170
DBMS_ROWID

The DBMS_ROWID package lets you create ROWIDs and obtain information about ROWIDs from PL/SQL programs and SQL statements. You can find the data block number, the object number, and other ROWID components without writing code to interpret the base-64 character external ROWID. DBMS_ROWID is intended for upgrading from Oracle database version 7 to Oracle database version 8.X.

> **Note:**
>
> DBMS_ROWID is not to be used with universal ROWIDs (UROWIDs).

This chapter contains the following topics:

- Security Model
- Types
- Exceptions
- Operational Notes
- Examples
- Summary of DBMS_ROWID Subprograms

DBMS_ROWID Security Model

This package runs with the privileges of calling user, rather than the package owner SYS.

DBMS_ROWID Types

There are four DBMS_ROWID types.

These are:

- Extension and restriction types
- Verification types
- Object types
- Conversion types

Extension and Restriction Type

The types are as follows:

- RESTRICTED—restricted ROWID
- EXTENDED—extended ROWID

For example:

```
rowid_type_restricted constant integer := 0;
rowid_type_extended   constant integer := 1;
```

> **Note:**
>
> Extended `ROWIDs` are only used in Oracle database version 8.X*i* and higher.

Verification Types

Table 170-1 Verification Types

Result	Description
`VALID`	Valid `ROWID`
`INVALID`	Invalid `ROWID`

For example:

```
rowid_is_valid   constant integer := 0;
rowid_is_invalid constant integer := 1;
```

Object Types

Table 170-2 Object Types

Result	Description
`UNDEFINED`	Object Number not defined (for restricted `ROWIDs`)

For example:

```
rowid_object_undefined constant integer := 0;
```

Conversion Types

Table 170-3 Conversion Types

Result	Description
`INTERNAL`	Convert to/from column of `ROWID` type
`EXTERNAL`	Convert to/from string format

For example:

```
rowid_convert_internal constant integer := 0;
rowid_convert_external constant integer := 1;
```

DBMS_ROWID Exceptions

This table describes the Exceptions raised by DBMS_ROWID subprograms.

Table 170-4 Exceptions

Exception	Description
ROWID_INVALID	Invalid rowid format
ROWID_BAD_BLOCK	Block is beyond end of file

For example:

```
ROWID_INVALID exception;
   pragma exception_init(ROWID_INVALID, -1410);

ROWID_BAD_BLOCK exception;
   pragma exception_init(ROWID_BAD_BLOCK, -28516);
```

DBMS_ROWID Operational Notes

These operation notes apply to DBMS_ROWID.

- Some of the functions in this package take a single parameter, such as a ROWID. This can be a character or a PL/SQL ROWID, either restricted or extended, as required.
- You can call the DBMS_ROWID functions and procedures from PL/SQL code, and you can also use the functions in SQL statements.

> **Note:**
>
> ROWID_INFO is a procedure. It can only be used in PL/SQL code.

- You can use functions from the DBMS_ROWID package just like built-in SQL functions; in other words, you can use them wherever you can use an expression. In this example, the ROWID_BLOCK_NUMBER function is used to return just the block number of a single row in the EMP table:

```
SELECT DBMS_ROWID.ROWID_BLOCK_NUMBER(rowid)
   FROM emp
   WHERE ename = 'KING';
```

- If Oracle returns the error "ORA:452, 0, 'Subprogram '%s' violates its associated pragma' for pragma restrict_references, it could mean the violation is due to:
  - A problem with the current procedure or function
  - Calling a procedure or function without a pragma or due to calling one with a less restrictive pragma
  - Calling a package procedure or function that touches the initialization code in a package or that sets the default values

DBMS_ROWID Examples

This example returns the ROWID for a row in the EMP table, extracts the data object number from the ROWID, using the ROWID_OBJECT function in the DBMS_ROWID package, then displays the object number:

```
DECLARE
  object_no   INTEGER;
  row_id      ROWID;
  ...
BEGIN
  SELECT ROWID INTO row_id FROM emp
    WHERE empno = 7499;
  object_no := DBMS_ROWID.ROWID_OBJECT(row_id);
  DBMS_OUTPUT.PUT_LINE('The obj. # is '|| object_no);
  ...
```

Summary of DBMS_ROWID Subprograms

This table lists the DBMS_ROWID subprograms and briefly describes them.

Table 170-5   DBMS_ROWID Package Subprograms

Subprogram	Description
ROWID_BLOCK_NUMBER Function	Returns the block number of a ROWID
ROWID_CREATE Function	Creates a ROWID, for testing only
ROWID_INFO Procedure	Returns the type and components of a ROWID
ROWID_OBJECT Function	Returns the object number of the extended ROWID
ROWID_RELATIVE_FNO Function	Returns the file number of a ROWID
ROWID_ROW_NUMBER Function	Returns the row number
ROWID_TO_ABSOLUTE_FNO Function	Returns the absolute file number associated with the ROWID for a row in a specific table
ROWID_TO_EXTENDED Function	Converts a ROWID from restricted format to extended
ROWID_TO_RESTRICTED Function	Converts an extended ROWID to restricted format
ROWID_TYPE Function	Returns the ROWID type: 0 is restricted, 1 is extended
ROWID_VERIFY Function	Checks if a ROWID can be correctly extended by the ROWID_TO_EXTENDED function

ROWID_BLOCK_NUMBER Function

This function returns the database block number for the input ROWID.

Syntax

```
DBMS_ROWID.ROWID_BLOCK_NUMBER (
   row_id      IN   ROWID,
   ts_type_in  IN   VARCHAR2 DEFAULT 'SMALLFILE')
  RETURN NUMBER;
```

Pragmas

```
   pragma RESTRICT_REFERENCES(rowid_block_number,WNDS,RNDS,WNPS,RNPS);
```

Parameters

Table 170-6   ROWID_BLOCK_NUMBER Function Parameters

Parameter	Description
`row_id`	`ROWID` to be interpreted
`ts_type_in`	The type of the tablespace (bigfile/smallfile) to which the row belongs

Examples

The example SQL statement selects the block number from a `ROWID` and inserts it into another table:

```
INSERT INTO T2 (SELECT dbms_rowid.rowid_block_number(ROWID, 'BIGFILE')
   FROM some_table
   WHERE key_value = 42);
```

ROWID_CREATE Function

This function lets you create a `ROWID`, given the component parts as parameters.

This is useful for testing `ROWID` operations, because only the Oracle Server can create a valid `ROWID` that points to data in a database.

Syntax

```
DBMS_ROWID.ROWID_CREATE (
   rowid_type    IN NUMBER,
   object_number IN NUMBER,
   relative_fno  IN NUMBER,
   block_number  IN NUMBER,
   row_number    IN NUMBER)
  RETURN ROWID;
```

Pragmas

```
pragma RESTRICT_REFERENCES(rowid_create,WNDS,RNDS,WNPS,RNPS);
```

Parameters

Table 170-7   ROWID_CREATE Function Parameters

Parameter	Description
`rowid_type`	Type (restricted or extended) Set the `rowid_type` parameter to 0 for a restricted `ROWID`. Set it to 1 to create an extended `ROWID`. If you specify `rowid_type` as 0, then the required `object_number` parameter is ignored, and `ROWID_CREATE` returns a restricted `ROWID`.
`object_number`	Data object number (`rowid_object_undefined` for restricted)
`relative_fno`	Relative file number

Table 170-7 (Cont.) ROWID_CREATE Function Parameters

Parameter	Description
`block_number`	Block number in this file
`row_number`	Returns row number in this block

Examples

Create a dummy extended `ROWID`:

```
my_rowid := DBMS_ROWID.ROWID_CREATE(1, 9999, 12, 1000, 13);
```

Find out what the `rowid_object` function returns:

```
obj_number := DBMS_ROWID.ROWID_OBJECT(my_rowid);
```

The variable `obj_number` now contains 9999.

ROWID_INFO Procedure

This procedure returns information about a `ROWID`, including its type (restricted or extended), and the components of the `ROWID`.

This is a procedure, and it cannot be used in a SQL statement.

Syntax

```
DBMS_ROWID.ROWID_INFO (
   rowid_in          IN   ROWID,
   rowid_type        OUT  NUMBER,
   object_number     OUT  NUMBER,
   relative_fno      OUT  NUMBER,
   block_number      OUT  NUMBER,
   row_number        OUT  NUMBER);
```

Pragmas

```
pragma RESTRICT_REFERENCES(rowid_info,WNDS,RNDS,WNPS,RNPS);
```

Parameters

Table 170-8 ROWID_INFO Procedure Parameters

Parameter	Description
`rowid_in`	`ROWID` to be interpreted. This determines if the `ROWID` is a restricted (0) or extended (1) `ROWID`.
`rowid_type`	Returns type (restricted/extended)
`object_number`	Returns data object number (`rowid_object_undefined` for restricted)
`relative_fno`	Returns relative file number
`block_number`	Returns block number in this file
`row_number`	Returns row number in this block

See Also:

"ROWID_TYPE Function"

Examples

This example reads back the values for the ROWID that you created in the ROWID_CREATE:

```
DBMS_ROWID.ROWID_INFO (
   my_rowid, rid_type, obj_num, file_num, block_num, row_num, 'BIGFILE');
```

ROWID_OBJECT Function

This function returns the data object number for an extended ROWID.

The function returns zero if the input ROWID is a restricted ROWID.

Syntax

```
DBMS_ROWID.ROWID_OBJECT (
   rowid_id IN ROWID)
  RETURN NUMBER;
```

Pragmas

```
   pragma RESTRICT_REFERENCES(rowid_object,WNDS,RNDS,WNPS,RNPS);
```

Parameters

Table 170-9 ROWID_OBJECT Function Parameters

Parameter	Description
row_id	ROWID to be interpreted

Note:

The ROWID_OBJECT_UNDEFINED constant is returned for restricted ROWIDs.

Examples

```
SELECT dbms_rowid.rowid_object(ROWID)
   FROM emp
   WHERE empno = 7499;
```

ROWID_RELATIVE_FNO Function

This function returns the relative file number of the ROWID specified as the IN parameter. (The file number is relative to the tablespace.)

Syntax

```
DBMS_ROWID.ROWID_RELATIVE_FNO (
   rowid_id      IN   ROWID,
   ts_type_in    IN   VARCHAR2 DEFAULT 'SMALLFILE')
  RETURN NUMBER;
```

Pragmas

```
   pragma RESTRICT_REFERENCES(rowid_relative_fno,WNDS,RNDS,WNPS,RNPS);
```

Parameters

Table 170-10 ROWID_RELATIVE_FNO Function Parameters

Parameter	Description
row_id	ROWID to be interpreted
ts_type_in	Type of the tablespace (bigfile/smallfile) to which the row belongs

Examples

The example PL/SQL code fragment returns the relative file number:

```
DECLARE
   file_number    INTEGER;
   rowid_val      ROWID;
BEGIN
   SELECT ROWID INTO rowid_val
     FROM dept
     WHERE loc = 'Boston';
   file_number :=
     dbms_rowid.rowid_relative_fno(rowid_val, 'SMALLFILE');
    ...
```

ROWID_ROW_NUMBER Function

This function extracts the row number from the ROWID IN parameter.

Syntax

```
DBMS_ROWID.ROWID_ROW_NUMBER (
   row_id IN ROWID)
  RETURN NUMBER;
```

Pragmas

```
   PRAGMA RESTRICT_REFERENCES(rowid_row_number,WNDS,RNDS,WNPS,RNPS);
```

Parameters

Table 170-11   ROWID_ROW_NUMBER Function Parameters

Parameter	Description
`row_id`	`ROWID` to be interpreted.

Examples

Select a row number:

```
SELECT dbms_rowid.rowid_row_number(ROWID)
   FROM emp
   WHERE ename = 'ALLEN';
```

ROWID_TO_ABSOLUTE_FNO Function

This function extracts the absolute file number from a `ROWID`, where the file number is absolute for a row in a given schema and table.

The schema name and the name of the schema object (such as a table name) are provided as `IN` parameters for this function.

Syntax

```
DBMS_ROWID.ROWID_TO_ABSOLUTE_FNO (
   row_id      IN ROWID,
   schema_name IN VARCHAR2,
   object_name IN VARCHAR2)
  RETURN NUMBER;
```

Pragmas

```
   pragma RESTRICT_REFERENCES(rowid_to_absolute_fno,WNDS,WNPS,RNPS);
```

Parameters

Table 170-12   ROWID_TO_ABSOLUTE_FNO Function Parameters

Parameter	Description
`row_id`	`ROWID` to be interpreted
`schema_name`	Name of the schema which contains the table
`object_name`	Table name

Examples

```
DECLARE
   abs_fno         INTEGER;
   rowid_val       CHAR(18);
   object_name     VARCHAR2(20) := 'EMP';
BEGIN
    SELECT ROWID INTO rowid_val
     FROM emp
     WHERE empno = 9999;
     abs_fno := dbms_rowid.rowid_to_absolute_fno(
     rowid_val, 'SCOTT', object_name);
```

> **Note:**
>
> For partitioned objects, the name must be a table name, not a partition or a sub/partition name.

ROWID_TO_EXTENDED Function

This function translates a restricted ROWID that addresses a row in a schema and table that you specify to the extended ROWID format.

Later, it may be removed from this package into a different place.

Syntax

```
DBMS_ROWID.ROWID_TO_EXTENDED (
   old_rowid       IN ROWID,
   schema_name     IN VARCHAR2,
   object_name     IN VARCHAR2,
   conversion_type IN INTEGER)
  RETURN ROWID;
```

Pragmas

```
   pragma RESTRICT_REFERENCES(rowid_to_extended,WNDS,WNPS,RNPS);
```

Parameters

Table 170-13 ROWID_TO_EXTENDED Function Parameters

Parameter	Description
old_rowid	ROWID to be converted
schema_name	Name of the schema which contains the table (optional)
object_name	Table name (optional).
conversion_type	The following constants are defined: ROWID_CONVERT_INTERNAL (:=0) ROWID_CONVERT_EXTERNAL (:=1)

Return Values

ROWID_TO_EXTENDED returns the ROWID in the extended character format. If the input ROWID is NULL, then the function returns NULL. If a zero-valued ROWID is supplied (00000000.0000.0000), then a zero-valued restricted ROWID is returned.

Examples

Assume that there is a table called RIDS in the schema SCOTT, and that the table contains a column ROWID_COL that holds ROWIDs (restricted), and a column TABLE_COL that point to other tables in the SCOTT schema. You can convert the ROWIDs to extended format with the statement:

```
UPDATE SCOTT.RIDS
   SET rowid_col =
   dbms_rowid.rowid_to_extended (
      rowid_col, 'SCOTT", TABLE_COL, 0);
```

Usage Notes

- If the schema and object names are provided as IN parameters, then this function verifies SELECT authority on the table named, and converts the restricted ROWID provided to an extended ROWID, using the data object number of the table. That ROWID_TO_EXTENDED returns a value, however, does not guarantee that the converted ROWID actually references a valid row in the table, either at the time that the function is called, or when the extended ROWID is actually used.
- If the schema and object name are not provided (are passed as NULL), then this function attempts to fetch the page specified by the restricted ROWID provided. It treats the file number stored in this ROWID as the absolute file number. This can cause problems if the file has been dropped, and its number has been reused prior to the migration. If the fetched page belongs to a valid table, then the data object number of this table is used in converting to an extended ROWID value. This is very inefficient, and Oracle recommends doing this only as a last resort, when the target table is not known. The user must still know the correct table name at the time of using the converted value.
- If an extended ROWID value is supplied, the data object number in the input extended ROWID is verified against the data object number computed from the table name parameter. If the two numbers do not match, the INVALID_ROWID exception is raised. If they do match, the input ROWID is returned.
- ROWID_TO_EXTENDED cannot be used with partition tables.

> **See Also:**
>
> The ROWID_VERIFY Function has a method to determine if a given ROWID can be converted to the extended format.

ROWID_TO_RESTRICTED Function

This function converts an extended ROWID into restricted ROWID format.

Syntax

```
DBMS_ROWID.ROWID_TO_RESTRICTED (
   old_rowid       IN ROWID,
   conversion_type IN INTEGER)
  RETURN ROWID;
```

Pragmas

```
   pragma RESTRICT_REFERENCES(rowid_to_restricted,WNDS,RNDS,WNPS,RNPS);
```

Parameters

Table 170-14 ROWID_TO_RESTRICTED Function Parameters

Parameter	Description
`old_rowid`	`ROWID` to be converted
`conversion_type`	The following constants are defined: `ROWID_CONVERT_INTERNAL` (:=0) `ROWID_CONVERT_EXTERNAL` (:=1)

ROWID_TYPE Function

This function returns 0 if the `ROWID` is a restricted `ROWID`, and 1 if it is extended.

Syntax

```
DBMS_ROWID.ROWID_TYPE (
   rowid_id IN ROWID)
  RETURN NUMBER;
```

Pragmas

```
   pragma RESTRICT_REFERENCES(rowid_type,WNDS,RNDS,WNPS,RNPS);
```

Parameters

Table 170-15 ROWID_TYPE Function Parameters

Parameter	Description
`row_id`	`ROWID` to be interpreted

Examples

```
IF DBMS_ROWID.ROWID_TYPE(my_rowid) = 1 THEN
   my_obj_num := DBMS_ROWID.ROWID_OBJECT(my_rowid);
```

ROWID_VERIFY Function

This function verifies the `ROWID`.

It returns 0 if the input restricted `ROWID` can be converted to extended format, given the input schema name and table name, and it returns 1 if the conversion is not possible.

> **Note:**
>
> You can use this function in a `WHERE` clause of a SQL statement, as shown in the example.

Syntax

```
DBMS_ROWID.ROWID_VERIFY (
   rowid_in        IN ROWID,
   schema_name     IN VARCHAR2,
   object_name     IN VARCHAR2,
   conversion_type IN INTEGER
  RETURN NUMBER;
```

Pragmas

```
   pragma RESTRICT_REFERENCES(rowid_verify,WNDS,WNPS,RNPS);
```

Parameters

Table 170-16 ROWID_VERIFY Function Parameters

Parameter	Description
rowid_in	ROWID to be verified
schema_name	Name of the schema which contains the table
object_name	Table name
conversion_type	The following constants are defined: ROWID_CONVERT_INTERNAL (:=0) ROWID_CONVERT_EXTERNAL (:=1)

Examples

Considering the schema in the example for the ROWID_TO_EXTENDED function, you can use the following statement to find bad ROWIDs prior to conversion. This enables you to fix them beforehand.

```
SELECT ROWID, rowid_col
   FROM SCOTT.RIDS
   WHERE dbms_rowid.rowid_verify(rowid_col, NULL, NULL, 0) =1;
```

> **See Also:**
>
> UTL_RAW , UTL_REF