152
DBMS_PREPROCESSOR

The DBMS_PREPROCESSOR package provides an interface to print or retrieve the source text of a PL/SQL unit in its post-processed form.

This package contains the following topics:

DBMS_PREPROCESSOR Overview

It is necessary to first understand the three styles of subprograms, in order to understand how `DBMS_PREPROCESSOR` works.

The following are the three styles of subprograms:

1. Subprograms that take a schema name, a unit type name, and the unit name.
2. Subprograms that take a `VARCHAR2` string which contains the source text of an arbitrary PL/SQL compilation unit.
3. Subprograms that take a `VARCHAR2` index-by table which contains the segmented source text of an arbitrary PL/SQL compilation unit.

Subprograms of the first style are used to print or retrieve the post-processed source text of a stored PL/SQL unit. The user must have the privileges necessary to view the original source text of this unit. The user must also specify the schema in which the unit is defined, the type of the unit, and the name of the unit. If the schema is null, then the current user schema is used. If the status of the stored unit is `VALID` and the user has the required privilege, then the post-processed source text is guaranteed to be the same as that of the unit the last time it was compiled.

Subprograms of the second or third style are used to generate post-processed source text in the current user schema. The source text is passed in as a single `VARCHAR2` string in the second style, or as a `VARCHAR2` index-by table in the third style. The source text can represent an arbitrary PL/SQL compilation unit. A typical usage is to pass the source text of an anonymous block and generate its post-processed source text in the current user schema. The third style can be useful when the source text exceeds the `VARCHAR2` length limit.

DBMS_PREPROCESSOR Operating Notes

These notes explain how DBMS_PREPROCESSOR works with the three subprogram styles.

- For subprograms of the first style, the status of the stored PL/SQL unit does not need to be `VALID`. Likewise, the source text passed in as a `VARCHAR2` string or a `VARCHAR2` index-by table may contain compile time errors. If errors are found when generating the post-processed source, the error message text will also appear at the end of the post-processed

source text. In some cases, the preprocessing can be aborted because of errors. When this happens, the post-processed source text will appear to be incomplete and the associated error message can help to indicate that an error has occurred during preprocessing.

- For subprograms of the second or third style, the source text can represent any arbitrary PL/SQL compilation unit. However, the source text of a valid PL/SQL compilation unit cannot include commonly used prefixes such as `CREATE OR REPLACE`. In general, the input source should be syntactically prepared in a way as if it were obtained from the `ALL_SOURCE` view. The following list gives some examples of valid initial syntax for some PL/SQL compilation units.

```
anonymous block   (BEGIN | DECLARE) ...
package           PACKAGE <name> ...
package body      PACKAGE BODY <name> ...
procedure         PROCEDURE <name> ...
function          FUNCTION <name> ...
type              TYPE <name> ...
type body         TYPE BODY <name> ...
trigger           (BEGIN | DECLARE) ...
```

  If the source text represents a named PL/SQL unit that is valid, that unit will not be created after its post-processed source text is generated.

- If the text of a wrapped PL/SQL unit is obtained from the `ALL_SOURCE` view, the keyword `WRAPPED` always immediately follows the name of the unit, as in this example:

```
PROCEDURE "some proc" WRAPPED
a000000
b2
...
```

  If such source text is presented to one of the GET_POST_PROCESSED_SOURCE Functions or to one of the PRINT_POST_PROCESSED_SOURCE Procedures, the exception `DBMS_PREPROCESSOR.WRAPPED_INPUT` is raised.

DBMS_PREPROCESSOR Data Structures

The `DBMS_PREPROCESSOR` package defines a `TABLE` type.

Table Types

SOURCE_LINES_T Table Type

DBMS_PREPROCESSOR SOURCE_LINES_T Table Type

This table type stores lines of post-processed source text. It is used to hold PL/SQL source text both before and after it is processed. It is especially useful in cases in which the amount of text exceeds 32K.

Syntax

```
TYPE source_lines_t IS
    TABLE OF VARCHAR2(32767) INDEX BY BINARY_INTEGER;
```

Summary of DBMS_PREPROCESSOR Subprograms

This table lists the `DBMS_PREPROCESSOR` subprograms and briefly describes them.

Table 152-1 DBMS_PREPROCESSOR Package Subprograms

Subprogram	Description
GET_POST_PROCESSED_SOURCE Functions	Returns the post-processed source text
PRINT_POST_PROCESSED_SOURCE Procedures	Prints post-processed source text

GET_POST_PROCESSED_SOURCE Functions

This overloaded function returns the post-processed source text. The different functionality of each form of syntax is presented along with the definition.

Syntax

Returns post-processed source text of a stored PL/SQL unit:

```
DBMS_PREPROCESSOR.GET_POST_PROCESSED_SOURCE (
   object_type    IN VARCHAR2,
   schema_name    IN VARCHAR2,
   object_name    IN VARCHAR2)
  RETURN source_lines_t;
```

Returns post-processed source text of a compilation unit:

```
DBMS_PREPROCESSOR.GET_POST_PROCESSED_SOURCE (
   source        IN VARCHAR2)
  RETURN source_lines_t;
```

Returns post-processed source text of an `INDEX-BY` table containing the source text of the compilation unit:

```
DBMS_PREPROCESSOR.GET_POST_PROCESSED_SOURCE (
   source        IN source_lines_t)
  RETURN source_lines_t;
```

Parameters

Table 152-2 GET_POST_PROCESSED_SOURCE Function Parameters

Parameter	Description
`object_type`	Must be one of `PACKAGE`, `PACKAGE BODY`, `PROCEDURE`, `FUNCTION`, `TYPE`, `TYPE`, `BODY` or `TRIGGER`. Case sensitive.
`schema_name`	The schema name. Case insensitive unless a quoted identifier is used. If `NULL`, use current schema.
`object_name`	The name of the object.The `object_type` is always case insensitive. Case insensitive unless a quoted identifier is used.
`source`	The source text of the compilation unit
`source_lines_t`	`INDEX-BY` table containing the source text of the compilation unit. The source text is a concatenation of all the non-`NULL` `INDEX-BY` table elements in ascending index order.

Return Values

The function returns an INDEX-BY table containing the lines of the post-processed source text starting from index 1.

Usage Notes

- Newline characters are not removed.
- Each line in the post-processed source text is mapped to a row in the INDEX-BY table.
- In the post-processed source, unselected text will have blank lines.

Exceptions

Table 152-3 GET_POST_PROCESSED_SOURCE Function Exceptions

Exception	Description
ORA-24234	Insufficient privileges or object does not exist
ORA-24235	Bad value for object type. Should be one of PACKAGE, PACKAGE BODY, PROCEDURE, FUNCTION, TYPE, TYPE, BODY or TRIGGER.
ORA-24236	The source text is empty
ORA-00931	Missing identifier. The object_name should not be NULL.
ORA-06502	Numeric or value error: • Character string buffer too small • A line is too long (> 32767 bytes)

PRINT_POST_PROCESSED_SOURCE Procedures

This overloaded procedure calls DBMS_OUTPUT.PUT_LINE to let you view post-processed source text. The different functionality of each form of syntax is presented along with the definition.

Syntax

Prints post-processed source text of a stored PL/SQL unit:

```
DBMS_PREPROCESSOR.PRINT_POST_PROCESSED_SOURCE (
   object_type    IN VARCHAR2,
   schema_name    IN VARCHAR2,
   object_name    IN VARCHAR2);
```

Prints post-processed source text of a compilation unit:

```
DBMS_PREPROCESSOR.PRINT_POST_PROCESSED_SOURCE (
   source         IN VARCHAR2);
```

Prints post-processed source text of an INDEX-BY table containing the source text of the compilation unit:

```
DBMS_PREPROCESSOR.PRINT_POST_PROCESSED_SOURCE (
   source         IN source_lines_t);
```

Parameters

Table 152-4 PRINT_POST_PROCESSED_SOURCE Procedure Parameters

Parameter	Description
object_type	Must be one of PACKAGE, PACKAGE BODY, PROCEDURE, FUNCTION, TYPE, TYPE, BODY or TRIGGER. Case sensitive.
schema_name	The schema name. Case insensitive unless a quoted identifier is used. If NULL, use current schema.
object_name	The name of the object.The object_type is always case insensitive. Case insensitive unless a quoted identifier is used.
source	The source text of the compilation unit
source_lines_t	INDEX-BY table containing the source text of the compilation unit. The source text is a concatenation of all the non-NULL INDEX-BY table elements in ascending index order.

Exceptions

Table 152-5 PRINT_POST_PROCESSED_SOURCE Procedure Exceptions

Exception	Description
ORA-24234	Insufficient privileges or object does not exist
ORA-24235	Bad value for object type. Should be one of PACKAGE, PACKAGE BODY, PROCEDURE, FUNCTION, TYPE, TYPE, BODY or TRIGGER.
ORA-24236	The source text is empty
ORA-00931	Missing identifier. The object_name should not be NULL.
ORA-06502	Numeric or value error: • Character string buffer too small • A line is too long (> 32767 bytes)

Usage Notes

The index-by table may contain holes. NULL elements are ignored when doing the concatenation.