

LocalDB

DLL Programmer's guide

index

LocalDB_Create()	page 3.
LocalDB_Delete()	page 3.
LocalDB_Open()	page 4.
LocalDB_Close()	page 4.
LocalDB_GetLastError()	page 5.
LocalDB_ExecuteQuery()	page 5.
LocalDB_GetFieldCount()	page 6.
LocalDB_GetFieldName()	page 6.
LocalDB_GetRowCount()	page 7.
LocalDB_GetData()	page 7.

function

```
int LocalDB_Create(char* szFullPath)
```

function description

Use this function to create a new database.

parameters

[in] szFullPath full path to the database file, i.e. c:\\test.db

return values

0 when everything went ok
not 0 when an error occurred. Get the error in textual form by calling
 LocalDB_GetLastError()

C/C++ example of usage

```
int nRet = LocalDB_Create("c:\\test.db");  
if (nRet != 0) {  
    fnAddMessage(_T("LocalDB_Create() failed"));  
    char *err = LocalDB_GetLastError();  
    fnAddMessage(CString(err));  
}
```

function

```
int LocalDB_Delete(char* szFullPath)
```

function description

Use this function to delete a database file

parameters

[in] szFullPath full path to the database file, i.e. c:\\test.db

return values

0 when everything went ok
not 0 when an error occurred. Get the error in textual form by calling
 LocalDB_GetLastError()

C/C++ example of usage

```
int nRet = LocalDB_Delete("c:\\test.db");  
if (nRet != 0) {  
    fnAddMessage(_T("LocalDB_Delete() failed"));  
    char *err = LocalDB_GetLastError();  
    fnAddMessage(CString(err));  
}
```

function

```
int LocalDB_Open(char* szFullPath)
```

function description

Use this function to open a database. The database file must exist.

parameters

[in] szFullPath full path to the database file, i.e. c:\\test.db

return values

0 when everything went ok
not 0 when an error occurred. Get the error in textual form by calling
 LocalDB_GetLastError()

C/C++ example of usage

```
int nRet = LocalDB_Open("c:\\test.db");  
if (nRet != 0) {  
    fnAddMessage(_T("LocalDB_Open() failed"));  
    char *err = LocalDB_GetLastError();  
    fnAddMessage(CString(err));  
}
```

function

```
void LocalDB_Close()
```

function description

Use this function to close a database

parameters

n/a

return values

n/a

function

```
void LocalDB_GetLastError(char* pszError)
```

function description

Use this function to get a textual description of the last error that has occurred.

parameters

[out] pszError buffer for textual error description
Note that you should allocate/deallocate this buffer. 255 bytes should be enough to hold all possible error messages.

return values

n/a

C/C++ example of usage

See example of LocalDB_Create()

function

```
int LocalDB_ExecuteQuery(char *szQuery)
```

function description

Use this function to execute an SQL query. After a successful calling of this function, you can retrieve the results of a query with the functions:

```
LocalDB_GetFieldCount()  
LocalDB_GetFieldName()  
LocalDB_GetRowCount()  
LocalDB_GetData()
```

parameters

[in] szQuery the SQL query

return values

0 when everything went ok
not 0 when an error occurred. Get the error in textual form by calling
 LocalDB_GetLastError()

C/C++ example of usage

```
int nRet = LocalDB_ExecuteQuery("UPDATE customers SET name='john' WHERE id=1");  
if (nRet != 0) {  
    fnAddMessage(_T("LocalDB_ExecuteQuery() failed"));  
    char *err = LocalDB_GetLastError();  
    fnAddMessage(CString(err));  
}
```

function

```
int LocalDB_GetFieldCount()
```

function description

Use this function to retrieve the number of fields (i.e. the number of columns) in an query result, after a successful SQL query.

parameters

n/a

return values

The number of fields

C/C++ example of usage

See example of `LocalDB_GetFieldName()`

function

```
int LocalDB_GetFieldName(char *szFieldName, int nIndex)
```

function description

You can use this function to retrieve field names, after a successful SQL query.

parameters

[out] szFieldName	buffer for field name Note that you should allocate/deallocate this buffer. 255 bytes should be enough to hold all possible error messages.
[in] nIndex	0-based index of field (get number of fields with <code>GetFieldCount()</code>)

return values

0	when everything went ok
not 0	when an error occurred. Get the error in textual form by calling <code>LocalDB_GetLastError()</code>

C/C++ example of usage

```
CString str = _T("");
for (int i=0; i < LocalDB_GetFieldCount(); i++) {
    char *name = LocalDB_GetFieldName(i);
    str += _T("[") + CString(name) + _T("] ");
}
fnAddMessage(_T("fields: ") + str);
```

function

```
int LocalDB_GetRowCount()
```

function description

Use this function to retrieve the number of rows in an query result, after a successful SQL query.

parameters

n/a

return values

The number of rows

C/C++ example of usage

See example of LocalDB_GetData()

function

```
int LocalDB_GetData(char *szData, int nRow, int nColumn)
```

function description

Use this function to retrieve row data, after a successful SQL query.

parameters

[out] szData	buffer to hold row element Note that you should allocate/deallocate this buffer. 8K bytes should normally be enough to hold all possible values.
[in] nRow	row number
[in] nColumn	column number

C/C++ example of usage

```
// get data
CString str = _T("");
for (int i=0; i < LocalDB_GetRowCount(); i++) {
    for (int j=0; j < LocalDB_GetFieldCount(); j++) {
        char *name = LocalDB_GetData(i,j);
        str += _T("[") + CString(name) + _T("] ");
    }
    fnAddMessage(_T("row: ") + str);
    str = _T("");
}
```