

# 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 <b>Databáza</b> alebo <b>Tabuka</b> .
connectString	in	Identifikátor typu <i>Text</i> , alebo textová konštanta obsahujúca pripájací string na databázu.
dbManIdent_Int	in	Odkaz na objekt typu <b>Proces</b> ( <b>D2000 DBManager</b> ), 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**.

Otváranie databázy je možné určiť 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** (**Auto Commit Disable**), 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](#)