103
DBMS_ILM

The `DBMS_ILM` package provides an interface for implementing Information Lifecycle Management (ILM) strategies using Automatic Data Optimization (ADO) policies.

This chapter contains the following topics:

> **See Also:**
>
> - *Oracle Database VLDB and Partitioning Guide* for information about managing Automatic Data Optimization (ADO) with this package
> - DBMS_ILM_ADMIN
> - DBMS_HEAT_MAP

DBMS_ILM Overview

To implement your ILM strategy, you can use Heat Map in Oracle Database to track data access and modification. You can also use Automatic Data Optimization (ADO) to automate the compression and movement of data between different tiers of storage within the database. The `DBMS_ILM` package supports immediate evaluation or execution of ADO related tasks. T

he package supports the following two ways for scheduling ADO actions.

- A database user schedules immediate ADO policy execution on a set of objects.
- A database user views the results of evaluation of ADO policies on a set of objects. The user then adds or deletes objects to this set and reviews the results of ADO policy evaluation again. The user repeats this step to determine the set of objects for ADO execution. The user can then schedule ADO actions for immediate execution on this set of objects.

The following procedures support the two usage modes. Before describing the procedures, we introduce the notion of an ADO task as an entity that helps to track a particular evaluation or (an evaluation and execution) of ADO policies. A particular ADO task could be in one of the following states.

- Inactive
- Active
- Completed

DBMS_ILM Security Model

The DBMS_ILM package runs under invoker's rights.

DBMS_ILM Constants

The table in this topic lists the constants used by the DBMS_ILM package.

Table 103-1 DBMS_ILM Constants

Constant	Value	Type	Description
ILM_ALL_POLICIES	'ALL POLICIES'	VARCHAR2(20)	Selects all ADO policies on an object
ILM_EXECUTION_OFFLINE	1	NUMBER	Specifies that the object may be offline while ADO action is performed
ILM_EXECUTION_ONLINE	2	NUMBER	Specifies that the object should be online while ADO action is performed
SCOPE_DATABASE	1	NUMBER	Selects all ADO policies in the database
SCOPE_SCHEMA	2	NUMBER	Selects all ADO policies in the current schema
SCHEDULE_IMMEDIATE	1	NUMBER	Schedules ADO task for immediate execution
ARCHIVE_STATE_ACTIVE	'0'	VARCHAR2(1)	Represents the value of the ORA_ARCHIVE_STATE column of a row-archival enabled table that would make the row active
ARCHIVE_STATE_ARCHIVED	'1'	VARCHAR2(1)	Represents the value of the ORA_ARCHIVE_STATE column of a row-archival enabled table that would make the row inactive

DBMS_ILM Exceptions

The table in this topic lists the exceptions raised by the DBMS_ILM package.

Table 103-2 DBMS_ILM Exceptions

Exception	Error Code	Description
INVALID_ARGUMENT_VALUE	38327	Invalid argument value
INVALID_ILM_DICTIONARY	38328	Inconsistent dictionary state
INTERNAL_ILM_ERROR	38329	Internal error
INSUFFICIENT_PRIVILEGES	38330	Insufficient privileges

Summary of DBMS_ILM Subprograms

Thi table lists and describes the `DBMS_ILM` package subprograms.

Table 103-3 DBMS_ILM Package Subprograms

Subprogram	Description
ADD_TO_ILM Procedure	Adds the object specified through the argument to a particular ADO task and evaluates the ADO policies on this object
ARCHIVESTATENAME Function	Returns the value of the `ORA_ARCHIVE_STATE` column of a row-archival enabled table
EXECUTE_ILM Procedure	Executes an ADO task.
EXECUTE_ILM_TASK Procedure	Executes an ADO task that has been evaluated previously
PREVIEW_ILM Procedure	Evaluates all ADO policies in the scope specified by means of an argument
REMOVE_FROM_ILM Procedure	Removes the object specified through the argument from a particular ADO task
STOP_ILM Procedure	Stops ADO-related jobs created for a particular ADO task

ADD_TO_ILM Procedure

This procedure adds the object specified through the argument to a particular ADO task and evaluates the ADO policies on this object.

The procedure can only be executed on an ADO task in an inactive state. The results of the ADO policy evaluation on this object can be viewed using the appropriate views depending on role and access (`USER_ILMTASKS` or `DBA_ILMTASKS`, `USER_ILMEVALUATIONDETAILS` or `DBA_ILMEVALUATIONDETAILS`, `USER_ILMRESULTS` or `DBA_ILMRESULTS`).

Syntax

```
DBMS_ILM.ADD_TO_ILM (
   task_id           IN    NUMBER,
   owner             IN    VARCHAR2,
   object_name       IN    VARCHAR2,
   subobject_name    IN    VARCHAR2 DEFAULT NULL);
```

Parameters

Table 103-4 *ADD_TO_ILM Procedure Parameters*

Parameter	Description
`task_id`	Identifies a particular ADO task
`owner`	Owner of the object
`object_name`	Name of the object
`subobject_name`	Name of the subobject (partition name in the case of partitioned tables)

ARCHIVESTATENAME Function

This function returns the value of the `ORA_ARCHIVE_STATE` column of a row-archival enabled table.

Syntax

```
DBMS_ILM.ARCHIVESTATENAME (
   value      IN  VARCHAR2)
 RETURN VARCHAR2;
```

Parameters

Table 103-5 *ARCHIVESTATENAME Function Parameters*

Parameter	Description
`value`	Value for which the archive state name is to be returned

Usage Notes

Returns `ARCHIVE_STATE_ACTIVE` for `0`, `ARCHIVE_STATE_ARCHIVED` for others

> **See Also:**
>
> "Using In-Database Archiving" in *Oracle Database VLDB and Partitioning Guide*

EXECUTE_ILM Procedure

This procedure executes an ADO task.

There are two overloads to this procedure. The first overload executes an ADO task for a set of objects without having evaluated them previously. The second overload executes ADO policies for a specific object.

Syntax

```
DBMS_ILM.EXECUTE_ILM (
   task_id             OUT    NUMBER,
   ilm_scope           IN     NUMBER DEFAULT SCOPE_SCHEMA,
   execution_mode      IN     NUMBER DEFAULT ILM_EXECUTION_ONLINE);

DBMS_ILM.EXECUTE_ILM (
   owner               IN     VARCHAR2,
   object_name         IN     VARCHAR2,
   task_id             OUT    NUMBER,
   subobject_name      IN     VARCHAR2 DEFAULT NULL,
   policy_name         IN     VARCHAR2 DEFAULT ILM_ALL_POLICIES,
   execution_mode      IN     NUMBER DEFAULT ILM_EXECUTION_ONLINE);
```

Parameters

Table 103-6 *EXECUTE_ILM Procedure Parameters*

Parameter	Description
task_id	Identifies a particular ADO task
ilm_scope	Determines the set of objects considered for ADO execution. The default is to consider only the objects in the schema.
execution_mode	Whether the ADO task be executed online (ILM_EXECUTION_ONLINE) or offline (ILM_EXECUTION_OFFLINE)
owner	Owner of the object
object_name	Name of the object
subobject_name	Name of the subobject (partition name in the case of partitioned tables)
policy_name	Name of the ADO policy to be evaluated on the object. The package constant ILM_ALL_POLICIES should be used if all ADO policies on an object should be evaluated.

Usage Notes

- The EXECUTE_ILM procedure can be used by users who want more control of when ADO is performed, and who do not want to wait until the next maintenance window.
- The procedure executes like a DDL in that it auto commits before and after the ADO task and related jobs are created.

EXECUTE_ILM_TASK Procedure

This procedure executes an ADO task that has been evaluated previously and moves it to an active state.

Syntax

```
DBMS_ILM.EXECUTE_ILM_TASK (
   task_id             IN     NUMBER,
   execution_mode      IN     NUMBER DEFAULT ILM_EXECUTION_ONLINE);
   execution_schedule  IN     NUMBER DEFAULT SCHEDULE_IMMEDIATE);
```

Parameters

Table 103-7 *EXECUTE_ILM_TASK Procedure Parameters*

Parameter	Description
task_id	Identifies a particular ADO task
execution_mode	Whether the ADO task be executed online (ILM_EXECUTION_ONLINE) or offline (ILM_EXECUTION_OFFLINE)
execution_schedule	Identifies when the ADO task should be executed.Currently, the only choice available is immediate scheduling of ADO jobs

PREVIEW_ILM Procedure

This procedure evaluates the ADO policies on the objects specified using the `ILM_SCOPE` argument.

It returns a number as `task_id` which identifies a particular ADO task. This can be used to view the results of the policy evaluation in the appropriate views depending on role and access (`USER_ILMTASKS` or `DBA_ILMTASKS`, `USER_ILMEVALUATIONDETAILS` or `DBA_ILMEVALUATIONDETAILS`, `USER_ILMRESULTS` or `DBA_ILMRESULTS`).

The `PREVIEW_ILM` procedure leaves the ADO task in an inactive state. Once you have previewed the results, you can add or delete objects to this task.

Syntax

```
DBMS_ILM.PREVIEW_ILM (
   task_id           OUT    NUMBER,
  ilm_scope          IN     NUMBER DEFAULT SCOPE_SCHEMA);
```

Parameters

Table 103-8 *PREVIEW_ILM Procedure Parameters*

Parameter	Description
`task_id`	Identifies a particular ADO task
`ilm_scope`	Identifies the scope of execution. Should be either `SCOPE_DATABASE` or `SCOPE_SCHEMA` as described in Constants

REMOVE_FROM_ILM Procedure

This procedure removes the object specified through the argument from a particular ADO task.

The procedure can only be executed on an ADO task in an inactive state.

Syntax

```
DBMS_ILM.REMOVE_FROM_ILM (
   task_id           IN    NUMBER,
   owner             IN    VARCHAR2,
   object_name       IN    VARCHAR2,
   subobject_name    IN    VARCHAR2 DEFAULT NULL);
```

Parameters

Table 103-9 *REMOVE_FROM_ILM Procedure Parameters*

Parameter	Description
`task_id`	Identifies a particular ADO task
`owner`	Owner of the object
`object_name`	Name of the object
`subobject_name`	Name of the subobject (partition name in the case of partitioned tables)

STOP_ILM Procedure

This procedure terminates ILM ADO jobs associated to a particular task Id or job name.

Syntax

```
DBMS_ILM.STOP_ILM (
   task_id               IN         NUMBER,
   p_drop_running_jobs   IN         BOOLEAN  DEFAULT FALSE),
   p_jobname             IN         VARCHAR2 DEFAULT NULL);
```

Parameters

Table 103-10 *STOP_ILM Procedure Parameters*

Parameter	Description
task_id	Number that uniquely identifies a particular ADO task
p_drop_running_jobs	Determines whether running jobs are dropped
p_jobname	Name of job to be terminated