

# SQL\_CONNECT

## SQL\_CONNECT action

**Function** The action establishes a connection to a database (opens a database).

**Declaration**

```
SQL_CONNECT dbObjIdent, handleIdent_Int, retCodeIdent_Int [TRANS
_transHandle_Int]
```

```
SQL_CONNECT connectString, handleIdent_Int, retCodeIdent_Int ON dbManIdent
```

**Parameters**

dbObjIdent	in	Reference to an object of the Database or Database table types.
connectString	in	Identifier of the Text type or a text constant containing a connection string to the database.
dbManIdent_Int	in	Reference to an object of the Process type (DbManager), which will execute the required commands.
handleIdent_Int	output	Identifier - the unique number (handle) of a connection.
retCodeIdent_Int	output	Return code identifier.
transHandle_Int	in	Identifier for an unique number (handle) of the connection to a database.

**Return code** The value of the parameter *transHandle\_Int*. See the table of error codes. It is possible to get extended error information.

**Description**

Identifier for a number (handle) of the connection which is used during another work with an opened database. Return code describes the action success (\_\_ERR\_NO\_ERROR).  
A unique number (identifier, handle) is assigned to each opening of a database by a script. It is necessary to close this number using the action [SQL\\_DISCONNECT](#), when having finished the work with the database. If all databases (opened by this script during its activity) are not to be closed by calling the action [SQL\\_DISCONNECT](#) after the script termination, they will be closed automatically. Handle is valid only within the frame of a script, which opened the database.

The action [SQL\\_CONNECT](#) always opens a database (not a table, unlike the actions [DB\\_CONNECT](#) and [PG\\_CONNECT](#)), because it allows to work with a whole database (all tables contained in it) via a connection created by this way. If an object of Database table is used to open a database, the database will be internally opened by means of its parent, so an object of Database.

Opened database may be determined by an object of the [Database](#) or [Database table](#) type (the first declaration type) or by so-called connect string (the second declaration type).  
In the first type, it is not necessary to enter the name of a process of [DbManager](#) type that will be interpret follow commands, because this is given by the parent relation between an object of Database type and the process. The command [SQL\\_CONNECT](#) will use the configuration parameters **User**, **Password** and **DSN** from the object configuration of particular object of Database type.  
The second declaration type allows creating the connection to an database on the basis of knowing the **Name**, **Password** and **DSN** using so-called connect string. It contains:  
"UID=user name;PWD=user password;DSN=DSN name".  
The keywords UID, PWD and DSN represent values of individual parameters. If some of them is not entered, it will be substituted by a null string. This declaration type requires to specifying, using a reference, a process of [DbManager](#) type, which will be provided for the execution of individual commands. The next optional parameters of connection string are related to optimization and are described together with the [DbManager](#) process.

If a database is opened using an object of the Database or Database Table type, it is able to enter *transHandle\_Int*. Thereby all [SQL\\_\\*](#) commands are to be executed within the frame of the specified [Connection](#). The connection must be created before using the action [DB\\_TRANS\\_OPEN](#). If the clause **TRANS** is not stated, there will be used the [Automatic connection](#).

If a database is opened by a connection string, the process [DbManager](#) always creates a new connection to the database. If the keyword **ACD** (**Auto Commit Disable**) is placed in the connection string, then the mode Auto Commit is disabled within the frame of this new connection, an it is necessary to execute the command Commit manually.

## Example

[Work with a database \(actions SQL\\_...\)](#).

```
INT _handle      ; handle to database
INT _retCode     ; return code
TEXT _name       ; product name
TEXT _type       ; product type
                ; parameterized 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
```

## Related topics

[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](#)

[All database related actions.](#)



### Related pages:

[Script actions](#)