

# DBMS\_WORKLOAD\_CAPTURE

The `DBMS_WORKLOAD_CAPTURE` package configures the Workload Capture system and produce the workload capture data.

Replay of this capture is implemented by way of the `DBMS_WORKLOAD_REPLAY` package.

This chapter contains the following topics:

- [Overview](#)
- [Security Model](#)
- [Summary of DBMS\\_WORKLOAD\\_CAPTURE Subprograms](#)



## See Also:

*Oracle Database Testing Guide* for more information about database replay

## DBMS\_WORKLOAD\_CAPTURE Overview

Since the capture infrastructure is instance wide (and also within an Oracle Real Application Clusters (Oracle RAC)), only one workload capture is being produced at any point in time. Thus capture interfaces do not need a state object passed in as a parameter since there is one single state at any point in time. This means that all subprograms cannot be methods of an object but are package wide PL/SQL subprograms.

## DBMS\_WORKLOAD\_CAPTURE Security Model

The security model describes the privileges needed for using `DBMS_WORKLOAD_CAPTURE`.

The following code describes the minimal set of privileges required to:

- Create directory objects
- Operate the interface provided by the `DBMS_WORKLOAD_CAPTURE` and `DBMS_WORKLOAD_REPLAY` packages
- Act as a replay client user (`wrc someuser/somepassword` or `wrc USER=someuser PASSWORD=somepassword`)

```
DROP USER rom1 CASCADE;
CREATE USER rom1 IDENTIFIED BY rom1;

GRANT EXECUTE ON DBMS_WORKLOAD_CAPTURE TO rom1;
GRANT EXECUTE ON DBMS_WORKLOAD_REPLAY TO rom1;

GRANT CREATE SESSION TO rom1;
GRANT CREATE ANY DIRECTORY TO rom1;
GRANT SELECT_CATALOG_ROLE TO rom1;
GRANT BECOME USER TO rom1;
```

Appropriate OS permissions are required to access and manipulate files and directories on both the capture and replay system. This means that the Oracle process(es) and the OS user performing the capture or replay must be able to access and manipulate at least one common directory accessible from the host where the instance is running. Additionally, the OS user performing the replay should be able to execute wrctl on hosts that are used for the replay clients and be able to access the file system appropriately to copy the capture to the replay clients' hosts if required.

## Summary of DBMS\_WORKLOAD\_CAPTURE Subprograms

This table lists the DBMS\_WORKLOAD\_CAPTURE package subprograms in alphabetical order.

**Table 220-1 DBMS\_WORKLOAD\_CAPTURE Package Subprograms**

Subprogram	Description
<a href="#">ADD_FILTER Procedures</a>	Adds a specified filter
<a href="#">DECRYPT_CAPTURE Procedure</a>	This procedure decrypts sensitive data in workload capture that was encrypted using various advanced encryption standards such as AES128, AES192, or AES256.
<a href="#">DELETE_CAPTURE_INFO Procedure</a>	Deletes the rows in the DBA_WORKLOAD_CAPTURES and DBA_WORKLOAD_FILTERS views that corresponds to the given workload capture ID
<a href="#">DELETE_FILTER Procedure</a>	Deletes a specified filter
<a href="#">ENCRYPT_CAPTURE Procedure</a>	This procedure encrypts sensitive data in workload capture using various advanced encryption standards such as AES128, AES192, or AES256.
<a href="#">EXPORT_AWR Procedure</a>	Exports the AWR snapshots associated with a given capture ID
<a href="#">FINISH_CAPTURE Procedure</a>	Finalizes the workload capture by signaling all connected sessions to stop capture, and stops future requests to the database from being captured
<a href="#">GET_CAPTURE_INFO Function</a>	Retrieves all the information regarding a workload capture present in the stipulated directory, imports the information into the DBA_WORKLOAD_CAPTURES and DBA_WORKLOAD_FILTERS views, and returns the appropriate DBA_WORKLOAD_CAPTURES.ID
<a href="#">IMPORT_AWR Function</a>	Imports the AWR snapshots associated with a given capture ID
<a href="#">REPORT Function</a>	Returns a report on the workload capture under consideration using one or more different sources
<a href="#">START_CAPTURE Procedure</a>	Initiates workload capture on all instances

### ADD\_FILTER Procedures

This procedure adds a filter to capture a subset of the workload.

#### Syntax

```
DBMS_WORKLOAD_CAPTURE.ADD_FILTER (
    fname          IN  VARCHAR2 NOT NULL,
    fattribute     IN  VARCHAR2 NOT NULL,
    fvalue         IN  VARCHAR2 NOT NULL);
```

```
DBMS_WORKLOAD_CAPTURE.ADD_FILTER (
    fname          IN  VARCHAR2 NOT NULL,
```

```

fattribute      IN   VARCHAR2 NOT NULL,
fvalue         IN   NUMBER NOT NULL);

```

## Parameters

**Table 220-2 ADD\_FILTER Procedure Parameters**

Parameter	Description
fname	Name for the filter to be added. Can be used to delete the filter later if it is not required. (Mandatory)
fattribute	Specifies the attribute on which the filter needs to be applied (Mandatory). The possible values are: <ul style="list-style-type: none"> <li>• INSTANCE_NUMBER - type NUMBER</li> <li>• USER - type STRING</li> <li>• MODULE - type STRING</li> <li>• ACTION - type STRING</li> <li>• PROGRAM - type STRING</li> <li>• SERVICE - type STRING</li> <li>• PDB_NAME - type STRING</li> </ul>
fvalue	Specifies the value to which the given attribute should be equal to for the filter to be considered active. Wildcards like '%' are acceptable for all attributes that are of type STRING. This means that the filter for a NUMBER attribute is parsed as "attribute = value", with the filter for a STRING attribute parsed as "attribute like value" (Mandatory).

## Usage Notes

- The workload capture filters work in either the `DEFAULT INCLUSION` or the `DEFAULT EXCLUSION` mode as determined by the `default_action` input to the [START\\_CAPTURE Procedure](#).
- `ADD_FILTER` adds a new filter that affects the next workload capture, and whether the filters are considered as `INCLUSION` filters or `EXCLUSION` filters depends on the value of the `default_action` input to [START\\_CAPTURE Procedure](#).
- Filters once specified are valid only for the next workload capture. If the same set of filters need to be used for subsequent capture, they need to be specified each time before the [START\\_CAPTURE Procedure](#) is executed.
- All the filters are listed in the `DBA_WORKLOAD_FILTERS` view.
- You can capture the workload for a particular PDB by specifying a filter of PDB type.

## Examples

- By default, a capture works in an `INCLUSION` mode, which records everything except for those requests that satisfy conditions of specified filters. For example, if you want to exclude all requests from `SCOTT`, you can add the following filter before starting a capture.

```
EXEC DBMS_WORKLOAD_CAPTURE.ADD_FILTER ('filter user1', 'USER', 'SCOTT');
```

- Multiple filters are evaluated according to the logical disjunction operator `OR`. Therefore, if you want to record workload for both `SCOTT` and `JOHN`, you add an additional filter:

```
EXEC DBMS_WORKLOAD_CAPTURE.ADD_FILTER ('filter user2', 'USER', 'JOHN');
```

- In a CDB, you exclude the workload of a particular PDB by the filter:

```
EXEC DBMS_WORKLOAD_CAPTURE.ADD_FILTER ('filter pdb workload', 'PDB_NAME',
'CDB1_PDB1');
```

- To use **DBMS\_APPLICATION\_INFO** to identify workload that is issued to the database:

```
DBMS_APPLICATION_INFO.SET_MODULE('ORDER_ENTRY', NULL);
-- run some SQL here
DBMS_APPLICATION_INFO.SET_ACTION('ORDER_ENTRY_LOG');
-- run logging SQL
```

- If having captured workload, you want to exclude the logging SQL from the captured, specify a filter for capture:

```
DBMS_WORKLOAD_CAPTURE.ADD_FILTER('filter logging operations', 'ACTION',
'ORDER_ENTRY_LOG');
```

- To filter out the full order entry transaction, define a filter:

```
DBMS_WORKLOAD_CAPTURE.ADD_FILTER('filter order entry', 'MODULE', 'ORDER_ENTRY');
```

## DECRYPT\_CAPTURE Procedure

This procedure decrypts sensitive data in workload capture that was encrypted using various advanced encryption standards such as AES128, AES192, or AES256.

### Syntax

```
DBMS_WORKLOAD_CAPTURE.DECRYPT_CAPTURE (
    src_dir          IN   VARCHAR2,
    dst_dir          IN   VARCHAR2);
```

### Parameters

**Table 220-3 DECRYPT\_CAPTURE Procedure Parameters**

Parameter	Description
src_dir	A directory object pointing to the workload capture to be decrypted. This parameter is case sensitive.
dst_dir	A directory object pointing to an OS path that has write permissions. The decrypted capture files will be written to this directory This parameter is case sensitive.

### Usage Notes

This procedure relies on a software keystore. The identifier is `oracle.rat.database_replay.encryption` (case-sensitive).

## DELETE\_CAPTURE\_INFO Procedure

This procedure deletes the rows in the `DBA_WORKLOAD_CAPTURES` and `DBA_WORKLOAD_FILTERS` views that corresponds to the given workload capture ID.

### Syntax

```
DBMS_WORKLOAD_CAPTURE.DELETE_CAPTURE_INFO
    capture_id      IN   NUMBER);
```

## Parameters

**Table 220-4 DELETE\_CAPTURE\_INFO Procedure Parameters**

Parameter	Description
capture_id	ID of the workload capture that needs to be deleted. Corresponds to DBA_WORKLOAD_CAPTURES.ID. (Mandatory)

## Usage Notes

Passing the ID of a capture that is in progress will first automatically stop that capture.

# DELETE\_FILTER Procedure

This procedure deletes a specified filter.

## Syntax

```
DBMS_WORKLOAD_CAPTURE.DELETE_FILTER (
    filter_name          IN  VARCHAR2(40) NOT NULL);
```

## Parameters

**Table 220-5 DELETE\_FILTER Procedure Parameters**

Parameter	Description
filter_name	Filter to be deleted

## Usage Notes

The `DELETE_FILTER` Procedure only affects filters that have not been used by any previous capture. Consequently, filters can be deleted only if they have been added using the [ADD\\_FILTER Procedures](#) after any capture has been completed. Filters that have been added using `ADD_FILTER` before a `START_CAPTURE` and `FINISH_CAPTURE` cannot be deleted anymore using this subprogram.

# ENCRYPT\_CAPTURE Procedure

This procedure encrypts sensitive data in workload capture using various advanced encryption standards such as AES128, AES192, or AES256. To encrypt capture on the fly, use encryption parameter in `START_CAPTURE` procedure.

## Syntax

```
DBMS_WORKLOAD_CAPTURE.ENCRYPT_CAPTURE
    src_dir      IN VARCHAR2,
    dst_dir      IN VARCHAR2,
    encryption   IN VARCHAR2 DEFAULT 'AES256');
```

## Parameters

**Table 220-6 ENCRYPT\_CAPTURE Procedure Parameters**

Parameter	Description
src_dir	A directory object pointing to the workload capture to be encrypted. The parameter value is case sensitive.
dst_dir	A directory object pointing to an OS path that has write permissions. The encrypted capture files will be written to this directory. The parameter value is case sensitive.
encryption	Specifies if sensitive data in workload capture is encrypted or not. The possible values are: AES128, AES192, and AES256. The default value is AES256.

## Usage Notes

For encrypted capture, the `ENCRYPT_CAPTURE` Procedure relies on a software keystore. The identifier is `oracle.rat.database_replay.encryption` (case-sensitive).

# EXPORT\_AWR Procedure

This procedure exports the AWR snapshots associated with a given capture ID.

## Syntax

```
DBMS_WORKLOAD_CAPTURE.EXPORT_AWR (  
    capture_id      IN NUMBER);
```

## Parameters

**Table 220-7 EXPORT\_AWR Procedure Parameters**

Parameter	Description
capture_id	ID of the capture whose AWR snapshots are to be exported. (Mandatory)

## Usage Notes

This procedure works only if the corresponding workload capture was performed in the current database (meaning that the corresponding row in `DBA_WORKLOAD_CAPTURES` was not created by calling the [GET\\_CAPTURE\\_INFO Function](#)) and the AWR snapshots that correspond to the original capture time period are still available.

## FINISH\_CAPTURE Procedure

This procedure signals all connected sessions to stop the workload capture and stops future requests to the database from being captured.

### Syntax

```
DBMS_WORKLOAD_CAPTURE.FINISH_CAPTURE
  timeout      IN      NUMBER DEFAULT 30
  reason       IN      VARCHAR2 DEFAULT NULL);
```

### Parameters

**Table 220-8 FINISH\_CAPTURE Procedure Parameters**

Parameter	Description
timeout	Specifies in seconds for how long the procedure should wait before it times out. Pass 0 if you want to cancel the current workload capture and not wait for any sessions to flush its capture buffers. Default value: 30 seconds
reason	Specifies a reason for calling the procedure. The reason appears in the column <code>ERROR_MESSAGE</code> of the view <code>DBA_WORKLOAD_CAPTURES</code> .

### Usage Notes

- By default, `FINISH_CAPTURE` waits for 30 seconds to receive a successful acknowledgement from all sessions in the database cluster before timing out.
- All sessions that either were in the middle of executing a user request or received a new user request, while `FINISH_CAPTURE` was waiting for acknowledgements, flush their buffers and send back their acknowledgement to `FINISH_CAPTURE`.
- If a database session remains idle (waiting for the next user request) throughout the duration of `FINISH_CAPTURE`, the session might have unflushed capture buffers and does not send its acknowledgement to `FINISH_CAPTURE`.

To avoid this, do not have sessions that remain idle (waiting for the next user request) while invoking `FINISH_CAPTURE`. Either close the database session(s) before running `FINISH_CAPTURE` or send new database requests to those sessions during `FINISH_CAPTURE`.

## GET\_CAPTURE\_INFO Function

This procedure retrieves all information regarding a workload capture present in the stipulated directory, imports the information into the `DBA_WORKLOAD_CAPTURES` and `DBA_WORKLOAD_FILTERS` views, and returns the appropriate `DBA_WORKLOAD_CAPTURES.ID`

### Syntax

```
DBMS_WORKLOAD_CAPTURE.GET_CAPTURE_INFO
  dir      IN      VARCHAR2)
RETURN NUMBER;
```

## Parameters

**Table 220-9 GET\_CAPTURE\_INFO Function Parameters**

Parameter	Description
dir	Name of the DIRECTORY object (case sensitive) where all the workload capture files are located (Mandatory)

## Usage Notes

If an appropriate row describing the capture in the stipulated directory already exists in DBA\_WORKLOAD\_CAPTURES, the [GET\\_CAPTURE\\_INFO Function](#) simply returns that row's DBA\_WORKLOAD\_CAPTURES.ID. If no existing row matches the capture present in the stipulated directory a new row is inserted to DBA\_WORKLOAD\_CAPTURES and that row's ID is returned.

# IMPORT\_AWR Function

This procedure imports the AWR snapshots associated with a given capture ID provided those AWR snapshots were exported earlier from the original capture system using the EXPORT\_AWR procedure.

## Syntax

```
DBMS_WORKLOAD_CAPTURE.IMPORT_AWR (
    capture_id      IN    NUMBER,
    staging_schema   IN    VARCHAR2,
    force_cleanup    IN    BOOLEAN DEFAULT FALSE)
RETURN NUMBER;
```

## Parameters

**Table 220-10 IMPORT\_AWR Function Parameters**

Parameter	Description
capture_id	ID of the capture whose AWR snapshots should be imported. (Mandatory)
staging_schema	Name of a valid schema in the current database which can be used as a staging area while importing the AWR snapshots from the capture directory to the SYS AWR schema. The SYS schema is not a valid input. (Mandatory, Case sensitive).



**Table 220-10 (Cont.) IMPORT\_AWR Function Parameters**

Parameter	Description
force_cleanup	<p>Values:</p> <ul style="list-style-type: none"> <li>TRUE - any AWR data present in the given staging_schema are removed before the actual import operation. All tables with names that match any of the tables in AWR are dropped before the actual import. This typically is equivalent to dropping all tables returned by the following SQL:</li> </ul> <pre>SELECT table_name FROM dba_tables WHERE owner = staging_schema AND table_name like 'WR_\$%';</pre> <p>Use this option only if you are sure that there are no important data in any such tables in the staging_schema.</p> <ul style="list-style-type: none"> <li>FALSE - (default) no tables dropped from the staging_schema prior to the import operation</li> </ul>

### Return Values

Returns the new randomly generated database ID that was used to import the AWR snapshots. The same value can be found in the AWR\_DBID column in the DBA\_WORKLOAD\_CAPTURES view.

### Usage Notes

IMPORT\_AWR fails if the staging\_schema provided as input contains any tables with the same name as any of the AWR tables, such as WRM\$\_SNAPSHOT or WRH\$\_PARAMETER. Please drop any such tables in the staging\_schema before invoking IMPORT\_AWR.

### Related Topics

- [EXPORT\\_AWR Procedure](#)  
This procedure exports the AWR snapshots associated with a given capture ID.

## REPORT Function

This function generates a report on the stipulated workload capture.

### Syntax

```
DBMS_WORKLOAD_CAPTURE.REPORT (
    capture_id      IN    NUMBER,
    format          IN    VARCHAR2)
RETURN CLOB;
```

## Parameters

**Table 220-11 REPORT Function Parameters**

Parameter	Description
capture_id	ID of the workload capture whose capture report is required. (Mandatory)  This relates to the directory that contains the workload capture on which the Report needs to be generated. Should be a valid DIRECTORY object that points to a valid directory in the host system that contains a workload capture.
format	Specifies the report format. Valid values are DBMS_WORKLOAD_CAPTURE.TYPE_TEXT and DBMS_WORKLOAD_CAPTURE.TYPE_HTML.(Mandatory)

## Return Values

The report body in the desired format returned as a CLOB.

**Table 220-12 Constants Used by Report Function**

Constant	Type	Value	Description
TYPE_HTML	VARCHAR2 (4)	'HTML'	Generates the HTML version of the report
TYPE_TEXT	VARCHAR2 (4)	'TEXT'	Used as input to the format argument to generate the text version of the report

# START\_CAPTURE Procedure

This procedure initiates workload capture on all instances.

## Syntax

```
DBMS_WORKLOAD_CAPTURE.START_CAPTURE (
    name           IN  VARCHAR2,
    dir            IN  VARCHAR2,
    duration       IN  NUMBER   DEFAULT NULL,
    default_action IN  VARCHAR2 DEFAULT 'INCLUDE',
    auto_unrestrict IN  BOOLEAN  DEFAULT TRUE,
    capture_sts    IN  BOOLEAN  DEFAULT FALSE,
    sts_cap_interval IN NUMBER  DEFAULT 300,
    plsqli_mode    IN  VARCHAR2 DEFAULT 'TOP_LEVEL',
    encryption     IN  VARCHAR2 DEFAULT NULL);
```

## Parameters

**Table 220-13 START\_CAPTURE Procedure Parameters**

Parameter	Description
name	Name of the workload capture. Allows the workload capture to be given a label, such as "Thanksgiving weekend" or "Christmas peak workload" for future reference. The workload capture's name is preserved along with the captured workload actions. (Mandatory)
dir	Name of the DIRECTORY object (case sensitive) where all the workload capture files are stored. Should contain enough space to hold all the workload capture files. (Mandatory)
duration	Optional input to specify the duration (in seconds) for which the workload needs to be captured. DEFAULT is NULL which means that workload capture continues until the user executes DBMS_WORKLOAD_CAPTURE.FINISH_CAPTURE.
default_action	<p>Can be either INCLUDE or EXCLUDE. Determines whether, by default, every user request should be captured or not. Also determines whether the workload filters specified should be considered as INCLUSION filters or EXCLUSION filters.</p> <ul style="list-style-type: none"> <li>If INCLUDE, by default all user requests to the database are captured, except for the part of the workload defined by the filters. In this case, all the filters specified using the <a href="#">ADD_FILTER Procedures</a> are treated as EXCLUSION filters, determining the workload that is not captured. (DEFAULT, and so all the filters specified are assumed to be EXCLUSION filters.)</li> <li>If EXCLUDE, by default no user request to the database is captured, except for the part of the workload defined by the filters. In this case, all the filters specified using the <a href="#">ADD_FILTER Procedures</a> are treated as INCLUSION filters, determining the workload that is captured.</li> </ul>
auto_unrestrict	<p>Can be either TRUE or FALSE.</p> <ul style="list-style-type: none"> <li>If TRUE, all instances started up in RESTRICTED mode using STARTUP RESTRICT are automatically unrestricted upon a successful START_CAPTURE. (DEFAULT)</li> <li>If FALSE, no database instance is automatically unrestricted.</li> </ul>
capture_sts	<p>If this parameter is TRUE, a SQL tuning set capture is also started in parallel with workload capture. The resulting SQL tuning set can be exported using the <a href="#">EXPORT_AWR Procedure</a> along with the AWR data. Currently, parallel STS capture is not supported in an Oracle RAC environment, so this parameter has no effect if used in that context. Capture filters defined using the DBMS_WORKLOAD_REPLAY interface do not apply to the SQL tuning set capture. The calling user must have the appropriate privileges ('ADMINISTER SQL TUNING SET').</p> <p>If starting SQL set capture fails, workload capture is stopped. The reason is stored in DBA_WORKLOAD_CAPTURES.ERROR_MESSAGE. The default value is FALSE.</p>
sts_cap_interval	Specifies the capture interval of the SQL set capture from the cursor cache in seconds. The default value is 300.

**Table 220-13 (Cont.) START\_CAPTURE Procedure Parameters**

Parameter	Description
<code>plsql_mode</code>	Specifies the PL/SQL capture mode: <ul style="list-style-type: none"> <li>• <code>TOP_LEVEL</code> — only top-level PL/SQL calls are captured</li> <li>• <code>EXTENDED</code> — both top-level PL/SQL calls and SQL called from PL/SQL are captured</li> </ul>
<code>encryption</code>	Specify if sensitive data in workload capture is encrypted or not. The possible values are: <ul style="list-style-type: none"> <li>• <code>NULL</code> — capture files are not encrypted</li> <li>• <code>AES128</code> — capture files are encrypted using AES128</li> <li>• <code>AES192</code> — capture files are encrypted using AES192</li> <li>• <code>AES256</code> — capture files is encrypted using AES256</li> </ul> The default value is <code>NULL</code> .

### Usage Notes

- All user requests sent to database after a successful invocation of `START_CAPTURE` are recorded in the given `dir` directory for the given duration provided that one was specified. If no duration was specified, the capture lasts indefinitely until the [FINISH\\_CAPTURE Procedure](#) is executed.
- A workload capture once started continues to record user requests across database instance shutdowns and startups for the specified duration, or until `FINISH_CAPTURE` is executed, whichever occurs first.
- One can use workload filters (as described with regard to the [ADD\\_FILTER Procedures](#)) to capture only a subset of the user requests sent to the database. By default, when no workload filters are defined, all user requests are captured.
- Workload that is initiated from Oracle Database background processes (such as `SMON`, `PMON`, `MMON`) and Oracle Database Scheduler Jobs (as detailed in the [DBMS\\_SCHEDULER](#) package) is not captured, no matter how the workload filters are defined. These activities should happen automatically on an appropriately configured replay system.
- By default, all database instances that were started up in `RESTRICTED` mode using `STARTUP RESTRICT` are `UNRESTRICTED` upon a successful invocation of `START_CAPTURE`. Use `FALSE` for the `auto_unrestrict` input parameter, if you do not want this behavior.
- It is important to have a well-defined starting point for the workload so that the replay system can be restored to that point before initiating a replay of the captured workload. To have a well-defined starting point for the workload capture, it is preferable not to have any active user sessions when `START_CAPTURE` is executed. If ongoing sessions have ongoing transactions, those transactions are not replayed properly in subsequent database replays, since only that part of the transaction whose calls were executed after `START_CAPTURE` are replayed.
- For encrypted capture, the `START_CAPTURE` Procedure relies on a software keystore. The identifier is `oracle.rat.database_replay.encryption` (case-sensitive).
- You must configure a software keystore in `auto-login` mode. Otherwise, if the database is bounced during capture, the capture is automatically terminated.