212

DBMS_UNDO_ADV

The Undo Advisor assists in correctly sizing the undo tablespace and sets the low threshold value of the Undo Retention Period for any Oracle Flashback requirements. The Undo Advisor can also be used to estimate the undo tablespace required for migration from manual to automatic undo management before actually creating the new undo tablespace.

The `DBMS_UNDO_ADVISOR` package provides subprograms to manage the execution of the Undo Advisor feature.

The Undo Advisor relies for its analysis on data collected in the Automatic Workload Repository (AWR). Thus, it is important that the AWR have adequate workload statistics available so that the Undo Advisor can make accurate recommendations. For newly created databases, adequate statistics may not be available immediately. In such cases, continue to use the default auto-extending undo tablespace until at least one workload cycle completes.

This chapter contains the following topic:

Summary of DBMS_UNDO_ADV Subprograms

This topic lists the `DBMS_UNDO_ADV` subprograms in alphabetical order and briefly describes them.

Table 212-1 DBMS_UNDO_ADV Package Subprograms

Subprogram	Description
BEST_POSSIBLE_RETENTION Function	Returns the best possible value for the `undo_retention` parameter that the current undo tablespace can satisfy in order to maximize the usage of the current undo tablespace based on the historical information of the given period.
LONGEST QUERY Function	This function returns the duration of the longest query, in seconds, for a given period.
RBU_MIGRATION Function	Estimates the undo tablespace needed for migration from manual to automatic undo management.
REQUIRED_RETENTION Function	Returns the required value for the `undo_retention` parameter to satisfy the longest query based on undo statistics available for a given period.
REQUIRED_UNDO_SIZE Function	Returns the required undo tablespace size in MB to satisfy a certain undo retention value based on undo statistics available for a given period.
UNDO_ADVISOR Function	Uses the advisor framework to check if there is any problem with the current instance. It also provides recommendations.
UNDO_AUTOTUNE Function	Determines whether auto-tuning of undo retention is enabled for the current undo tablespace.

Table 212-1 (Cont.) DBMS_UNDO_ADV Package Subprograms

Subprogram	Description
UNDO_HEALTH Function	Checks whether there is any problem with the current setting of undo retention and undo tablespace size based on the historical information of a given period and provides recommendations to fix the problem.
UNDO_INFO Function	Retrieves information about the undo tablespace of the current instance.

BEST_POSSIBLE_RETENTION Function

This function returns the best possible value for the `undo_retention` parameter that the current undo tablespace can satisfy in order to maximize the usage for the current undo tablespace based on the historical information of a given period.

Syntax

Viewing the output using the historical information in memory:

```
DBMS_UNDO_ADV.BEST_POSSIBLE_RETENTION()
RETURN NUMBER;
```

Viewing the output using start time and end time:

```
DBMS_UNDO_ADV.BEST_POSSIBLE_RETENTION(
    start_time IN DATE,
    end_time   IN DATE)
RETURN NUMBER;
```

Viewing the output using begin and end AWR snapshot ID:

```
DBMS_UNDO_ADV.BEST_POSSIBLE_RETENTION(
    begin_snap IN NUMBER,
    end_snap   IN NUMBER)
RETURN NUMBER;
```

Parameters

Table 212-2 BEST_POSSIBLE_RETENTION Function Parameters

Parameter	Description
`start_time`	Start time of the given period.
`end_time`	End time of the given period.
`begin_snap`	Begin snapshot identifier. It is based on historical information in AWR from the `begin_snap` identifier.
`end_snap`	End snapshot identifier. It is based on historical information in AWR until the `end_snap` identifier.

RBU_MIGRATION Function

This function estimates the undo tablespace needed for migration from manual to automatic undo management.

If you are currently using manual undo management (rollback segments) to manage undo space, then Oracle recommends migrating to automatic undo management. You must first create an undo tablespace, before opening a newly upgraded database. The required size of undo tablespace depends on the system workload and Flashback requirements.

> **Note:**
>
> The `RBU_MIGRATION` function should be called only when `undo_management = manual`.

Syntax

Viewing the output using the historical information in memory:

```
DBMS_UNDO_ADV.RBU_MIGRATION()
RETURN NUMBER;
```

Viewing the output using the Start Time and End Time:

```
DBMS_UNDO_ADV.RBU_MIGRATION(
    START_TIME IN DATE,
    END_TIME   IN DATE)
RETURN NUMBER;
```

Parameters

Table 212-3 RBU_MIGRATION Function Parameters

Parameter	Description
`START_TIME`	Start time of the given period.
`END_TIME`	End time of the given period.

LONGEST_QUERY Function

This function returns the duration of the longest query, in seconds, for a given period.

Syntax

Viewing the output using the historical information in memory:

```
DBMS_UNDO_ADV.LONGEST_QUERY()
RETURN NUMBER;
```

Viewing the output using start time and end time:

```
DBMS_UNDO_ADV.LONGEST_QUERY(
    start_time IN DATE,
    end_time   IN DATE)
RETURN NUMBER;
```

Viewing the output using begin and end AWR snapshot ID:

```
DBMS_UNDO_ADV.LONGEST_QUERY(
    begin_snap IN NUMBER,
    end_snap   IN NUMBER)
RETURN NUMBER;
```

Parameters

Table 212-4 LONGEST_QUERY Function Parameters

Parameter	Description
`start_time`	Start time for the specified period.
`end_time`	End time for the specified period.
`begin_snap`	Begin snapshot identifier. It is based on historical information in AWR from the `begin_snap` identifier.
`end_snap`	End snapshot identifier. It is based on historical information in AWR until the `end_snap` identifier.

REQUIRED_RETENTION Function

This function returns the required value for the `undo_retention` parameter to satisfy the longest query based on undo statistics available for a given period.

Syntax

Viewing the output using the historical information in memory:

```
DBMS_UNDO_ADV.REQUIRED_RETENTION()
RETURN NUMBER;
```

Viewing the output using start time and end time:

```
DBMS_UNDO_ADV.REQUIRED_RETENTION(
    start_time IN DATE,
    end_time   IN DATE)
RETURN NUMBER;
```

Viewing the output using begin and end AWR snapshot ID:

```
DBMS_UNDO_ADV.REQUIRED_RETENTION(
    begin_snap IN NUMBER,
```

```
    end_snap   IN NUMBER)
RETURN NUMBER;
```

Parameters

Table 212-5 REQUIRED_RETENTION Function Parameters

Parameter	Description
start_time	Start time of the given period.
end_time	End time of the given period.
begin_snap	Begin snapshot identifier. It is based on historical information in AWR from the begin_snap identifier.
end_snap	End snapshot identifier. It is based on historical information in AWR until the end_snap identifier.

REQUIRED_UNDO_SIZE Function

This function returns the required undo tablespace size (in MB) to satisfy certain undo retention value based on undo statistics available for a given period.

> **Note:**
>
> Zero will be returned if the information about the given period is not available.

Syntax

Viewing the output using the historical information in memory:

```
DBMS_UNDO_ADV.REQUIRED_UNDO_SIZE(
    retention IN NUMBER)
RETURN NUMBER;
```

Viewing the output using start time and end time:

```
DBMS_UNDO_ADV.REQUIRED_UNDO_SIZE(
    retention  IN NUMBER,
    start_time IN DATE,
    end_time   IN DATE)
RETURN NUMBER;
```

Viewing the output using begin and end AWR snapshot ID:

```
DBMS_UNDO_ADV.REQUIRED_UNDO_SIZE(
    retention  IN NUMBER,
    begin_snap IN NUMBER,
    end_snap   IN NUMBER)
RETURN NUMBER;
```

Parameters

Table 212-6 REQUIRED_UNDO_SIZE Function Parameters

Parameter	Description
retention	Retention value you want to set for the undo_retention init.ora parameter.
start_time	Start time of the given period.
end_time	End time of the given period.
begin_snap	Begin snapshot identifier. It is based on historical information in AWR from the begin_snap identifier.
end_snap	End snapshot identifier. It is based on historical information in AWR until the end_snap identifier.

UNDO_ADVISOR Function

This function uses the advisor framework to check if there is any problem with the current instance and provide recommendations.

> **Note:**
>
> This function should be used when undo_management is set to auto.

Syntax

Viewing the output using the historical information in memory:

```
DBMS_UNDO_ADV.UNDO_ADVISOR(
    instance_id IN NUMBER)
RETURN VARCHAR2;
```

Viewing the output using start time and end time:

```
DBMS_UNDO_ADV.UNDO_ADVISOR(
    start_time  IN DATE,
    end_time    IN DATE,
    instance_id IN NUMBER)
RETURN VARCHAR2;
```

Viewing the output using begin and end AWR snapshot ID:

```
DBMS_UNDO_ADV.UNDO_ADVISOR(
    begin_snap  IN NUMBER,
    end_snap    IN NUMBER,
    instance_id IN NUMBER)
RETURN VARCHAR2;
```

Parameters

Table 212-7 UNDO_ADVISOR Function Parameters

Parameter	Description
start_time	Start time of the given period.
end_time	End time of the given period.
begin_snap	Begin snapshot identifier. It is based on historical information in AWR from the begin_snap identifier.
end_snap	End snapshot identifier. It is based on historical information in AWR until the end_snap identifier.
instance_id	Instance ID of the current instance.

UNDO_AUTOTUNE Function

This function finds out whether the auto-tuning of undo retention is enabled for the current undo tablespace.

Syntax

```
DBMS_UNDO_ADV.UNDO_AUTOTUNE(
     chk OUT BOOLEAN)
RETURN BOOLEAN;
```

Parameters

Table 212-8 UNDO_AUTOTUNE Function Parameters

Parameter	Description
chk	TRUE if auto-tuning of undo retention is enabled, FALSE otherwise.

UNDO_HEALTH Function

Checks whether there is any problem with the current setting of undo retention and undo tablespace size based on the historical information of a given period and provides recommendations to fix the problem.

If the return value is 0, no problem is found. Otherwise, parameter prob and reco are the problem and recommendation on fixing the problem.

Syntax

Viewing the output using the historical information in memory:

```
DBMS_UNDO_ADV.UNDO_HEALTH(
    prob OUT VARCHAR2,
    reco OUT VARCHAR2,
    rtn1 OUT VARCHAR2,
```

```
    retn OUT NUMBER,
    utbs OUT NUMBER);
```

Viewing the output using start time and end time:

```
DBMS_UNDO_ADV.UNDO_HEALTH(
    prob OUT VARCHAR2,
    reco OUT VARCHAR2,
    rtn1 OUT VARCHAR2,
    retn OUT NUMBER,
    utbs OUT NUMBER)
RETURN NUMBER;
```

Viewing the output using begin and end AWR snapshot ID:

```
DBMS_UNDO_ADV.UNDO_HEALTH(
    begin_snap IN  NUMBER,
    end_snap   IN  NUMBER,
    prob       OUT VARCHAR2,
    reco       OUT VARCHAR2,
    rtn1       OUT VARCHAR2,
    retn       OUT NUMBER,
    utbs       OUT NUMBER)
RETURN NUMBER;
```

Parameters

Table 212-9 UNDO_HEALTH Function Parameters

Parameter	Description
`start_time`	Start time of the given period.
`end_time`	End time of the given period.
`begin_snap`	Begin snapshot identifier. It is based on historical information in AWR from the `begin_snap` identifier.
`end_snap`	End snapshot identifier. It is based on historical information in AWR until the `end_snap` identifier.
`prob`	Problem that is being diagnosed. For example, long running query may fail or undo tablespace cannot satisfy `undo_retention`.
`reco`	Recommendation for fixing the problem.
`rtn1`	Rationale for the recommendation.
`retn`	The numerical value of retention if the recommendation is to change retention.
`utbs`	The numerical value of undo tablespace size (in MB) if the recommendation is to change undo tablespace size.

UNDO_INFO Function

This function retrieves information about the undo tablespace of the current instance.

This function returns the undo tablespace name, maximum possible size for the undo tablespace, current undo retention value, and boolean values to verify if it is auto-extensible or if the undo tablespace has guaranteed undo retention.

Syntax

```
DBMS_UNDO_ADV.UNDO_INFO(
    tbs_name       OUT VARCHAR2,
    tbs_size       OUT NUMBER,
    tbs_autoextend OUT BOOLEAN,
    tbs_retention  OUT NUMBER,
    tbs_guarantee  OUT BOOLEAN)
RETURN BOOLEAN;
```

Parameters

Table 212-10 UNDO_INFO Function Parameters

Parameter	Description
`tbs_name`	Name of the current undo tablespace the instance is using.
`tbs_size`	The size of the undo tablespace in MB, if the undo tablespace is `fixed-sized`. If the tablespace is `auto_extensible`, it is the maximum possible size of the undo tablespace in MB.
`tbs_autoextend`	`TRUE` if the undo tablespace is extensible, `FALSE` otherwise.
`tbs_retention`	The value of the `undo_retentioninit.ora` parameter.
`tbs_guarantee`	`TRUE` if the undo tablespace has guaranteed retention, `FALSE` otherwise.