65
DBMS_DB_VERSION

The `DBMS_DB_VERSION` package specifies the Oracle version numbers and other information useful for simple conditional compilation selections based on Oracle versions.

> **See Also:**
>
> *Oracle Database PL/SQL Language Reference* regarding conditional compilation

This package contains the following topics:

- Overview
- Constants
- Examples

DBMS_DB_VERSION Overview

The `DBMS_DB_VERSION` package specifies the Oracle version numbers and other information useful for simple conditional compilation selections based on Oracle versions.

The package for the Oracle Database 21c version is shown below.

```
PACKAGE DBMS_DB_VERSION IS
   VERSION CONSTANT PLS_INTEGER := 21; -- RDBMS version number
   RELEASE CONSTANT PLS_INTEGER := 0;  -- RDBMS release number
   ver_le_9_1     CONSTANT BOOLEAN := FALSE;
   ver_le_9_2     CONSTANT BOOLEAN := FALSE;
   ver_le_9       CONSTANT BOOLEAN := FALSE;
   ver_le_10_1    CONSTANT BOOLEAN := FALSE;
   ver_le_10_2    CONSTANT BOOLEAN := FALSE;
   ver_le_10      CONSTANT BOOLEAN := FALSE;
   ver_le_11_1    CONSTANT BOOLEAN := FALSE;
   ver_le_11_2    CONSTANT BOOLEAN := FALSE;
   ver_le_11      CONSTANT BOOLEAN := FALSE;
   ver_le_12_1    CONSTANT BOOLEAN := FALSE;
   ver_le_12_2    CONSTANT BOOLEAN := FALSE;
   ver_le_12      CONSTANT BOOLEAN := FALSE;
   ver_le_18      CONSTANT BOOLEAN := FALSE;
   ver_le_19      CONSTANT BOOLEAN := FALSE;
   ver_le_20      CONSTANT BOOLEAN := FALSE;
   ver_le_21      CONSTANT BOOLEAN := TRUE;
END DBMS_DB_VERSION;
```

The boolean constants follow a naming convention. Each constant gives a name for a boolean expression. For example:

- `VER_LE_9_1` represents version <= 9 and release <= 1
- `VER_LE_10_2` represents version <= 10 and release <= 2

- VER_LE_10 represents version <= 10

A typical usage of these boolean constants is:

```
$IF DBMS_DB_VERSION.VER_LE_10 $THEN
   version 10 and earlier code
$ELSIF DBMS_DB_VERSION.VER_LE_11 $THEN
   version 11 code
$ELSE
   version 12 and later code
$END
```

This code structure will protect any reference to the code for version 12. It also prevents the controlling package constant DBMS_DB_VERSION.VER_LE_11 from being referenced when the program is compiled under version 10. A similar observation applies to version 11. This scheme works even though the static constant VER_LE_11 is not defined in version 10 database because conditional compilation protects the $ELSIF from evaluation if DBMS_DB_VERSION.VER_LE_10 is TRUE.

DBMS_DB_VERSION Constants

The DBMS_DB_VERSION package contains different constants for different Oracle Database releases.

The Oracle Database 21*c* version of the DBMS_DB_VERSION package uses the constants shown in the following table.

Table 65-1 DBMS_DB_VERSION Constants

Name	Type	Value	Description
VERSION	PLS_INTEGER	21	Current version
RELEASE	PLS_INTEGER	0	Current release
VER_LE_9	BOOLEAN	FALSE	Version <= 9
VER_LE_9_1	BOOLEAN	FALSE	Version <= 9 and release <= 1
VER_LE_9_2	BOOLEAN	FALSE	Version <= 9 and release <= 2
VER_LE_10	BOOLEAN	FALSE	Version <= 10
VER_LE_10_1	BOOLEAN	FALSE	Version <= 10 and release <= 1
VER_LE_10_2	BOOLEAN	FALSE	Version <=10 and release <= 2
VER_LE_11	BOOLEAN	FALSE	Version <= 11
VER_LE_11_1	BOOLEAN	FALSE	Version <=11 and release <= 1
VER_LE_11_2	BOOLEAN	FALSE	Version <=11 and release <= 2
VER_LE_12	BOOLEAN	FALSE	Version <=12
VER_LE_12_1	BOOLEAN	FALSE	Version <=12 and release <= 1
VER_LE_12_2	BOOLEAN	FALSE	Version <=12 and release <= 2
VER_LE_18	BOOLEAN	FALSE	Version <=18
VER_LE_19	BOOLEAN	FALSE	Version <=19
VER_LE_20	BOOLEAN	FALSE	Version <=20
VER_LE_21	BOOLEAN	TRUE	Version <=21

DBMS_DB_VERSION Examples

This example uses conditional compilation to guard new features.

```
CREATE OR REPLACE PROCEDURE whetstone IS

 -- Notice that conditional compilation constructs
 -- can interrupt a regular PL/SQL statement.
 -- You can locate a conditional compilation directive anywhere
 -- there is whitespace in the regular statement.

 SUBTYPE my_real IS
    $IF DBMS_DB_VERSION.VER_LE_9 $THEN NUMBER
                                 $ELSE BINARY_DOUBLE
    $END;

 t  CONSTANT my_real := $IF DBMS_DB_VERSION.VER_LE_9 $THEN 0.499975
                                                     $ELSE 0.499975d
                        $END;

 t2 CONSTANT my_real := $if DBMS_DB_VERSION.VER_LE_9 $THEN 2.0
                                                     $ELSE 2.0d
                        $END;

 x  CONSTANT my_real := $IF DBMS_DB_VERSION.VER_LE_9 $THEN 1.0
                                                     $ELSE 1.0d
                        $END;

 y  CONSTANT my_real := $IF DBMS_DB_VERSION.VER_LE_9 $THEN 1.0
                                                     $ELSE 1.0d
                        $END;

 z  MY_REAL;

 PROCEDURE P(x IN my_real, y IN my_real, z OUT NOCOPY my_real) IS
   x1 my_real;
   y1 my_real;
 BEGIN
   x1 := x;
   y1 := y;
   x1 := t * (x1 + y1);
   y1 := t * (x1 + y1);
   z := (x1 + y1)/t2;
 END P;
BEGIN
 P(x, y, z);
 DBMS_OUTPUT.PUT_LINE ('z = '|| z);
END whetstone;
/
```