222

DBMS_WORKLOAD_REPOSITORY

The `DBMS_WORKLOAD_REPOSITORY` package lets you manage the Automatic Workload Repository (AWR) by performing operations, such as, managing snapshots and baselines.

The chapter contains the following topics:

- Examples
- Data Structures
- Summary of DBMS_WORKLOAD_REPOSITORY Subprograms

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about the Automatic Workload Repository

DBMS_WORKLOAD_REPOSITORY Examples

This example shows how to generate an AWR text report with the `DBMS_WORKLOAD_REPOSITORY` package for database ID 1557521192, instance ID 1, snapshot IDs 5390 and 5391, and with default options.

```
-- make sure to set line size appropriately
-- set linesize 152
SELECT output FROM TABLE(
   DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_TEXT(
     1557521192,  1, 5390, 5392) ) ;
```

You can call the `DBMS_WORKLOAD_REPOSITORY` packaged functions directly as in the example, but Oracle recommends you use the corresponding supplied SQL script (`awrrpt.sql` in this case) for the packaged function, which prompts the user for required information.

DBMS_WORKLOAD_REPOSITORY Data Structures

The `DBMS_WORKLOAD_REPOSITORY` package defines an object and associated table types.

OBJECT Types

- AWR_BASELINE_METRIC_TYPE Object Type

TABLE Types

- AWR_BASELINE_METRIC_TYPE_TABLE Table Type
- AWRRPT_INSTANCE_LIST_TYPE Table Type

DBMS_WORKLOAD_REPOSITORY AWR_BASELINE_METRIC_TYPE Object Type

This type shows the values of the metrics corresponding to a baseline.

Syntax

```
TYPE awr_baseline_metric_type AS OBJECT (
   baseline_name      VARCHAR2(64),
   dbid               NUMBER NOT NULL,
   instance_number    NUMBER NOT NULL,
   beg_time           DATE NOT NULL,
   end_time           DATE NOT NULL,
   metric_id          NUMBER NOT NULL,
   metric_name        VARCHAR2(64) NOT NULL,
   metric_unit        VARCHAR2(64) NOT NULL,
   num_interval       NUMBER NOT NULL,
   interval_size      NUMBER NOT NULL,
   average            NUMBER NOT NULL,
   minimum            NUMBER NOT NULL,
   maximum            NUMBER NOT NULL);
```

Fields

Table 222-1 AWR_BASELINE_METRIC_TYPE Fields

Field	Description
baseline_name	Name of the Baseline
dbid	Database ID for the snapshot
instance_number	Instance number for the snapshot
beg_time	Begin time of the interval
end_time	End time of the interval
metric_id	Metric ID
metric_name	Metric name
metric_unit	Unit of measurement
num_interval	Number of intervals observed
interval_size	Interval size (in hundredths of a second)
average	Average over the period
minimum	Minimum value observed
maximum	Maximum value observed

DBMS_WORKLOAD_REPOSITORY AWR_BASELINE_METRIC_TYPE_TABLE Table Type

This type is used by the SELECT_BASELINE_METRIC Function.

Syntax

```
CREATE TYPE awr_baseline_metric_type_table AS TABLE OF awr_baseline_metric_type;
```

Related Topics

- SELECT_BASELINE_METRIC Function
  This table function shows the values of the metrics corresponding to a baseline for all the snapshots.

DBMS_WORKLOAD_REPOSITORY AWRRPT_INSTANCE_LIST_TYPE Table Type

This type provides an alternative to a comma-separated list.

Syntax

```
CREATE TYPE awrrpt_instance_list_type AS TABLE OF NUMBER;
```

Summary of DBMS_WORKLOAD_REPOSITORY Subprograms

This table lists the `DBMS_WORKLOAD_REPOSITORY` subprograms and briefly describes them.

Table 222-2 DBMS_WORKLOAD_REPOSITORY Package Subprograms

Subprogram	Description
ADD_COLORED_SQL Procedure	Adds a colored SQL ID
ASH_GLOBAL_REPORT_HTML Function	Displays a global or Oracle Real Application Clusters (Oracle RAC) ASH Spot report in HTML format.
ASH_GLOBAL_REPORT_TEXT Function	Displays a global or Oracle Real Application Clusters (Oracle RAC) ASH Spot report in Text format.
ASH_REPORT_ANALYTICS Function	Displays the ASH Analytics active report
ASH_REPORT_HTML Function	Displays the ASH report in HTML
ASH_REPORT_TEXT Function	Displays the ASH report in text
AWR_DIFF_REPORT_HTML Function	Displays the AWR Diff-Diff report in HTML
AWR_DIFF_REPORT_TEXT Function	Displays the AWR Diff-Diff report in text
AWR_GLOBAL_DIFF_REPORT_HTML Functions	Displays the Global AWR Compare Periods Report in HTML
AWR_GLOBAL_DIFF_REPORT_TEXT Functions	Displays the Global AWR Compare Periods Report in text
AWR_EXP Procedure	Extracts AWR data from the AWR schema into a dump file.
AWR_GLOBAL_REPORT_HTML Functions	Displays the Global AWR report in HTML
AWR_GLOBAL_REPORT_TEXT Functions	Displays the Global AWR report in text
AWR_IMP Procedure	Loads the AWR data from a dump file into the `SYS` schema.
AWR_REPORT_HTML Function	Displays the AWR report in HTML
AWR_REPORT_TEXT Function	Displays the AWR report in text
AWR_SET_REPORT_THRESHOLDS Procedure	Configures specified report thresholds, including the number of rows in the report

Table 222-2 (Cont.) DBMS_WORKLOAD_REPOSITORY Package Subprograms

Subprogram	Description
AWR_SQL_REPORT_HTML Function	Displays the AWR SQL Report in HTML format
AWR_SQL_REPORT_TEXT Function	Displays the AWR SQL Report in text format
CONTROL_RESTRICTED_SNAPSHOT Procedure	Controls the AWR snapshot creation for a database in the *restricted session* mode.
CREATE_BASELINE Functions & Procedures	Creates a single baseline
CREATE_BASELINE_TEMPLATE Procedures	Creates a baseline template
CREATE_REMOTE_SNAPSHOT Function and Procedure	Creates a manual *remote* snapshot immediately using the Remote Management Framework (RMF)
CREATE_SNAPSHOT Function and Procedure	Creates a manual *local* snapshot immediately
DROP_BASELINE Procedure	Drops a previously-defined baseline
DROP_BASELINE_TEMPLATE Procedure	Removes a baseline template that is no longer needed
DROP_SNAPSHOT_RANGE Procedure	Drops a range of snapshots
LOCAL_AWR_DBID Function	Returns the database identifier for the local AWR database
MODIFY_BASELINE_WINDOW_SIZE Procedure	Modifies the window size for the Default Moving Window Baseline
MODIFY_SNAPSHOT_SETTINGS Procedures	Modifies the snapshot settings
PURGE_SQL_DETAILS Procedure	Purges SQL details, specifically rows from `WRH$_SQLTEXT` and `WRH$_SQL_PLAN` that do not have corresponding rows (DBID, SQL_ID) in `WRH$_SQLSTAT`.
REGISTER_REMOTE_DATABASE Procedures	Registers a remote database in the AWR using the Remote Management Framework (RMF)
REMOVE_COLORED_SQL Procedure	Removes a colored SQL ID
RENAME_BASELINE Procedure	Renames a baseline
SELECT_BASELINE_DETAILS Function	Shows the values of the metrics corresponding to a baseline for a range of snapshots
SELECT_BASELINE_METRIC Function	Shows the values of the metrics corresponding to a baseline for all the snapshots
UNREGISTER_REMOTE_DATABASE Procedures	Removes all the statistics, metadata, and partitions of a remote database from the AWR using the Remote Management Framework (RMF)
UPDATE_DATAFILE_INFO Procedure	Updates the data file and tablespace information stored in the AWR with the current information in the database
UPDATE_OBJECT_INFO Procedure	Updates rows of `WRH$_SEG_STAT_OBJ` table that represent objects in the local database

ADD_COLORED_SQL Procedure

This procedure adds a colored SQL ID.

If an SQL ID is colored, it will be captured in every snapshot, independent of its level of activities (so that it does not have to be a `TOP SQL`). Capture occurs if the SQL is found in the cursor cache at snapshot time.To uncolor the SQL, invoke the REMOVE_COLORED_SQL Procedure.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.ADD_COLORED_SQL(
   sql_id          IN VARCHAR2,
   dbid            IN NUMBER DEFAULT NULL);
```

Parameters

Table 222-3 *ADD_COLORED_SQL Procedure Parameters*

Parameter	Description
`sql_id`	13-character external SQL ID
`dbid`	Optional DBID, defaults to Local `DBID`

ASH_GLOBAL_REPORT_HTML Function

This table function displays a global or Oracle Real Application Clusters (Oracle RAC) ASH Spot report in HTML format.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.ASH_GLOBAL_REPORT_HTML(
   l_dbid           IN NUMBER,
   l_inst_num       IN VARCHAR2((1023),
   l_btime          IN DATE,
   l_etime          IN DATE,
   l_options        IN NUMBER    DEFAULT 0,
   l_slot_width     IN NUMBER    DEFAULT 0,
   l_sid            IN NUMBER    DEFAULT NULL,
   l_sql_id         IN VARCHAR2  DEFAULT NULL,
   l_wait_class     IN VARCHAR2  DEFAULT NULL,
   l_service_hash   IN NUMBER    DEFAULT NULL,
   l_module         IN VARCHAR2  DEFAULT NULL,
   l_action         IN VARCHAR2  DEFAULT NULL,
   l_client_id      IN VARCHAR2  DEFAULT NULL,
   l_plsql_entry    IN VARCHAR2  DEFAULT NULL,
   l_data_src       IN NUMBER    DEFAULT 0,
   l_container      IN VARCHAR2  DEFAULT NULL)
 RETURN awrrpt_html_type_table PIPELINED;
```

Parameters

Table 222-4 *ASH_GLOBAL_REPORT_HTML Parameters*

Parameter	Description
l_dbid	Database identifier
l_inst_num	List of instances (such as '1,2,3'), or NULL to report on all instances in the database
l_btime	The 'begin time'
l_etime	The 'end time'
l_options	Report level (currently not used)
l_slot_width	Specifies (in seconds) how wide the slots used in the "Top Activity" section of the report should be. This argument is optional, and if it is not specified the time interval between l_btime and l_etime is appropriately split into not more than 10 slots.
l_sid	Session ID (see Usage Notes)
l_sql_id	SQL ID (see Usage Notes)
l_wait_class	Wait class name (see Usage Notes)
l_service_hash	Service name hash (see Usage Notes)
l_module	Module name (see Usage Notes)
l_action	Action name (see Usage Notes)
l_client_id	Client ID for end-to-end backtracing (see Usage Notes)
l_plsql_entry	PL/SQL entry point (see Usage Notes)
l_data_src	Ignored since the report works off of data on disk only
l_container	Name of the container for which report activity is limited. Valid values other than NULL (default) should be taken from container names in V$CONTAINERS. Behavior is as follows: • If NULL: When connected to a root container the report is on all containers. When connected to a PDB the report is on only that PDB. • If not NULL: When connected to a root container the report is on activity from the specified container. When connected to a PDB the report is the same as NULL value for l_container regarding the connected PDB. Note: If while connected to a PDB you request information from another PDB this produces an empty report.

Return Values

The output will be one column of VARCHAR2(1500).

Usage Notes

- You can call the function directly but Oracle recommends you use the ashrpti.sql script which prompts users for the required information.
- The unspecified optional arguments are used to generate an ASH Reports that specify 'report targets' such as a SQL statement, or a session, or a particular Service/Module combination. These arguments are specified to restrict the ASH rows that would be used to generate the report. For example, to generate an ASH report on a particular SQL

statement, such as SQL_ID 'abcdefghij123 ' pass that sql_id value to the l_sql_id argument:

```
l_sql_id =>  'abcdefghij123'
```

Any combination of those optional arguments can be passed in, and only rows in ASH that satisfy all of those 'report targets' will be used. If multiple 'report targets' are specified, AND conditional logic is used to connect them. For example, to generate an ASH report on MODULE "PAYROLL" and ACTION "PROCESS", use the following predicate:

```
l_module =>  'PAYROLL', l_action =>  'PROCESS'
```

Valid SQL wildcards can be used in all the arguments that are of type VARCHAR2.

Table 222-5 ASH_REPORT_HTML: Wildcards Allowed (or Not) in Arguments

Argument Name	Comment	Wildcard Allowed
l_sid	Session ID (for example, V$SESSION.SID)	No
l_sql_id	SQL ID (for example, V$SQL.SQL_ID)	Yes
l_wait_class	Wait class name (for example, V$EVENT_NAME.WAIT_CLASS)	Yes
l_service_hash	Service name hash (for example, V$ACTIVE_SERVICES.NAME_HASH)	No
l_module	Module name (for example, V$SESSION.MODULE)	Yes
l_action	Action name (for example, V$SESSION.ACTION)	Yes
l_client_id	Client ID for end-to-end backtracing (for example, V$SESSION.CLIENT_IDENTIFIER)	Yes
l_data_src	Wildcards are not allowed for l_data_src as it is of numeric datatype	No

ASH_GLOBAL_REPORT_TEXT Function

This table function displays a global or Oracle Real Application Clusters (Oracle RAC) ASH Spot report in text format.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.ASH_GLOBAL_REPORT_TEXT(
   l_dbid          IN VARCHAR2(1023),
   l_inst_num      IN NUMBER,
   l_btime         IN DATE,
   l_etime         IN DATE,
   l_options       IN NUMBER    DEFAULT 0,
   l_slot_width    IN NUMBER    DEFAULT 0,
   l_sid           IN NUMBER    DEFAULT NULL,
   l_sql_id        IN VARCHAR2  DEFAULT NULL,
   l_wait_class    IN VARCHAR2  DEFAULT NULL,
   l_service_hash  IN NUMBER    DEFAULT NULL,
   l_module        IN VARCHAR2  DEFAULT NULL,
   l_action        IN VARCHAR2  DEFAULT NULL,
   l_client_id     IN VARCHAR2  DEFAULT NULL,
   l_plsql_entry   IN VARCHAR2  DEFAULT NULL,
   l_data_src      IN NUMBER    DEFAULT 0,
   l_container     IN VARCHAR2  DEFAULT NULL)
 RETURN awrrpt_text_type_table PIPELINED;
```

Parameters

Table 222-6 *ASH_GLOBAL_REPORT_TEXT Parameters*

Parameter	Description
l_dbid	Database identifier
l_inst_num	List of instances (such as '1,2,3'), or NULL to report on all instances in the database
l_btime	The 'begin time'
l_etime	The 'end time'
l_options	Report level (currently not used)
l_slot_width	Specifies (in seconds) how wide the slots used in the "Top Activity" section of the report should be. This argument is optional, and if it is not specified the time interval between l_btime and l_etime is appropriately split into not more than 10 slots.
l_sid	Session ID (see Usage Notes)
l_sql_id	SQL ID (see Usage Notes)
l_wait_class	Wait class name (see Usage Notes)
l_service_hash	Service name hash (see Usage Notes)
l_module	Module name (see Usage Notes)
l_action	Action name (see Usage Notes)
l_client_id	Client ID for end-to-end backtracing (see Usage Notes)
l_plsql_entry	PL/SQL entry point (see Usage Notes)
l_data_src	Ignored since the report works off of data on disk only
l_container	Name of the container for which report activity is limited. Valid values other than NULL (default) should be taken from container names in V$CONTAINERS. Behavior is as follows: • If NULL: When connected to a root container the report is on all containers. When connected to a PDB the report is on only that PDB. • If not NULL: When connected to a root container the report is on activity from the specified container. When connected to a PDB the report is the same as NULL value for l_container regarding the connected PDB. Note: If while connected to a PDB you request information from another PDB this produces an empty report.

Return Values

The output will be one column of VARCHAR2(320).

Usage Notes

- You can call the function directly but Oracle recommends you use the ashrpti.sql script which prompts users for the required information.
- The unspecified optional arguments are used to generate an ASH Reports that specify 'report targets' such as a SQL statement, or a session, or a particular Service/Module combination. These arguments are specified to restrict the ASH rows that would be used to generate the report. For example, to generate an ASH report on a particular SQL

statement, such as SQL_ID 'abcdefghij123 ' pass that SQL_ID value to the l_sql_id argument:

```
l_sql_id =>  'abcdefghij123'
```

Table 222-7 ASH_GLOBAL_REPORT_TEXT: Wildcards Allowed (or Not) in Arguments

Argument Name	Comment	Wildcard Allowed
l_sid	Session ID (for example, V$SESSION.SID)	No
l_sql_id	SQL ID (for example, V$SQL.SQL_ID)	Yes
l_wait_class	Wait class name (for example, V$EVENT_NAME.WAIT_CLASS)	Yes
l_service_hash	Service name hash (for example, V$ACTIVE_SERVICES.NAME_HASH)	No
l_module	Module name (for example, V$SESSION.MODULE)	Yes
l_action	Action name (for example, V$SESSION.ACTION)	Yes
l_client_id	Client ID for end-to-end backtracing (for example, V$SESSION.CLIENT_IDENTIFIER)	Yes
l_plsql_entry	PL/SQL entry point (for example, "SYS.DBMS_LOB.*")	Yes
l_data_src	Wildcards are not allowed for l_data_src as it is of numeric datatype	No

- Any combination of those optional arguments can be passed in, and only rows in ASH that satisfy all of those 'report targets' will be used. If multiple 'report targets' are specified, AND conditional logic is used to connect them. For example, to generate an ASH report on MODULE "PAYROLL" and ACTION "PROCESS", use the following predicate:

  ```
  l_module =>  'PAYROLL', l_action =>  'PROCESS'
  ```

  Valid SQL wildcards can be used in all the arguments that are of type VARCHAR2.

ASH_REPORT_ANALYTICS Function

This function returns the ASH Analytics active report.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.ASH_REPORT_ANALYTICS(
   dbid           IN NUMBER DEFAULT NULL,
   inst_id        IN NUMBER DEFAULT NULL,
   begin_time     IN DATE,
   end_time       IN DATE,
   report_level   IN VARCHAR2 DEFAULT NULL,
   filter_list    IN VARCHAR2 DEFAULT NULL)
 RETURN CLOB;
```

Parameters

Table 222-8 ASH_REPORT_ANALYTICS Parameters

Parameter	Description
dbid	Database identifier. If its value is set to NULL, then the database identifier for the local database is used. Its default value is NULL.
inst_id	Instance number of the database for which the statistics are required. If its value is set to NULL, then the statistics for the local database are returned. Its default value is NULL.
begin_time	The start time of the interval for which the ASH report is required.
end_time	The end time of the interval for which the ASH report is required.
report_level	Describes the list of components to build.
filter_list	Describes the list of filters to apply. Its default value is NULL (no filters to apply).

Return Values

Returns the ASH Analytics active report.

ASH_REPORT_HTML Function

This table function displays the ASH Spot report in HTML.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.ASH_REPORT_HTML(
   l_dbid          IN NUMBER,
   l_inst_num      IN NUMBER,
   l_btime         IN DATE,
   l_etime         IN DATE,
   l_options       IN NUMBER    DEFAULT 0,
   l_slot_width    IN NUMBER    DEFAULT 0,
   l_sid           IN NUMBER    DEFAULT NULL,
   l_sql_id        IN VARCHAR2  DEFAULT NULL,
   l_wait_class    IN VARCHAR2  DEFAULT NULL,
   l_service_hash  IN NUMBER    DEFAULT NULL,
   l_module        IN VARCHAR2  DEFAULT NULL,
   l_action        IN VARCHAR2  DEFAULT NULL,
   l_client_id     IN VARCHAR2  DEFAULT NULL,
   l_plsql_entry   IN VARCHAR2  DEFAULT NULL,
   l_data_src      IN NUMBER    DEFAULT 0,
   l_container     IN VARCHAR2  DEFAULT NULL)
 RETURN awrrpt_html_type_table PIPELINED;
```

Parameters

Table 222-9 *ASH_REPORT_HTML Parameters*

Parameter	Description
l_dbid	Database identifier
l_inst_num	Instance number

Table 222-9 (Cont.) *ASH_REPORT_HTML Parameters*

Parameter	Description
l_btime	The 'begin time'
l_etime	The 'end time'
l_options	Report level (currently not used)
l_slot_width	Specifies (in seconds) how wide the slots used in the "Top Activity" section of the report should be. This argument is optional, and if it is not specified the time interval between l_btime and l_etime is appropriately split into not more than 10 slots.
l_sid	Session ID (see Usage Notes)
l_sql_id	SQL ID (see Usage Notes)
l_wait_class	Wait class name (see Usage Notes)
l_service_hash	Service name hash (see Usage Notes)
l_module	Module name (see Usage Notes)
l_action	Action name (see Usage Notes)
l_client_id	Client ID for end-to-end backtracing (see Usage Notes)
l_plsql_entry	PL/SQL entry point (see Usage Notes)
l_data_src	Can be used to specify a data source (see Usage Notes) • 1 => memory (V$ACTIVE_SESION_HISTORY) • 2 => disk (DBA_HIST_ACTIVE_SESS_HISTORY) • 0 => both. This is the default value. Here, the begin and end time parameters are used to get the samples from the appropriate data source, which can be memory, disk, or both.
l_container	Name of the container for which report activity is limited. Valid values other than NULL (default) should be taken from container names in V$CONTAINERS. Behavior is as follows: • If NULL: When connected to a root container the report is on all containers. When connected to a PDB the report is on only that PDB. • If not NULL: When connected to a root container the report is on activity from the specified container. When connected to a PDB the report is the same as NULL value for l_container regarding the connected PDB. Note: If while connected to a PDB you request information from another PDB this produces an empty report.

Return Values

The output will be one column of VARCHAR2(500).

Usage Notes

- You can call the function directly but Oracle recommends you use the ashrpti.sql script which prompts users for the required information.
- By default, the report uses the begin and end time parameters (l_btime and l_etime, respectively) to find all rows in that time range either from memory, or disk, or both. However, using l_data_src, one can explicitly specify one of those data sources. For

example, to generate an ASH report on all rows between `l_btime` and `l_time` found in memory, use

```
l_data_src =>  1
```

Similarly, to generate a report on samples found only on disk, use

```
l_data_src => 2
```

- The unspecified optional arguments are used to generate an ASH Reports that specify 'report targets' such as a SQL statement, or a session, or a particular Service/Module combination. These arguments are specified to restrict the ASH rows that would be used to generate the report. For example, to generate an ASH report on a particular SQL statement, such as SQL_ID 'abcdefghij123 ' pass that `sql_id` value to the `l_sql_id` argument:

```
l_sql_id =>  'abcdefghij123'
```

Any combination of those optional arguments can be passed in, and only rows in ASH that satisfy all of those 'report targets' will be used. If multiple 'report targets' are specified, `AND` conditional logic is used to connect them. For example, to generate an ASH report on `MODULE` "PAYROLL" and `ACTION` "PROCESS", use the following predicate:

```
l_module =>  'PAYROLL', l_action =>  'PROCESS'
```

Valid SQL wildcards can be used in all the arguments that are of type `VARCHAR2`.

Table 222-10 ASH_REPORT_HTML: Wildcards Allowed (or Not) in Arguments

Argument Name	Comment	Wildcard Allowed
l_sid	Session ID (for example, V$SESSION.SID)	No
l_sql_id	SQL ID (for example, V$SQL.SQL_ID)	Yes
l_wait_class	Wait class name (for example, V$EVENT_NAME.WAIT_CLASS)	Yes
l_service_hash	Service name hash (for example, V$ACTIVE_SERVICES.NAME_HASH)	No
l_module	Module name (for example, V$SESSION.MODULE)	Yes
l_action	Action name (for example, V$SESSION.ACTION)	Yes
l_client_id	Client ID for end-to-end backtracing (for example, V$SESSION.CLIENT_IDENTIFIER)	Yes

ASH_REPORT_TEXT Function

This table function displays the ASH Spot report in text.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.ASH_REPORT_TEXT(
   l_dbid          IN NUMBER,
   l_inst_num      IN NUMBER,
   l_btime         IN DATE,
   l_etime         IN DATE,
   l_options       IN NUMBER    DEFAULT 0,
   l_slot_width    IN NUMBER    DEFAULT 0,
   l_sid           IN NUMBER    DEFAULT NULL,
   l_sql_id        IN VARCHAR2  DEFAULT NULL,
```

```
   l_wait_class    IN VARCHAR2  DEFAULT NULL,
   l_service_hash  IN NUMBER    DEFAULT NULL,
   l_module        IN VARCHAR2  DEFAULT NULL,
   l_action        IN VARCHAR2  DEFAULT NULL,
   l_client_id     IN VARCHAR2  DEFAULT NULL,
   l_plsql_entry   IN VARCHAR2  DEFAULT NULL,
   l_data_src      IN NUMBER    DEFAULT 0,
   l_container     IN VARCHAR2  DEFAULT NULL)
 RETURN awrrpt_text_type_table PIPELINED;
```

Parameters

Table 222-11 *ASH_REPORT_TEXT Parameters*

Parameter	Description
l_dbid	Database identifier
l_inst_num	Instance number
l_btime	The 'begin time'
l_etime	The 'end time'
l_options	Report level (currently not used)
l_slot_width	Specifies (in seconds) how wide the slots used in the "Top Activity" section of the report should be. This argument is optional, and if it is not specified the time interval between l_btime and l_etime is appropriately split into not more than 10 slots.
l_sid	Session ID (see Usage Notes)
l_sql_id	SQL ID (see Usage Notes)
l_wait_class	Wait class name (see Usage Notes)
l_service_hash	Service name hash (see Usage Notes)
l_module	Module name (see Usage Notes)
l_action	Action name (see Usage Notes)
l_client_id	Client ID for end-to-end backtracing (see Usage Notes)
l_plsql_entry	PL/SQL entry point (see Usage Notes)
l_data_src	Can be used to specify a data source (see Usage Notes) • 1 => memory (V$ACTIVE_SESION_HISTORY) • 2 => disk (DBA_HIST_ACTIVE_SESS_HISTORY) • 0 => both. This is the default value. Here, the begin and end time parameters are used to get the samples from the appropriate data source, which can be memory, disk, or both.
l_container	Name of the container for which report activity is limited. Valid values other than NULL (default) should be taken from container names in V$CONTAINERS. Behavior is as follows: • If NULL: When connected to a root container the report is on all containers. When connected to a PDB the report is on only that PDB. • If not NULL: When connected to a root container the report is on activity from the specified container. When connected to a PDB the report is the same as NULL value for l_container regarding the connected PDB. Note: If while connected to a PDB you request information from another PDB this produces an empty report.

Return Values

The output will be one column of `VARCHAR2(80)`.

Usage Notes

- You can call the function directly but Oracle recommends you use the `ashrpti.sql` script which prompts users for the required information.
- By default, the report uses the begin and end time parameters (`l_btime` and `l_etime`, respectively) to find all rows in that time range either from memory, or disk, or both. However, using `l_data_src`, one can explicitly specify one of those data sources. For example, to generate an ASH report on all rows between `l_btime` and `l_time` found in memory, use

  ```
  l_data_src =>  1
  ```

  Similarly, to generate a report on samples found only on disk, use

  ```
  l_data_src => 2
  ```

- The unspecified optional arguments are used to generate an ASH Reports that specify 'report targets' such as a SQL statement, or a session, or a particular Service/Module combination. These arguments are specified to restrict the ASH rows that would be used to generate the report. For example, to generate an ASH report on a particular SQL statement, such as SQL_ID '`abcdefghij123` ' pass that `SQL_ID` value to the `l_sql_id` argument:

  ```
  l_sql_id =>  'abcdefghij123'
  ```

Table 222-12 ASH_REPORT_TEXT: Wildcards Allowed (or Not) in Arguments

Argument Name	Comment	Wildcard Allowed
`l_sid`	Session ID (for example, `V$SESSION.SID`)	No
`l_sql_id`	SQL ID (for example, `V$SQL.SQL_ID`)	Yes
`l_wait_class`	Wait class name (for example, `V$EVENT_NAME.WAIT_CLASS`)	Yes
`l_service_hash`	Service name hash (for example, `V$ACTIVE_SERVICES.NAME_HASH`)	No
`l_module`	Module name (for example, `V$SESSION.MODULE`)	Yes
`l_action`	Action name (for example, `V$SESSION.ACTION`)	Yes
`l_client_id`	Client ID for end-to-end backtracing (for example, `V$SESSION.CLIENT_IDENTIFIER`)	Yes
`l_plsql_entry`	PL/SQL entry point (for example, "`SYS.DBMS_LOB.*`")	Yes
`l_data_src`	Wildcards are not allowed for `l_data_src` as it is of numeric datatype	No

- Any combination of those optional arguments can be passed in, and only rows in ASH that satisfy all of those 'report targets' will be used. If multiple 'report targets' are specified, `AND` conditional logic is used to connect them. For example, to generate an ASH report on `MODULE` "`PAYROLL`" and `ACTION` "`PROCESS`", use the following predicate:

  ```
  l_module =>  'PAYROLL', l_action =>  'PROCESS'
  ```

Valid SQL wildcards can be used in all the arguments that are of type VARCHAR2.

AWR_DIFF_REPORT_HTML Function

This table function displays the AWR Compare Periods report in HTML.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_DIFF_REPORT_HTML(
   dbid1     IN NUMBER,
   inst_num1 IN NUMBER,
   bid1      IN NUMBER,
   eid1      IN NUMBER,
   dbid2     IN NUMBER,
   inst_num2 IN NUMBER,
   bid2      IN NUMBER,
   eid2      IN NUMBER)
  RETURN awrdrpt_text_type_table PIPELINED;
```

Parameters

Table 222-13 *AWR_DIFF_REPORT_HTML Parameters*

Parameter	Description
dbid1	1st database identifier
inst_num1	1st instance number
bid1	1st beginning snapshot ID
eid1	1st ending snapshot ID
dbid2	2nd database identifier
inst_num2	2nd instance number
bid2	2nd beginning snapshot ID
eid2	2nd ending snapshot ID

Return Values

The output will be one column of VARCHAR2(500).

Usage Notes

You can call the function directly but Oracle recommends you use the awrddrpt.sql script which prompts users for the required information.

AWR_DIFF_REPORT_TEXT Function

This table function displays the AWR Compare Periods report in text.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_DIFF_REPORT_TEXT(
   dbid1     IN NUMBER,
   inst_num1 IN NUMBER,
   bid1      IN NUMBER,
   eid1      IN NUMBER,
   dbid2     IN NUMBER,
```

```
   inst_num2 IN NUMBER,
   bid2      IN NUMBER,
   eid2      IN NUMBER)
  RETURN awrdrpt_text_type_table PIPELINED;
```

Parameters

Table 222-14 *AWR_DIFF_REPORT_TEXT Parameters*

Parameter	Description
dbid1	1st database identifier
inst_num1	1st instance number
bid1	1st beginning snapshot ID
eid1	1st ending snapshot ID
dbid2	2nd database identifier
inst_num2	2nd instance number
bid2	2nd beginning snapshot ID
eid2	2nd ending snapshot ID

Return Values

The output will be one column of `VARCHAR2(500)`.

Usage Notes

You can call the function directly but Oracle recommends you use the `awrddrpt.sql` script which prompts users for the required information.

AWR_EXP Procedure

This procedure extracts AWR data from the AWR schema and dump the information into a file. You can specify the snapshot range for the data that you want to extract.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_EXP(
   dmpfile              IN   VARCHAR2   DEFAULT 'awrdat',
   dmpdir               IN   VARCHAR2   DEFAULT 'DATA_PUMP_DIR',
   dbid                 IN   NUMBER     DEFAULT NULL,
   bid                  IN   NUMBER     DEFAULT 1,
   eid                  IN   NUMBER     DEFAULT 1000000);
```

Parameters

Table 222-15 *AWR_EXP Procedure Parameters*

Parameter	Description
dmpfile	The prefix for the name of the extract dump file and log file. The name of the dump file where all the data from the AWR table will be written is dmpfile.dmp. The `dmpfile.log` log file shows the status of the export job. The default value for the prefix is `awrdat`.

Table 222-15 (Cont.) *AWR_EXP Procedure Parameters*

Parameter	Description
dmpdir	Name of the Directory Object for the file system directory where the extract dump file will be written. The list of Directory Objects can be queried using the DBA_DIRECTORIES view, and a new directory object can be created using the following command: create directory dmpdir as '/directory/path'. The default value is DATA_PUMP_DIR.
dbid	The database ID for the snapshots that you want to export. The default value is NULL, for the local database ID.
bid	The begin snapshot ID for snapshots to be exported. The default value is 1.
eid	The end Snapshot Id for snapshots to be exported. The default value is 10000000.

AWR_GLOBAL_DIFF_REPORT_HTML Functions

This table function displays Global AWR Compare Periods Report in HTML format.

The first overload accepts comma-separated lists of instance numbers for inst_num1 and inst_num2. No leading zeroes are allowed and there is a limit of 1023 characters.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_GLOBAL_DIFF_REPORT_HTML (
   dbid1         IN    NUMBER,
   inst_num1     IN    AWRRPT_INSTANCE_LIST_TYPE,
   bid1          IN    NUMBER,
   eid1          IN    NUMBER,
   dbid2         IN    NUMBER,
   inst_num2     IN    AWRRPT_INSTANCE_LIST_TYPE,
   bid2          IN    NUMBER,
   eid2          IN    NUMBER)
  RETURN awrrpt_html_type_table PIPELINED;

DBMS_WORKLOAD_REPOSITORY.AWR_GLOBAL_DIFF_REPORT_HTML (
   dbid1         IN    NUMBER,
   inst_num1     IN    VARCHAR2,
   bid1          IN    NUMBER,
   eid1          IN    NUMBER,
   dbid2         IN    NUMBER,
   inst_num2     IN    VARCHAR2,
   bid2          IN    NUMBER,
   eid2          IN    NUMBER)
  RETURN awrrpt_html_type_table PIPELINED;
```

Parameters

Table 222-16 *AWR_GLOBAL_DIFF_REPORT_HTML Function Parameters*

Parameter	Description
dbid1	1st database identifier
inst_num1	1st list of instance numbers. If set to NULL, all instances for which begin and end snapshots are available, and which have not been restarted between snapshots, will be included in the report.
bid1	1st beginning snapshot ID
eid1	1st ending snapshot ID
dbid2	2nd database identifier
inst_num2	2nd list of instance numbers to be included in report. If set to NULL, all instances for which begin and end snapshots are available, and which have not been restarted between snapshots, will be included in the report.
bid2	2nd beginning snapshot ID
eid2	2nd ending snapshot ID

Return Values

The output will be one column of VARCHAR2(1500).

AWR_GLOBAL_DIFF_REPORT_TEXT Functions

This table function displays Global AWR Compare Periods Report in text format.

The first overload accepts comma-separated lists of instance numbers for inst_num1 and inst_num2. No leading zeroes are allowed and there is a limit of 1023 characters.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_GLOBAL_DIFF_REPORT_TEXT (
   dbid1         IN    NUMBER,
   inst_num1     IN    AWRRPT_INSTANCE_LIST_TYPE,
   bid1          IN    NUMBER,
   eid1          IN    NUMBER,
   dbid2         IN    NUMBER,
   inst_num2     IN    AWRRPT_INSTANCE_LIST_TYPE,
   bid2          IN    NUMBER,
   eid2          IN    NUMBER)
  RETURN awrdrpt_text_type_table PIPELINED;

DBMS_WORKLOAD_REPOSITORY.AWR_GLOBAL_DIFF_REPORT_TEXT (
   dbid1         IN    NUMBER,
   inst_num1     IN    VARCHAR2,
   bid1          IN    NUMBER,
   eid1          IN    NUMBER,
   dbid2         IN    NUMBER,
   inst_num2     IN    VARCHAR2,
   bid2          IN    NUMBER,
   eid2          IN    NUMBER)
  RETURN awrdrpt_text_type_table PIPELINED;
```

Parameters

Table 222-17 *AWR_GLOBAL_DIFF_REPORT_TEXT Functions Parameters*

Parameter	Description
dbid1	1st database identifier
inst_num1	1st list of instance numbers. If set to NULL, all instances for which begin and end snapshots are available, and which have not been restarted between snapshots, will be included in the report.
bid1	1st beginning snapshot ID
eid1	1st ending snapshot ID
dbid2	2nd database identifier
inst_num2	2nd list of instance numbers to be included in report. If set to NULL, all instances for which begin and end snapshots are available, and which have not been restarted between snapshots, will be included in the report.
bid2	2nd beginning snapshot ID
eid2	2nd ending snapshot ID

Return Values

The output will be one column of VARCHAR2(320).

AWR_GLOBAL_REPORT_HTML Functions

This table function displays the Global AWR report in HTML.

The first overload accepts a comma-separated list of instance numbers. No leading zeroes are allowed and there is a limit of 1023 characters.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_GLOBAL_REPORT_HTML (
   l_dbid        IN    NUMBER,
   l_inst_num    IN    AWRRPT_INSTANCE_LIST_TYPE,
   l_bid         IN    NUMBER,
   l_eid         IN    NUMBER,
   l_options     IN    NUMBER DEFAULT 0)
  RETURN awrrpt_html_type_table PIPELINED;

DBMS_WORKLOAD_REPOSITORY.AWR_GLOBAL_REPORT_HTML (
   l_dbid       IN    NUMBER,
   l_inst_num   IN    VARCHAR2,
   l_bid        IN    NUMBER,
   l_eid        IN    NUMBER,
   l_options    IN    NUMBER DEFAULT 0)
  RETURN awrrpt_html_type_table PIPELINED;
```

Parameters

Table 222-18 *AWR_GLOBAL_REPORT_HTML Function Parameters*

Parameter	Description
l_dbid	Database identifier
l_inst_num	List of instance numbers to be included in report. If set to NULL, all instances for which begin and end snapshots are available, and which have not been restarted between snapshots, will be included in the report.
l_bid	Beginning snapshot ID
l_eid	Ending snapshot ID
l_options	Report level (currently not used)

Return Values

The output will be one column of VARCHAR2(1500).

AWR_GLOBAL_REPORT_TEXT Functions

This table function displays the Global AWR report in text.

The first overload accepts a comma-separated list of instance numbers. No leading zeroes are allowed and there is a limit of 1023 characters

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_GLOBAL_REPORT_TEXT(
   l_dbid        IN    NUMBER,
   l_inst_num    IN    AWRRPT_INSTANCE_LIST_TYPE,
   l_bid         IN    NUMBER,
   l_eid         IN    NUMBER,
   l_options     IN    NUMBER DEFAULT 0)
  RETURN awrdrpt_text_type_table PIPELINED;

DBMS_WORKLOAD_REPOSITORY.AWR_GLOBAL_REPORT_TEXT(
   l_dbid       IN    NUMBER,
   l_inst_num   IN    VARCHAR2,
   l_bid        IN    NUMBER,
   l_eid        IN    NUMBER,
   l_options    IN    NUMBER DEFAULT 0)
  RETURN awrdrpt_text_type_table PIPELINED;
```

Parameters

Table 222-19 *AWR_GLOBAL_REPORT_TEXT Function Parameters*

Parameter	Description
l_dbid	Database identifier
l_inst_num	List of instance numbers to be included in report. If set to NULL, all instances for which begin and end snapshots are available, and which have not been restarted between snapshots, will be included in the report.

Table 222-19 (Cont.) *AWR_GLOBAL_REPORT_TEXT Function Parameters*

Parameter	Description
l_bid	Beginning snapshot ID
l_eid	Ending snapshot ID
l_options	A flag to specify to control the output of the report. Currently, not used.

Return Values

The output will be one column of VARCHAR2(320).

AWR_IMP Procedure

This procedure loads the AWR data from a dump file into the SYS schema.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_IMP(
   dmpfile              IN   VARCHAR2   DEFAULT 'awrdat',
   dmpdir               IN   VARCHAR2   DEFAULT 'DATA_PUMP_DIR',
   new_dbid             IN   NUMBER     DEFAULT NULL);
```

Parameters

Table 222-20 *AWR_IMP Procedure Parameters*

Parameter	Description
dmpfile	The prefix for the name of the dump file and log file. This file will be the source of the imported AWR data. The default value is awrdat.
dmpdir	The name of the Directory Object for the file system directory where the load dump file is located. The default value is DATA_PUMP_DIR.
new_dbid	The database ID to be used instead of existing database ID.

AWR_REPORT_HTML Function

This table function displays the AWR report in HTML.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_HTML(
   l_dbid       IN    NUMBER,
   l_inst_num   IN    NUMBER,
   l_bid        IN    NUMBER,
   l_eid        IN    NUMBER,
   l_options    IN    NUMBER DEFAULT 0)
 RETURN awrrpt_text_type_table PIPELINED;
```

Parameters

Table 222-21 *AWR_REPORT_HTML Parameters*

Parameter	Description
l_dbid	Database identifier
l_inst_num	Instance number
l_bid	Beginning snapshot ID
l_eid	Ending snapshot ID
l_options	A flag to specify to control the output of the report. Currently, Oracle supports one value: • l_options - 8. Displays the ADDM specific portions of the report. These sections include the Buffer Pool Advice, Shared Pool Advice, and PGA Target Advice.

Return Values

The output will be one column of VARCHAR2(1500).

Usage Notes

You can call the function directly but Oracle recommends you use the awrrpt.sql script which prompts users for the required information.

AWR_REPORT_TEXT Function

This table function displays the AWR report in text.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_TEXT(
   l_dbid       IN    NUMBER,
   l_inst_num   IN    NUMBER,
   l_bid        IN    NUMBER,
   l_eid        IN    NUMBER,
   l_options    IN    NUMBER DEFAULT 0)
 RETURN awrrpt_text_type_table PIPELINED;
```

Parameters

Table 222-22 AWR_REPORT_TEXT Parameters

Parameter	Description
l_dbid	Database identifier
l_insT_num	Instance number
l_bid	Beginning snapshot ID
l_eid	Ending snapshot ID

Table 222-22 (Cont.) AWR_REPORT_TEXT Parameters

Parameter	Description
l_options	A flag to specify to control the output of the report. Currently, Oracle supports one value: • l_options - 8. Displays the ADDM specific portions of the report. These sections include the Buffer Pool Advice, Shared Pool Advice, and PGA Target Advice.

Return Values

The output will be one column of VARCHAR2(80).

Usage Notes

You can call the function directly but Oracle recommends you use the awrrpt.sql script which prompts users for the required information.

AWR_SET_REPORT_THRESHOLDS Procedure

This procedure configure specified report thresholds, including the number of rows in the report.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_SET_REPORT_THRESHOLDS(
   top_n_events         IN   NUMBER DEFAULT NULL,
   top_n_files          IN   NUMBER DEFAULT NULL,
   top_n_segments       IN   NUMBER DEFAULT NULL,
   top_n_services       IN   NUMBER DEFAULT NULL,
   top_n_sql            IN   NUMBER DEFAULT NULL,
   top_n_sql_max        IN   NUMBER DEFAULT NULL,
   top_sql_pct          IN   NUMBER DEFAULT NULL,
   shmem_threshold      IN   NUMBER DEFAULT NULL,
   versions_threshold   IN   NUMBER DEFAULT NULL,
   top_n_disks          IN   NUMBER DEFAULT NULL,
   outlier_pct          IN   NUMBER DEFAULT NULL,
   outlier_cpu_pct      IN   NUMBER DEFAULT NULL);
```

Parameters

Table 222-23 *AWR_SET_REPORT_THRESHOLDS Procedure Parameters*

Parameter	Description
top_n_events	Number of most significant wait events to be included
top_n_files	Number of most active files to be included
top_n_segments	Number of most active segments to be included
top_n_services	Number of most active services to be included
top_n_sql	Number of most significant SQL statements to be included
top_n_sql_max	Number of SQL statements to be included if their activity is greater than that specified by top_sql_pct

Table 222-23 (Cont.) *AWR_SET_REPORT_THRESHOLDS Procedure Parameters*

Parameter	Description
top_sql_pct	Significance threshold for SQL statements between top_n_sql and top_n_sql_max
shmem_threshold	Shared memory low threshold
versions_threshold	Plan version count low threshold
top_n_disks	Number of cell disks with most I/O
outlier_pct	Percentage of maximum capacity before displaying outliers for Exadata sections
outlier_cpu_pct	Threshold for mean percentage CPU to display outliers

User Notes

- The effect of each setting depends on the type of report being generated as well as on the underlying AWR data. Not all settings are meaningful for each report type. Invalid settings (such as negative numbers) are ignored.
- Settings are effective only in the context of the session that executes the AWR_SET_REPORT_THRESHOLDS procedure. For example, to get a report that lists top 12 segments as compared to the default, one can invoke as follows:

```
DBMS_WORKLOAD_REPOSITORY.AWR_SET_REPORT_THRESHOLDS (top_n_segments=>12);
```

AWR_SQL_REPORT_HTML Function

This table function displays the AWR SQL Report in HTML format.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_SQL_REPORT_HTML(
   l_dbid       IN    NUMBER,
   l_inst_num   IN    NUMBER,
   l_bid        IN    NUMBER,
   l_eid        IN    NUMBER,
   l_sqlid      IN    VARCHAR2,
   l_options    IN    NUMBER DEFAULT 0,
   l_con_dbid  IN    NUMBER DEFAULT NULL)
 RETURN awrrpt_html_type_table PIPELINED;
```

Parameters

Table 222-24 *AWR_SQL_REPORT_HTML Parameters*

Parameter	Description
l_dbid	Database identifier
l_inst_num	Instance number
l_bid	Beginning snapshot ID
l_eid	Ending snapshot ID

Table 222-24 (Cont.) *AWR_SQL_REPORT_HTML Parameters*

Parameter	Description
l_sqlid	SQL ID of statement to be analyzed
l_options	A flag to specify to control the output of the report. Currently, not used.
l_con_dbid	Container database identifier. If NULL (default), then l_con_dbid is set equal to l_dbid

Return Values

The output will be one column of VARCHAR2(500).

Usage Notes

You can call the function directly but Oracle recommends you use the awrsqrpt.sql script which prompts users for the required information.

AWR_SQL_REPORT_TEXT Function

This table function displays the AWR SQL Report in text format.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.AWR_SQL_REPORT_TEXT(
   l_dbid       IN    NUMBER,
   l_inst_num   IN    NUMBER,
   l_bid        IN    NUMBER,
   l_eid        IN    NUMBER,
   l_sqlid      IN    VARCHAR2,
   l_options    IN    NUMBER DEFAULT 0,
   l_con_dbid  IN    NUMBER DEFAULT NULL)
 RETURN awrrpt_text_type_table PIPELINED;
```

Parameters

Table 222-25 *AWR_SQL_REPORT_TEXT Parameters*

Parameter	Description
l_dbid	Database identifier
l_inst_num	Instance number
l_bid	Beginning snapshot ID
l_eid	Ending snapshot ID
l_sqlid	SQL ID of statement to be analyzed
l_options	Flag to specify to control the output of the report. Currently, not used.
l_con_dbid	Container database identifier. If NULL (default), then l_con_dbid is set equal to l_dbid

Return Values

The output will be one column of VARCHAR2(120).

Usage Notes

You can call the function directly but Oracle recommends you use the `awrsqrpt.sql` script which prompts users for the required information.

CONTROL_RESTRICTED_SNAPSHOT Procedure

This procedure controls the AWR snapshot creation for a database in the *restricted session* mode. If this procedure is not used, then by default, the AWR snapshots cannot be created for a database in the *restricted session* mode. This procedure affects the local database on which it is executed.

> **Note:**
>
> To enable AWR snapshot creation for an Oracle RAC in the *restricted session* mode, this procedure must be executed on every database instance in the Oracle RAC.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.CONTROL_RESTRICTED_SNAPSHOT(
   allow IN BOOLEAN);
```

Parameters

Table 222-26 CONTROL_RESTRICTED_SNAPSHOT Parameters

Parameter	Description
`allow`	This parameter can have one of the following values: • `TRUE`: AWR snapshots can be created for the database in the *restricted session* mode. • `FALSE`: AWR snapshots cannot be created for the database in the *restricted session* mode.

CREATE_BASELINE Functions and Procedures

This function and procedure creates a baseline.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE(
   start_snap_id    IN  NUMBER,
   end_snap_id      IN  NUMBER,
   baseline_name    IN  VARCHAR2,
   dbid             IN  NUMBER DEFAULT NULL,
   expiration       IN  NUMBER DEFAULT NULL);

DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE(
   start_snap_id    IN  NUMBER,
   end_snap_id      IN  NUMBER,
   baseline_name    IN  VARCHAR2,
   dbid             IN  NUMBER DEFAULT NULL,
   expiration       IN  NUMBER DEFAULT NULL)
 RETURN NUMBER;
```

```
DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE(
   start_time       IN  DATE,
   end_time         IN  DATE,
   baseline_name    IN  VARCHAR2,
   dbid             IN  NUMBER DEFAULT NULL,
   expiration       IN  NUMBER DEFAULT NULL);

DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE(
   start_time       IN  DATE,
   end_time         IN  DATE,
   baseline_name    IN  VARCHAR2,
   dbid             IN  NUMBER DEFAULT NULL,
   expiration       IN  NUMBER DEFAULT NULL);
 RETURN NUMBER;
```

Parameters

Table 222-27 CREATE_BASELINE Function & Procedure Parameters

Parameter	Description
start_snap_id	Start snapshot sequence number for the baseline.
end_snap_id	End snapshot sequence number for the baseline.
start_time	Start time for the baseline.
end_time	End time for the baseline.
baseline_name	Name of baseline.
dbid	Database Identifier for baseline. If NULL, this takes the database identifier for the local database. Defaults to NULL.
expiration	Expiration in number of days for the baseline. If NULL, then expiration is infinite, meaning do not drop baseline ever. Defaults to NULL.

Exceptions

- An error will be returned if this baseline name already exists in the system.
- The snapshot range that is specified for this interface must be an existing pair of snapshots in the database. An error will be returned if the inputted snapshots do not exist in the system.

Examples

This example creates a baseline (named 'oltp_peakload_bl') between snapshots 105 and 107 for the local database:

```
EXECUTE DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE (start_snap_id => 105,
                                                  end_snap_id   => 107,
                                                  baseline_name => 'oltp_peakload_bl');
```

If you query the DBA_HIST_BASELINE view after the CREATE BASELINE action, you will see the newly created baseline in the Workload Repository.

CREATE_BASELINE_TEMPLATE Procedures

This procedure specifies a template for how they would like baselines to be created for future time periods.

Syntax

Specifies a template for generating a baseline for a single time period in the future.

```
DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE_TEMPLATE(
   start_time              IN DATE,
   end_time                IN DATE,
   baseline_name           IN VARCHAR2,
   template_name           IN VARCHAR2,
   expiration              IN NUMBER,
   dbid                    IN NUMBER   DEFAULT NULL);
```

Specifies a template for creating and dropping baseline based on repeating time periods:

```
DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE_TEMPLATE(
   day_of_week             IN VARCHAR2,
   hour_in_day             IN NUMBER,
   duration                IN NUMBER,
   start_time              IN DATE,
   end_time                IN DATE,
   baseline_name_prefix    IN VARCHAR2,
   template_name           IN VARCHAR2,
   expiration              IN NUMBER,
   dbid                    IN NUMBER   DEFAULT NULL);
```

Parameters

Table 222-28 CREATE_BASELINE_TEMPLATE Procedure Parameters

Parameter	Description
start_time	Start Time for the baseline to be created'
end_time	End Time for the baseline to be created
baseline_name	Name of baseline to be created
template_name	Name for the template
expiration	Expiration in number of days for the baseline. If NULL, then expiration is infinite, meaning do not drop baseline ever. Defaults to NULL
dbid	Database ID for which the baseline template needs to be used. If NULL, this takes the database identifier of the local database. Defaults to NULL.
day_of_week	Day of week that the baseline should repeat on. Specify one of the following values: SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY.
hour_in_day	Value of 0-23 to specify the Hour in the Day the baseline should start
duration	Duration (in number of hours) after hour in the day that the baseline should last
baseline_name_prefix	Name for baseline prefix. When creating the baseline, the name of the baseline will be the prefix appended with the date information.

CREATE_REMOTE_SNAPSHOT Function and Procedure

This function and procedure create a remote snapshot using the Remote Management Framework (RMF). The function returns the snapshot ID.

> **Note:**
>
> A multitenant container database is the only supported architecture in Oracle Database 21c and later releases. While the documentation is being revised, legacy terminology may persist. In most cases, "database" and "non-CDB" refer to a CDB or PDB, depending on context. In some contexts, such as upgrades, "non-CDB" refers to a non-CDB from a previous release.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.CREATE_REMOTE_SNAPSHOT(
   node_id     IN NUMBER,
   flush_level IN VARCHAR2 DEFAULT 'BESTFIT');

DBMS_WORKLOAD_REPOSITORY.CREATE_REMOTE_SNAPSHOT(
   node_id     IN NUMBER,
   flush_level IN VARCHAR2 DEFAULT 'BESTFIT')
 RETURN NUMBER;

DBMS_WORKLOAD_REPOSITORY.CREATE_REMOTE_SNAPSHOT(
   node_name      IN VARCHAR2,
   topology_name  IN VARCHAR2 DEFAULT NULL,
   flush_level    IN VARCHAR2 DEFAULT 'BESTFIT');

DBMS_WORKLOAD_REPOSITORY.CREATE_REMOTE_SNAPSHOT(
   node_name      IN VARCHAR2,
   topology_name  IN VARCHAR2 DEFAULT NULL,
   flush_level    IN VARCHAR2 DEFAULT 'BESTFIT')
 RETURN NUMBER;
```

Parameters

Table 222-29 CREATE_REMOTE_SNAPSHOT Parameters

Parameter	Description
node_id	RMF node identifier of the database for which the snapshot needs to be created.
node_name	RMF node name of the database for which the snapshot needs to be created.
topology_name	RMF topology name of the database for which the snapshot needs to be created.

Table 222-29 (Cont.) CREATE_REMOTE_SNAPSHOT Parameters

Parameter	Description
flush_level	The flush level can be one of the following: • BESTFIT: Uses the default value depending on the type of snapshot being taken. • LITE: Lightweight snapshot. Only the most important statistics are collected. This is default for a pluggable database (PDB) and application container. • TYPICAL: Regular snapshot. Most of the statistics are collected. This is default for a container database root (CDB root) and non-CDB database. • ALL: Heavyweight snapshot. All the possible statistics are collected. This consumes a considerable amount of disk space and takes a long time to create.

Examples

This example creates a remote snapshot of the database having the RMF node identifier of 10:

```
EXECUTE DBMS_WORKLOAD_REPOSITORY.CREATE_REMOTE_SNAPSHOT(10);
```

If you query the DBA_HIST_SNAPSHOT view after executing the above procedure, you will see a new snapshot record added to the Automatic Workload Repository (AWR).

CREATE_SNAPSHOT Function and Procedure

This function and procedure create a snapshot. The function returns the snapshot ID. If both, the database ID and the database name are not specified in this subprogram, then the snapshot is created for the local database on which the subprogram is executed.

> **Note:**
>
> A multitenant container database is the only supported architecture in Oracle Database 21c and later releases. While the documentation is being revised, legacy terminology may persist. In most cases, "database" and "non-CDB" refer to a CDB or PDB, depending on context. In some contexts, such as upgrades, "non-CDB" refers to a non-CDB from a previous release.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT(
   flush_level IN VARCHAR2 DEFAULT 'BESTFIT',
   dbid        IN NUMBER   DEFAULT NULL,
   source_name IN VARCHAR2 DEFAULT NULL);

DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT(
   flush_level IN VARCHAR2 DEFAULT 'BESTFIT',
```

```
   dbid        IN NUMBER   DEFAULT NULL,
   source_name IN VARCHAR2 DEFAULT NULL)
 RETURN NUMBER;
```

Parameters

Table 222-30 CREATE_SNAPSHOT Parameters

Parameter	Description
flush_level	The flush level can be one of the following: • BESTFIT: Uses the default value depending on the type of snapshot being taken. • LITE: Lightweight snapshot. Only the most important statistics are collected. This is default for a pluggable database (PDB) and application container. • TYPICAL: Regular snapshot. Most of the statistics are collected. This is default for a container database root (CDB root) and non-CDB database. • ALL: Heavyweight snapshot. All the possible statistics are collected. This consumes a considerable amount of disk space and takes a long time to create.
dbid	Database ID of the database for which the snapshot needs to be created.
source_name	Name of the database for which the snapshot needs to be created.

Examples

This example creates a snapshot of the local database with the flush level of ALL:

```
EXECUTE DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT('ALL');
```

If you query the DBA_HIST_SNAPSHOT view after executing the above procedure, you will see a new snapshot record added to the Automatic Workload Repository (AWR).

DROP_BASELINE Procedure

This procedure drops a previously-defined baseline.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.DROP_BASELINE(
   baseline_name  IN  VARCHAR2,
   cascade        IN  BOOLEAN  DEFAULT FALSE,
   dbid           IN  NUMBER   DEFAULT NULL);
```

Parameters

Table 222-31 DROP_BASELINE Parameters

Parameter	Description
baseline_name	Name of baseline to drop from the system
cascade	If TRUE, the pair of snapshots associated with the baseline will also be dropped. Otherwise, only the baseline is removed.
dbid	Database ID for which the baseline needs to be dropped (defaults to local DBID).

Examples

This example drops the baseline 'oltp_peakload_bl' without dropping the underlying snapshots:

```
EXECUTE DBMS_WORKLOAD_REPOSITORY.DROP_BASELINE (
       baseline_name => 'oltp_peakload_bl');
```

If you query the DBA_HIST_BASELINE view after the DROP_BASELINE action, you will see the specified baseline definition is removed. You can query the DBA_HIST_SNAPSHOT view to find that the underlying snapshots are left intact.

DROP_BASELINE_TEMPLATE Procedure

This procedure removes a template that is no longer needed.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.DROP_BASELINE_TEMPLATE(
   template_name    IN VARCHAR2,
   dbid             IN NUMBER   DEFAULT NULL);
```

Parameters

Table 222-32 DROP_BASELINE_TEMPLATE Procedure Parameters

Parameter	Description
template_name	Name of the template to remove
dbid	Database ID for which the baseline template needs to be dropped. If NULL, this takes the database identifier of the local database. Defaults to NULL.

DROP_SNAPSHOT_RANGE Procedure

This procedure drops a range of snapshots.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.DROP_SNAPSHOT_RANGE(
   low_snap_id    IN  NUMBER,
   high_snap_id   IN  NUMBER,
   dbid           IN  NUMBER   DEFAULT NULL);
```

Parameters

Table 222-33 DROP_SNAPSHOT_RANGE Procedure Parameters

Parameter	Description
low_snap_id	Low snapshot id of snapshots to drop.
high_snap_id	High snapshot id of snapshots to drop.
dbid	Database id (defaults to local DBID).

Examples

This example drops the range of snapshots between snapshot id 102 to 105 for the local database:

```
EXECUTE DBMS_WORKLOAD_REPOSITORY.DROP_SNAPSHOT_RANGE(102, 105);
```

If you query the `dba_hist_snapshot` view after the `Drop Snapshot` action, you will see that snapshots 102 to 105 are removed from the Workload Repository.

LOCAL_AWR_DBID Function

This function returns the database identifier for the local AWR database.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.LOCAL_AWR_DBID()
 RETURN NUMBER;
```

MODIFY_BASELINE_WINDOW_SIZE Procedure

This procedure modifies the window size for the Default Moving Window Baseline.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.MODIFY_BASELINE_WINDOW_SIZE(
   window_size     IN   NUMBER,
   dbid            IN   NUMBER   DEFAULT NULL);
```

Parameters

Table 222-34 MODIFY_BASELINE_WINDOW_SIZE Procedure Parameters

Parameter	Description
window_size	New Window size for the default Moving Window Baseline, in number of days.
dbid	Database ID (defaults to local DBID).

Usage Notes

The window size must be less than or equal to the AWR retention setting. If the window size needs to be greater than the retention setting, the MODIFY_SNAPSHOT_SETTINGS Procedures can be used to adjust the retention setting. A moving window can be set to a maximum of 13 weeks.

MODIFY_SNAPSHOT_SETTINGS Procedures

This procedure controls three aspects of snapshot generation.

- The `INTERVAL` setting affects how often snapshots are automatically captured.
- The `RETENTION` setting affects how long snapshots are retained in the Workload Repository.
- The number of SQL captured for each Top criteria. If the user manually specifies a value for Top N SQL, the AWR SQL collection will use the user-specified number for both automatic and manual snapshots.

There are two overloads. The first takes a `NUMBER` and the second takes a `VARCHAR2` for the `topnsql` argument. The differences are described under the Parameters description.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS(
   retention       IN  NUMBER    DEFAULT  NULL,
   interval        IN  NUMBER    DEFAULT  NULL,
   topnsql         IN  NUMBER    DEFAULT  NULL,
   dbid            IN  NUMBER    DEFAULT  NULL,
   tablespace_name     IN  VARCHAR2  DEFAULT  NULL);

DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS(
   retention       IN  NUMBER    DEFAULT NULL,
   interval        IN  NUMBER    DEFAULT NULL,
   topnsql         IN  VARCHAR2,
   dbid            IN  NUMBER    DEFAULT  NULL,
   tablespace_name     IN  VARCHAR2  DEFAULT  NULL);
```

Parameters

Table 222-35 MODIFY_SNAPSHOT_SETTINGS Procedure Parameters

Parameter	Description
`retention`	New retention time (in minutes). The specified value must be in the range of `MIN_RETENTION` (1 day) to `MAX_RETENTION` (100 years). If `ZERO` is specified, snapshots will be retained forever. A large system-defined value will be used as the interval setting. If `NULL` is specified, the old value for retention is preserved. NOTE: The retention setting must be greater than or equal to the window size of the '`SYSTEM_MOVING_WINDOW`' baseline. If the retention needs to be less than the window size, the MODIFY_BASELINE_WINDOW_SIZE Procedure can be used to adjust the window size.
`interval`	New interval setting between each snapshot, in units of minutes. The specified value must be in the range `MIN_INTERVAL` (10 minutes) to `MAX_INTERVAL` (1 year). If `ZERO` is specified, automatic and manual snapshots will be disabled. A large system-defined value will be used as the retention setting. If `NULL` is specified, the current value is preserved.

Table 222-35 (Cont.) MODIFY_SNAPSHOT_SETTINGS Procedure Parameters

Parameter	Description
topnsql	• If NUMBER: Top N SQL size. The number of Top SQL to flush for each SQL criteria (Elapsed Time, CPU Time, Parse Calls, Shareable Memory, Version Count). The value for this setting will not be affected by the statistics/flush level and will override the system default behavior for the AWR SQL collection. The setting will have a minimum value of 30 and a maximum value of 50,000. Specifying NULL will keep the current setting. • If VARCHAR2: Users are allowed to specify the following values: (DEFAULT, MAXIMUM, N), where N is the number of Top SQL to flush for each SQL criteria. Specifying DEFAULT will revert the system back to the default behavior of Top 30 for statistics level TYPICAL and Top 100 for statistics level ALL. Specifying MAXIMUM will cause the system to capture the complete set of SQL in the cursor cache. Specifying the number N is equivalent to setting the Top N SQL with the NUMBER type. Specifying NULL for this argument will keep the current setting.
dbid	Database identifier in AWR for which to modify the snapshot settings. If NULL is specified, the local dbid will be used. Defaults to NULL.
tablespace_name	Specify a user-defined tablespace for storing AWR data (snapshot data). If this parameter is not used, then AWR data is stored in the SYSAUX tablespace by default.

Examples

This example changes the interval setting to one hour and the retention setting to two weeks for the local database:

```
EXECUTE DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS(
  interval  =>  60,
  retention =>  20160);
```

If you query the DBA_HIST_WR_CONTROL table after this procedure is executed, you will see the changes to these settings.

PURGE_SQL_DETAILS Procedure

This procedure purges SQL details, specifically rows from WRH$_SQLTEXT, WRH$_SQL_PLAN, and WRH$_SQL_BIND_METADATA that do not have corresponding rows (DBID, SQL_ID) in WRH$_SQLSTAT.

The subprogram calls for the DBID for which to run the purge. If the DBID is not specified, the database DBID is used. You can constrain runtime by specifying the maximum number of rows to purge per table. If no maximum is specified, the subprograms tries to purge all applicable rows.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.PURGE_SQL_DETAILS(
   numrows IN NUMBER DEFAULT NULL,
   dbid    IN NUMBER DEFAULT NULL);
```

Parameters

Table 222-36 PURGE_SQL_DETAILS Procedure Parameters

Parameter	Description
numrows	Number of rows
dbid	Database ID (default to local DBID)

REGISTER_REMOTE_DATABASE Procedures

This procedure registers a remote database in the Automatic Workload Repository (AWR) using the Remote Management Framework (RMF).

Syntax

```
DBMS_WORKLOAD_REPOSITORY.REGISTER_REMOTE_DATABASE(
   node_id  IN NUMBER);

DBMS_WORKLOAD_REPOSITORY.REGISTER_REMOTE_DATABASE(
   node_name      IN VARCHAR2,
   topology_name  IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 222-37 REGISTER_REMOTE_DATABASE Parameters

Parameter	Description
node_id	RMF node identifier of the database that needs to be registered in the AWR.
node_name	RMF node name of the database that needs to be registered in the AWR.
topology_name	RMF topology name of the database that needs to be registered in the AWR.

Examples

This example registers the remote database having the RMF node identifier of 10 in the AWR:

```
EXECUTE DBMS_WORKLOAD_REPOSITORY.REGISTER_REMOTE_DATABASE(10);
```

REMOVE_COLORED_SQL Procedure

This procedure removes a colored SQL ID. After a SQL is uncolored, it will no longer be captured in a snapshot automatically, unless it makes the TOP list.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.REMOVE_COLORED_SQL(
   sql_id         IN VARCHAR2,
   dbid           IN NUMBER DEFAULT NULL);
```

Parameters

Table 222-38 *REMOVE_COLORED_SQL Procedure Parameters*

Parameter	Description
sql_id	13-character external SQL ID
dbid	Optional dbid, defaults to Local DBID

RENAME_BASELINE Procedure

This procedure renames a baseline.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.RENAME_BASELINE(
   old_baseline_name     IN   VARCHAR2,
   new_baseline_name     IN VARCHAR2,
   dbid                  IN NUMBER   DEFAULT NULL);
```

Parameters

Table 222-39 RENAME_BASELINE Procedure Parameters

Parameter	Description
old_baseline_name	Old baseline name.
new_baseline_name	New baseline name.
dbid	Database ID for which the baseline needs to be renamed (defaults to local DBID).

SELECT_BASELINE_DETAILS Function

This table function shows the values of the metrics corresponding to a baseline for a range of snapshots.

This table function returns an object of AWR_BASELINE_METRIC_TYPE Object Type.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.SELECT_BASELINE_DETAILS(
   l_baseline_id    IN  NUMBER,
   l_begin_snap     IN  NUMBER DEFAULT NULL,
   l_end_snap       IN  NUMBER DEFAULT NULL,
   l_dbid           IN  NUMBER DEFAULT NULL)
 RETURN awrbl_details_type_table PIPELINED;
```

Parameters

Table 222-40 SELECT_BASELINE_DETAILS Function Parameters

Parameter	Description
l_baseline_id	ID of the baseline for which the statistics need to be retrieved. Specifying the value 0 returns the statistics for the moving window baseline.
l_begin_snap	Start snapshot sequence number for the baseline.
l_end_snap	End snapshot sequence number for the baseline.
l_dbid	Database identifier for the baseline. If its value is set to NULL, then the database identifier for the local database is used. Its default value is NULL.

SELECT_BASELINE_METRIC Function

This table function shows the values of the metrics corresponding to a baseline for all the snapshots.

This table function returns an object of AWR_BASELINE_METRIC_TYPE Object Type.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.SELECT_BASELINE_METRIC(
   l_baseline_name     IN VARCHAR2,
   l_dbid              IN NUMBER DEFAULT NULL,
   l_instance_num      IN NUMBER DEFAULT NULL)
 RETURN awr_metric_type_table PIPELINED;
```

Parameters

Table 222-41 SELECT_BASELINE_METRIC Function Parameters

Parameter	Description
l_baseline_name	Name of the baseline for which the metrics need to be viewed.
l_dbid	Database identifier for the baseline. If set to NULL, the database identifier for the local database is used. Default is NULL.
l_instance_num	The instance number for which the metrics need to be viewed. If set to NULL, metrics for the local database instance are shown. Default is NULL.

UNREGISTER_REMOTE_DATABASE Procedures

This procedure removes all the statistics, metadata, partitions, and so on of a remote database from the Automatic Workload Repository (AWR). After executing this procedure, the remote database cannot be used for any AWR operations, such as creating remote snapshots.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.UNREGISTER_REMOTE_DATABASE(
   node_id        IN NUMBER,
   remote_check   IN BOOLEAN  DEFAULT TRUE);
```

```
DBMS_WORKLOAD_REPOSITORY.UNREGISTER_REMOTE_DATABASE(
   node_name      IN VARCHAR2,
   topology_name  IN VARCHAR2 DEFAULT NULL,
   remote_check   IN BOOLEAN  DEFAULT TRUE);
```

Parameters

Table 222-42 UNREGISTER_REMOTE_DATABASE Parameters

Parameter	Description
node_id	Identifier of the remote database whose data needs to be removed from the AWR.
node_name	Name of the remote database whose data needs to be removed from the AWR.
topology_name	RMF topology name of the remote database.
remote_check	If set to TRUE, the remote database's metadata is validated before removing its data from the AWR. This option requires the remote database to be available. If set to FALSE, the remote database's data is removed from the AWR without validating its metadata. This option must be selected to unregister a remote database that is not available (it is offline or there is a network outage).

Examples

This example removes the AWR data related to the remote database having the database identifier of 10:

```
EXECUTE DBMS_WORKLOAD_REPOSITORY.UNREGISTER_REMOTE_DATABASE(10);
```

UPDATE_DATAFILE_INFO Procedure

This procedure updates the data file and tablespace information stored in the Automatic Workload Repository (AWR) with the current information in the database. This procedure is useful when a data file or a tablespace has been moved or renamed. As this change is not always captured in the next snapshot, AWR report may not show the most current information.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.UPDATE_DATAFILE_INFO();
```

UPDATE_OBJECT_INFO Procedure

This procedure updates rows of WRH$_SEG_STAT_OBJ table that represent objects in the local database. It attempts to determine the current names for all object belonging to the local database, except those with 'MISSING' and/or 'TRANSIENT' values in the name columns.

The amount of work performed at each invocation of this routine may be controlled by setting the input parameter.

Syntax

```
DBMS_WORKLOAD_REPOSITORY.UPDATE_OBJECT_INFO(
   maxrows    IN  NUMBER  DEFAULT 0);
```

Parameters

Table 222-43 UPDATE_OBJECT_INFO Procedure Parameters

Parameter	Description
`maxrows`	Maximum number of rows to be updated. Default= `0`, meaning there is no limit.