



DaoX Xtras 1.0

- **Version Notes:**

28/2/2003 1.0

First Release version.

Fix some bugs on memory leaking.

This version is not Shockwave-safe.

The version only work on win32 platform with MFC 6.0 and MS DAO 3.60 installed.

13/2/2003 Alpha 1.0

First alpha version.

This version of Xtras is not Shockwave-safe.

The Xtras only work on win32 platform with MFC 6.0 and MS DAO 3.60 installed.

- **Code Example:**

1. **Make a Query:**

```
tempList = []

..*****|*****
--          Make query
..*****

--Create an instead of DaoX Xtras
set mDatabase = New(xtra "DaoX","")
--Open the database
mDatabase.daox_opendb(the moviepath & "testing.mdb")

--Make a Query on the database
mRecordset = mDatabase.daox_db_query("SELECT * FROM test")

--Loop Through the RecordSet
repeat while (not mDatabase.daox_isEOF(mRecordset))
  --Fetch the result into a list
  tempList = mDatabase.daox_fetch_list(mRecordset)

  --Print the value refer to field "key1"
  put "Value in record:" & tempList["key1"] & "," & tempList["key2"]

  --Move to next record
  mDatabase.daox_MoveNext(mRecordset)
end repeat

put "Total Reocrd Number is:" & mDatabase.daox_num_rows(mRecordset)
--Close the database after use
mDatabase.daox_closedb()
set mDatabase = 0
```

2. **Update Record:**

```
..*****|*****
--          Edit a Record
..*****

--Start Edit RecordSet
mDatabase.daox_Edit(mRecordset)
--Update the current record for field "key1" with value "abc"
mDatabase.daox_setValue(mRecordset,"key1","abc")
--Update the RecordSet into database
mDatabase.daox_Update(mRecordset)
```

3. Add a Record:

```

_*****
--          Add a Record
_*****

--*****First Create a recordset for a table*****

--Create a list equal to number of field of the table.
--Fill in the value you want.
tempValue = ["", "ac", "1234567"]

--Insert the value into the database
mDatabase.daox_AddNew(mRecordset, tempValue)

```

4. Delete a Record:

```

_*****
--          Delete a Record
_*****

mDatabase.daox_MoveFirst(mRecordset)

--Loop Through the RecordSet
repeat while (not mDatabase.daox_isEOF(mRecordset))
  --Fetch the result into a list
  tempList = mDatabase.daox_fetch_list(mRecordset)

  -- check if the field value equal to abc, delete it
  if tempList["key1"] = "ac" then
    mDatabase.daox_Delete(mRecordset)
    -- mDatabase.daox_Update(mRecordset)
  end if

  --Move to next record
  mDatabase.daox_MoveNext(mRecordset)
end repeat

```

- **Functions Reference:**

New

Syntax

Object = New(xtra "DaoX", string **option**)

Description

This function creates an instance of a DaoX Xtras. All functions call reference to this object.

daox_opendb

Syntax

Void daox_opendb (object **myDaoxXtras**, string **strConnection**)

Void myDaoxXtras.daox_opendb (string **strConnection**)

Description

This function opens a connection to the database. "strConnection" is string that hold the path to the database.

daox_closedb

Syntax

Void daox_closedb (object **myDaoxXtras**)

Void myDaoxXtras.daox_closedb ()

Description

This functions use to close the connection to the database. If you end using the database, you should use this function to close the database connect. In order, system can free up the resource.

daox_db_isOpen

Syntax

Boolean daox_db_isOpen (object **myDaoxXtras**)

Boolean myDaoxXtras.daox_db_isOpen ()

Description

Check if the database is opened. Return TRUE or False.

daox_db_query

Syntax

Integer **daox_db_query**(object **myDaoxXtras**, string **strQuery**)

Integer **myDaoxXtras.daox_db_query**(string **strQuery**)

Description

This function makes a query on the database and returns a reference to the record set that created. Return void if no record set created. **strQuery** is a standard SQL query string, you can use SELECT, INSERT or other valid SQL query.

daox_fetch_list

Syntax

PropertyList **daox_fetch_list**(object **myDaoxXtras**, integer **mRecordSet**)

PropertyList **myDaoxXtras.daox_fetch_list**(integer **mRecordSet**)

Description

This function will fetch the value into a list for current record. **mRecordSet** is the reference to the record set. Return a property list.

daox_MovePrev

Syntax

Void **daox_MovePrev**(object **myDaoxXtras**, integer **mRecordSet**)

Void **myDaoxXtras.daox_MovePrev**(integer **mRecordSet**)

Description

This function will move the record set pointer to previous position. **mRecordSet** is the reference to the record set.

daox_MoveNext

Syntax

Void **daox_MoveNext** (object **myDaoxXtras**, integer **mRecordSet**)

Void **myDaoxXtras.daox_MoveNext** (integer **mRecordSet**)

Description

This function will move the record set pointer to next position. **mRecordSet** is the reference to the record set.

daox_MoveFirst

Syntax

Void **daox_MoveFirst** (object **myDaoxXtras**, integer **mRecordSet**)

Void myDaoxXtras.daox_MoveFirst (integer **mRecordSet**)

Description

This function will move the record set pointer to start position of the record set. **mRecordSet** is the reference to the record set.

daox_MoveLast

Syntax

Void daox_MoveLast (object **myDaoxXtras**, integer **mRecordSet**)

Void myDaoxXtras.daox_MoveLast (integer **mRecordSet**)

Description

This function will move the record set pointer to last position of the record set. **mRecordSet** is the reference to the record set.

daox_isEOF

Syntax

Boolean daox_isEOF (object **myDaoxXtras**, integer **mRecordSet**)

Boolean myDaoxXtras.daox_isEOF (integer **mRecordSet**)

Description

Check if the record set pointer is at the end of record set. **mRecordSet** is the reference to the record set.

daox_num

Syntax

Integer daox_num_(rows object **myDaoxXtras**, integer **mRecordSet**)

Integer myDaoxXtras.daox_num_rows(integer **mRecordSet**)

Description

Return the record count of the record set. **mRecordSet** is the reference to the record set.

daox_Update

Syntax

Void daox_Update (object **myDaoxXtras**, integer **mRecordSet**)

Void myDaoxXtras.daox_Update (integer **mRecordSet**)

Description

Update the record set value in buffer into the database. It should use after **doax_Edit** and **doax_setValue** in order to update the value in the database. **mRecordSet** is the reference to the record set.

doax_Edit

Syntax

Void doax_Edit (object **myDaoxXtras**, integer **mRecordSet**)

Void myDaoxXtras.doax_Edit (integer **mRecordSet**)

Description

Start edit the recordset in buffer. Should use before **doax_setValue**. Must use **doax_Update** to update the value into database before and record set navigation function, otherwise the update will discard.

doax_setValue

Syntax

Void doax_setValue (object **myDaoxXtras**, integer **mRecordSet**, string **fieldName**, String **Value**)

Void myDaoxXtras.doax_setValue (integer **mRecordSet**, string **fieldName**, String **Value**)

Description

Set the value with the field name specified. Must use **doax_Update** to update the value into database before and record set navigation function, otherwise the update will discard. **fieldname** is the field name you want to update. **Value** is value you want to fill in.

doax_AddNew

Syntax

Void doax_AddNew (object **myDaoxXtras**, integer **mRecordSet**, list **listValue**)

Void myDaoxXtras.doax_AddNew (integer **mRecordSet**, list **listValue**)

Description

Insert a new record into the database. Create a list with number of elements equal to the number of field of the record set. Then use this function to insert into the database.

doax_Delete

Syntax

Void daox_Delete (object **myDaoxXtras**, integer **mRecordSet**)

Void myDaoxXtras.daox_Delete (integer **mRecordSet**)

Description

Delete the current record which the record set pointer point to. Must use **doax_Update** to update the value into database before and record set navigation function, otherwise the update will discard.

DaoX Xtras . Copyright 2003 DigiCrafts Co. All Rights Reserved.