4)	First additional parameter
P2TEXT	VARCHAR2	Description of second parameter
P2	NUMBER	Second additional parameter
P2RAW	RAW(4)	Second additional parameter
P3TEXT	VARCHAR2	Description of third parameter
P3	NUMBER	Third additional parameter
P3RAW	RAW(4)	Third additional parameter
WAIT_TIME	NUMBER	A non-zero value is the session's last wait time. A zero value means the session is currently waiting
SECONDS_IN _ WAIT	NUMBER	The seconds in wait
STATE	VARCHAR2	Wait state: ■ 0: WAITING (the session is currently waiting) ■ -2: WAITED UNKNOWN TIME (duration of last wait is unknown) ■ -1: WAITED SHORT TIME (last wait <1/100th of a second) ■ >0: WAITED KNOWN TIME (WAIT_TIME = duration of last wait)

V$SESSTAT

This view lists user session statistics. To find the name of the statistic associated with each statistic number (STATISTIC#), query the V$STATNAME view.

See Also: "V$STATNAME" on page 3-109 and Appendix C, "Statistics Descriptions".

Column	Datatype	Description
SID	NUMBER	Session identifier
STATISTIC#	NUMBER	Statistic number (identifier)
		Note: Statistics numbers are not guaranteed to remain constant from one release to another. Therefore, you should rely on the statistics name rather than its number in your applications.
VALUE	NUMBER	Statistic value

V$SESS_IO

This view lists I/O statistics for each user session.

Column	Datatype	Description
SID	NUMBER	Session identifier
BLOCK_GETS	NUMBER	Block gets for this session
CONSISTENT_GETS	NUMBER	Consistent gets for this session
PHYSICAL_READS	NUMBER	Physical reads for this session
BLOCK_CHANGES	NUMBER	Block changes for this session
CONSISTENT_CHANGES	NUMBER	Consistent changes for this session

V$SGA

This view contains summary information on the system global area (SGA).

Column	Datatype	Description
NAME	VARCHAR2	SGA component group
VALUE	NUMBER	Memory size in bytes

V$SGASTAT

This view contains detailed information on the system global area (SGA).

Column	Datatype	Description
NAME	VARCHAR2	SGA component name
BYTES	NUMBER	Memory size in bytes

Column	Datatype	Description
POOL	VARCHAR2	Designates the pool in which the memory in NAME resides: ■ LARGE POOL: Memory is allocated from the large pool ■ SHARED POOL: Memory is allocated from the shared pool

V$SHARED_POOL_RESERVED

This fixed view lists statistics that help you tune the reserved pool and space within the shared pool.

Column	Datatype	Description
The following columns of V$SHARED_POOL_RESERVED are valid only if the initialization parameter SHARED_POOL_RESERVED_SIZE is set to a valid value. **See Also:** "SHARED_POOL_RESERVED_SIZE" on page 1-111.		
FREE_SPACE	NUMBER	Total amount of free space on the reserved list
AVG_FREE_SIZE	NUMBER	Average size of the free memory on the reserved list
FREE_COUNT	NUMBER	Number of free pieces of memory on the reserved list
MAX_FREE_SIZE	NUMBER	Size of the largest free piece of memory on the reserved list
USED_SPACE	NUMBER	Total amount of used memory on the reserved list
AVG_USED_SIZE	NUMBER	Average size of the used memory on the reserved list
USED_COUNT	NUMBER	Number of used pieces of memory on the reserved list
MAX_USED_SIZE	NUMBER	Size of the largest used piece of memory on the reserved list
REQUESTS	NUMBER	Number of times that the reserved list was searched for a free piece of memory
REQUEST_MISSES	NUMBER	Number of times the reserved list did not have a free piece of memory to satisfy the request, and started flushing objects from the LRU list
LAST_MISS_SIZE	NUMBER	Request size of the last request miss, when the reserved list did not have a free piece of memory to satisfy the request and started flushing objects from the LRU list
MAX_MISS_SIZE	NUMBER	Request size of the largest request miss, when the reserved list did not have a free piece of memory to satisfy the request and started flushing objects from the LRU list
The following columns of V$SHARED_POOL_RESERVED contain values which are valid even if SHARED_POOL_RESERVED_SIZE is not set.		
REQUEST_FAILURES	NUMBER	Number of times that no memory was found to satisfy a request (that is, the number of times the error ORA-4031 occurred)
LAST_FAILURE_SIZE	NUMBER	Request size of the last failed request (that is, the request size for the last ORA-4031 error)

Column	Datatype	Description
ABORTED_REQUEST _ THRESHOLD	NUMBER	Minimum size of a request which signals an ORA-4031 error without flushing objects
ABORTED_REQUESTS	NUMBER	Number of requests that signalled an ORA-4031 error without flushing objects
LAST_ABORTED_SIZE	NUMBER	Last size of the request that returned an ORA-4031 error without flushing objects from the LRU list

V$SHARED_SERVER

This view contains information on the shared server processes.

Column	Datatype	Description
NAME	VARCHAR2	Name of the server
PADDR	RAW(4)	Server's process address
STATUS	VARCHAR2	Server status: EXEC (executing SQL) WAIT (ENQ) (waiting for a lock), WAIT (SEND) (waiting to send data to user) WAIT (COMMON) (idle; waiting for a user request) WAIT (RESET) (waiting for a circuit to reset after a break) QUIT (terminating)
MESSAGES	NUMBER	Number of messages processed
BYTES	NUMBER	Total number of bytes in all messages
BREAKS	NUMBER	Number of breaks
CIRCUIT	RAW(4)	Address of circuit currently being serviced
IDLE	NUMBER	Total idle time in hundredths of a second
BUSY	NUMBER	Total busy time in hundredths of a second
REQUESTS	NUMBER	Total number of requests taken from the common queue in this server's lifetime

V$SORT_SEGMENT

This view contains information about every sort segment in a given instance. The view is only updated when the tablespace is of the TEMPORARY type.

Column	Datatype	Description
TABLESPACE_NAME	VARCHAR2(31)	Name of tablespace

Column	Datatype	Description
SEGMENT_FILE	NUMBER	File number of the first extent
SEGMENT_BLOCK	NUMBER	Block number of the first extent
EXTENT_SIZE	NUMBER	Extent size
CURRENT_USERS	NUMBER	Number of active users of the segment
TOTAL_EXTENTS	NUMBER	Total number of extents in the segment
TOTAL_BLOCKS	NUMBER	Total number of blocks in the segment
RELATIVE_FNO	NUMBER	Relative file number of the sort segment header
USED_EXTENTS	NUMBER	Extents allocated to active sorts
USED_BLOCKS	NUMBER	Blocks allocated to active sorts
FREE_EXTENTS	NUMBER	Extents not allocated to any sort
FREE_BLOCKS	NUMBER	Blocks not allocated to any sort
ADDED_EXTENTS	NUMBER	Number of extent allocations
EXTENT_HITS	NUMBER	Number of times an unused extent was found in the pool
FREED_EXTENTS	NUMBER	Number of deallocated extents
FREE_REQUESTS	NUMBER	Number of requests to deallocate
MAX_SIZE	NUMBER	Maximum number of extents ever used
MAX_BLOCKS	NUMBER	Maximum number of blocks ever used
MAX_USED_SIZE	NUMBER	Maximum number of extents used by all sorts
MAX_USED_BLOCKS	NUMBER	Maximum number of blocks used by all sorts
MAX_SORT_SIZE	NUMBER	Maximum number of extents used by an individual sort
MAX_SORT_BLOCKS	NUMBER	Maximum number of blocks used by an individual sort

V$SORT_USAGE

This view describes sort usage.

Column	Datatype	Description
USER	VARCHAR2(30)	User who requested temporary space
SESSION_ADDR	RAW(4)	Address of shared SQL cursor
SESSION_NUM	NUMBER	Serial number of session
SQLADDR	RAW(4)	Address of SQL statement
SQLHASH	NUMBER	Hash value of SQL statement

Column	Datatype	Description
TABLESPACE	VARCHAR2(31)	Tablespace in which space is allocated
CONTENTS	VARCHAR2(9)	Indicates whether tablespace is TEMPORARY / PERMANENT
SEGFILE#	NUMBER	File number of initial extent
SEGBLK#	NUMBER	Block number of the initial extent
EXTENTS	NUMBER	Extents allocated to the sort
BLOCKS	NUMBER	Extents in blocks allocated to the sort
SEGRFNO#	NUMBER	Relative file number of initial extent

V$SQL

This view lists statistics on shared SQL area without the GROUP BY clause and contains one row for each child of the original SQL text entered.

Column	Datatype	Description
SQL_TEXT	VARCHAR2(1000)	The first eighty characters of the SQL text for the current cursor
SHARABLE_MEM	NUMBER	Amount of shared memory, in bytes used by this child cursor
PERSISTENT_MEM	NUMBER	Fixed amount of memory, in bytes, used for the lifetime of this child cursor
RUNTIME_MEM	NUMBER	Fixed amount of memory required during the execution of this child cursor
SORTS	NUMBER	The number of sorts that was done for this child cursor
LOADED_VERSIONS	NUMBER	1 if context heap is loaded, 0 otherwise
OPEN_VERSIONS	NUMBER	1 if the child cursor is locked, 0 otherwise
USERS_OPENING	NUMBER	The number of users executing the statement
EXECUTIONS	NUMBER	The number of executions that took place on this object since it was brought into the library cache
USERS_EXECUTING	NUMBER	The number of users executing the statement
LOADS	NUMBER	The number of times the object was loaded or reloaded
FIRST_LOAD_TIME	VARCHAR2(19)	The time stamp of the parent creation time
INVALIDATIONS	NUMBER	The number of times this child cursor has been invalidated
PARSE_CALLS	NUMBER	The number of parse calls for this child cursor
DISK_READS	NUMBER	The number of disk reads for this child cursor
BUFFER_GETS	NUMBER	The number of buffer gets for this child cursor
ROWS_PROCESSED	NUMBER	The total number of rows the parsed SQL statement returns

Column	Datatype	Description
COMMAND_TYPE	NUMBER	The Oracle command type definition
OPTIMIZER_MODE	VARCHAR2(10)	Mode under which the SQL statement is executed
OPTIMIZER_COST	NUMBER	The cost of this query given by the optimizer
PARSING_USER_ID	NUMBER	The user ID of the user who originally built this child cursor
PARSING_SCHEMA_ID	NUMBER	The schema ID that was used to originally build this child cursor
KEPT_VERSIONS	NUMBER	Indicates whether this child cursor has been marked to be kept pinned in cache using the DBMS_SHARED_POOL package
ADDRESS	RAW(4)	The address of the handle to the parent for this cursor
TYPE_CHK_HEAP	RAW(4)	The descriptor of the type check heap for this child cursor
HASH_VALUE	NUMBER	The hash value of the parent statement in the library cache
CHILD_NUMBER	NUMBER	The number of this child cursor
MODULE	VARCHAR2(64)	Contains the name of the module that was executing at the time that the SQL statement was first parsed as set by calling DBMS_ APPLICATION_INFO.SET_MODULE
MODULE_HASH	NUMBER	The hash value of the module that is named in the MODULE column
ACTION	VARCHAR2(64)	Contains the name of the action that was executing at the time that the SQL statement was first parsed as set by calling DBMS_ APPLICATION_INFO.SET_ACTION
ACTION_HASH	NUMBER	The hash value of the action that is named in the ACTION column
SERIALIZABLE_ABORTS	NUMBER	The number of times the transaction fails to serialize, producing ORA-8177 errors, per cursor
OUTLINE_CATEGORY	VARCHAR2(64)	If an outline was applied during construction of the cursor, this column displays the category of that outline. Otherwise the column is left blank.

V$SQL_BIND_DATA

For each distinct bind variable in each cursor owned by the session querying this view, this view describes:

- Actual bind data, if the bind variable is user defined
- The underlying literal, if the CURSOR_SHARING parameter is set to FORCE and the bind variable is system generated. (System-generated binds have a value in the SHARED_FLAG2 column of 256.)

Column	Datatype	Description
CURSOR_NUM	NUMBER	Cursor number for this bind

Column	Datatype	Description
POSITION	NUMBER	Bind position
DATATYPE	NUMBER	Bind datatype
SHARED_MAX_LEN	NUMBER	Shared maximum length for this bind from the shared cursor object associated with this bind
PRIVATE_MAX_LEN	NUMBER	Private maximum length for this bind sent from the client
ARRAY_SIZE	NUMBER	Maximum number of array elements (for array binds only)
PRECISION	NUMBER	Precision (for numeric binds)
SCALE	NUMBER	Scale (for numeric binds)
SHARED_FLAG	NUMBER	Shared bind data flags
SHARED_FLAG2	NUMBER	Shared bind data flags (continued)
BUF_ADDRESS	RAW(4)	Bind buffer memory address
BUF_LENGTH	NUMBER	Bind buffer length
VAL_LENGTH	NUMBER	Actual bind value length
BUF_FLAG	NUMBER	Bind buffer flags
INDICATOR	NUMBER	Bind indicator
VALUE	VARCHAR2(4000)	Contents of the bind buffer

V$SQL_BIND_METADATA

For each distinct bind variable in each cursor owned by the session querying this view, this view describes:

- Bind metadata provided by the client, if the bind variable is user defined
- Metadata based on the underlying literal, if the CURSOR_SHARING parameter is set to FORCE and the bind variable is system-generated.

Column	Datatype	Description
ADDRESS	RAW(4)	Memory address of the child cursor that owns this bind variable
POSITION	NUMBER	Bind position
DATATYPE	NUMBER	Bind datatype
MAX_LENGTH	NUMBER	Maximum length of the bind value
ARRAY_LEN	NUMBER	Maximum number of array elements (for array binds only)
BIND_NAME	VARCHAR2(30)	User-defined or system-generated bind variable name (if used)

V$SQL_CURSOR

This view displays debugging information for each cursor associated with the session querying this view.

Column	Datatype	Description
CURNO	NUMBER	Cursor number
FLAG	NUMBER	Flags set in the cursor
STATUS	VARCHAR2(9)	Status of the cursor; that is, what state the cursor is in
PARENT_HANDLE	RAW(4)	Pointer to the parent cursor handle
PARENT_LOCK	RAW(4)	Pointer to the parent cursor lock
CHILD_LOCK	RAW(4)	Pointer to the child cursor lock
CHILD_PIN	RAW(4)	Pointer to the child cursor pin
PERS_HEAP_MEM	NUMBER	Total amount of memory allocated from persistent heap for this cursor
WORK_HEAP_MEM	NUMBER	Total amount of memory allocated from the work heap for this cursor
BIND_VARS	NUMBER	Total number of bind positions in the query currently parsed into this cursor
DEFINE_VARS	NUMBER	Total number of define variables in the query currently parsed into this cursor
BIND_MEM_LOC	VARCHAR2(64)	Which memory heap the bind variables are stored in: either the UGA or the CGA
INST_FLAG	VARCHAR2(64)	Instantiation object flags
INST_FLAG2	VARCHAR2(64)	Instantiation object flags (continued)

V$SQL_SHARED_MEMORY

This view displays information about the cursor shared memory snapshot. Each SQL statement stored in the shared pool has one or more child objects associated with it. Each child object has a number of parts, one of which is the context heap, which holds, among other things, the query plan.

Column	Datatype	Description
SQL_TEXT	VARCHAR2(1000)	The SQL text of the shared cursor child object that this row is displaying information for
HASH_VALUE	NUMBER	The hash value of the above SQL text in the shared pool
HEAP_DESC	RAW(4)	The address of the descriptor for the context heap of the child cursor described in this row

Column	Datatype	Description
STRUCTURE	VARCHAR2(16)	If the memory chunk described in this row was allocated using a comment of the form "X : Y", then this is the "X" part of the comment
FUNCTION	VARCHAR2(16)	Similar to the STRUCTURE column, this is the "Y" field of the comment
COMMENT	VARCHAR2(16)	This is the whole comment field that was supplied when this memory chunk was allocated
CHUNK_PTR	RAW(4)	This is the starting address of the allocated memory chunk
CHUNK_SIZE	NUMBER	The amount of memory allocated for this chunk
ALLOC_CLASS	VARCHAR2(8)	Class of memory that this chunk of memory belongs to. It will usually be either FREEABLE or PERMANENT
CHUNK_TYPE	NUMBER	An index into a table of callback functions that tell the server how to recreate this chunk of memory should it need to be removed from the shared pool based on an LRU algorithm
SUBHEAP_DESC	RAW(4)	If the parent heap of this context heap is itself a subheap, then this is the address of the descriptor of the parent heap

V$SQLAREA

This view lists statistics on shared SQL area and contains one row per SQL string. It provides statistics on SQL statements that are in memory, parsed, and ready for execution.

Column	Datatype	Description
SQL_TEXT	VARCHAR2(1000)	The first eighty characters of the SQL text for the current cursor
SHARABLE_MEM	NUMBER	Amount of shared memory used by a cursor. If multiple child cursors exist, the sum of all shared memory used by all child cursors.
PERSISTENT_MEM	NUMBER	Fixed amount of memory used for the lifetime of an open cursor. If multiple child cursors exist, the fixed sum of memory used for the lifetime of all the child cursors.
RUNTIME_MEM	NUMBER	Fixed amount of memory required during execution of a cursor. If multiple child cursors exist, the fixed sum of all memory required during execution of all the child cursors.
SORTS	NUMBER	The sum of the number of sorts that was done for all the children
VERSION_COUNT	NUMBER	The number of children that are present in the cache under this parent
LOADED_VERSIONS	NUMBER	The number of children that are present in the cache AND have their context heap (KGL heap 6) loaded
OPEN_VERSIONS	NUMBER	The number of child cursors that are currently open under this current parent
USERS_OPENING	NUMBER	The number of users that have any of the child cursors open

Column	Datatype	Description
EXECUTIONS	NUMBER	The total number of executions, totalled over all the children
USERS_EXECUTING	NUMBER	The total number of users executing the statement over all children
LOADS	NUMBER	The number of times the object was loaded or reloaded
FIRST_LOAD_TIME	VARCHAR2(19)	The time stamp of the parent creation time
INVALIDATIONS	NUMBER	The total number of invalidations over all the children
PARSE_CALLS	NUMBER	The sum of all parse calls to all the child cursors under this parent
DISK_READS	NUMBER	The sum of the number of disk reads over all child cursors
BUFFER_GETS	NUMBER	The sum of buffer gets over all child cursors
ROWS_PROCESSED	NUMBER	The total number of rows processed on behalf of this SQL statement
COMMAND_TYPE	NUMBER	The Oracle command type definition
OPTIMIZER_MODE	VARCHAR2(10)	Mode under which the SQL statement is executed
PARSING_USER_ID	NUMBER	The user ID of the user that has parsed the very first cursor under this parent
PARSING_SCHEMA_ID	NUMBER	The schema ID that was used to parse this child cursor
KEPT_VERSIONS	NUMBER	The number of child cursors that have been marked to be kept using the DBMS_SHARED_POOL package
ADDRESS	RAW(4)	The address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	The hash value of the parent statement in the library cache
MODULE	VARCHAR2(64)	Contains the name of the module that was executing at the time that the SQL statement was first parsed as set by calling DBMS_APPLICATION_INFO.SET_MODULE
MODULE_HASH	NUMBER	The hash value of the module that is named in the MODULE column
ACTION	VARCHAR2(64)	Contains the name of the action that was executing at the time that the SQL statement was first parsed as set by calling DBMS_APPLICATION_INFO.SET_ACTION
ACTION_HASH	NUMBER	The hash value of the action that is named in the ACTION column
SERIALIZABLE_ABORTS	NUMBER	The number of times the transaction fails to serialize, producing ORA-8177 errors, totalled over all the children

V$SQLTEXT

This view contains the text of SQL statements belonging to shared SQL cursors in the SGA.

Column	Datatype	Description
ADDRESS	RAW	Used with HASH_VALUE to identify uniquely a cached cursor
HASH_VALUE	NUMBER	Used with ADDRESS to identify uniquely a cached cursor
PIECE	NUMBER	Number used to order the pieces of SQL text
SQL_TEXT	VARCHAR2	A column containing one piece of the SQL text
COMMAND_TYPE	NUMBER	Code for the type of SQL statement (SELECT, INSERT, etc.)

V$SQLTEXT_WITH_NEWLINES

This view is identical to the V$SQLTEXT view except that, to improve legibility, V$SQLTEXT_WITH_NEWLINES does not replace newlines and tabs in the SQL statement with spaces.

See Also: "V$SQLTEXT" on page 3-108.

Column	Datatype	Description
ADDRESS	RAW	Used with HASH_VALUE to identify uniquely a cached cursor
HASH_VALUE	NUMBER	Used with ADDRESS to identify uniquely a cached cursor
PIECE	NUMBER	Number used to order the pieces of SQL text
SQL_TEXT	VARCHAR2	A column containing one piece of the SQL text
COMMAND_TYPE	NUMBER	Code for the type of SQL statement (SELECT, INSERT, etc.)

V$STATNAME

This view displays decoded statistic names for the statistics shown in the V$SESSTAT and V$SYSSTAT tables.

On some platforms, the NAME and CLASS columns contain additional operating system-specific statistics.

See Also:

- "V$SESSTAT" on page 3-98 and "V$SYSSTAT" on page 3-111.
- Appendix C, "Statistics Descriptions" for a description of each statistic.
- Your operating system-specific Oracle documentation.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number. **Note:** Statistics numbers are not guaranteed to remain constant from one release to another. Therefore, you should rely on the statistics name rather than its number in your applications.
NAME	VARCHAR2	Statistic name
CLASS	NUMBER	A number representing one or more statistics class. The following class numbers are additive: ■ 1, User ■ 2, Redo ■ 4, Enqueue ■ 8, Cache ■ 16, OS ■ 32, Parallel Server ■ 64, SQL ■ 128, Debug

V$SUBCACHE

This view displays information about the subordinate caches currently loaded into library cache memory. The view walks through the library cache, printing out a row for each loaded subordinate cache per library cache object.

Column	Datatype	Description
OWNER_NAME	VARCHAR2(64)	Owner of object containing these cache entries
NAME	VARCHAR2(1000)	Object Name
TYPE	NUMBER	Object Type
HEAP_NUM	NUMBER	Heap number containing this subordinate cache
CACHE_ID	NUMBER	Subordinate cache ID
CACHE_CNT	NUMBER	Number of entries for this cache in this object
HEAP_SZ	NUMBER	Amount of extent space allocated to this heap
HEAP_ALOC	NUMBER	Amount of extent space allocated from this heap
HEAP_USED	NUMBER	Amount of space utilized in this heap

V$SYSSTAT

This view lists system statistics. To find the name of the statistic associated with each statistic number (STATISTIC#), query the V$STATNAME view.

See Also: "V$STATNAME" on page 3-109 and Appendix C, "Statistics Descriptions".

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number **Note:** Statistics numbers are not guaranteed to remain constant from one release to another. Therefore, you should rely on the statistics name rather than its number in your applications.
NAME	VARCHAR2(64)	Statistic name
CLASS	NUMBER	A number representing one or more statistics class. The following class numbers are additive: ■ 1, User ■ 2, Redo ■ 4, Enqueue ■ 8, Cache ■ 16, OS ■ 32, Parallel Server ■ 64, SQL ■ 128, Debug
VALUE	NUMBER	Statistic value

V$SYSTEM_CURSOR_CACHE

This view displays similar information to the V$SESSION_CURSOR_CACHE view except that this information is system wide.

See Also: "V$SESSION_CURSOR_CACHE" on page 3-94.

Column	Datatype	Description
OPENS	NUMBER	Cumulative total of cursor opens
HITS	NUMBER	Cumulative total of cursor open hits
HIT_RATIO	NUMBER	Ratio of the number of times you found an open cursor divided by the number of times you looked for a cursor

V$SYSTEM_EVENT

This view contains information on total waits for an event. Note that the TIME_WAITED and AVERAGE_WAIT columns will contain a value of zero on those platforms that do not support a fast timing mechanism. If you are running on one of these platforms and you want this column to reflect true wait times, you must set TIMED_STATISTICS to TRUE in the parameter file. Please remember that doing this will have a small negative effect on system performance.

See Also: "TIMED_STATISTICS" on page 1-118.

Column	Datatype	Description
EVENT	VARCHAR2(64)	The name of the wait event
TOTAL_WAITS	NUMBER	The total number of waits for this event
TOTAL_TIMEOUTS	NUMBER	The total number of timeouts for this event
TIME_WAITED	NUMBER	The total amount of time waited for this event, in hundredths of a second
AVERAGE_WAIT	NUMBER	The average amount of time waited for this event, in hundredths of a second

V$SYSTEM_PARAMETER

This view contains information on system parameters.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(64)	Parameter name
TYPE	NUMBER	Parameter type; 1 = Boolean, 2 = string, 3 = integer
VALUE	VARCHAR2(512)	Value assigned to the parameter
ISDEFAULT	VARCHAR2(9)	Is the value assigned to the parameter the default
ISSES_MODIFIABLE	VARCHAR2(5)	Whether the parameter can be modified by ALTER SESSION
ISSYS_MODIFIABLE	VARCHAR2(9)	Whether the parameter can be modified by ALTER SYSTEM
ISMODIFIED	VARCHAR2(8)	Indicates how the parameter was modified. If an ALTER SESSION was performed, the value will be MODIFIED. If an ALTER SYSTEM (which will cause all the currently logged in sessions' values to be modified) was performed the value will be SYS_MODIFIED
ISADJUSTED	VARCHAR2(5)	Indicates that the rdbms adjusted the input value to a more suitable value (e.g., the parameter value should be prime, but the user input a non-prime number, so the rdbms adjusted the value to the next prime number)

Column	Datatype	Description
DESCRIPTION	VARCHAR2(64)	Descriptive text about the parameter

V$TABLESPACE

This view displays tablespace information from the control file.

Column	Datatype	Description
TS#	NUMBER	Tablespace number
NAME	VARCHAR2 (30)	Tablespace name

V$TEMPFILE

This view displays tempfile information.

Column	Datatype	Description
FILE#	NUMBER	The absolute file number
CREATION_CHANGE#	NUMBER	The creation System Change Number
CREATION_TIME	DATE	The creation time
TS#	NUMBER	The tablespace number
RFILE#	NUMBER	The relative file number in tablespace
STATUS	VARCHAR2(7)	The status of the file (offline/online)
ENABLED	VARCHAR2(10)	Enabled for read and/or write
BYTES	NUMBER	The size of the file in bytes (from File Header)
BLOCKS	NUMBER	The size of the file in blocks (from File Header)
CREATE_BYTES	NUMBER	The creation size of the file (in bytes)
BLOCK_SIZE	NUMBER	The block size for the file
NAME	VARCHAR2(513)	The name of the file

V$TEMPORARY_LOBS

This view displays temporary lobs.

Column	Datatype	Description
SID	NUMBER	Session ID

Column	Datatype	Description
CACHE_LOBS	NUMBER	Number cache temp lobs
NOCACHE_LOBS	NUMBER	Number of nocache temp lobs

V$TEMP_EXTENT_MAP

This view displays the status of each unit for all temporary tablespaces.

Column	Datatype	Description
TABLESPACE_NAME	NUMBER	Name of tablespace this unit belongs to
FILE_ID	NUMBER	Absolute file number
BLOCK_ID	NUMBER	Begin block number for this unit
BYTES	NUMBER	Bytes in extent
BLOCKS	NUMBER	Blocks in extent
OWNER	NUMBER	Which instance own this unit (string)
RELATIVE_FNO	NUMBER	The relative file number

V$TEMP_EXTENT_POOL

This view displays the state of temporary space cached and used for a given instance. Note that loading of the temporary space cache is lazy, and that instances can be dormant. Use GV$TEMP_EXTENT_POOL for information about all instances.

Column	Datatype	Description
TABLESPACE_NAME	VARCHAR2(30)	Name of the tablespace
FILE_ID	NUMBER	Absolute file number
EXTENTS_CACHED	NUMBER	How many extents have been cached
EXTENTS_USED	NUMBER	How many extents are actually being used
BLOCKS_CACHED	NUMBER	How many blocks are cached
BLOCKS_USED	NUMBER	How many blocks are used
BYTES_CACHED	NUMBER	How many bytes are cached
BYTES_USED	NUMBER	How many bytes used
RELATIVE_FNO	NUMBER	The relative file number

V$TEMP_PING

The view V$TEMP_PING displays the number of blocks pinged per datafile. This information in turn can be used to determine access patterns to existing datafiles and deciding new mappings from datafile blocks to PCM locks.

Column	Datatype	Description
FILE_NUMBER	NUMBER	Number of the datafile
FREQUENCY	NUMBER	The frequency
X_2_NULL	NUMBER	Number of lock conversions from Exclusive-to-NULL for all blocks in the file
X_2_NULL_FORCED _ WRITE	NUMBER	Number of forced writes that occur for blocks of the specified file due to Exclusive-to-NULL conversions
X_2_NULL_FORCED _ STALE	NUMBER	Number of times a block in the file was made STALE due to Exclusive-to-NULL conversions
X_2_S	NUMBER	Number of lock conversions from Exclusive-to-Shared for all blocks in the file
X_2_S_FORCED_WRITE	NUMBER	Number of forced writes that occur for blocks of the specified file due to Exclusive-to-Shared conversions
X_2_SSX	NUMBER	Number of lock conversions from Exclusive-to-Sub Shared Exclusive for all blocks in the file
X_2_SSX_FORCED _WRITE	NUMBER	Number of forced writes that occur for blocks of the specified file due to Exclusive-to-Sub Shared Exclusive conversions
S_2_NULL	NUMBER	Number of lock conversions from Shared-to-NULL for all blocks in the file
S_2_NULL_FORCED _ STALE	NUMBER	Number of times a block in the file was made STALE due to Shared-to-NULL conversions
SS_2_NULL	NUMBER	Number of lock conversions from Sub Shared-to-NULL for all blocks in the file
WRB	NUMBER	Number of times the instance received a write single buffer cross instance call for this file
WRB_FORCED_WRITE	NUMBER	Number of blocks written due to write single buffer cross instance calls for this file
RBR	NUMBER	Number of times the instance received a reuse block range cross instance call for this file
RBR_FORCED_WRITE	NUMBER	Number of blocks written due to reuse block range cross instance calls for this file
RBR_FORCED_STALE	NUMBER	Number of times a block in this file was made STALE due to reuse block range cross instance calls
CBR	NUMBER	Number of times the instance received a checkpoint block range cross instance call for this file

Column	Datatype	Description
CBR_FORCED_WRITE	NUMBER	Number of blocks in this file which were written due to checkpoint cross range cross instance calls
NULL_2_X	NUMBER	Number of lock conversions from NULL-to-Exclusive for all blocks of the specified file
S_2_X	NUMBER	Number of lock conversions from Shared-to-Exclusive for all blocks of the specified file
SSX_2_X	NUMBER	Number of lock conversions from Sub Shared Exclusive-to-Exclusive for all blocks of the specified file
NULL_2_S	NUMBER	Number of lock conversions from NULL-to-Shared for all blocks of the specified file
NULL_2_SS	NUMBER	Number of lock conversions from NULL-to-Sub Shared for all blocks of the specified file
OP_2_SS	NUMBER	Number of pcm locks ss locks opened. 0 in Oracle 8.1

V$TEMP_SPACE_HEADER

This view displays aggregate information per file per temporary tablespace regarding how much space is currently being used and how much is free as per the space header.

Column	Datatype	Description
TABLESPACE_NAME	VARCHAR2(30)	The name of the temporary tablespace
FILE_ID	NUMBER	The absolute file number
BYTES_USED	NUMBER	How many bytes are in use
BLOCKS_USED	NUMBER	How many blocks are in use
BYTES_FREE	NUMBER	How many bytes are free
BLOCKS_FREE	NUMBER	How many blocks are free
RELATIVE_FNO	NUMBER	The relative file number for the file

V$TEMPSTAT

This view contains information about file read / write statistics.

Column	Datatype	Description
FILE#	NUMBER	Number of the file
PHYRDS	NUMBER	Number of physical reads done

Column	Datatype	Description
PHYWRTS	NUMBER	Number of times DBWR is required to write
PHYBLKRD	NUMBER	Number of physical blocks read
PHYBLKWRT	NUMBER	Number of blocks written to disk; which may be the same as PHYWRTS if all writes are single blocks
READTIM	NUMBER	Time (in hundredths of a second) spent doing reads if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
WRITETIM	NUMBER	Time (in hundredths of a second) spent doing writes if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
AVGIOTIM	NUMBER	The average time (in hundredths of a second) spent on I/O, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
LSTIOTIM	NUMBER	The time (in hundredths of a second) spent doing the last I/O, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
MINIOTIM	NUMBER	The minimum time (in hundredths of a second) spent on a single I/O, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
MAXIOWTM	NUMBER	The maximum time (in hundredths of a second) spent doing a single write, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE
MAXIORTM	NUMBER	The maximum time (in hundredths of a second) spent doing a single read, if the TIMED_STATISTICS parameter is TRUE; 0 if FALSE

V$THREAD

This view contains thread information from the control file.

Column	Datatype	Description
THREAD#	NUMBER	Thread number
STATUS	VARCHAR2	Thread status: OPEN, CLOSED
ENABLED	VARCHAR2	Enabled status: DISABLED, (enabled) PRIVATE, or (enabled) PUBLIC
ENABLE_CHANGE#	NUMBER	SCN at which thread was enabled
ENABLE_TIME	DATE	Time of enable SCN
DISABLE_CHANGE#	NUMBER	SCN at which thread was disabled
DISABLE_TIME	DATE	Time of disable SCN
GROUPS	NUMBER	Number of log groups assigned to this thread
INSTANCE	VARCHAR2	Instance name, if available
OPEN_TIME	DATE	Last time the thread was opened
CURRENT_GROUP#	NUMBER	Current log group

Column	Datatype	Description
SEQUENCE#	NUMBER	Sequence number of current log
CHECKPOINT_CHANGE#	NUMBER	SCN at last checkpoint
CHECKPOINT_TIME	DATE	Time of last checkpoint

V$TIMER

This view lists the elapsed time in hundredths of seconds. Time is measured since the beginning of the epoch, which is operating system specific, and wraps around to 0 again whenever the value overflows four bytes (roughly 497 days).

Column	Datatype	Description
HSECS	NUMBER	Elapsed time in hundredths of a second

V$TRANSACTION

This view lists the active transactions in the system.

Column	Datatype	Description
ADDR	RAW(4)	Address of transaction state object
XIDUSN	NUMBER	Undo segment number
XIDSLOT	NUMBER	Slot number
XIDSQN	NUMBER	Sequence number
UBAFIL	NUMBER	Undo block address (UBA) filenum
UBABLK	NUMBER	UBA block number
UBASQN	NUMBER	UBA sequence number
UBAREC	NUMBER	UBA record number
STATUS	VARCHAR2(16)	Status
START_TIME	VARCHAR2(20)	Start time (wall clock)
START_SCNB	NUMBER	Start system change number (SCN) base
START_SCNW	NUMBER	Start SCN wrap
START_UEXT	NUMBER	Start extent number
START_UBAFIL	NUMBER	Start UBA file number
START_UBABLK	NUMBER	Start UBA block number

Column	Datatype	Description
START_UBASQN	NUMBER	Start UBA sequence number
START_UBAREC	NUMBER	Start UBA record number
SES_ADDR	RAW(4)	User session object address
FLAG	NUMBER	Flag
SPACE	VARCHAR2(3)	"Yes", if a space transaction
RECURSIVE	VARCHAR2(3)	"Yes", if a recursive transaction
NOUNDO	VARCHAR2(3)	"Yes" if a no undo transaction
PTX	VARCHAR 2(3)	YES if parallel transaction, otherwise set to NO
PRV_XIDUSN	NUMBER	Previous transaction undo segment number
PRV_XIDSLT	NUMBER	Previous transaction slot number
PRV_XIDSQN	NUMBER	Previous transaction sequence number
PTX_XIDUSN	NUMBER	Rollback segment number of the parent XID
PTX_XIDSLT	NUMBER	Slot number of the parent XID
PTX_XIDSQN	NUMBER	Sequence number of the parent XID
DSCN_B	NUMBER	Dependent SCN base
DSCN_W	NUMBER	Dependent SCN wrap
USED_UBLK	NUMBER	Number of undo blocks used
USED_UREC	NUMBER	Number of undo records used
LOG_IO	NUMBER	Logical I/O
PHY_IO	NUMBER	Physical I/O
CR_GET	NUMBER	Consistent gets
CR_CHANGE	NUMBER	Consistent changes

V$TRANSACTION_ENQUEUE

V$TRANSACTION_ENQUEUE displays locks owned by transaction state objects.

Column	Datatype	Description
ADDR	RAW(4)	Address of lock state object
KADDR	RAW(4)	Address of lock
SID	NUMBER	Identifier for session holding or acquiring the lock
TYPE	VARCHAR2(2)	Type of lock. TX = transaction enqueue.

Column	Datatype	Description
ID1	NUMBER	Lock identifier #1 (depends on type)
ID2	NUMBER	Lock identifier #2 (depends on type)
LMODE	NUMBER	Lock mode in which the session holds the lock: ■ 0, None ■ 1, Null (NULL) ■ 2, Row-S (SS) ■ 3, Row-X (SX) ■ 4, Share (S) ■ 5, S/Row-X (SSX) ■ 6, Exclusive (X)
REQUEST	NUMBER	Lock mode in which the process requests the lock: ■ 0, None ■ 1, Null (NULL) ■ 2, Row-S (SS) ■ 3, Row-X (SX) ■ 4, Share (S) ■ 5, S/Row-X (SSX) ■ 6, Exclusive (X)
CTIME	NUMBER	Time since current mode was granted
BLOCK	NUMBER	The lock is blocking another lock

V$TYPE_SIZE

This view lists the sizes of various database components for use in estimating data block capacity.

Column	Datatype	Description
COMPONENT	VARCHAR2	Component name, such as segment or buffer header
TYPE	VARCHAR2	Component type
DESCRIPTION	VARCHAR2	Description of component
TYPE_SIZE	NUMBER	Size of component

V$VERSION

Version numbers of core library components in the Oracle server. There is one row for each component.

Column	Datatype	Description
BANNER	VARCHAR2	Component name and version number

V$WAITSTAT

This view lists block contention statistics. This table is only updated when timed statistics are enabled.

Column	Datatype	Description
CLASS	VARCHAR2	Class of block
COUNT	NUMBER	Number of waits by this OPERATION for this CLASS of block
TIME	NUMBER	Sum of all wait times for all the waits by this OPERATION for this CLASS of block

4

Database Limits

This chapter lists the limits of values associated with database functions and objects. Limits exist on several levels in the database. There is usually a hard-coded limit in the database that cannot be exceeded. This value may be further restricted for any given operating system.

Database limits are divided into four categories:

- Datatype Limits
- Physical Database Limits
- Logical Database Limits
- Process and Runtime Limits

See Also: Your operating system-specific Oracle documentation.

Datatype Limits

Datatypes	Limit	Comments
BFILE	Maximum size: 4 GB Maximum size of file name: 255 characters Maximum size of directory name: 30 characters Maximum number of open BFILEs: see Comments	The maximum number of BFILEs is limited by the value of SESSION_MAX_OPEN_FILES initialization parameter, which is itself limited by the maximum number of open files the operating system will allow.
BLOB	Maximum size: 4 GB	The number of LOB columns per table is limited only by the maximum number of columns per table (that is, 1000).[a]
CHAR	2000 bytes maximum	
CHAR VARYING	4000 bytes	
CLOB	4 GB maximum	The number of LOB columns per table is limited only by the maximum number of columns per table (that is, 1000).[a]
Literals (characters or numbers in SQL or PL/SQL)	4000 characters maximum	
LONG	2^{31}-1 bytes (2 GB) maximum	Only one LONG column allowed per table.
NCHAR	2000 bytes	
NCHAR VARYING	4000 bytes	
NCLOB	4 GB maximum	The number of LOB columns per table is limited only by the maximum number of columns per table (that is, 1000).[a]

a The absolute maximum number of columns in a table is 1000. However, when you create an object table (or a relational table with columns of object, nested table, varray, or REF type), Oracle maps the columns of the user-defined types to relational columns, creating in effect "hidden columns" that count toward the 1000-column limit. For details on how Oracle calculates the total number of columns in such a table, please refer to *Oracle8i Administrator's Guide*.

Datatypes	Limit	Comments
NUMBER	999...(38 9's) x10^{125} maximum value	Can be represented to full 38-digit precision (the mantissa).
	-999...(38 9's) x10^{125} minimum value	Can be represented to full 38-digit precision (the mantissa).
Precision	38 significant digits	
RAW	2000 bytes maximum	
VARCHAR	4000 bytes maximum	
VARCHAR2	4000 bytes maximum	

[a] The absolute maximum number of columns in a table is 1000. However, when you create an object table (or a relational table with columns of object, nested table, varray, or REF type), Oracle maps the columns of the user-defined types to relational columns, creating in effect "hidden columns" that count toward the 1000-column limit. For details on how Oracle calculates the total number of columns in such a table, please refer to *Oracle8i Administrator's Guide*.

Physical Database Limits

Item	Type of Limit	Limit Value
Database Block Size	Minimum	2048 bytes; must be a multiple of operating system physical block size.
	Maximum	Operating system dependent; never more than 32 KB.
Database Blocks	Minimum in initial extent of a segment.	2 blocks
	Maximum per datafile	Platform dependent; typically 2^{22}-1 blocks.
Controlfiles	Number of control files	1 minimum; 2 or more (on separate devices) strongly recommended.
	Size of a control file	Dependent on operating system and database creation options; maximum of 20,000 x (database block size).

Item	Type of Limit	Limit Value
Database files	Maximum per tablespace	Operating system dependent; usually 1022.
	Maximum per database	65533. May be less on some operating systems. Limited also by size of database blocks and by the DB_FILES initialization parameter for a particular instance.
Database extents	Maximum	2 GB, regardless of the maximum file size allowed by the operating system.
Database file size	Maximum	Operating system dependent. Limited by maximum operating system file size; typically 2^{22} or 4M blocks.
MAXEXTENTS	Default value	Derived from tablespace default storage or DB_BLOCK_SIZE initialization parameter.
	Maximum	Unlimited
Redo Log Files	Maximum number of logfiles	Limited by value of LOG_FILES initialization parameter or MAXLOGFILES parameter in the CREATE DATABASE statement. Control file can be resized to allow more entries; ultimately an operating system limit.
	Maximum number of logfiles per group	Unlimited
Redo Log File Size	Minimum size	50K bytes
	Maximum size	Operating system limit; typically 2GB.
Tablespaces	Maximum number per database	64K Number of tablespaces cannot exceed the number of database files, as each tablespace must include at least one file.

Logical Database Limits

Item	Type	Limit
GROUP BY clause	Maximum length	The GROUP BY expression and all of the nondistinct aggregates functions (e.g., SUM, AVG) must fit within a single database block.

Item	Type	Limit
Indexes	Maximum per table	Unlimited
	total size of indexed column	40% of the database block size minus some overhead.
Columns	Per table	1000 columns maximum
	Per index (or clustered index)	32 columns maximum
	Per bitmapped index	30 columns maximum
Constraints	Maximum per column	Unlimited
Nested Queries	Maximum number	255
Partitions	Maximum length of linear partitioning key	4KB - overhead
	Maximum number of columns in partition key	16 columns
	Maximum number of partitions allowed per table or index	64K-1 partitions
Rollback Segments	Maximum number per database	No limit; limited within a session by the `MAX_ROLLBACK_SEGMENTS` initialization parameter
Rows	Maximum number per table	Unlimited
SQL Statement Length	Maximum length of statements	64K maximum; particular tools may impose lower limits.
Stored Packages	Maximum size	PL/SQL and Developer/2000 may have limits on the size of stored procedures they can call. The limits typically range from 2000 to 3000 lines of code.
		See Also: Your PL/SQL or Developer/2000 documentation for details.
Trigger Cascade Limit	Maximum value	Operating system dependent, typically 32.
Users and Roles	Maximum	2,147,483,638
Tables	Maximum per clustered table	32 tables
	Maximum per database	Unlimited

Process and Runtime Limits

Item	Type	Limit
Instances per database	Maximum number of OPS instances per database	Operating system dependent
Locks	Row-level	Unlimited
	Distributed Lock Manager	Operating system dependent
SGA size	Maximum value	Operating system dependent; typically 2 to 4 GB for 32-bit operating systems, and > 4 GB for 64-bit operating systems.
Advanced Queuing Processes	Maximum per instance	10
Job Queue Processes	Maximum per instance	36
I/O Slave Processes	Maximum per background process (DBWR, LGWR, etc.)	15
	Maximum per Backup session	15
Sessions	Maximum per instance	32K; limited by the `PROCESSES` and `SESSIONS` initialization parameters.
LCK Processes	Maximum per instance	10
MTS Servers	Maximum per instance	Unlimited within constraints set by the `PROCESSES` and `SESSIONS` initialization parameters, for instance.
Dispatchers	Maximum per instance	Unlimited within constraints set by `PROCESSES` and `SESSIONS` initialization parameters, for instance.
Parallel Execution Slaves	Maximum per instance	Unlimited within constraints set by `PROCESSES` and `SESSIONS` initialization parameters, for instance.
Backup Sessions	Maximum per instance	Unlimited within constraints set by `PROCESSES` and `SESSIONS` initialization parameters, for instance.

5

SQL Scripts

This chapter describes the SQL scripts that are required for optimal operation of Oracle.

The SQL scripts are described in the following sections:

- Creating the Data Dictionary
- Creating Additional Data Dictionary Structures
- The "NO" Scripts
- Migration Scripts

> **Note:** Comments within the SQL scripts themselves contain more detailed information and examples.

Creating the Data Dictionary

Oracle creates the data dictionary automatically when you create a database. Thereafter, whenever the database is in operation, Oracle updates the data dictionary in response to every DDL statement.

The data dictionary base tables are the first objects created in any Oracle database. They are created in the SYSTEM tablespace and must remain there. The data dictionary base tables store information about all user-defined objects in the database.

Table 5–1 lists required scripts, which are run automatically when you create a database. They are described here because you might need to run them again when upgrading to a new release of Oracle. To run these scripts you must be connected to Oracle as the user SYS.

See Also:

- Your release notes and *Oracle8i Migration* to learn when it is necessary to run these scripts.
- Your operating system specific Oracle documentation for the exact names and locations of these scripts on your system.
- *Oracle8i Administrator's Guide* for more information about scripts with names starting with DBMS.

Table 5–1 Creating the Data Dictionary Scripts

Script Name	Needed For	Description
CATALOG.SQL	All databases	Creates the data dictionary and public synonyms for many of its views
		Grants PUBLIC access to the synonyms
CATPROC.SQL	All databases	Runs all scripts required for or used with PL/SQL

Creating Additional Data Dictionary Structures

Oracle supplies other scripts that create additional structures you can use in managing your database and creating database applications. These scripts are listed in Table 5–2.

See Also: Your operating system specific Oracle documentation for the exact names and locations of these scripts on your system.

Table 5–2 Creating Additional Data Dictionary Structures

Script Name	Needed For	Run By	Description
CATBLOCK.SQL	Performance management	SYS	Creates views that can dynamically display lock dependency graphs
CATEXP7.SQL	Exporting data to Oracle7	SYS	Creates the dictionary views needed for the Oracle7 Export utility to export data from Oracle8*i* in Oracle7 Export file format
CATHS.SQL	Heterogeneous Services	SYS	Installs packages for administering heterogeneous services
CATIO.SQL	Performance management	SYS	Allows I/O to be traced on a table-by-table basis
CATOCTK.SQL	Security	SYS	Creates the Oracle Cryptographic Toolkit package
CATPARR.SQL	Oracle Parallel Server	SYS or SYSDBA	Creates parallel server data dictionary views
CATQUEUE.SQL	Advanced Queuing		Creates the dictionary objects required for Advanced Queuing.
CATREP.SQL	Advanced Replication	SYS	Runs all SQL scripts for enabling database replication
CATRMAN.SQL	Recovery Manager	RMAN or any user with GRANT_ RECOVERY_ CATALOG_OWNER role	Creates recovery manager tables and views (schema) to establish an external recovery catalog for the backup, restore, and recovery functionality provided by the Recovery Manager (RMAN) utility
DBMSIOTC.SQL	Storage management	Any user	Analyzes chained rows in index-organized tables
DBMSOTRC.SQL	Performance management	SYS or SYSDBA	Enables and disables generation of Oracle Trace output
DBMSPOOL.SQL	Performance management	SYS or SYSDBA	Enables DBA to lock PL/SQL packages, SQL statements, and triggers into the shared pool
USERLOCK.SQL	Concurrency control	SYS or SYSDBA	Provides a facility for user-named locks that can be used in a local or clustered environment to aid in sequencing application actions
UTLBSTAT.SQL and UTLESTAT.SQL	Performance monitoring	SYS	Respectively start and stop collecting performance tuning statistics

Table 5–2 Creating Additional Data Dictionary Structures

Script Name	Needed For	Run By	Description
UTLCHN1.SQL	Storage management	Any user	For use with Oracle8*i*. Creates tables for storing the output of the ANALYZE command with the CHAINED ROWS option. Can handle both physical and logical rowids.
UTLCONST.SQL	Year 2000 compliance	Any user	Provides functions to validate that CHECK constraints on date columns are year 2000 compliant
UTLDTREE.SQL	Metadata management	Any user	Creates tables and views that show dependencies between objects
UTLEXPT1.SQL	Constraints	Any user	For use with Oracle8*i*. Creates the default table (EXCEPTIONS) for storing exceptions from enabling constraints. Can handle both physical and logical rowids.
UTLHTTP.SQL	Web access	SYS or SYSDBA	PL/SQL package to retrieve data from Internet or intranet web servers via HTTP protocol
UTLIP.SQL	PL/SQL	SYS	Used primarily for migration, upgrade, and downgrade operations. It invalidates all existing PL/SQL modules by altering certain dictionary tables so that subsequent recompilations will occur in the format required by the database. It also reloads the packages STANDARD and DBMS_STANDARD, which are necessary for any PL/SQL compilations.
UTLIRP.SQL	PL/SQL	SYS	Used to change from 32-bit to 64-bit word size or vice versa. This script recompiles existing PL/SQL modules in the format required by the new database. It first alters some data dictionary tables. Then it reloads the packages STANDARD and DBMS_STANDARD, which are necessary for using PL/SQL. Finally, it triggers a recompilation of all PL/SQL modules, such as packages, procedures, and types.
UTLLOCKT.SQL	Performance monitoring	SYS or SYSDBA	Displays a lock wait-for graph, in tree structure format
UTLPG.SQL	Data conversion	SYS or SYSDBA	Provides a package that converts IBM/370 VS COBOL II

Table 5–2 Creating Additional Data Dictionary Structures

Script Name	Needed For	Run By	Description
UTLPWDMG.SQL	Security	SYS or SYSDBA	Creates PL/SQL function for default password complexity verification. Sets the default password profile parameters and enables password management features.
UTLRP.SQL	PL/SQL	SYS	Recompiles all existing PL/SQL modules that were previously in an INVALID state, such as packages, procedures, and types.
UTLSAMPL.SQL	Examples	SYS or any user with DBA role	Creates sample tables, such as EMP and DEPT, and users, such as SCOTT
UTLSCLN.SQL	Advanced Replication	Any user	Copies a snapshot schema from another snapshot site
UTLTKPROF.SQL	Performance management	SYS	Creates the TKPROFER role to allow the TKPROF profiling utility to be run by non-DBA users
UTLVALID.SQL	Partitioned tables	Any user	Creates table required for storing output of ANALYZE TABLE ...VALIDATE STRUCTURE of a partitioned table
UTLXPLAN.SQL	Performance management	Any user	Creates the table PLAN_TABLE, which holds output from the EXPLAIN PLAN command

The "NO" Scripts

The scripts in Table 5–3 are used to remove dictionary information for various optional services or components.

> **See Also:** *Oracle8i Migration* for more information about these scripts.

Table 5–3 The NO Scripts

Script Name	Needed For	Run By	Description
CATNOADT.SQL	Objects	SYS	Drops views and synonyms on dictionary metadata that relate to object types
CATNOAUD.SQL	Security	SYS	Drops views and synonyms on auditing metadata
CATNOHS.SQL	Heterogeneous Services	SYS	Removes Heterogeneous Services dictionary metadata
CATNOPRT.SQL	Partitioning	SYS	Drops views and synonyms on dictionary metadata that relate to partitioned tables and indexes
CATNOQUEUE.SQL	Advanced Queuing	SYS	Removes Advanced Queuing dictionary metadata
CATNORMN.SQL	Recovery Manager	Owner of recovery catalog	Removes recovery catalog schema
CATNOSVM.SQL	Server Manager	SYS	Removes Oracle7 Server Manager views and synonyms
CATNSNMP.SQL	Distributed management	SYS	Drops the DBSNMP user and SNMPAGENT role

Migration Scripts

The scripts in Table 5–4 are useful when migrating to another version or release.

See Also: *Oracle8i Migration*.

Table 5–4 Migration Scripts

Script Name	Needed For	Run By	Description
DROPCAT6.SQL	Removing legacy metadata	SYS	Drops Oracle6 data dictionary catalog views
DROPCAT5.SQL	Removing legacy metadata	SYS	Drops Oracle5 data dictionary catalog views

Table 5–4 Migration Scripts

Script Name	Needed For	Run By	Description
R070304.SQL	Replication	SYS	Performs a post-CATREP.SQL replication upgrade
RM804813.SQL	Recovery Manager	Owner of the Recovery Catalog Tables	Upgrades the recovery catalog from either 8.0.4 or 8.0.5 to 8.1.3
U703040.SQL	Migration from Oracle7	SYS or SYSDBA	Creates new Oracle8*i* dictionary metadata

Java Scripts

The scripts in Table 5–5 are useful only on the Oracle8*i* JServer platform.

See Also: *Oracle8i Java Developer's Guide.*

Table 5–5 Java Scripts

Script Name	Description
INITJVM.SQL	Initializes the JServer by installing the core Java class libraries and Oracle-specific Java classes into the Oracle server, and initializing Java-related packages and settings
RMJVM.SQL	Removes all elements of the JServer
UTLJAVARM.SQL	Needed only if you are upgrading from release 8.1.4. For information on this script, see *Oracle8i Migration*.

A

Oracle Wait Events

This appendix describes the event name, wait time, and parameters for wait events displayed by the V$SESSION_WAIT and V$SYSTEM_EVENT views.

Topics covered include:

- Displaying Wait Events
- Wait Events and Parameters
- Parameter Descriptions
- Wait Event Descriptions

Displaying Wait Events

Information about wait events is displayed in three dynamic performance views:

- V$SESSION_WAIT displays the events for which sessions have just completed waiting or are currently waiting.
- V$SYSTEM_EVENT displays the total number of times all the sessions have waited for the events in that view.
- V$SESSION_EVENT is similar to V$SYSTEM_EVENT, but displays all waits for each session.

> **See Also:** "V$SESSION_EVENT" on page 3-94, "V$SESSION_WAIT" on page 3-97, and "V$SYSTEM_EVENT" on page 3-112.

Many of these wait events are tied to the internal implementation of Oracle and therefore are subject to change or deletion without notice. Application developers should be aware of this and write their code to tolerate missing or extra wait events.

This appendix describes the event name, wait time, and parameters for each event.

The following SQL statement displays all Oracle events:

```
SELECT * FROM V$EVENT_NAME;
```

Wait Events and Parameters

The following wait events are present in the Oracle server. The columns P1, P2, and P3 represent parameters for the wait event. Selected parameters and wait events are further described in the sections that follow.

Event Name	P1	P2	P3
alter system set mts_dispatcher	waited	not used	not used
batched allocate scn lock request	not used	not used	not used
BFILE check if exists			
BFILE check if open			
BFILE closure			
BFILE get length			
BFILE get name object			

Event Name	P1	P2	P3
BFILE get path object			
BFILE internal seek			
BFILE open			
BFILE read			
buffer being modified waits			
buffer busy due to global cache	file#	block#	id
buffer busy waits	file#	block#	id
buffer deadlock	dba	class*10+mode	flag
buffer for checkpoint	buffer#	dba	state*10+mode
buffer latch	latch addr	chain#	not used
buffer read retry	file#	block#	not used
checkpoint completed	not used	not used	not used
checkpoint range buffer not saved	not used	not used	not used
Contacting SCN server or SCN lock master			
control file parallel write	files	blocks	requests
control file sequential read	file#	block#	blocks
control file single write	file#	block#	blocks
conversion file read	file#	block#	blocks
db file parallel read	files	blocks	requests
db file parallel write	files	blocks	requests
db file scattered read	file#	block#	blocks
db file sequential read	file#	block#	blocks
db file single write	file#	block#	blocks
debugger command	not used	not used	not used
DFS db file lock	file#	not used	not used
DFS lock handle	type | mode	id1	id2
direct path write	file number	first dba	block cnt
dispatcher shutdown			
dispatcher timer	sleep time	not used	not used
DLM generic wait event			

Event Name	P1	P2	P3
dupl. cluster key	dba	not used	not used
enqueue	name | mode	id1	id2
file identify	fib	file name	opcode
file open	fib	iov	0
free buffer waits	file#	block#	set-id#
free global transaction table entry	tries	not used	not used
free process state object	not used	not used	not used
global cache bg acks			
global cache cr request			
global cache freelist wait	lenum	not used	not used
global cache lock busy	file#	block#	lenum
global cache lock cleanup	file#	block#	lenum
global cache lock null to s	file#	block#	lenum
global cache lock null to x	file#	block#	lenum
global cache lock open null	file#	block#	class
global cache lock open s	file#	block#	lenum
global cache lock open x	file#	block#	lenum
global cache lock s to x	file#	block#	lenum
global cache multiple locks			
global cache pending ast			
imm op	msg ptr	not used	not used
inactive session	session#	waited	not used
inactive transaction branch	branch#	waited	not used
index block split	rootdba	level	childdba
instance recovery	undo segment#	not used	not used
instance state change	layer	value	waited
IO clients wait for LMON to join GMS group			
io done	msg ptr	not used	not used
kcl bg acks	count	loops	not used

Event Name	P1	P2	P3
kdi: Done Message Dequeue-Coord			
ktpr: Done Message Dequeue-Coord			
latch activity	address	number	process#
latch free	address	number	tries
library cache load lock	object address	lock address	10*mask+namespace
library cache lock	handle address	lock address	10*mode+namespace
library cache pin	handle address	pin address	10*mode+namespace
LMON wait for LMD to inherit communication channels			
local write wait			
lock manager wait for dlmd to shutdown			
lock manager wait for remote message	waittime	not used	not used
log buffer space	not used	not used	not used
log file parallel write	files	blocks	requests
log file sequential read	log#	block#	blocks
log file single write	log#	block#	blocks
log file switch (archiving needed)	not used	not used	not used
log file switch (checkpoint incomplete	not used	not used	not used
log file switch (clearing log file)	not used	not used	not used
log file switch completion	not used	not used	not used
log file sync	buffer#	not used	not used
log switch / archive	thread#	not used	not used
name-service call wait			
Null event			
on-going SCN fetch to complete	not used	not used	not used
PAR RECOV: Dequeue msg-Slave			
PAR RECOV: Wait for reply-Query Coord			
parallel execution create server	nservers	sleeptime	enqueue
parallel execution dequeue wait	reason	sleeptime / senderid	passes

Event Name	P1	P2	P3
Parallel Execution Idle Wait-Slaves wait event			
parallel execution qref latch	function	sleeptime	qref
parallel execution server shutdown	nalive	sleeptime	loop
parallel execution signal server	serial	error	nbusy
pending global transaction(s)	scans	not used	not used
pipe get	handle address	buffer length	timeout
pipe put	handle address	record length	timeout
PL/SQL lock timer	duration	not used	not used
pmon rdomain attach			
pmon timer	duration	not used	not used
process startup	type	process#	waited
queue messages	queue id	process#	wait time
rdbms ipc message	timeout	not used	not used
rdbms ipc message block	not used	not used	not used
rdbms ipc reply	from_process	timeout	not used
redo wait	not used	not used	not used
refresh controlfile command			
reliable message			
Replication Dequeue			
retry contact SCN lock master			
row cache lock	cache id	mode	request
scginq AST call	not used	not used	not used
secondary event			
single-task message	not used	not used	not used
slave exit	nalive	sleeptime	loop
slave wait			
smon timer	sleep time	failed	not used
sort segment request			
SQL*Net break/reset to client	driver id	break?	not used

Event Name	P1	P2	P3
SQL*Net break/reset to dblink	driver id	break?	not used
SQL*Net message from client	driver id	#bytes	not used
SQL*Net message from dblink	driver id	#bytes	not used
SQL*Net message to client	driver id	#bytes	not used
SQL*Net message to dblink	driver id	#bytes	not used
SQL*Net more data from client	driver id	#bytes	not used
SQL*Net more data from dblink	driver id	#bytes	not used
SQL*Net more data to client	driver id	#bytes	not used
SQL*Net more data to dblink	driver id	#bytes	not used
switch logfile command	not used	not used	not used
Test if message present			
timer in sksawat	not used	not used	not used
trace continue			
trace unfreeze			
trace writer flush			
trace writer I/O			
transaction	undo seg# | slot#	wrap#	count
unbound tx	not used	not used	not used
undo segment extension	segment#	not used	not used
undo segment recovery	segment#	tx flags	not used
undo segment tx slot	segment#	not used	not used
virtual circuit status	circuit#	status	not used
Wait for a parallel reco to abort			
Wait for a undo record			
wait for checking DLM domain			
Wait for credit-free buffer			
Wait for credit-need buffer to send			
Wait for credit-send blocked			
wait for DLM			
wait for DLM latch	latchtype	gets	immediate

Event Name	P1	P2	P3
wait for DLM process allocation			
wait for DLM reconfiguration to complete			
wait for gms de-registration			
wait for gms registration			
wait for influx DLM latch	latchtype	latchaddr	not used
wait for lmd and pmon to attach DLM			
wait for lock db to become frozen			
wait for lock db to unfreeze			
wait for ownership of group-owned lock			
wait for pmon to exit			
wait for reconfiguration to start			
wait for recovery domain attach			
wait for recovery domain latch in kjpr			
wait for recovery validate to complete			
wait for register recovery to complete			
wait for send buffers to send DLM message			
Wait for slaves to ACK-Query Coord			
Wait for slaves to join-Query Coord			
Wait for stopper event to be increased			
wait for tickets to send DLM message			
wait to run in thread run			
waiting in scheduler enter			
waiting in scheduler enter2			
waiting in scheduler shutdown			
waiting in scheduler system stop			
waiting in thread check			
waiting in thread check2			
waiting in thread end wait			
waiting in thread run (queued)			
WMON goes to sleep	not used	not used	not used

Event Name	P1	P2	P3
write complete waits	file#	block#	id
writes stopped by instance recovery or database suspension	by thread #	our thread#	not used

Parameter Descriptions

block#

This is the block number of the block for which Oracle needs to wait. The block number is relative to the start of the file. To find the object to which this block belongs, enter these SQL statements:

```
select name, kind
from ext_to_obj_view
where file#  = file#
     and lowb  <= block#
     and highb >= block#;
```

blocks

The number of blocks that is being either read from or written to the file. The block size is dependent on the file type:

- database files have a block size of DB_BLOCK_SIZE
- logfiles and controlfiles have a block size that is equivalent to the physical block size of the platform

break?

If the value for this parameter equals 0, a reset was sent to the client. A non-zero value indicates that a break was sent to the client.

class

The class of the block describes how the contents of the block are used. For example, class 1 represents data block, class 4 represents segment header.

dba

The initials "dba" represents the data block address. A dba consists of a file number and a block number.

driver id

The address of the disconnect function of the driver that is currently being used.

file#

The following query returns the name of the database file:

```
select *
from v$datafile
where file# = file#;
```

id1

The first identifier (*id1*) of the enqueue or global lock takes its value from P2 or P2RAW. The meaning of the identifier depends on the name (P1).

id2

The second identifier (*id2*) of the enqueue or global lock takes its value from P3 or P3RAW. The meaning of the identifier depends on the name (P1).

lenum

The relative index number into V$LOCK_ELEMENT.

mode

The *mode* is usually stored in the low order bytes of P1 or P1RAW and indicates the mode of the enqueue or global lock request. This parameter has one of the following values:

Table A–1 Lock Mode Values

Mode Value	Description
1	Null mode
2	Sub-Share
3	Sub-Exclusive
4	Share
5	Share / Sub-Exclusive
6	Exclusive

Use the following SQL statement to retrieve the name of the lock and the mode of the lock request:

```
select chr(bitand(p1,-16777216)/16777215)||
       chr(bitand(p1, 16711680)/65535) "Lock",
       bitand(p1, 65536) "Mode"
from v$session_wait
where event = 'DFS enqueue lock acquisition';
```

name and type

The name or "type" of the enqueue or global lock can be determined by looking at the two high order bytes of P1 or P1RAW. The name is always two characters. Use the following SQL statement to retrieve the lock name.

```
select chr(bitand(p1,-16777216)/16777215)||
      chr(bitand(p1,16711680)/65535) "Lock"
from v$session_wait
where event = 'DFS enqueue lock acquisition';
```

namespace

The name of the object namespace as it is displayed in V$DB_OBJECT_CACHE view.

requests

The number of I/Os that are "requested". This differs from the number of blocks in that one request could potentially contain multiple blocks.

session#

The number of the inactive session. Use the following SQL statement to find more information about the session:

```
select *
from v$session
where sid = session#;
```

waited

This is the total amount of time the session has waited for this session to die.

Wait Event Descriptions

This section describes some of the more common Oracle events in more detail.

alter system set mts_dispatchers

A session has issued a statement ALTER SYSTEM SET MTS_DISPATCHERS = <string> and is waiting for the dispatchers to get started.

Wait Time: The session will wait 1/100 of a second and check to see if the new dispatchers have started else the session will wait again.

Parameters:

waited The number of times that the session has waited 1/100 of second.

batched allocate scn lock request

A session is waiting on another process to allocate a system change number (SCN). If the foreground timed out waiting on a process to get the SCN, the foreground will get the SCN.

Wait Time: The wait time is 1 second on the assumption that an SCN allocation should normally need much less than that.

Parameters: None

BFILE check if exists

The session waits to check if an external large object (LOB) exists.

Wait Time: The total elapsed time for the **exists** call.

Parameters:

session# See "session#" on page A-11.

waited See "waited" on page A-11.

BFILE check if open

The session waits to check if an external large object (LOB) has already been opened.

Wait Time: The total elapsed time for the **isopen** call.

Parameters:

session# See "session#" on page A-11.

waited See "waited" on page A-11.

BFILE closure

The session waits for an external large object (LOB) to close.

Wait Time: The total elapsed time for the **close** call.

Parameters:

session# See "session#" on page A-11.

waited See "waited" on page A-11.

BFILE get length

The session waits on a call to check the size of an external large object (LOB).

Wait Time: The total elapsed time for the call to check the LOB size.

Parameters:

session# See "session#" on page A-11.

waited See "waited" on page A-11.

BFILE get name object

The session waits on a call to find or generate the external name of a external large object.

Wait Time: The total elapse time for **make external file name** to complete.

Parameters:

session# See "session#" on page A-11.

waited See "waited" on page A-11.

BFILE get path object

The session is waiting on a call to find or generate the external path name of an external large object (LOB).

Wait Time: The total elapsed time for **make external path** to complete.

Parameters:

session# See "session#" on page A-11.

waited See "waited" on page A-11.

BFILE internal seek

The session waits for a positioning call within the external large object (LOB) to complete.

Wait Time: The total elapse time for the **seek** to complete.

Parameters:

session# See "session#" on page A-11.

waited See "waited" on page A-11.

BFILE open

The session waits to check if an external large object (LOB) has already been opened.

Wait Time: The total elapsed time for the **isopen** call.

Parameters:

session# See "session#" on page A-11.

waited See "waited" on page A-11.

BFILE read

The session waits for a read from a external large object (LOB) to complete.

Wait Time: The total elapse time for the **read** to complete.

Parameters:

session# See "session#" on page A-11.

waited See "waited" on page A-11.

buffer busy waits

Wait until a buffer becomes available. This event happens because a buffer is either being read into the buffer cache by another session (and the session is waiting for that read to complete) or the buffer is the buffer cache, but in a incompatible mode (that is, some other session is changing the buffer).

Wait Time: Normal wait time is 1 second. If the session was waiting for a buffer during the last wait, then the next wait will be 3 seconds.

Parameters:

file#	See "file#" on page A-10.
block#	See "block#" on page A-9.
id	The buffer busy wait event is called from different places in the session.

buffer deadlock

Oracle does not really wait on this event; the foreground only yields the CPU. Thus, the chances of catching this event are very low. This is not an application induced deadlock, but an assumed deadlock by the cache layer. The cache layer cannot get a buffer in a certain mode within a certain amount of time.

Wait Time: 0 seconds. The foreground process only yields the CPU and will usually be placed at the end of the CPU run queue.

Parameters:

class	See "class" on page A-9.
mode	See "mode" on page A-10.
flag	The flag points to the internal flags used by the session to get this block.
dba	See "dba" on page A-9.

buffer for checkpoint

The buffer could not be checkpointed, because some process is modifying it. This means that after the wait, the DBWR will scan the whole buffer cache again. This could happen during a database close or after a user does a local checkpoint. During this situation the database cannot be closed.

Wait Time: 1 second

Parameters:

dba	See "dba" on page A-9.
state	State refers to the status of the buffer contents.
mode	See "mode" on page A-10.
buffer#	This is the index of the block in the buffer cache (V$BH).

buffer latch

The session waits on the buffer hash chain latch. Primarily used in the dump routines.

Wait Time: 1 second

Parameters:

latch addr The virtual address in the SGA where this latch is located. Use the following command to find the name of this latch:

```
select *
from v$latch a, v$latchname b
where addr = latch addr
and a.latch# = b.latch#;
```

chain# The index into array of buffer hash chains. When the chain is 0xffffffff, the foreground waits on the LRU latch.

buffer read retry

This event occurs only if the instance is mounted in shared mode (Parallel Server). During the read of the buffer, the contents changed. This means that either:

- the version number, dba, or the incarnation and sequence number stored in the block no longer match
- the checksum on the block does not match the checksum in the block

The block will be re-read (this may fail up to 3 times), then corruption is assumed and the corrupt block is dumped in the trace file.

Wait Time: The wait time is the elapsed time of the read.

Parameters:

file#	See "file#" on page A-10.
block#	See "block#" on page A-9.

checkpoint completed

A session waits for a checkpoint to complete. This could happen, for example, during a close database or a local checkpoint.

Wait Time: 5 seconds

Parameters: None

checkpoint range buffer not saved

During a range checkpoint operation a buffer was found that was not saved or written. Either:

- the session will wait on this event if the write batch is empty and it is the first time that the session waited on this event in the range checkpoint operation
- the current range checkpoint operation will be aborted and a new one will be started to complete the operation

Wait Time: 10 milliseconds

Parameters: None

control file parallel write

This event occurs while the session is writing physical blocks to all controlfiles. This happens when:

- the session starts a controlfile transaction (to make sure that the controlfiles are up to date in case the session crashes before committing the controlfile transaction)
- the session commits a transaction to a controlfile
- changing a generic entry in the controlfile, the new value is being written to all controlfiles

Wait Time: The wait time is the time it takes to finish all writes to all controlfiles.

Parameters:

files	The number of controlfiles to which the session is writing.
blocks	The number of blocks that the session is writing to the controlfile.
requests	The number of I/O requests which the session wants to write.

control file sequential read

Reading from the controlfile. This happens in many cases. For example, while:

- making a backup of the controlfiles
- sharing information (between instances) from the controlfile
- reading other blocks from the controlfiles
- reading the header block

Wait Time: The wait time is the elapsed time of the read.

Parameters:

file#	The controlfile from which the session is reading.
block#	Block number in the controlfile from where the session starts to read. The block size is the physical block size of the port (usually 512 bytes, some UNIX ports have 1 or 2 Kilobytes).
blocks	The number of blocks that the session is trying to read.

control file single write

This wait is signaled while the controlfile's shared information is written to disk. This is an atomic operation protected by an enqueue (CF), so that only one session at a time can write to the entire database.

Wait Time: The wait time is the elapsed time of the write.

Parameters:

file#	This identifies the controlfile to which the session is currently writing.
block#	Block number in the controlfile where the write begins. The block size is the as the physical block size of the port (usually 512 bytes, some UNIX ports have 1 or 2 Kilobytes).
blocks	The number of blocks that the session is trying to read.

conversion file read

This event occurs during a the creation of a Version 7 controlfile as part of converting a database to Version 7 from Version 6.

Wait Time: The wait time is the elapsed time of the read.

Parameters:

file#	The controlfile to which the session is currently writing.
block#	Block number in the controlfile where the write begins. The block size is the as the physical block size of the port (usually 512 bytes, some UNIX ports have 1 or 2 Kilobytes).
blocks	The number of blocks that the session is trying to read.

db file parallel read

This happens during recovery. Database blocks that need to be changed as part of recovery are read in parallel from the database.

Wait Time: Wait until all of the I/Os are completed.

Parameters:

files	This indicates the number of files to which the session is reading.
blocks	This indicates the total number of blocks to be read.
requests	This indicates the total number of I/O requests, which will be the same as blocks.

db file parallel write

This event occurs in the DBWR. It indicates that the DBWR is performing a parallel write to files and blocks. The parameter *requests* indicates the real number of I/Os that are being performed. When the last I/O has gone to disk, the wait ends.

Wait Time: Wait until all of the I/Os are completed.

Parameters:

files	This indicates the number of files to which the session is writing.
blocks	This indicates the total number of blocks to be written.

requests This indicates the total number of I/O requests, which will be the same as blocks.

db file scattered read

Similar to **db file sequential read**, except that the session is reading multiple data blocks.

Wait Time: The wait time is the actual time it takes to do all of the I/Os.

Parameters:

file# See "file#" on page A-10.

block# See "block#" on page A-10.

blocks The number of blocks that the session is trying to read from the *file#* starting at *block#*.

db file sequential read

The session waits while a sequential read from the database is performed. This event is also used for rebuilding the controlfile, dumping datafile headers, and getting the database file headers.

Wait Time: The wait time is the actual time it takes to do the I/O.

Parameters:

file# See "file#" on page A-10.

block# See"block#" on page A-10.

blocks This is the number of blocks that the session is trying to read (should be 1).

db file single write

This event is used to wait for the writing of the file headers.

Wait Time: The wait time is the actual time it takes to do the I/O.

Parameters:

file# See "file#" on page A-10.

block# See"block#" on page A-10.

blocks This is the number of blocks that the session is trying to write in *file#* starting at *block#*.

DFS db file lock

This event occurs only for the DBWR in the Oracle Parallel Server. Each DBWR of every instance holds a global lock on each file in shared mode. The instance that is trying to offline the file will escalate the global lock from shared to exclusive. This signals the other instances to synchronize their SGAs with the controlfile before the file can be taken offline. The name of this lock is **DF** (see Appendix B, "Oracle Enqueue Names" for more information).

Wait Time: 1 second in loop. The DBWR is waiting in a loop (sleep, check) for the other instances to downgrade to NULL mode. During this time, the DBWR cannot perform other tasks such as writing buffers.

Parameter:

file See "file#" on page A-10.

DFS lock handle

The session waits for the lock handle of a global lock request. The lock handle identifies a global lock. With this lock handle, other operations can be performed on this global lock (to identify the global lock in future operations such as conversions or release). The global lock is maintained by the DLM.

Wait Time: The session waits in a loop until it has obtained the lock handle from the DLM. Inside the loop there is a wait of 0.5 seconds.

Parameters:

name See "name and type" on page A-11.

mode See "mode" on page A-10.

id1 See "id1" on page A-10.

id2 See "id2" on page A-10.

The session needs to get the lock handle.

direct path read

During Direct Path operations the data is asynchronously read from the database files. At some stage the session needs to make sure that all outstanding asynchronous I/O have been completed to disk. This can also happen if during a direct read no more slots are available to store outstanding load requests (a load request could consist of multiple I/Os).

Wait Time: 10 seconds. The session will be posted by the completing asynchronous I/O. It will never wait the entire 10 seconds. The session waits in a tight loop until all outstanding I/Os have completed.

Parameters:

descriptor address	This is a pointer to the I/O context of outstanding direct I/Os on which the session is currently waiting.
first dba	The dba of the oldest I/O in the context referenced by the descriptor address.
block cnt	Number of valid buffers in the context referenced by the descriptor address.

direct path write

During Direct Path operations, the data is asynchronously written to the database files. At some stage the session needs to make sure that all outstanding asynchronous I/O have been completed to disk. This can also happen if, during a direct write, no more slots are available to store outstanding load requests (a load request could consist of multiple I/Os).

Wait Time: 10 seconds. The session will be posted by the completing asynchronous I/O. It will never wait the entire 10 seconds. The session waits in a tight loop until all outstanding I/Os have completed.

Parameters:

descriptor address	This is a pointer to the I/O context of outstanding direct I/Os on which the session is currently waiting.
first dba	The dba of the oldest I/O in the context referenced by the descriptor address.
block cnt	Number of valid buffers in the context referenced by the descriptor address.

dispatcher shutdown

During shutdown immediate or normal, the shutdown process must wait for all the dispatchers to shutdown. As each dispatcher is signaled, the session that causes the shutdown is waits on this event until the requested dispatcher is no longer alive.

Wait Time: 1 second

Parameter:

waited Indicates the cumulative wait time. After 5 minutes, the session writes to the alert and trace files to indicate that there might be a problem.

dispatcher timer

This basically means that the dispatcher is idle and waiting for some work to arrive.

Wait Time: 60 seconds

Parameter:

sleep time The intended sleep time. The dispatcher will return to work sooner if it is posted by either data arriving on the network or by a post from a shared server process to send data back to the client.

duplicate cluster key

It is possible for a race condition to occur when creating a new cluster key. If it is found that another process has put the cluster key into the data/index block, then the session waits and retries. The retry should then find a valid cluster key.

Wait Time: 0.01 seconds

Parameter:

dba The dba of the block into which the session is trying to insert a cluster key.

enqueue

The session is waiting for a local enqueue. The wait is dependent on the name of the enqueue (see Appendix B, "Oracle Enqueue Names").

Wait Time: Depends on the enqueue name.

Parameters:

name	See "name and type" on page A-11.
mode	See "mode" on page A-10.

file identify

The time it takes to identify a file so that it can be opened later.

file open

The time it takes to open the file.

free buffer waits

This will happen if:

- All buffer gets have been suspended. This could happen when a file was read-only and is now read-write. All the existing buffers need to be invalidated since they are not linked to lock elements (needed when mounted parallel (shared)). So cache buffers are not assigned to data block addresses until the invalidation is finished.
- The session moved some dirty buffers to the dirty queue and now this dirty queue is full. The dirty queue needs to be written first. The session will wait on this event and try again to find a free buffer
- This also happens after inspecting **free buffer inspected** buffers. If no free buffer is found, Oracle waits for one second, and then tries to get the buffer again (depends on the context). For more information, see free buffer inspected on page C-7.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-10.
block#	See "block#" on page A-9.

free global transaction table entry

The session is waiting for a free slot in the global transaction table (used by the Distributed Database option). It will wait for 1 second and try again.

Wait Time: 1 second

Parameter:

tries	The number of times the session tried to find a free slot in the global transaction table.

free process state object

Used during the creation of a process. The session will scan the process table and look for a free process slot. If none can be found, PMON is posted to check if all the processes currently in the process table are still alive. If there are dead processes, PMON will clean them and make the process slot available to new processes. The waiting process will then rescan the process table to find the new slot.

Wait Time: 1 second

Parameters: None

global cache freelist wait

All releasable locks are used and a new one has been requested. To make a lock element available, a lock element is pinged.

Wait Time: The duration of the lock get operation to ping the lock element.

Parameter:

lenum	See "lenum" on page A-8.

global cache lock busy

The session waits to convert a buffer up from Shared Current to Exclusive Current status.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-10.
block#	See"block#" on page A-10.
lenum	See "lenum" on page A-8.

global cache lock cleanup

PMON is waiting for an LCK process to cleanup the lock context after a foreground process died while doing a global cache lock operation.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-10.
block#	See"block#" on page A-10.
lenum	See "lenum" on page A-8.

global cache lock null to s

The session waits for a lock convert from NULL to SHARED mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-10.
block#	See"block#" on page A-10.
class	See "class" on page A-9.

global cache lock null to x

The session waits for a lock convert from NULL to EXCLUSIVE mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-10.
block#	See"block#" on page A-10.
lenum	See"lenum" on page A-10.

global cache lock open null

The session waits for a lock get in NULL mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-10.
block#	See"block#" on page A-10.
class	See"class" on page A-10.

global cache lock open s

The session waits for a lock get in SHARED mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-10.
block#	See"block#" on page A-10.
class	See"class" on page A-10.

global cache lock open x

The session waits for a lock get in EXCLUSIVE mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-10.
block#	See"block#" on page A-10.
lenum	See"lenum" on page A-10.

global cache lock s to x

The session waits for a lock convert from SHARED to EXCLUSIVE mode on the block identified by file# and block#.

Wait Time: 1 second

Parameters:

file#	See "file#" on page A-10.
block#	See"block#" on page A-10.
lenum	See"lenum" on page A-10.

inactive session

This event is used for two purposes:

- Switching sessions

  If a time-out period has been specified, then wait that amount of time for the session to be detached.

- Killing sessions

  From either KILL SESSION or internal request. Having posted a session that it should kill itself, wait for up to 1 minute for the session to die.

Wait Time: 1 second

Parameters:

session#	See "session#" on page A-11.
waited	See "waited" on page A-11.

inactive transaction branch

The session waits for a transaction branch that is currently used by another session.

Wait Time: 1 second

Parameters:

branch#	The serial number of the transaction for which the session is waiting.

waited See "waited" on page A-11.

index block split

While trying to find an index key in an index block, Oracle noticed that the index block was being split. Oracle will wait for the split to finish and try to find the key again.

Wait Time: The session will yield the CPU, so there is no actual waiting time.

Parameters:

rootdba The root of the index.

level This is the level of the block that the session is trying to split in the index. The leaf blocks are level 0. If the level is > 0, it is a branch block. (The root block can be considered a special branch block).

childdba The block that the session is trying to split.

instance recovery

The session waits for SMON to finish the instance, transaction recovery, or sort segment cleanup.

Wait Time: The wait time can vary and depends on the amount of recovery needed.

Parameter:

undo segment# If the value is 0, SMON is probably performing instance recovery. If P1 > 0, use this query to find the undo segment:

```
select *
from v$rollstat
where usn = undo segment#;
```

instance state change

The session waits for SMON to enable or disable cache or transaction recovery. This usually happens during ALTER DATABASE OPEN or CLOSE.

Wait Time: Wait time depends on the amount of time the action takes (that is, the amount of recovery needed).

Parameters:

layer	This value can be 1 or 2. If 1, it means that the transaction layer wants transaction recovery to be performed. If 2, it means that cache recovery will be performed.
value	This value can be 0 (disable) or 1 (enable).
waited	The number of seconds waited so far.

io done

The session waits for an I/O to complete or it waits for a slave process to become available to submit the I/O request. This event occurs on platforms that do not support asynchronous I/O.

Wait Time: 50 milliseconds

Parameter:

msg ptr	A pointer to the I/O request.

kcl bg acks

The session waits for the background LCK process(es) to finish what they are doing. For example:

- lock recovery
- initializing the locks (start up)
- finalizing the locks (shut down)

Wait Time: 10 seconds

Parameters:

count	The number of LCK processes that have finished.
loops	The number times the process had to wait for the LCK processes to finish what they were doing.

latch activity

This event is used as part of the process of determining whether a latch needs to be cleaned.

Wait Time: 0.05 to 0.1 seconds

Parameters:

address The address of the latch that is being checked.

number The latch number of the latch that has activity. To find more information on the latch, use this SQL command:

```
select *
from v$latchname
where latch# = number;
```

process# If this is 0, it is the first phase of the in-flux tests.

latch free

The process waits for a latch that is currently busy (held by another process).

Wait Time: The wait time increases exponentially and does not include spinning on the latch (active waiting). The maximum wait time also depends on the number of latches that the process is holding. There is an incremental wait of up to 2 seconds.

Parameters:

address The address of the latch for which the process is waiting.

number The latch number that indexes in the V$LATCHNAME view.To find more information on the latch, use this SQL command:

```
select *
from v$latchname
where latch# = number;
```

tries A count of the number of times the process tried to get the latch (slow with spinning) and the process has to sleep.

library cache load lock

The session tries to find the load lock for the database object so that it can load the object. The load lock is always obtained in Exclusive mode, so that no other process can load the same object. If the load lock is busy the session will wait on this event until the lock becomes available.

Wait Time: 3 seconds (1 second for PMON)

Parameters:

object address	Address of the object being loaded.
lock address	Address of load lock being used.
mask	Indicates which data pieces of the object that needs to be loaded.

library cache lock

This event controls the concurrency between clients of the library cache. It acquires a lock on the object handle so that either:

- one client can prevent other clients from accessing the same object
- the client can maintain a dependency for a long time (e.g., no other client can change the object)

This lock is also obtained to locate an object in the library cache.

Wait Time: 3 seconds (1 second for PMON)

Parameters:

handle address	Address of the object being loaded.
lock address	Address of the load lock being used. This is not the same thing as a latch or an enqueue, it is a State Object.
mode	Indicates the data pieces of the object which need to be loaded.
namespace	See "namespace" on page A-11.

library cache pin

This event manages library cache concurrency. Pinning an object causes the heaps to be loaded into memory. If a client wants to modify or examine the object, the client must acquire a pin after the lock.

Wait Time: 3 seconds (1 second for PMON)

Parameters:

handle address	Address of the object being loaded.

pin address	Address of the load lock being used. This is not the same thing as a latch or an enqueue, it is basically a State Object.
mode	Indicates which data pieces of the object that needs to be loaded.
namespace	See "namespace" on page A-11.

lock manager wait for remote message

The lock manager waits for a message from a remote lock manager in the same configuration.

Wait Time: The elapsed time of the wait

Parameter:

waittime	The elapsed time of the actual wait.

log buffer space

Waiting for space in the log buffer because the session is writing data into the log buffer faster than LGWR can write it out. Consider making the log buffer bigger if it is small, or moving the log files to faster disks such as striped disks.

Wait Time: Usually 1 second, but 5 seconds if it is waiting for a Switch Logfile to complete.

Parameters: None

log file parallel write

Writing redo records to the redo log files from the log buffer.

Wait Time: Time it takes for the I/Os to complete. Even though redo records are written in parallel, the parallel write is not complete until the last I/O is on disk.

Parameters:

files	Number of files to be written.
blocks	Number of blocks to be written.
requests	Number of I/O requests.

log file sequential read

Waiting for the read from this logfile to return. This is used to read redo records from the log file.

Wait Time: Time it takes to complete the physical I/O (read).

Parameters:

log#	The relative sequence number of the logfiles within a log group (used only when dumping the logfiles).
block#	See "block#" on page A-9.
blocks	The number of blocks to read.

log file single write

Waiting for the write to this logfile to complete. This event is used while updating the header of the logfile. It is signaled when adding a log file member and when incrementing sequence numbers.

Wait Time: Time it takes for the physical I/O (write) to complete.

Parameters:

log#	This is the number of the group/log to which the session is currently writing.
block#	See "block#" on page A-9.
blocks	The number of blocks to write.

log file switch (archiving needed)

Waiting for a log switch because the log that the LGWR will be switching into has not been archived yet. Check the alert file to make sure that archiving has not stopped due to a failed archive write. To speed archiving, consider adding more archive processes or putting the archive files on striped disks.

Wait Time: 1 second

Parameters: None

log file switch (checkpoint incomplete)

Waiting for a log switch because the session cannot wrap into the next log. Wrapping cannot be performed because the checkpoint for that log has not completed.

Wait Time: 1 second

Parameters: None

log file switch (clearing log file)

Waiting for a log switch because the log is being cleared due to a `CLEAR LOGFILE` command or implicit clear logfile executed by recovery.

Wait Time: 1 second

Parameters: None

log file switch completion

Waiting for a log switch to complete.

Wait Time: 1 second

Parameters: None

log file sync

When a user session commits, the session's redo information needs to be flushed to the redo logfile. The user session will post the LGWR to write the log buffer to the redo log file. When the LGWR has finished writing, it will post the user session.

Wait Time: The wait time includes the writing of the log buffer and the post.

Parameter:

buffer#	The number of the physical buffer in the redo log buffer that needs to be sync'ed

log switch/archive

Used as part of the `ALTER SYSTEM ARCHIVE LOG CHANGE` *scn* command. The session waits for the current log from all open threads to be archived.

Wait Time: Wait for up to 10 seconds.

Parameter:

thread#	The thread number of the thread that is currently archiving its current log.

on-going SCN fetch to complete

Another session is fetching the SCN (system change number). This session waits for the other session finish fetching the SCN.

Wait Time: 1 second

Parameters: None

parallel execution create server

Used when creating or starting a parallel execution slave.

Wait Time: The time it takes to start all of the requested parallel execution slaves.

Parameters:

nservers	The number of parallel execution slaves that are being started.
sleeptime	Time it takes to get the processes started. The process should be started within *sleeptime*.
enqueue	The number of blocks to read.

parallel execution dequeue wait

The process is waiting for a message during a parallel execute.

Wait Time: The wait time depends on how quickly the message arrives. Wait times can vary, but it will normally be a short period of time.

Parameters:

reason	The reason for dequeuing.
sleeptime	The amount of time that the session slept.
loop	The total number of times that the session has slept.

parallel execution qref latch

Each parallel execution process has a parallel execution qref latch, which needs to be acquired before the queue buffers can be manipulated.

Wait Time: Wait up to 1 second.

Parameters:

function	Indicates the type of wait that the session is doing.
sleeptime	The amount of time that the session waits (in hundredths of a second).
qref	The address of the process queue for which the session is waits.

parallel execution server shutdown

During normal or immediate shutdown the parallel execution slaves are posted to shutdown cleanly. If any parallel execution slaves are still alive after 10 seconds, they are killed.

Wait Time: Wait up to 0.5 seconds.

Parameters:

nalive	The number of parallel execution slaves that are still running.
sleeptime	The total sleeptime since the session started to wait on this event.
loop	The number of times the session waited for this event.

parallel execution signal server

This event occurs only in Exclusive mode. The query coordinator is signalling the Query Slaves that an error has occurred.

Wait Time: 0.5 seconds

Parameters:

serial	The serial number of the slave process queue.
error	The error that has occurred.
nbusy	The number of slave processes that are still busy.

pending global transaction(s)

This event should happen only during testing. The session waits for pending transactions to clear.

Wait Time: 30 seconds

Parameter:

scans	Number of times the session has scanned the PENDING_TRANS$ table.

pipe get

The session waits for a message to be received on the pipe or for the pipe timer to expire.

Wait Time: There is a 5 second wake up (check) and the pipe timer set by the user.

Parameters:

handle address	The library cache object handle for this pipe.
buffer length	The length of the buffer.
timeout	The pipe timer set by the user.

pipe put

The session waits for the pipe send timer to expire or for space to be made available in the pipe.

Wait Time: There is the 5 second wakeup (check) and the user-supplied timeout value.

Parameters:

handle address	The library cache object handle for this pipe.
record length	The length of the record or buffer that has been put into the pipe.
timeout	The pipe timer set by the user.

PL/SQL lock timer

This event is called through the DBMSLOCK.SLEEP procedure or USERLOCK.SLEEP procedure. This event will most likely originate from procedures written by a user.

Wait Time: The wait time is in hundredths of seconds and is dependent on the user context.

Parameter:

duration The duration that the user specified in the DBMS_LOCK.SLEEP or USER_LOCK.SLEEP procedures.

pmon rdomain attach

This is the main wait event for PMON. When PMON is idle, it is waiting on this event.

pmon timer

This is the main wait event for PMON. When PMON is idle, it is waiting on this event.

Wait Time: Up to 3 seconds, if not posted before.

Parameter:

duration The actual amount of time that the PMON is trying to sleep.

process startup

Wait for a Multi-Threaded Server (Shared Server), Dispatcher, or other background process to start.

Wait Time: Wait up to 1 second for a background process to start. If timed out, then re-wait until 5 minutes have passed and signal an error. If the process has started, the event will acknowledge this.

Parameters:

type The process type that was started.

process# The process number of the process being started.

waited Cumulative time waited for the process to start.

queue messages

The session is waiting on an empty OLTP queue (Advanced Queuing) for a message to arrive so that the session can dequeue that message.

Wait Time: The amount of time that the session wants to wait is determined by the parameter *wait time*.

Parameters:

queue id	The ID of the OLTP queue for which this session is waiting.
process#	The process number of the process in which this session runs.
wait time	The intended wait time for this session.

rdbms ipc message

The background processes (LGWR, DBWR, LCK0) use this event to indicate that they are idle and are waiting for the foreground processes to send them an IPC message to do some work.

Wait Time: Up to 3 seconds. The parameter *timeout* shows the true sleep time.

Parameter:

timeout	The amount of time that the session waits for an IPC message.

rdbms ipc message block

This event indicates that all message blocks are in use and that the session had to wait for a message block to become available.

Wait Time: Wait up to 60 seconds.

Parameters: None

rdbms ipc reply

This event is used to wait for a reply from one of the background processes.

Wait Time: The wait time is specified by the user and is indicated by the parameter *timeout*.

Parameters:

from_process	The background process for which the session is waiting. The wait is for a reply to an IPC message sent by the session.
timeout	The amount of time in seconds that this process will wait for a reply.

redo wait

Defined but not used by the code.

row cache lock

The session is trying to get a data dictionary lock.

Wait Time: Wait up to 60 seconds.

Parameters:

cache id	The CACHE# column value in the V$ROWCACHE view.
mode	See "mode" on page A-10.
request	The pipe timer set by the user.

scginq AST call

Called by the session to find the highest lock mode that is held on a resource.

Wait Time: Wait up to 0.2 seconds, but the wait will continue until the NULL mode Acquisition AST has fired.

Parameters: None

single-task message

When running single task, this event indicates that the session waits for the client side of the executable.

Wait Time: Total elapsed time that this session spent in the user application.

Parameters: None

smon timer

This is the main idle event for SMON. SMON will be waiting on this event most of the time until it times out or is posted by another process.

Wait Time: 5 minutes (300 seconds)

Parameters:

sleeptime	The amount of time that SMON tries to wait on this event in seconds.
failed	The number of times SMON was posted when there some kind of error.

SQL*Net break/reset to client

The server sends a break or reset message to the client. The session running on the server waits for a reply from the client.

Wait Time: The actual time it takes for the break or reset message to return from the client.

Parameters:

driver id	See "driver id" on page A-10.
break?	See "break?" on page A-9.

SQL*Net break/reset to dblink

Same as **SQL*Net break/reset to client**, but in this case, the break / reset message is sent to another server process over a database link.

Wait Time: The actual time it takes for the break or reset message to return from the other server process.

Parameters:

driver id	See "driver id" on page A-10.
break?	See "break?" on page A-9.

SQL*Net message from client

The server process (foreground process) waits for a message from the client process to arrive.

Wait Time: The time it took for a message to arrive from the client since the last message was sent to the client.

Parameters:

driver id	See "driver id" on page A-10.
#bytes	The number of bytes received by the server (foreground process) from the client.

SQL*Net message from dblink

The session waits while the server process (foreground process) receives messages over a database link from another server process.

Wait Time: The time it took for a message to arrive from another server (foreground process) since a message was sent to the other foreground process.

Parameters:

driver id	See "driver id" on page A-10.
#bytes	The number of bytes received by the server (foreground process) from another foreground process over a database link.

SQL*Net message to client

The server (foreground process) is sending a message to the client.

Wait Time: The actual time the **send** takes.

Parameters:

driver id	See "driver id" on page A-10.
#bytes	The number of bytes sent by the server process to the client.

SQL*Net message to dblink

The server process (foreground process) is sending a message over a database link to another server process.

Wait Time: The actual time the **send** takes.

Parameters:

driver id	See "driver id" on page A-10.
#bytes	The number of bytes sent by the server process to another server process over a database link.

SQL*Net more data from client

The server is performing another send to the client. The previous operation was also a send to the client.

Wait Time: The time waited depends on the time it took to receive the data (including the waiting time).

Parameters:

driver id	See "driver id" on page A-10.
#bytes	The number of bytes received from the client.

SQL*Net more data from dblink

The foreground process is expecting more data from a data base link.

Wait Time: The total time it takes to read the data from the database link (including the waiting time for the data to arrive).

Parameters:

driver id	See "driver id" on page A-10.
#bytes	The number of bytes received.

SQL*Net more data to client

The server process is sending more data / messages to the client. The previous operation to the client was also a **send**.

Wait Time: The actual time it took for the **send** to complete.

Parameters:

driver id	See "driver id" on page A-10.

#bytes The number of bytes that are being sent to the client.

SQL*Net more data to dblink

The event indicates that the server is sending data over a database link again. The previous operation over this database link was also a **send**.

Wait Time: The actual time it takes to send the data to the other server.

Parameters:

driver id See "driver id" on page A-10.

#bytes The number of bytes that are sent over the database link to the other server process.

switch logfile command

The session waits on the user command `SWITCH LOGFILE` to complete.

Wait Time: 5 seconds

Parameters: None

timer in sksawat

The session waits for the Archiver (ARCH) asynchronous I/O to complete.

Wait Time: 0.01 seconds

Parameters: None

transaction

Wait for a blocking transaction to be rolled back. Continue waiting until the transaction has been rolled back.

Wait Time: 1 second

Parameters:

undo seg# The rollback segment ID.

slot# The slot ID inside the rollback segment.

wrap# The sequence number that is incremented for each transaction.

count The number of times that the session has waited on this transaction.

unbound tx

The session waits to see if there are any transactions that have been started but do not have a Rollback Segment associated with them.

Wait Time: 1 second

Parameters: None

undo segment extension

The undo segment is being extended or shrunk. The session must wait until the operation on the undo segment has finished.

Wait Time: 0.01 seconds

Parameter:

segment# The ID of the rollback segment that is being extended or shrunk.

undo segment recovery

PMON is rolling back a dead transaction. The wait continues until rollback finishes.

Wait Time: 3 seconds

Parameters:

segment# The ID of the rollback segment that contains the transaction that is being rolled back.

tx flags The transaction flags (options) set for the transaction that is being rolled back.

undo segment tx slot

Wait for a transaction slot to become available within the selected rollback segment. Continue waiting until the slot is available.

Wait Time: 1 second

Parameters:

segment# The ID of the rollback segment that contains the transaction that is being rolled back.

virtual circuit status

The session waits for a virtual circuit to return a message type indicated by *status*.

Wait Time: 30 seconds

Parameters:

circuit# Indicates the virtual circuit# being waited on.

status Indicates what the session is waiting for.

WMON goes to sleep

WMON is the UNIX-specific Wait Monitor, that can be used to reduce the number of system calls related to setting timers for posting or waiting in Oracle. You need to set an initialization parameter that enables the WMON process.

Wait Time: Depends on the next timeout.

Parameters: None

write complete waits

The session waits for a buffer to be written. The write is caused by normal aging or by a cross-instance call.

Wait Time: 1 second

Parameters:

file# The rollback segment id that contains the transaction that is being rolled back.

block# The transaction flags (options) set for the transaction that is being rolled back.

id Identifies the reason for waiting.

writes stopped by instance recovery or database suspension

The session is blocked until the instance that started Instance Recovery is finished.

Wait Time: 5 seconds

Parameters:

bythread# The rollback segment id that contains the transaction that is being rolled back.

ourthread# The current instance thread number.

B

Oracle Enqueue Names

This appendix lists Oracle enqueues. **Enqueues** are shared memory structures (locks) that serialize access to database resources. They can be associated with a session or transaction. Enqueue names are displayed in the LOCK_TYPE column of the DBA_LOCKS and DBA_LOCK_INTERNAL data dictionary views.

A **resource** uniquely identifies an object that can be locked by different sessions within an instance (local resource) or between instances (global resource). Each session that tries to lock the resource will have an **enqueue** on the resource.

Note: The names of enqueues and their definitions may change from release to release.

See Also:

- *Oracle8i Application Developer's Guide - Advanced Queuing* for more information on enqueues.
- "DBA_LOCK_INTERNAL" on page 2-114 and "DBA_LOCKS" on page 2-114.

The Oracle enqueues are:

- BL, Buffer Cache Management
- CF, Controlfile Transaction
- CI, Cross-instance Call Invocation
- CU, Bind Enqueue
- DF, Datafile

- DL, Direct Loader Index Creation
- DM, Database Mount
- DR, Distributed Recovery
- DX, Distributed TX
- FS, File Set
- IN, Instance Number
- IR, Instance Recovery
- IS, Instance State
- IV, Library Cache Invalidation
- JQ, Job Queue
- KK, Redo Log "Kick"
- L[A-P], Library Cache Lock
- MR, Media Recovery
- N[A-Z], Library Cache Pin
- PF, Password File
- PI, Parallel Slaves
- PR, Process Startup
- PS, Parallel Slave Synchronization
- Q[A-Z], Row Cache
- RT, Redo Thread
- SC, System Commit Number
- SM, SMON
- SQ, Sequence Number Enqueue
- SR, Synchronized Replication
- SS, Sort Segment
- ST, Space Management Transaction
- SV, Sequence Number Value
- TA, Transaction Recovery

- TM, DML Enqueue
- TS, Temporary Segment (also TableSpace)
- TT, Temporary Table
- TX, Transaction
- UL, User-defined Locks
- UN, User Name
- US, Undo Segment, Serialization
- WL, Being Written Redo Log
- XA, Instance Attribute Lock
- XI, Instance Registration Lock

C

Statistics Descriptions

This appendix describes the statistics stored in the V$SESSTAT and V$SYSSTAT dynamic performance tables. These statistics are useful in identifying and correcting performance problems.

Displaying Statistics

The V$SESSTAT view displays statistics on a per-session basis and is valid only for the session currently connected. When a session disconnects, all statistics for the session are updated in V$SYSSTAT. The values for the statistics are cleared until the next session uses them.

The V$STATNAME view contains all of the statistics for an Oracle release.

Many of these statistics are tied to the internal implementation of Oracle and therefore are subject to change or deletion without notice, even between patch releases. Application developers should be aware of this and write their code to tolerate missing or extra statistics.

> **See Also:** "V$SESSTAT" on page 3-98, "V$STATNAME" on page 3-109, and "V$SYSSTAT" on page 3-111 for more information on these views.

Statistics Descriptions

This section describes the statistics stored in the V$SESSTAT and V$SYSSTAT views. The statistics are listed here in alphabetical order.

The CLASS column contains a number representing one or more statistics class. The following class numbers are additive:

- 1, User

- 2, Redo
- 4, Enqueue
- 8, Cache
- 16, OS
- 32, Parallel Server
- 64, SQL
- 128, Debug

For example, a class value of 72 represents a statistic that relates to SQL statements and caching.

Some statistics are populated only if the TIMED_STATISTICS initialization parameter is set to TRUE. Those statistics are flagged in the right-hand column.

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
background checkpoints completed	8	The number of checkpoints completed by the background process. This statistic is incremented when the background process successfully advances the thread checkpoint.	
background checkpoints started	8	The number of checkpoints started by the background process. This statistics can be larger than "background checkpoints completed" if a new checkpoint overrides an incomplete checkpoint or if a checkpoint is currently under way. This statistic includes only checkpoints of the redo thread. It does not include: ■ Individual file checkpoints for operations such as offline or begin backup ■ Foreground (user-requested) checkpoints (for example, performed by ALTER SYSTEM CHECKPOINT LOCAL statements)	
branch node splits	128	Number of times an index branch block was split because of the insertion of an additional value.	
buffer is not pinned count	72	Number of times a buffer was free when visited. This statistic is useful only for internal debugging purposes.	
buffer is pinned count	72	Number of times a buffer was pinned when visited. This statistic is useful only for internal debugging purposes.	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
bytes received via SQL*Net from client	1	Total number of bytes received from the client over Net8.	
bytes received via SQL*Net from dblink	1	Total number of bytes received from a database link over Net8.	
bytes sent via SQL*Net to client	1	Total number of bytes sent to the client from the foreground process(es).	
bytes sent via SQL*Net to dblink	1	Total number of bytes sent over a database link.	
Cached Commit SCN referenced	128	This statistic is useful only for internal debugging purposes.	
calls to get snapshot scn: kcmgss	32	Number of times a snapshot system change number (SCN) was allocated. The SCN is allocated at the start of a transaction.	
calls to kcmgas	128	Number of calls to routine kcmgas to get a new SCN.	
calls to kcmgcs	128	Number of calls to routine kcmgcs to get a current SCN.	
calls to kcmgrs	128	Number of calls to routine kcsgrs to get a recent SCN.	
change write time	8	Elapsed redo write time for changes made to CURRENT blocks in 10s of milliseconds. This statistic is populated only if the `TIME_STATISTICS` parameter is set to `TRUE`.	
cleanouts and rollbacks - consistent read gets	128	Number of consistent gets that require both block rollbacks and block cleanouts. See "consistent gets" below.	
cleanouts only - consistent read gets	128	Number of consistent gets that require only block cleanouts, no rollbacks. See "consistent gets" below.	
cluster key scan block gets	64	Number of blocks obtained in a cluster scan.	
cluster key scans	64	Number of cluster scans that were started.	
cold recycle reads	8	Number of buffers that were read through the least recently used end of the recycle cache with fast aging strategy.	
commit cleanout failures: block lost	8	Number of times Oracle attempted a cleanout at commit but could not find the correct block due to forced write, replacement, or switch CURRENT.	
commit cleanout failures: buffer being written	8	Number of times Oracle attempted a cleanout at commit, but the buffer was currently being written.	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
commit cleanout failures: callback failure	8	Number of times the cleanout callback function returns FALSE.	
commit cleanout failures: cannot pin	8	Total number of times a commit cleanout was performed but failed because the block could not be pinned.	
commit cleanout failures: hot backup in progress	8	Number of times Oracle attempted block cleanout at commit during hot backup. The image of the block needs to be logged before the buffer can be made dirty.	
commit cleanout failures: write disabled	8	Number of times a cleanout block at commit was performed but the writes to the database had been temporarily disabled.	
commit cleanouts	8	Total number of times the cleanout block at commit function was performed.	
commit cleanouts successfully completed	8	Number of times the cleanout block at commit function completed successfully.	
Commit SCN cached	128	Number of times the system change number of a commit operation was cached	
consistent changes	8	Number of times a user process has applied rollback entries to perform a consistent read on the block. Work loads that produce a great deal of consistent changes can consume a great deal of resources. The value of this statistic should be small in relation to the "consistent gets" statistic.	
consistent gets	8	Number of times a consistent read was requested for a block. See also the "consistent changes" and "session logical reads" statistics.	
CPU used by this session	1	Amount of CPU time (in 10s of milliseconds) used by a session from the time a user call starts until it ends. If a user call completes within 10 milliseconds, the start and end user-call time are the same for purposes of this statistics, and 0 milliseconds are added. A similar problem can exist in the reporting by the operating system, especially on systems that suffer from many context switches.	✓
CPU used when call started	128	The CPU time used when the call is started. See also "CPU used by this session", above.	✓

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
CR blocks created	8	Number of CURRENT blocks cloned to create CR (consistent read) blocks. The most common reason for cloning is that the buffer is held in a incompatible mode.	
current blocks converted for CR	8	Number CURRENT blocks converted to CR state	
cursor authentications	128	Number of privilege checks conducted during execution of an operation	
data blocks consistent reads - undo records applied	128	Number of undo records applied to data blocks that have been rolled back for consistent read purposes.	
db block changes	8	Closely related to "consistent changes", this statistic counts the total number of changes that were part of an update or delete operation that were made to all blocks in the SGA. Such changes generate redo log entries and hence become permanent changes to the database if the transaction is committed. This approximates total database work. It statistic indicates the rate at which buffers are being dirtied (on a per-transaction or per-second basis, for example).	
db block gets	8	Number of times a CURRENT block was requested. See also "consistent gets", above.	
DBWR buffers scanned	8	Total number of dirty and clean buffers Oracle looks at when scanning LRU sets for dirty buffers to clean. Divide by "DBWR lru scans" to find the average number of buffers scanned.	
DBWR checkpoint buffers written	8	Number of buffers that were written for checkpoints.	
DBWR checkpoints	8	Number of times the DBWR was asked to scan the cache and write all blocks marked for a checkpoint or the end of recovery. This statistic is always larger than "background checkpoints completed".	
DBWR cross instance writes	40	Oracle Parallel Server only: Number of blocks written to satisfy a lock request from another instance	
DBWR free buffers found	8	Number of clean buffers that DBWR found when it was requested to make free buffers. Divide by "DBWR make free requests" to find the average number of reusable buffers at the end of each LRU.	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
DBWR lru scans	8	Number of times that DBWR scans the LRU queue looking for buffers to write. This count includes scans to fill a batch being written for another purpose (such as a checkpoint). This statistic is always greater than or equal to "DBWR make free requests".	
DBWR make free requests	8	Number of requests to DBWR to make some free buffers for the LRU.	
DBWR revisited being-written buffer	8	Number of times that DBWR tried to save a buffer for writing and found that it was already in the write batch. This statistic measures the amount of "useless" work that DBWR had to do in trying to fill the batch. Many sources contribute to a write batch. If the same buffer from different sources is considered for adding to the write batch, then all but the first attempt will be "useless" because the buffer is already marked as being written.	
DBWR summed scan depth	8	The current **scan depth** (number of buffers examined by DBWR) is added to this statistic every time DBWR scans the LRU for dirty buffers. Divide by "DBWR lru scans" to find the average scan depth.	
DBWR transaction table writes	8	Number of rollback segment headers written by DBWR. This statistic indicates how many "hot" buffers were written, causing a user process to wait while the write completed.	
DBWR undo block writes	8	Number of rollback segment blocks written by DBWR.	
DDL statements parallelized	32	Number of DDL statements that were executed in parallel.	
deferred (CURRENT) block cleanout applications	128	Number of times cleanout records are deferred, piggyback with changes, always current get.	
DFO trees parallelized	32	Number of times a serial execution plan was converted to a parallel plan.	
dirty buffers inspected	8	Number of dirty buffers found by the user process while the it is looking for a buffer to reuse.	
DML statements parallelized	32	Number of DML statements that were executed in parallel.	
enqueue conversions	4	Total number of conversions of the state of table or row lock	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
enqueue deadlocks	4	Total number of deadlocks between table or row locks in different sessions.	
enqueue releases	4	Total number of table or row locks released	
enqueue requests	4	Total number of table or row locks acquired	
enqueue timeouts	4	Total number of table and row locks (acquired and converted) that timed out before they could complete	
enqueue waits	4	Total number of waits that occurred during an enqueue convert or get because the enqueue get was deferred	
exchange deadlocks	8	Number of times that a process detected a potential deadlock when exchanging two buffers and raised an internal, restartable error. Index scans are the only operations that perform exchanges.	
execute count	64	Total number of calls (user and recursive) that executed SQL statements.	
free buffer inspected	8	Number of buffers skipped over from the end of an LRU queue in order to find a reusable buffer. The difference between this statistic and "dirty buffers inspected" is the number of buffers that could not be used because they had a user, a waiter, or were being read or written, or because they were busy or needed to be written after rapid aging out.	
free buffer requested	8	Number of times a reusable buffer or a free buffer was requested to create or load a block.	
global cache blocks corrupt	40	OPS only: Number of blocks that encountered a corruption or checksum failure during interconnect.	
global cache convert time	40	OPS only: Total time elapsed during lock converts.	
global cache convert timeouts	40	Number of times lock converts in the global cache timed out.	
global cache converts	40	Number of lock converts in the global cache.	
global cache cr block receive time	40	Total amount of time foreground processes waited for a CR block to be sent through the interconnect. This statistic divided by "global cache cr blocks received" = time waited per block	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
global cache cr block serve time	40	Total amount of time the BSP process took to construct consistent-read (CR) blocks. This statistic divided by "global cache cr blocks served" = construction time per CR block	
global cache cr blocks received	40	Total number of blocks received.	
global cache cr blocks served	40	Total number of blocks constructed by the BSP process.	
global cache cr requests blocked	40	Number of times foreground attempt to request a cr block and failed.	
global cache cr timeouts	40	Number of times a foreground process requested a consistent-read (CR) block when the request timed out.	
global cache defers	40	Number of times a lock was requested and the holder of the lock deferred the release.	
global cache freelist waits	40	System configured with fewer lock elements than buffers. Number of times foreground has to wait for a lock element.	
global cache get time	40	Total time spent waiting. this divided by global cache gets = time waited per request.	
global cache gets	40	Number of locks acquired	
global cache cr block send time	40	Total time spent by the BSP process in sending constructed consistent-read (CR) blocks. This statistic divided by "global cache cr blocks served" = send time per CR block.	
global cache cr block log flushes	40	Number of log flushes of the consistent-read block	
global cache cr block log flush time	40	Total time spent by the BSP process in log flushes after sending a constructed consistent-read (CR) block. This statistic divided by "global cache cr blocks served" = log flush time per CR block.	
global cache prepare failures	40	Number of times a failure occurred during preparation for interconnect transfer.	
global lock async converts	32	Total number of asynchronous global lock converts.	
global lock async gets	32	Total number of asynchronous global lock gets.	
global lock convert time	32	Total elapsed time in 10s of milliseconds of all synchronous (non-asynchronous) global lock converts.	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
global lock get time	32	Total elapsed time in 10s of milliseconds of all synchronous (non-asynchronous) global lock gets.	
global lock releases	32	Total number of synchronous global lock releases.	
global lock sync converts	32	Total number of synchronous global lock converts.	
global lock sync gets	32	Total number of synchronous global lock gets.	
hot buffers moved to head of LRU	8	When a hot buffer reaches the tail of its replacement list, Oracle moves it back to the head of the list to keep it from being reused. This statistic counts such moves.	
immediate (CR) block cleanout applications	128	Number of times cleanout records are applied immediately during consistent-read requests	
immediate (CURRENT) block cleanout applications	128	Number of times cleanout records are applied immediately during current gets. Compare this statistic with "deferred (CURRENT) block cleanout applications", above.	
index fast full scans (direct read)	64	Number of fast full scans initiated using direct read.	
index fast full scans (full)	64	Number of fast full scans initiated for full segments.	
index fast full scans (rowid ranges)	64	Number of fast full scans initiated with rowid endpoints specified.	
instance recovery database freeze count	32	Number of times the database is frozen during instance recovery.	
kcmccs called get current scn	32	Number of times the kernel got the CURRENT SCN when there was a need to casually confirm the SCN.	
kcmgss read scn without going to DLM	32	Number of times the kernel got a snapshot SCN without going to the distributed lock manager (DLM)	
kcmgss waited for batching	32	Number of times a database process is blocked waiting for a snapshot SCN.	
leaf node splits	128	Number of times an index leaf node was split because of the insertion of an additional value.	
logons cumulative	1	Total number of logons since the instance started. This statistic is useful only in V$SYSSTAT. It gives an instance overview of all processes that logged on.	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
logons current	1	Total number of current logons. This statistic is useful only in V$SYSSTAT.	
messages received	128	Number of messages sent and received between background processes	
messages sent	128		
native hash arithmetic execute	64	Number of hash operations performed using native arithmetic rather than Oracle NUMBERs.	
native hash arithmetic fail	64	Number of has operations performed using native arithmetic that failed, requiring the hash operation to be performed with Oracle NUMBERs.	
next scns gotten without going to DLM	32	Number of system change numbers obtained without going to the distributed lock manager or server	
no buffer to keep pinned count	72	Number of times a visit to a buffer attempted, but the buffer was not found where expected. Like "buffer is not pinned count" and "buffer is pinned count", this statistic is useful only for internal debugging purposes.	
no work - consistent read gets	128	Number consistent gets that require neither block cleanouts nor rollbacks. See "consistent gets" above.	
opened cursors cumulative	1	In V$SYSSTAT: Total number of cursors opened since the instance started. In V$SESSTAT: Total number of cursors opened since the start of the session.	
opened cursors current	1	Total number of current open cursors.	
opens of replaced files	8	Total number of files that had to be reopened because they were no longer in the process file cache.	
opens requiring cache replacement	8	Total number of file opens that caused a current file in the process file cache to be closed.	
OS All other sleep time	16	Time spent sleeping for reasons other than misses in the data segment (see "OS Data page fault sleep time"), kernel page faults (see "OS Kernel page fault sleep time"), misses in the text segment (see "OS Text page fault sleep time"), or waiting for an OS locking object (see "OS User lock wait sleep time"). An example of such a reason is expiration of quanta.	
OS Chars read and written	16	Number of bytes read and written	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
OS Data page fault sleep time	16	Time spent sleeping due to misses in the data segment	
OS Input blocks	16	Number of read I/Os	
OS Involuntary context switches	16	Number of context switches that were enforced by the operating system	
OS Kernel page fault sleep time	16	Time spent sleeping due to OS kernel page faults	
OS Major page faults	16	Number of page faults that resulted in I/O	
OS Messages received	16	Number of messages received	
OS Messages sent	16	Number of messages sent	
OS Minor page faults	16	Number of page faults that did not result in an actual I/O	
OS Other system trap CPU time	16	Total amount of time to process system traps (as distinct from system calls)	
OS Output blocks	16	Number of write I/Os	
OS Process heap size	16	Size of area in memory allocated by the process. Typically this represents memory obtained by way of malloc().	
OS Process stack size	16	Size of the process stack segment.	
OS Signals received	16	Number of signals received	
OS Swaps	16	Number of swap pages	
OS System call CPU time	16	Total amount of time spent executing in system mode	
OS System calls	16	Number of system calls	
OS Text page fault sleep time	16	Time spent sleeping due to misses in the text segment.	
OS User level CPU time	16	Total amount of time spent executing in user mode	
OS User lock wait sleep time	16	Total amount of time sleeping while waiting for an OS locking object	
OS Voluntary context switches	16	Number of voluntary context switches (for example, when a process gives up the CPU by a SLEEP() system call).	
OS Wait-cpu (latency) time	16	Time spent sleeping while waiting for a CPU to become available	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
Parallel operations downgraded 1 to 25 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers.	
Parallel operations downgraded 25 to 50 pct	32		
Parallel operations downgraded 50 to 75 pct	32		
Parallel operations downgraded 75 to 99 pct	32		
Parallel operations downgraded to serial	32	Number of times parallel execution was requested but execution was serial because of insufficient parallel execution servers.	
Parallel operations not downgraded	32	Number of times parallel execution was executed at the requested degree of parallelism.	
parse count (hard)	64	Total number of parse calls (real parses). A hard parse is a very expensive operation in terms of memory use, because it requires Oracle to allocate a workheap and other memory structures and then build a parse tree.	
parse count (total)	64	Total number of parse calls (hard and soft). A soft parse is a check on an object already in the shared pool, to verify that the permissions on the underlying object have not changed.	
parse time cpu	64	Total CPU time used for parsing (hard and soft) in 10s of milliseconds.	✓
parse time elapsed	64	Total elapsed time for parsing, in 10s of milliseconds. Subtract "parse time cpu" from the this statistic to determine the total waiting time for parse resources.	✓
physical reads	8	Total number of data blocks read from disk. This number equals the value of "physical reads direct" plus all reads into buffer cache.	
physical reads direct	8	Number of reads directly from disk, bypassing the buffer cache. For example, in high bandwidth, data-intensive operations such as parallel query, reads of disk blocks bypass the buffer cache to maximize transfer rates and to prevent the premature aging of shared data blocks resident in the buffer cache.	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
physical writes	8	Total number of data blocks written to disk. This number equals the value of "physical writes direct" plus all writes from buffer cache.	
physical writes direct	8	Number of writes directly to disk, bypassing the buffer cache (as in a direct load operation).	
physical writes non checkpoint	8	Number of times a buffer is written for reasons other than advancement of the checkpoint. Used as a metric for determining the I/O overhead imposed by setting the FAST_START_IO_TARGET parameter to limit recovery I/Os. Essentially this statistic measures the number of writes that would have occurred had there been no checkpointing. Subtracting this value from "physical writes" gives the extra I/O for checkpointing.	
pinned buffers inspected	8	Number of times a user process, when scanning the tail of the replacement list looking for a buffer to reuse, encountered a cold buffer that was pinned or had a waiter that was about to pin it. This occurrence is uncommon, because a cold buffer should not be pinned very often.	
prefetched blocks	8	Number of contiguous and noncontiguous blocks that were prefetched.	
prefetched blocks aged out before use	8	Number of contiguous and noncontiguous blocks that were prefetched but aged out before use.	
process last non-idle time	128	The last time this process executed.	✓
PX local messages recv'd	32	Number of local messages received for parallel execution within the instance local to the current session.	
PX local messages sent	32	Number of local messages sent for parallel execution within the instance local to the current session.	
PX remote messages recv'd	32	Number of remote messages received for parallel execution within the instance local to the current session.	
PX remote messages sent	32	Number of remote messages sent for parallel execution within the instance local to the current session.	
queries parallelized	32	Number of SELECT statements executed in parallel.	
recovery array read time	8	Elapsed time of I/O during recovery.	
recovery array reads	8	Number of reads performed during recovery.	
recovery blocks read	8	Number of blocks read during recovery.	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
recursive calls	1	Number of recursive calls generated at both the user and system level. Oracle maintains tables used for internal processing. When Oracle needs to make a change to these tables, it internally generates an internal SQL statement, which in turn generates a recursive call.	
recursive cpu usage	1	Total CPU time used by non-user calls (recursive calls). Subtract this value from "CPU used by this session" to determine how much CPU time was used by the user calls.	
redo blocks written	2	Total number of redo blocks written. This statistic divided by "redo writes" equals number of blocks per write.	
redo buffer allocation retries	2	Total number of retries necessary to allocate space in the redo buffer. Retries are needed either because the redo writer has fallen behind or because an event such as a log switch is occurring.	
redo entries	2	Number of times a redo entry is copied into the redo log buffer.	
redo log space requests	2	Number of times the active log file is full and Oracle must wait for disk space to be allocated for the redo log entries. Such space is created by performing a log switch. Log files that are small in relation to the size of the SGA or the commit rate of the work load can cause problems. When the log switch occurs, Oracle must ensure that all committed dirty buffers are written to disk before switching to a new log file. If you have a large SGA full of dirty buffers and small redo log files, a log switch must wait for DBWR to write dirty buffers to disk before continuing. Also examine the **log file space** and **log file space switch** wait events in V$SESSION_WAIT.	
redo log space wait time	2	Total elapsed waiting time for "redo log space requests" in 10s of milliseconds.	✓
redo log switch interrupts	2	Number of times that another instance asked this instance to advance to the next log file.	
redo ordering marks	2	Number of times that a system change number was allocated to force a redo record to have an higher SCN than a record generated in another thread using the same block.	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
redo size	2	Total amount of redo generated in bytes.	
redo synch time	8	Elapsed time of all "redo synch writes" calls in 10s of milliseconds.	✓
redo synch writes	8	Number of times a change being applied to the log buffer must be written out to disk due to a commit. The log buffer is a circular buffer that LGWR periodically flushes. Usually, redo that is generated and copied into the log buffer need not be flushed out to disk immediately.	
redo wastage	2	Number of bytes wasted because redo blocks needed to be written before they are completely full. Early writing may be needed to commit transactions, to be able to write a database buffer, or to switch logs.	
redo write time	2	Total elapsed time of the write from the redo log buffer to the current redo log file in 10s of milliseconds.	✓
redo writer latching time	2	Elapsed time in 10s of milliseconds needed by LWGR to obtain and release each copy latch.	✓
redo writes	2	Total number of writes by LGWR to the redo log files. "redo blocks written" divided by this statistic equals the number of blocks per write.	
remote instance undo block writes	40	Number of times this instance wrote a rollback segment so that another instance could read it.	
remote instance undo header writes	40	Number of times this instance wrote a undo header block so that another instance could read it.	
rollback changes - undo records applied	128	Number of undo records applied to user-requested rollback changes (not consistent-read rollbacks).	
rollbacks only - consistent read gets	128	Number of consistent gets that require only block rollbacks, no block cleanouts. See "consistent gets" above.	
rows fetched via callback	64	Rows fetched via callback. This statistics is useful primarily for internal debugging purposes.	
serializable aborts	1	Number of times a SQL statement in a serializable isolation level had to abort.	
session connect time	1	The connect time for the session in 10s of milliseconds. This value is useful only in V$SESSTAT. It is the wall clock time since the logon to this session occurred.	✓

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
session cursor cache count	64	Total number of cursors cached. This statistic is incremented only if SESSION_CACHED_CURSORS > 0. This statistic is the most useful in V$SESSTAT. If the value for this statistic in V$SESSTAT is close to the setting of the SESSION_CACHED_CURSORS parameter, the value of the parameter should be increased.	
session cursor cache hits	64	Number of hits in the session cursor cache. A hit means that the SQL statement did not have to be reparsed. Subtract this statistic from "parse count (total)" to determine the real number of parses that occurred.	
session logical reads	1	The sum of "db block gets" plus "consistent gets", described above.	
session pga memory	1	Current PGA size for the session. This statistic is useful only in V$SESSTAT; it has no meaning in V$SYSSTAT.	
session pga memory max	1	Peak PGA size for the session. This statistic is useful only in V$SESSTAT; it has no meaning in V$SYSSTAT.	
session stored procedure space	1	Amount of memory this session is using for stored procedures.	
session uga memory	1	Current UGA size for the session. This statistic is useful only in V$SESSTAT; it has no meaning in V$SYSSTAT.	
session uga memory max	1	Peak UGA size for a session. This statistic is useful only in V$SESSTAT; it has no meaning in V$SYSSTAT.	
sorts (disk)	64	Number of sort operations that required at least one disk write. Sorts that require I/O to disk are quite resource intensive. Try increasing the size of the initialization parameter SORT_AREA_SIZE. For more information, see "SORT_ AREA_SIZE" on page 1-112.	
sorts (memory)	64	Number of sort operations that were performed completely in memory and did not require any disk writes. You cannot do much better than memory sorts, except maybe no sorts at all. Sorting is usually caused by selection criteria specifications within table join SQL operations.	
sorts (rows)	64	Total number of rows sorted.	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
SQL*Net roundtrips to/from client	1	Total number of Net8 messages sent to and received from the client.	
SQL*Net roundtrips to/from dblink	1	Total number of Net8 messages sent over and received from a database link.	
summed dirty queue length	8	The sum of the dirty LRU queue length after every write request. Divide by **write requests** to get the average queue length after write completion.	
switch current to new buffer	8	Number of times the CURRENT block moved to a different buffer, leaving a CR block in the original buffer.	
table fetch by rowid	64	Number of rows that are fetched using a ROWID (usually recovered from an index). This occurrence of table scans usually indicates either non-optimal queries or tables without indexes. Therefore, this statistic should increase as you optimize queries and provide indexes in the application.	
table fetch continued row	64	Number of times a chained or migrated row is encountered during a fetch. Retrieving rows that span more than one block increases the logical I/O by a factor that corresponds to the number of blocks than need to be accessed. Exporting and re-importing may eliminate this problem. Evaluate the settings for the storage parameters PCTFREE and PCTUSED. This problem cannot be fixed if rows are larger than database blocks (for example, if the LONG datatype is used and the rows are extremely large).	
table scan blocks gotten	64	During scanning operations, each row is retrieved sequentially by Oracle. This statistic counts the number of blocks encountered during the scan. This statistic tells you the number of database blocks that you had to get from the buffer cache for the purpose of scanning. Compare this value with the value of "consistent gets" to determine how much of the consistent read activity can be attributed to scanning.	
table scan rows gotten	64	Number of rows that are processed during scanning operations.	
table scans (cache partitions)	64	Number of range scans performed on tables that have the CACHE option enabled.	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
table scans (direct read)	64	Number of table scans performed with direct read (bypassing the buffer cache).	
table scans (long tables)	64	Long (or conversely short) tables can be defined as tables that do not meet the short table criteria as described in table scans (short tables) below.	
table scans (rowid ranges)	64	During parallel query, the number of table scans conducted with specified ROWID ranges	
table scans (short tables)	64	Long (or conversely short) tables can be defined by optimizer hints coming down into the row source access layer of Oracle. The table must have the CACHE option set.	
total file opens	8	Total number of file opens performed by the instance. Each process needs a number of files (control file, log file, database file) in order to work against the database.	
transaction lock background get time	128	This statistic is useful only for internal debugging purposes.	
transaction lock background gets	128	This statistic is useful only for internal debugging purposes.	
transaction lock foreground requests	128	This statistic is useful only for internal debugging purposes.	
transaction lock foreground wait time	128	This statistic is useful only for internal debugging purposes.	
transaction rollbacks	128	Number of transactions being successfully rolled back.	
transaction tables consistent read rollbacks	128	Number of times rollback segment headers are rolled back to create consistent read blocks.	
transaction tables consistent reads - undo records applied	128	Number of undo records applied to transaction tables that have been rolled back for consistent read purposes.	
Unnecessary process cleanup for SCN batching	32	Total number of times that the process cleanup was performed unnecessarily because the session or process did not get the next batched SCN. The next batched SCN went to another session instead.	

Table C–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
user calls	1	Number of user calls such as login, parse, fetch, or execute. When determining activity, the ratio of user calls to RPI calls, give you an indication of how much internal work gets generated as a result of the type of requests the user is sending to Oracle.	
user commits	1	Number of user commits. When a user commits a transaction, the redo generated that reflects the changes made to database blocks must be written to disk. Commits often represent the closest thing to a user transaction rate.	
user rollbacks	1	Number of times users manually issue the ROLLBACK statement or an error occurs during a user's transactions.	
write clones created in background	8	Number of times a a background or foreground process clones a CURRENT buffer that is being written. The clone becomes the new, accessible CURRENT buffer, leaving the original buffer (now the clone) to complete writing.	
write clones created in foreground	8		

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W