

# DB\_UPDATE

## Akcia DB\_UPDATE a DBS\_UPDATE

### Funkcia

Modifikácia existujúceho riadku (riadkov) v databáze.

### Deklarácia

```
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]
```

alebo

```
DB_UPDATE handleIdent_Int, structIdent, retCodeIdent_Int [ORAHINT  
hintIdent_Str]  
  
DBS_UPDATE dbObjIdent, structIdent, retCodeIdent_Int [TRANS  
transHandle_Int] [ORAHINT hintIdent_Str]
```

### Parametre

handleIdent_Int	in	Identifikátor typu <i>Int</i> - ( <i>handle</i> ) spojenia s databázou ( <i>DB_CONNECT</i> ).
dbObjIdent	in	Odkaz na objekt typu <i>Tabuka</i> .
rowIdent	in	Identifikátor jedného riadku štruktúry (vkladaný riadok).
structIdent	in	Identifikátor celej štruktúry (vkladané riadky).
retCodeIdent_Int	out	Návratová hodnota typu <i>Int</i> - úspešnosť akcie.
strExpression_Str	in	Výraz typu <i>String</i> , ktorý identifikuje modifikované riadky. Ak je výraz parametrizovaný, tak kľúčové slovo <b>BINDIN</b> a hodnoty parametrov ( <i>structRowIdent</i> alebo <i>varIdent1</i> , <i>varIdent2</i> , ...) sú povinné.
varIdent1, varIdent2, ...	in	zoznam objektov, konštánt alebo <b>lokálnych premenných</b> , ktoré špecifikujú hodnoty parametrov <b>parametrizovaného</b> SQL výrazu <i>strExpression_Str</i> .
structRowIdent	in	Odkaz na riadok <b>lokálnej premennej</b> typu <i>Record</i> alebo na riadok <b>štruktúrovanej premennej</b> . Hodnoty riadkov špecifikujú hodnoty parametrov <b>parametrizovaného</b> SQL výrazu <i>strExpression_Str</i> .
transHandle_Int	in	Identifikátor <b>Pripojenia</b> na databázu.
hintIdent_Str	in	Výraz typu <i>String</i> , ktorý definuje Oracle SQL hint. Ten je použitý ako inštrukcia pre optimalizátor výkonu SQL príkazu. Zapisuje hodnotu bez obaľujúcich znakov <i>/*+ &lt;orahint &gt;*</i> . Príklad použitia je uvedený <b>tu</b> .

### Návratový kód

Hodnota parametra *retCodeIdent\_Int* - pozri tabuku [chybových stavov](#). Je možné získa [rozšírenú informáciu o chybe](#).

## Popis

Databáza musí byť otvorená s prístupom *\_DB\_MODIFY*. Podobne ako pri akcii *DB\_READ* (*DBS\_READ*), môžu byť modifikované riadky určené dvoma spôsobmi:

1. Nastavením kúovej položky v riadku štruktúry (*rowIdent*), resp. v celej štruktúre (*structIdent*). Nastavenie musí byť realizované pred akciou **DB\_UPDATE**. Položka musí byť označená ako kú pri definícii objektu typu *Databáza*. Potom sa v databáze hľadá riadok, ktorý má zhodnú hodnotu pre kúovú položku a tento je modifikovaný. Vkladaný riadok alebo štruktúra musia mať korektný typ štruktúry. Všetky položky každého vkladaneho riadku musia mať platné hodnoty.
2. Hodnotou výrazu *strExpression\_Str*, ktorý predstavuje klauzulu **WHERE** pre SQL príkaz **UPDATE**, ktorý vykonáva zmenu databázy. V tomto prípade je hodnota kúovej položky v štruktúre nepoužitá.

Výhodou akcie **DBS\_UPDATE** pri práci s tabukou je možnosť vynechať jej otvorenie a zatvorenie (jednoduchší zápis).

**Pre D2000 verziu 5:** Nevýhoda akcie **DBS\_UPDATE** je v rýchlosti. Každé volanie akcie **DBS\_UPDATE** má za následok, v rámci jej obsluhy v procese **D2000 DBManager**, nutnosť databázu otvoriť a následne zatvoriť, čo môže byť asovo náročné a predstavuje z databázového hľadiska pomerne neštandardný postup. Nutnosť otvárania a zatvárania databázy je možné potlačiť v rámci transakčného spracovania tak, že za príkaz uvedieme parameter **TRANS**.

**Pre D2000 verziu 6 a vyššie:** [Optimalizácia](#) procesu **D2000 DBManager** (recyklovanie spojení, predpripravené spojenia) spôsobila, že akcia **DBS\_UPDATE** sa vykoná rovnako rýchlo ako akcia **DB\_UPDATE** a navyše sa ušetrí čas potrebný na spracovanie explicitného príkazu na otvorenie databázy **DB\_CONNECT**.

Potreba otvorenia a zatvorenia databázy sa môže eliminovať v rámci transakčného spracovania tak, že za príkazom nasleduje parameter **TRANS**.

## Príklad

Príklad práce s databázou (akcie *DB\_...*).

## Súvisiace odkazy

[DB\\_CONNECT](#)  
[DB\\_DELETE](#)  
[DB\\_DISCONNECT](#)  
[DB\\_INSERT](#)  
[DB\\_INSERT](#)  
[DB\\_READ](#)  
[DB\\_READ\\_BLOB](#)  
[DB\\_UPDATE\\_BLOB](#)  
  
[DB\\_TRANS\\_OPEN](#)  
[DB\\_TRANS\\_COMMIT](#)  
[DB\\_TRANS\\_ROLLBACK](#)  
[DB\\_TRANS\\_CLOSE](#)

Všetky databázové akcie



Súvisiace stránky:

[Akcie v skriptoch](#)