

OWA_OPT_LOCK

The `OWA_OPT_LOCK` package contains subprograms that impose optimistic locking strategies so as to prevent lost updates.

This chapter contains the following topics:

- [Overview](#)
- [Types](#)
- [Summary of OWA_OPT_LOCK Subprograms](#)

See Also:

For more information about implementation of this package:

- [Oracle Fusion Middleware Administrator's Guide for Oracle HTTP Server](#)
- [Oracle Fusion Middleware User's Guide for mod_plsql](#)

OWA_OPT_LOCK Overview

The `OWA_OPT_LOCK` package contains subprograms that impose optimistic locking strategies, so as to prevent lost updates.

It checks if the row that the user is interested in updating has been changed by someone else in the meantime.

The PL/SQL Gateway cannot use conventional database locking schemes because HTTP is a stateless protocol. The `OWA_OPT_LOCK` package gives you two ways of dealing with the lost update problem:

- The hidden fields method stores the previous values in hidden fields in the HTML page. When the user requests an update, the PL/SQL Gateway checks these values against the current state of the database. The update operation is performed only if the values match. To use this method, call the `owa_opt_lock.store_values` procedure.
- The checksum method stores a checksum rather than the values themselves. To use this method, call the `owa_opt_lock.checksum` function.

These methods are optimistic. They do not prevent other users from performing updates, but they do reject the current update if an intervening update has occurred.

OWA_TEXT Types

This datatype is a PL/SQL table intended to hold ROWIDs.

```
TYPE VCARRAY IS TABLE OF VARCHAR2(2000) INDEX BY BINARY_INTEGER
```

Note that this is different from the `OWA_TEXT.VC_ARR` DATA TYPE.

Summary of OWA_OPT_LOCK Subprograms

This table lists the OWA_OPT_LOCK subprograms and briefly describes them.

Table 252-1 OWA_OPT_LOCK Package Subprograms

Subprogram	Description
CHECKSUM Functions	Returns the checksum value
GET_ROWID Function	Returns the ROWID value
STORE_VALUES Procedure	Stores unmodified values in hidden fields for later verification
VERIFY_VALUES Function	Verifies the stored values against modified values

CHECKSUM Functions

This function returns a `checksum` value for a specified string, or for a row in a table. For a row in a table, the function calculates the `checksum` value based on the values of the columns in the row. This function comes in two versions.

The first version returns a `checksum` based on the specified string. This is a "pure" 32-bit `checksum` executed by the database and based on the Internet 1 protocol.

The second version returns a `checksum` based on the values of a row in a table. This is a "impure" 32-bit checksum based on the Internet 1 protocol.

Syntax

```
OWA_OPT_LOCK.CHECKSUM(  
    p_buff      IN      VARCHAR2)  
    RETURN NUMBER;  
  
OWA_OPT_LOCK.CHECKSUM(  
    p_buff      IN      VARCHAR2)  
    RETURN NUMBER PARALLEL_ENABLE DETERMINISTIC;  
  
OWA_OPT_LOCK.CHECKSUM(  
    p_owner      IN      VARCHAR2,  
    p_tname      IN      VARCHAR2,  
    p_rowid      IN      ROWID)  
    RETURN NUMBER;
```

Parameters

Table 252-2 CHECKSUM Function Parameters

Parameter	Description
<code>p_buff</code>	The nstring where you want to calculate the checksum.
<code>p_owner</code>	The owner of the table.
<code>p_tname</code>	The table name.
<code>p_rowid</code>	The row in <code>p_tname</code> where you want to calculate the checksum value. Use the GET_ROWID Function to convert VCARRAY values to proper rowids.

GET_ROWID Function

This function returns the ROWID datatype from the specified OWA_OPT_LOCK.VCARRAY DATA TYPE.

Syntax

```
OWA_OPT_LOCK.GET_ROWID(  
    p_old_values      IN      varray)  
RETURN ROWID;  
  
OWA_OPT_LOCK.GET_ROWID(  
    p_old_values      IN      varray)  
RETURN ROWID PARALLEL_ENABLE DETERMINISTIC;
```

Parameters

Table 252-3 GET_ROWID Function Parameters

Parameter	Description
p_old_values	This parameter is usually passed in from an HTML form.

STORE_VALUES Procedure

This procedure stores the column values of the row that you want to update later. The values are stored in hidden HTML form elements.

Syntax

```
OWA_OPT_LOCK.STORE_VALUES(  
    p_owner      IN      VARCHAR2,  
    p_tname      IN      VARCHAR2,  
    p_rowid      IN      ROWID);
```

Parameters

Table 252-4 STORE_VALUES Procedure Parameters

Parameter	Description
p_owner	The owner of the table.
p_tname	The name of the table.
p_rowid	The row where you want to store values.

Usage Notes

Before updating the row, compare these values with the current row values to ensure that the values in the row have not been changed. If the values have changed, you can warn the users and let them decide if the update should take place.

The procedure generates series of hidden form elements:

- One hidden form element is created for the table owner. The name of the element is "old_p_tname", where p_tname is the name of the table. The value of the element is the owner name.

- One hidden form element is created for the table name. The name of the element is "old_p_tname", where p_tname is the name of the table. The value of the element is the table name.
- One element is created for each column in the row. The name of the element is "old_p_tname", where p_tname is the name of the table. The value of the element is the column value.

See also the [VERIFY_VALUES Function](#).

VERIFY_VALUES Function

This function verifies whether values in the specified row have been updated since the last query.

Use this function with the [STORE_VALUES Procedure](#).

Syntax

```
OWA_OPT_LOCK.VERIFY_VALUES(  
    p_old_values IN varray)  
RETURN BOOLEAN;
```

Parameters

Table 252-5 *VERIFY_VALUES Function Parameters*

Parameter	Description
p_old_values	<p>A PL/SQL table containing the following information:</p> <ul style="list-style-type: none">• p_old_values(1) specifies the owner of the table.• p_old_values(2) specifies the table.• p_old_values(3) specifies the rowid of the row to verify. <p>The remaining indexes contain values for the columns in the table.</p> <p>Typically, this parameter is passed in from the HTML form, where you have previously called the STORE_VALUES Procedure to store the row values on hidden form elements.</p>

Return Values

TRUE if no other update has been performed, otherwise FALSE.