209

DBMS_TTS

The DBMS_TTS package checks if the transportable set is self-contained. All violations are inserted into a temporary table that can be selected from the view `TRANSPORT_SET_VIOLATIONS`.

This chapter contains the following topics:

> **See Also:**
>
> - *Oracle Database Administrator's Guide*
> - *Oracle Database Upgrade Guide*

DBMS_TTS Security Model

Only users having the `execute_catalog_role` can execute this procedure. This role is initially only assigned to user SYS.

DBMS_TTS Exceptions

The `DBMS_TTS` package creates exceptions for missing or invalid transportable tablespaces.

```
ts_not_found  EXCEPTION;
PRAGMA exception_init(ts_not_found, -29304);
ts_not_found_num NUMBER := -29304;

invalid_ts_list  EXCEPTION;
PRAGMA exception_init(invalid_ts_list, -29346);
invalid_ts_list_num NUMBER := -29346;

sys_or_tmp_ts     EXCEPTION;
PRAGMA exception_init(sys_or_tmp_ts, -29351);
sys_or_tmp_ts_num NUMBER := -29351;
```

DBMS_TTS Operational Notes

With respect to transportable tablespaces, disabled and enabled referential integrity constraints are handled differently.

- A disabled referential integrity constraint does not violate the transportability rules and is dropped during the import phase.

- An enabled referential integrity constraint violates the transportability rules if it references a table in a tablespace outside the transportable set.

Summary of DBMS_TTS Subprograms

The two procedures listed in the table are designed to be called by database administrators.

Table 209-1 DBMS_TTS Package Subprograms

Subprogram	Description
DOWNGRADE Procedure	Downgrades transportable tablespace-related data
TRANSPORT_SET_CHECK Procedure	Checks if a set of tablespaces (to be transported) is self-contained

DOWNGRADE Procedure

This procedure downgrades transportable tablespace related data in releases earlier than Oracle Database 10g.

Syntax

```
DBMS_TTS.DOWNGRADE;
```

> **Note:**
>
> This procedure has no effect in Oracle Database 10g and later releases. It is included only for backward compatibility to support references to it in existing scripts.

TRANSPORT_SET_CHECK Procedure

This procedure checks if a set of tablespaces (to be transported) is self-contained. After calling this procedure, the user may select from a view to see a list of violations, if there are any.

Syntax

```
DBMS_TTS.TRANSPORT_SET_CHECK (
   ts_list          IN CLOB,
   incl_constraints IN BOOLEAN DEFAULT FALSE,
   full_check       IN BOOLEAN DEFAULT FALSE);
```

Parameters

Table 209-2 TRANSPORT_SET_CHECK Procedure Parameters

Parameter	Description
`ts_list`	List of one or more tablespaces, separated by comma.
`incl_constraints`	`TRUE` if you want to count in referential integrity constraints when examining if the set of tablespaces is self-contained. (The `incl_constraints` parameter is a default so that `TRANSPORT_SET_CHECK` will work if it is called with only the `ts_list` argument.)

Table 209-2 (Cont.) TRANSPORT_SET_CHECK Procedure Parameters

Parameter	Description
full_check	Indicates whether a full or partial dependency check is required. If TRUE, treats all IN and OUT pointers (dependencies) and captures them as violations if they are not self-contained in the transportable set. The parameter should be set to TRUE for TSPITR or if a strict version of transportable is desired. By default the parameter is set to FALSE. It will only consider OUT pointers as violations.

Examples

If the view does not return any rows, then the set of tablespaces is self-contained. For example,

```
SQLPLUS> EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK('foo,bar', TRUE);
SQLPLUS> SELECT * FROM TRANSPORT_SET_VIOLATIONS;
```