167
DBMS_RESUMABLE

With the DBMS_RESUMABLE package, you can suspend large operations that run out of space or reach space limits after executing for a long time, fix the problem, and make the statement resume execution. In this way you can write applications without worrying about running into space-related errors.

This chapter contains the following topics:

- Operational Notes
- Summary of DBMS_RESUMABLE Subprograms

DBMS_RESUMABLE Operational Notes

When you suspend a statement, you should log the suspension in the alert log. You should also register a procedure to be executed when the statement is suspended. Using a view, you can monitor the progress of the statement and indicate whether the statement is currently executing or suspended.

Suspending a statement automatically results in suspending the transaction. Thus all transactional resources are held during a statement suspend and resume. When the error condition disappears, the suspended statement automatically resumes execution. A resumable space allocation can be suspended and resumed multiple times during execution.

A suspension timeout interval is associated with resumable space allocations. A resumable space allocation that is suspended for the timeout interval (the default is two hours) wakes up and returns an exception to the user. A suspended statement may be forced to throw an exception using the DBMS_RESUMABLE.ABORT() procedure.

Summary of DBMS_RESUMABLE Subprograms

This table lists the DBMS_RESUMABLE subprograms and briefly describes them.

Table 167-1 ***DBMS_RESUMABLE Package Subprograms***

Subprogram	Description
ABORT Procedure	Aborts a suspended resumable space allocation
GET_SESSION_TIMEOUT Function	Returns the current timeout value of the resumable space allocations for a session with session_id
GET_TIMEOUT Function	Returns the current timeout value of resumable space allocations for the current session
SET_SESSION_TIMEOUT Procedure	Sets the timeout of resumable space allocations for a session with session_id
SET_TIMEOUT Procedure	Sets the timeout of resumable space allocations for the current session
SPACE_ERROR_INFO Function	Looks for space-related errors in the error stack, otherwise returning FALSE

ABORT Procedure

This procedure aborts a suspended resumable space allocation.

The parameter session_id is the session ID in which the statement is executed. For a parallel DML/DDL, session_id is any session ID that participates in the parallel DML/DDL. This operation is guaranteed to succeed. The procedure can be called either inside or outside of the AFTER SUSPEND trigger.

Syntax

```
DBMS_RESUMABLE.ABORT (
   session_id  IN NUMBER);
```

Parameters

Table 167-2 ABORT Procedure Parameters

Parameter	Description
session_id	The session identifier of the resumable space allocation.

Usage Notes

To call an ABORT procedure, you must be the owner of the session with session_id, have ALTER SYSTEM privileges, or be a DBA.

GET_SESSION_TIMEOUT Function

This function returns the current timeout value of resumable space allocations for a session with session_id.

Syntax

```
DBMS_RESUMABLE.GET_SESSION_TIMEOUT (
   session_id  IN NUMBER)
RETURN NUMBER;
```

Parameters

Table 167-3 GET_SESSION_TIMEOUT Function Parameters

Parameter	Description
session_id	The session identifier of the resumable space allocation.

Return Values

Table 167-4 GET_SESSION_TIMEOUT Function Return Values

Return Value	Description
NUMBER	The current timeout value of resumable space allocations for a session with session_id.The timeout is returned in seconds.

Usage Notes

If session_id does not exist, the GET_SESSION_TIMEOUT function returns -1.

GET_TIMEOUT Function

This function returns the current timeout value of resumable space allocations for the current session.

Syntax

```
DBMS_RESUMABLE.GET_TIMEOUT
 RETURN NUMBER;
```

Return Values

Table 167-5 GET_TIMEOUT Function Return Values

Return Value	Description
NUMBER	The current timeout value of resumable space allocations for the current session. The returned value is in seconds.

Usage Notes

If the current session is not resumable enabled, the GET_TIMEOUT function returns -1.

SET_SESSION_TIMEOUT Procedure

This procedure sets the timeout of resumable space allocations for a session with session_id.

The new timeout setting applies to the session immediately. If session_id does not exist, no operation occurs.

Syntax

```
DBMS_RESUMABLE.SET_SESSION_TIMEOUT (
   session_id   IN NUMBER,
   timeout      IN NUMBER);
```

Parameters

Table 167-6 SET_SESSION_TIMEOUT Procedure Parameters

Parameter	Description
session_id	The session identifier of the resumable space allocation.
timeout	The timeout of the resumable space allocation.

SET_TIMEOUT Procedure

This procedure sets the timeout of resumable space allocations for the current session. The new timeout setting applies to the session immediately.

Syntax

```
DBMS_RESUMABLE.SET_TIMEOUT (
   timeout   IN NUMBER);
```

Parameters

Table 167-7 SET_TIMEOUT Procedure Parameters

Parameter	Description
timeout	The timeout of the resumable space allocation.

SPACE_ERROR_INFO Function

This function looks for space-related errors in the error stack.

If it cannot find a space related error, it will return FALSE. Otherwise, TRUE is returned and information about the particular object that causes the space error is returned.

Syntax

```
DBMS_RESUMABLE.SPACE_ERROR_INFO
   error_type        OUT VARCHAR2,
   object_type       OUT VARCHAR2,
   object_owner      OUT VARCHAR2,
   table_space_name  OUT VARCHAR2,
   object_name       OUT VARCHAR2,
   sub_object_name   OUT VARCHAR2)
RETURN BOOLEAN;
```

Parameters

Table 167-8 SPACE_ERROR_INFO Function Parameters

Parameter	Description
error_type	The space error type. It will be one of the following: • NO MORE SPACE • MAX EXTENTS REACHED • SPACE QUOTA EXCEEDED

Table 167-8 (Cont.) SPACE_ERROR_INFO Function Parameters

Parameter	Description
`object_type`	The object type. It will be one of the following: • TABLE • INDEX • CLUSTER • TABLE SPACE • ROLLBACK SEGMENT • UNDO SEGMENT • LOB SEGMENT • TEMP SEGMENT • INDEX PARTITION • TABLE PARTITION • LOB PARTITION • TABLE SUBPARTITION • INDEX SUBPARTITION • LOB SUBPARTITION The type can also be `NULL` if it does not apply.
`object_owner`	The owner of the object. `NULL` if it cannot be determined.
`table_space_name`	The table space where the object resides. `NULL` if it cannot be determined.
`object_name`	The name of rollback segment, temp segment, table, index, or cluster.
`sub_object_name`	The partition name or sub-partition name of `LOB, TABLE,` or `INDEX.` `NULL` if it cannot be determined.