

SQL_CONNECT

Akcia SQL_CONNECT

Funkcia

Akcia nadviaže spojenie s databázou (otvorenie databázy).

Deklarácia

```
SQL_CONNECT dbObjIdent, handleIdent_Int, retCodeIdent_Int [TRANS  
_transHandle_Int]  
  
SQL_CONNECT connectString, handleIdent_Int, retCodeIdent_Int ON dbManIdent
```

Parametre

dbObjIdent	in	Odkaz na objekt typu Databáza alebo Tabuka .
connectString	in	Identifikátor typu <i>Text</i> , alebo textová konštanta obsahujúca pripájací string na databázu.
dbManIdent_Int	in	Odkaz na objekt typu Proces (D2000 DBManager), ktorý bude realizovať požadované príkazy.
handleIdent_Int	out	Identifikátor typu <i>Int</i> - jednoznačné číslo (handle) spojenia.
retCodeIdent_Int	out	Identifikátor typu <i>Int</i> - návratový kód.
transHandle_Int	in	Identifikátor typu <i>Int</i> - jednoznačné číslo (handle) spojenia s databázou.

Návratový kód

Hodnota parametra *retCodeIdent_Int* - pozri tabuku [chybových stavov](#). Je možné získať [rozšírenú informáciu o chybe](#).

Popis

Identifikátor pre číslo spojenia, alebo handle sa používa pri ďalšej práci s otvorenou databázou. Návratový kód popisuje úspešnosť akcie (*_ERR_NO_ERROR*).

Každé otvorenie databázy skriptom má pridelené jednoznačné číslo (identifikátor, handle). Toto je potrebné po ukončení práce s databázou zatvoriť akciou [SQL_DISCONNECT](#). Ak po skončení skriptu nebudú všetky databázy uzatvorené volaním akcie [SQL_DISCONNECT](#) (tie, ktoré skript pri svojej inosti otvoril), budú interne automaticky zatvorené. Handle platí len v rámci skriptu, ktorý databázu otvoril.

Akcia [SQL_CONNECT](#) má vždy charakter otvorenia databázy (nie tabuky na rozdiel od akcií [DB_CONNECT](#) a [PG_CONNECT](#)), lebo prostredníctvom takto vytvoreného spojenia je možné pracovať s celou databázou (všetkými v nej obsiahnutými tabukami). Ak sa na otvorenie použije objekt typu [Tabuka](#), interne bude databáza otváraná prostredníctvom jeho rodia, teda objektu typu [Databáza](#).

Otváranú databázu je možné uri objektom typu [Databáza](#) alebo [Tabuka](#) (prvý spôsob zápisu) alebo tzv. pripájacím stringom (druhý spôsob zápisu).

Pri prvom spôsobe nie je potrebné zadať meno procesu [D2000 DBManager](#), ktorý bude následne príkazy interpretovať, lebo toto je dané rodičovským vzťahom medzi objektom typu [Databáza](#) a týmto procesom. Akcia [SQL_CONNECT](#) využije z konfigurácie objektu typu [Databáza](#) konfigurované parametre **Užívateľ**, **Heslo** a **DSN**.

Druhý spôsob zápisu umožňuje vytvoriť spojenie s databázou na základe znalosti **Mena**, **Hesla** a **DSN** pomocou tzv. pripájacieho stringu (parameter *connectString*). Jeho obsah je nasledovný:

"UID=meno užívateľa;PWD=heslo užívateľa;DSN=meno DSN".

Kúové slová *UID*, *PWD* a *DSN* označujú hodnoty jednotlivých parametrov. Ak niektorý z nich nie je uvedený, bude nahradený prázdny reťazcom. Pri tomto spôsobe je nutné špecifikovať odkazom proces [D2000 DBManager](#), ktorý bude zabezpečovať vykonávanie jednotlivých príkazov. Ďalšie voliteľné parametre pripájacieho stringu sa týkajú optimalizácie a sú popísané v kapitole [Transakčná práca s databázou](#).

V prípade otvárania databázy prostredníctvom objektu typu [Databáza](#) alebo [Tabuka](#), je možné uviesť parameter *transHandle_Int*. Týmto budú všetky akcie [SQL_*](#) vykonávané v rámci určitého [pripojenia](#). Toto pripojenie je potrebné najprv vytvoriť akciou [DB_TRANS_OPEN](#). Ak klauzula **TRANS** nie je uvedená, použije sa [automatické pripojenie](#).

V prípade otvárania databázy prostredníctvom pripájacieho stringu (parameter *connectString*), proces [D2000 DBManager](#) vždy vytvára nové pripojenie na databázu. Ak sa v pripájacom stringu nachádza kúové slovo **ACD** (**A**uto **C**ommit **D**isable), je v rámci nového pripojenia vypnutý režim **Auto Commit**, teda je potrebné "commit" vykonať manuálne pomocou SQL príkazu **COMMIT**.

Príklad

Príklad práce s databázou (akcie SQL_...).

```
INT _handle      ; handle to database
INT _retCode     ; return code
TEXT _name       ; product name
TEXT _type       ; product type
                ; parametrized SQL command
TEXT _sql = "SELECT Name, Type FROM Products WHERE ID>= #PAR# AND ID<=
#PAR#"

SQL_CONNECT MyDatabase, _handle, _retCode
SQL_PREPARE _handle, _retCode, _sql BINDOUT _name, _type
SQL_BINDIN _handle, _retCode, 1, 100 ; read all products between 1 and
100

DO_LOOP
    SQL_FETCH _handle, _retCode
    EXIT_LOOP _retCode # _ERR_NO_ERROR
    ; data processing goes here
END_LOOP

SQL_FREE _handle
SQL_DISCONNECT _handle
```

Súvisiace odkazy

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

[SQL_DISCONNECT](#)
[SQL_EXEC_DIRECT](#)
[SQL_EXEC_PROC](#)

[SQL_PREPARE](#)
[SQL_BINDIN](#)
[SQL_FETCH](#)
[SQL_FREE](#)

[SQL_SELECT](#)

[Všetky databázové akcie](#)



Súvisiace stránky:

[Akcie v skriptoch](#)