

DBS_UPDATE

DB_UPDATE and DBS_UPDATE actions

Function	Modification of the given existing row (rows) in the database.		
Declaration	<pre>DB_UPDATE handleIdent_Int, rowIdent, retCodeIdent_Int [WHERE strExpression_Str [BINDIN varIdent1, varIdent2, ...]] [ORAHINT hintIdent_Str] DB_UPDATE handleIdent_Int, rowIdent, retCodeIdent_Int [WHERE strExpression_Str [BINDIN structRowIdent]] [ORAHINT hintIdent_Str] DBS_UPDATE dbObjIdent, rowIdent, retCodeIdent_Int [WHERE strExpression_Str [BINDIN varIdent1, varIdent2, ...]] [TRANS transHandle_Int] [ORAHINT hintIdent_Str] DBS_UPDATE dbObjIdent, rowIdent, retCodeIdent_Int [WHERE strExpression_Str [BINDIN structRowIdent]] [TRANS transHandle_Int] [ORAHINT hintIdent_Str]</pre> <p>or</p> <pre>DB_UPDATE handleIdent_Int, structIdent, retCodeIdent_Int [ORAHINT hintIdent_Str] DBS_UPDATE dbObjIdent, structIdent, retCodeIdent_Int [TRANS transHandle_Int] [ORAHINT hintIdent_Str]</pre>		
Parameters	handl eIdent _Int	in	Identifier of <i>Int</i> type (handle) of the connection with the database (DB_CONNECT).
	dbObj Ident	in	Reference to an object of Database table type.
	rowId ent	in	One structure row identifier (a row to insert).
	structI dent	in	Identifier of entire structure (the rows to insert).
	retCo delde nt_Int	o ut	Return value of <i>Int</i> type - action success.
	strEx pressi on_Str	in	Expression of <i>String</i> type, which identifies modified rows. If the expression is parameterized , the keyword BINDIN and the values of parameters (<i>structRowIdent</i> or <i>varIdent1</i> , <i>varIdent2</i> , ...) are mandatory.
	varIde nt1, varIde nt2, ...	in	List of objects, constants or local variables , which will specify the values of parameters of parameterized SQL expression <i>strExpression_Str</i> .
	struct RowI dent	in	Reference to a row of local variable of <i>Record</i> type or to a row of structured variable . The row's values will specify the values of parameters of parameterized SQL expression <i>strExpression_Str</i> .
	trans Handle _Int	in	Identifier of the Connection to the database.
	hintId ent_S tr	in	Expression of <i>String</i> type that defines Oracle SQL hint. It is used as an instruction for the performance optimizer of SQL command. The value is used without the opening and terminating characters <i>/*+ <orahint> */</i> . The example is mentioned here .
Return code			

The value of the parameter *transHandle_Int*. See the table of [error codes](#). It is possible to get [extended error information](#).

Description

The database must be opened with the access `_DB_MODIFY`. Analogous to the action `DB_READ` (`DBS_READ`), modified rows may be determined by two ways:

1. Setting the key item in a structure row (`rowIdent`) or in the entire structure (`structIdent`). The setting must be performed before the action `DB_UPDATE`. The item must be signed as the key during the definition of an object of [Database](#) type. Then in the database, there is found a row which key item value is equal to the key item and is modified. The inserted row or structure must contain a proper type of structure. Each item of the inserted row must contain a valid value.
2. The value of the expression `strExpression_Str` which represents

WHERE

clause for the SQL command UPDATE executing a database change. In this case, a key item value is not used.

The advantage of the action `DBS_UPDATE` during the work with a table is the possibility to leave out its opening and closing (simpler notation).

For D2000 v5.00: A disadvantage of the action `DBS_UPDATE` is in speed. Each `DBS_UPDATE` call results in necessity to open and close the database in DBManager - it can be a time-consuming operation and it is a comparatively nonstandard method in term of databases. The need to open and close the database may be eliminated in the scope of transaction processing so that the command is followed by the parameter

TRANS

For D2000 v6.00 and higher: DBManager [optimization](#) (connection recycling, predefined connections) causes, that the action `DBS_UPDATE` is executes as quick as the action `DB_UPDATE` and moreover there is saved a time required for execution of the action `DB_CONNECT` to open the database.

The need to open and close the database may be eliminated in the scope of transaction processing so that the command is followed by the parameter

TRANS

Example

[Work with a database table \(actions DB_ ...\).](#)

Related topics

[DB_CONNECT](#)
[DB_DELETE](#)
[DB_DISCONNECT](#)
[DB_INSERT](#)
[DB_INSUPD](#)
[DB_READ](#)
[DB_READ_BLOB](#)
[DB_UPDATE_BLOB](#)

[DB_TRANS_OPEN](#)
[DB_TRANS_COMMIT](#)
[DB_TRANS_ROLLBACK](#)
[DB_TRANS_CLOSE](#)

[All database related actions](#)



Related pages:

[Script actions](#)