

dmmADO - Xtra for Director and Authorware

List of functions

new	XTRA initialisation.
dbRegistration	Registration of XTRA dmmADO.x32.
dbActiveSQL	Insertion and activation of an SQL command, whose type is SELECT.
dbBof	A test to check whether the actual record is the first one in the database.
dbClearBookmarks	All bookmarks deletion.
dbClose	Database file closure.
dbColumnCount	Number of fields in the database.
dbColumns	Returned values of database fields.
dbConnection	Connection of the XTRA to database.
dbDeleteBookmark	Bookmark deletion.
dbDisconnection	Cancellation of connection between the XTRA and database.
dbEof	A test to check whether the actual record is the last one in the database.
dbErrorDialog	A preselection giving choice to show the dialogue window with error statement.
dbErrorLog	A preselection giving choice to save errors to log file.
dbErrorMsg	An error occurred while working with mdb file.
dbExecSQL	Insertion and activation of an SQL command, whose type is INSERT, UPDATE or DELETE.
dbFirst	A jump on the first record of the database.
dbGetADOVersion	Finding out of version of the database engine.
dbGetBookmarks	Information about what bookmarks are setted.
dbGetFieldByName	A given value of the item defined by its name.
dbGetFieldByNumber	A given value of the item defined by its number in the database.
dbGetFields	Returned values of the actual record.
dbGoToBookmark	A jump on an already create bookmark in the database.
dbInsertBookmark	Placement of a bookmark on a certain record in the database.
dbIsNoError	Test made to show whether an error occurred during work with the XTRA.
dbLast	A jump on the last record of the database.
dbMoveBy	A jump n records ahead in the database.
dbNext	A jump on the next record of the database.
dbOpen	Database file opening.
dbPrior	A jump on the previous record of the database.
dbRecordCount	Number of records of the database.
dbRecordNo	Record number of the actual record in the database.
dbXtraVersion	Finding out of version of XTRA dmmADO.x32.

dmmADO - Xtra for Director and Authorware

new

It is necessary to initiate the dmmADO XTRA before the first use.

If the library dmmADO.x32 is located in the XTRAS folder it is enough to insert

```
global ado
ado=new(xtra "dmmADO")
```

Example - Director

```
global ado
openXlib the pathName&"dmmADO.x32"
ado=new(xtra "dmmADO")
```

Example - Authorware

```
ado:=NewObject("dmmADO")
```

void=dbRegistration(name: string, code:string)

This function has to be called before the first use of the dmmADO immediately after its initialisation. Unless the right registration name and number are inserted, an announcement "this is a demo version" will appear.

Parameters

Type of name is string, for the demo version name = "dmm".

Type of code is string, for the demo version code= "demo". The chain for commercial version is unique.

To register the user will receive parameters code and name.

Example - Director

```
global ado
ado.dbRegistration("dmm", "demo")
```

Example - Authorware

```
CallObject(ado,"dbRegistration","dmm","demo")
```

boolean=dbActiveSQL(sql: string)

The function puts through an SQL command, whose type is SELECT. We are not about to explain the principle of SELECT commands and SQL language. We advise to read appropriate manuals. If the SQL order has been well accomplished, the function returns true. If an error occurred during the SQL order, the function returns false.

Parameters

The only parameter in this function is the SQL command SELECT.

Example - Director

```
global ado
if ado.dbActiveSQL("SELECT * FROM data WHERE ordNumber='abc' ") then
end if
```

Example - Authorware

```
CallObject(ado,"dbActiveSQL","SELECT * FROM data WHERE ordNumber='abc' ")
```

boolean=dbBof()

The function gives "true" in case the actual record is the first one in the database. Otherwise the function gives "false".

Parameters

There are no parameters in this function.

Example - Director

```
global ado
if not ado.dbBof() then
    airPlane=ado.dbFields()
end if
```

Example - Authorware

```
CallObject(ado,"dbBof")
```

dmmADO - Xtra for Director and Authorware

void=dbClearBookmarks()

The functions deletes all created bookmarks.

Parameters

There are no parametres in this function.

Example - Director

```
global ado
ado.dbClearBookmarks()
```

Example - Authorware

```
CallObject(ado,"dbClearBookmarks")
```

void=dbClose()

This function closes and finishes work with an mdb file.

Parameters

There are no parametres in this function.

Example - Director

```
global ado
ado.dbClose()
```

Example - Authorware

```
CallObject(ado,"dbClose")
```

integer=dbColumnCount()

The function gives number of fields in the database.

Parameters

There are no parametres in this function.

Example - Director

```
global ado
count=ado.dbColumnCount()
```

Example - Authorware

```
count:=CallObject(ado; "dbColumnCount")
```

void=dbColumns()

The function gives names of fields in the database. The values are in Abstract Data Types format.

[#field1: name1, #field2: name2, #field3: name3, ...]

For example [#field1:"ordNumber", #field2:"company", #field3:"type"]

Parameters

There are no parametres in this function.

Example - Director

```
global ado
fieldsName=ado.dbColumns()
put fieldsName.field1
put fieldsName.field2
put fieldsName.field3
```

Example - Authorware

```
fieldsName:=CallObject(ado,"dbColumns")
```

dmmADO - Xtra for Director and Authorware

void=dbConnection(connectionString: string)

This function connects dmmADO to the chosen database. To be able to get connected to the database, you have to enter connection string characteristic for the given database.

Parameters

The parameter of the function is connectionString. By this parameter the connection to database is defined. If you need to connect to a database you have to know what shape connectionString has for the given database.

Example - Director

```
global ado
ado.dbConnection("Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\Okc.mdb; Mode=ReadWrite;Persist Security Info=True")
```

Example - Authorware

```
CallObject(ado; "dbConnection", "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\Okc.mdb; Mode=ReadWrite;Persist Security Info=True")
```

void=dbDeleteBookmark(name: string)

The function deletes a bookmark of the given name.

Parameters

The parameter of the function is name, which represents name of the bookmark.

Example - Director

```
global ado
ado.dbDeleteBookmark("test1")
```

Example - Authorware

```
CallObject(ado; "dbDeleteBookmark", "test1")
```

void=dbDisconnection()

Cancellation of connection between the XTRA and database.

Parameters

There are no parameters in this function.

Example - Director

```
global ado
ado.Disconnection()
```

Example - Authorware

```
CallObject(ado, "Disconnection")
```

boolean=dbEof()

The function gives "true" in case the actual record is the last one in the database. Otherwise the function gives "false" .

Parameters

There are no parameters in this function.

Example - Director

```
global ado
if not ado.dbEof() then
    airPlane=ado.dbFields()
end if
```

Example - Authorware

```
CallObject(ado, "dbEof")
```

dmmADO - Xtra for Director and Authorware

void=dbErrorDialog(dialog: boolean)

Using this function we can set whether a dialogue window, showing error statement appears at the moment of an error or not. This setting only functions in Autor mode and it simplifies the application debugging. If you decide not to set the dialogue window you can check the errors using the function dbErrorMsg().

Parameters

Dialog is the only parameter of the function, whose type is boolean. If the value is true, the dialogue window with error statement is opened with every error in the XTRA. For the parameter value false the window is not shown. Default setting is dialog=false.

Example - Director

```
global ado
ado.dbErrorDialog(true)
```

Example - Authorware

```
CallObject(ado,"dbErrorDialog", true)
```

void=dbErrorLog(logFile: string, logSave: boolean)

Using this function we can set whether an error is saved to text log file at the time of the error or not. This setting only functions in Autor mode and it simplifies the application debugging.

Parameters

The function has two parameters. The parameter logFile represents name of the log file that we want to save the errors to. Type of the logSave parameter is boolean. If its value is true, every error in the XTRA is saved to log file. If its value is false nothing will be saved. Default setting is logSave=false.

Example - Director

```
global ado
ado.dbErrorLog(the pathName&"log.txt", true)
```

Example - Authorware

```
CallObject(ado; dbErrorLog, FileLocation ^ "log.txt", true)
```

string=dbErrorMsg()

The function gives text of the error that occurred when working with the database. If the operation ran correctly, it gives an empty chain.

Parameters

There are no parameters in this function.

Example - Director

```
global ado
ado.dbOpen(the pathName&"data.mdb", "abc")
error=ado.dbErrorMsg()
if error="" then airPlane=ado.dbFields()
```

Example - Authorware

```
error:=CallObject(ado,"dbErrorMsg")
```

boolean=dbExecSQL(sql: string)

The function puts through an SQL command, whose type is INSERT, UPDATE or DELETE. We are not about to explain the principle of INSERT, UPDATE or DELETE commands and SQL language. We advise to read appropriate manuals. If the SQL order has been well accomplished, the function returns true. If an error occurred during the SQL order, the function returns false.

Parameters

The only parameter in this function is the SQL command INSERT, UPDATE or DELETE.

dmmADO - Xtra for Director and Authorware

Example - Director

```
global ado
if ado.dbExecSQL("INSERT INTO data (ordNumber, company) VALUES ('dmmMBD27', 'Studio dmm') ") then
end if

if ado.dbExecSQL("UPDATE data SET registration='YU-AKO' WHERE ordNumber='A001' ") then
end if

if ado.dbExecSQL("DELETE FROM data") then
end if
```

Example - Authorware

```
CallObject(ado; "dbExecSQL","UPDATE data SET registration='YU-AKO' WHERE ordNumber='A001' ")
```

void=dbFirst()

The function sets the first record of the database as the actual one.

Parameters

There are no parameters in this function.

Example - Director

```
global ado
ado.dbFirst()
```

Example - Authorware

```
CallObject(ado,"dbFirst")
```

string=dbGetADOVersion()

The function returns number of version of the database engine for work with database files Microsoft Access (mdb), that is installed on the computer. Number of version is returned in the string format.

Parameters

There are no parameters in this function.

Example - Director

```
global ado
version=ado.dbGetADOVersion()
```

Example - Authorware

```
version:=CallObject(ado,"dbGetADOVersion")
```

void=dbGetBookmarks()

The function gives a list of defined bookmarks in the Abstract Data Types format ["name1", "name2", "name3", ...]

Parameters

There are no parameters in this function.

Example - Director

```
global ado
listBookmarks=ado.dbGetBookmarks()
```

Example - Authorware

```
listBookmarks:=CallObject(ado,"dbGetBookmarks")
```

dmmADO - Xtra for Director and Authorware

value=dbGetFieldByName(fieldName: string)

The function gives value of one item of the actual record in the database. The item is defined by its name. The function gives value in the format in which the item in the database is defined. If, for example the item ordNumber is defined as integer, the given value is integer too. When an error occurs in database, the function gives back an empty text item.

Parameters

The parameter of the function is fieldName, whose type is string and we insert name of the item defined in the database.

Example - Director

```
global ado
ordNumber=ado.dbGetFieldByName("ordNumber")
company=ado.dbGetFieldByName("company")
type=ado.dbGetFieldByName("type")
```

Example - Authorware

```
ordNumber:=CallObject(ado,"dbGetFieldByName","ordNumber")
company:=CallObject(ado,"dbGetFieldByName","company")
type:=CallObject(ado,"dbGetFieldByName","type")
```

value=dbGetFieldByNumber(fieldNo: string)

The function gives value of one item of the actual record in the database. The item is defined by its number in the database. The function gives value in the format in which the item in the database is defined. If, for example the item ordNumber is defined as integer, the given value is integer too. When an error occurs in database, the function gives back an empty text item.

Parameters

The parameter of the function is fieldNo, whose type is integer and we insert number of the item in the database. The parameter ranges from 1 (the first item) to the number dbColumnCount (the last item).

Example - Director

```
global ado
ordNumber=ado.dbGetFieldByNumber(1)
company=ado.dbGetFieldByNumber(2)
type=ado.dbGetFieldByNumber(3)
```

Example - Authorware

```
ordNumber:=CallObject(ado,"dbGetFieldByNumber", 1)
company:=CallObject(ado,"dbGetFieldByNumber", 2)
type:=CallObject(ado,"dbGetFieldByNumber", 3)
```

list=dbGetFields()

The function gives values of the actual record in the database. The values are in Abstract Data Types format.

[#fieldByName1: value1, #fieldByName2: value2, #fieldByName3: value3, ...]

For example [#ordNumber:"A227", #company:"PAMIR AIR", #type:"B-707-324C"]

Parameters

There are no parameters in this function.

Example - Director

```
global ado
airPlane=ado.dbGetFields()
put airPlane.ordNumber
put airPlane.company
put airPlane.type
```

Example - Authorware

```
airPlane:=CallObject(ado,"dbGetFields")
```

void=dbGoToBookmark(name: string)

The function sets as actual record the record that is represented by the given bookmark.

dmmADO - Xtra for Director and Authorware

Parameters

The parameter of the function is name, which represents name of the bookmark.

Example - Director

```
global ado
ado.dbGoToBookmark("test1")
```

Example - Authorware

```
CallObject(ado,"dbGoToBookmark","test1")
```

void=dbInsertBookmark(name: string)

The function creates and sets bookmark for the actual record of the database. In an application we can create a list of such bookmarks and then visit records that seemed somehow interesting to us.

Parameters

The parameter of the function is name, which represents name of the bookmark. This name must be unique. If we insert an already existing name, the existing bookmark will be rewritten.

Example - Director

```
global ado
ado.dbInsertBookmark("test1")
```

Example - Authorware

```
CallObject(ado,"dbInsertBookmark","test1")
```

boolean=dbIsNoError()

The function gives back true, if no error occurred during an operation. If an error occurs the function gives back false.

Parameters

There are no parameters in this function.

Example - Director

```
global ado
ado.dbOpen(the pathName&"data.mdb", "abc")
if ado.dbIsNoError() then airPlane=mdb.dbFields()
```

Example - Authorware

```
error:=CallObject(ado,"dbIsNoError")
```

void=dbLast()

The function sets the last record of the database as the actual one.

Parameters

There are no parameters in this function.

Example - Director

```
global ado
ado.dbLast()
```

Example - Authorware

```
CallObject(ado,"dbLast")
```

void=dbMoveBy(distance:integer)

The function jumps n records ahead in the database.

Parameters

Parameter of this function is distance. Distance equals how many records in the database we want to jump. If distance is >0 we go ahead, if it is <0 we go back.

dmmADO - Xtra for Director and Authorware

Example - Director

```
global ado
ado.dbMoveBy(17)
ado.dbMoveBy(-2)
```

Example - Authorware

```
CallObject(ado,"dbMoveBy"; 22)
```

void=dbNext()

The function sets the next record of the database as the actual one.

Parameters

There are no parameters in this function.

Example - Director

```
global ado
ado.dbNext()
```

Example - Authorware

```
CallObject(ado,"dbNext")
```

void=dbOpen(mdbFile, mdbPassword: string)

This function opens the mdb file for additional use. If the file is not opened, we cannot work with it.

Parameters

There are 2 parameters in this function. mdbFile is name of mdb file, and mdbPassword is password for the mdb file. If the password was not use when creating the mdb file, we insert an empty chain.

Example - Director

```
global ado
ado.dbOpen(the pathName&"data.mdb", "abc")
```

Example - Authorware

```
CallObject(ado; "dbOpen", FileLocation ^ "data.mdb", "abc")
```

void=dbPrior()

The function sets the previous record of the database as the actual one.

Parameters

There are no parameters in this function.

Example - Director

```
global ado
ado.dbPrior()
```

Example - Authorware

```
CallObject(ado,"dbPrior")
```

integer=dbRecordCount()

The function gives number of records in the database.

Parameters

There are no parameters in this function.

Example - Director

```
global ado
count=ado.dbRecordCount()
```

dmmADO - Xtra for Director and Authorware

Example - Authorware

```
count:=CallObject(ado,"dbRecordCount")
```

integer=dbRecordNo()

The function gives record number of the actual record in the database.

Parameters

There are no parameters in this function.

Example - Director

```
global ado  
number=ado.dbRecordNo()
```

Example - Authorware

```
number:=CallObject(ado,"dbRecordNo")
```

list=dbXtraVersion()

The function returns information about the XTRA dmmADO. The function returns values in the format Abstract Data Types:

```
[#fileType: "Xtra (32)",  
#CompanyName: "Studio dmm",  
#FileDescription: "XTRA dmmADO for work with database files Microsoft Access",  
#FileVersion: "1.9.0.21", #InternalName: "dmmADO",  
#LegalCopyRight: "© 1992-2009 Studio dmm",  
#LegalTradeMarks: "Studio dmm",  
#OriginalFileName: "dmmADO.x32",  
#productName: "dmmADO",  
#productVersion: "1.0.0.0"]
```

The meaning of the items is clear and it is not necessary to describe it closer.

Parameters

There are no parameters in this function.

Example - Director

```
global ado  
version=ado.dbXtraVersion()
```

Example - Authorware

```
version:=CallObject(ado,"dbXtraVersion")
```