

# MySQLMobile.NET

*Programmer's guide*



## index

SetEncoding().....	page 3.
EnableCompression().....	page 3.
Connect().....	page 4.
Disconnect().....	page 5.
SelectDB().....	page 6.
ExecuteQuery().....	page 7.
GetLastInsertID().....	page 8.
GetLastServerMessage().....	page 9.
GetFieldCount().....	page 10.
GetFieldName().....	page 11.
GetRowCount().....	page 12.
GetData().....	page 13.
SetBufferSize().....	page 14.
Using binary data.....	page 14.

**function**

SetEncoding(codePage)

**function description**

Use this function to set the character set that should be used.

The default value is 437 (Extended ASCII set: OEM United States).

You can find a list of possibilities in the Help for *encoding.GetEncoding* in Visual Studio.

Note: this function is only available in the .NET control

**parameters**

[in] codePage (int)      codePage

**exceptions**

This function will throw an exception when the MySQLMobile.dll is not present, or could not be found.

**VB.NET example of usage**

```
' Set encoding  
mySQLMobile.SetEncoding(437)
```

**function**

EnableCompression (enable)

**function description**

Use this function to enable/disable compression in MySQLMobile. Compression is ON by default, so you don't have to call this function to enable it.

Make sure you call this function before calling any other MySQLMobile function.

**parameters**

[in] enable (bool)	true	→	enable compression
	false	→	disable compression

**exceptions**

This function will throw an exception when the MySQLMobile.dll is not present, or could not be found.

## function

Connect(host, port, username, password)

## function description

Use this function to connect to a mySQL database.

## parameters

[in] host (String)	host to connect to
[in] port (Integer/long)	port to connect to (default mySQL port is 3306)
[in] username (String)	username to access database
[in] password (String)	password to access database

## exceptions

This function will throw a MySQLMobileException if one or more of the following conditions are true:

- host is null or empty
- port is invalid
- username and/or password is null
- could not connect or authenticate to the MySQL server

The exception.Message will contain a detailed description of the problem.

## VB.NET example of usage

```
' Connect to mySQL
Try
    nRet = mySQLMobile.Connect("192.168.0.14", 3306, "john", "mypass")
    fnAddMessage("Connect() successful")
Catch ex As MySQLMobileException
    fnAddMessage("MySQLMobile exception: " + ex.Message)
End Try

Private Sub fnAddMessage(ByVal str As String)
    Dim MyCrLf As String = ChrW(13) + ChrW(10)
    TextBox1.Text = TextBox1.Text + str + MyCrLf
End Sub
```

**function**

Disconnect()

**function description**

Use this function to disconnect from a mySQL database.

**parameters**

n/a

**exceptions**

This function will throw a MySQLMobileException when disconnecting fails for some reason. The exception.Message will contain a detailed description of the problem.

**VB.NET example of usage**

```
Try
    mysqlMobile.Disconnect()
    fnAddMessage("Disconnect() successful")
Catch ex As MySQLMobileException
    fnAddMessage(ex.Message)
End Try
```

**function**

SelectDB(databaseName)

**function description**

Use this function to select a database to work with.

**parameters**

[in] databaseName (String)      database name

**exceptions**

This function will throw a MySQLMobileException if:

- databaseName is null or empty
- no database by that name could be found

The exception.Message will contain a detailed description of the problem.

**VB.NET example of usage**

```
Try
    mysqlMobile.SelectDB("test")
    fnAddMessage("SelectDB() successful")
Catch ex As MySQLMobileException
    fnAddMessage(ex.Message)
End Try
```

## function

ExecuteQuery(queryString)

## 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:

```
GetLastInsertId()  
GetLastServermessage()  
GetFieldCount()  
GetSingleField() or GetFields()  
GetRowCount()  
GetSingleRecord() or GetRowData()
```

Due to memory and/or bandwidth limitation of mobile devices, it is recommended to use the LIMIT keyword in SELECT queries. This allows you to go through tables step by step with for example 50 records at a time.

Also note that at this moment there is no support for BLOB fields so it is recommended to avoid using these.

## parameters

[in] queryString (String)      the SQL query

## exceptions

This function will throw a MySQLMobileException if:

- queryString is null or empty
- the query contains an error on database level

The exception.Message will contain a detailed description of the problem.

## VB.NET example of usage

```
Try  
    mysqlMobile.ExecuteQuery("UPDATE mytable SET name='john' WHERE id=1")  
    fnAddMessage("ExecuteQuery() successful")  
    fnAddMessage(mysqlMobile.GetLastServerMessage())  
Catch ex As MySQLMobileException  
    fnAddMessage(ex.Message)  
End Try
```

See the section on function GetSingleField() for an example of a SELECT query

**function**`GetLastInsertId()`**function description**

Use this function to get the id of the last inserted row, after a successful SQL query. As an alternative you could also fire the SQL query “SELECT LAST\_INSERT\_ID()”, which will give the same result.

**return values**

the last inserted id as Integer

**exceptions**

A MySQLMobile exception will be thrown when an internal error has occurred. The exception.Message will contain a detailed description of the problem.

**VB.NET example of usage**

```
Dim lastId As Integer
Try
    mysqlMobile.ExecuteNonQuery("INSERT INTO my_table (name) VALUES ('john')")
    lastId = mysqlMobile.GetLastInsertId()
Catch ex As MySQLMobileException
    fnAddMessage(ex.Message)
End Try
```

**function**

GetLastServerMessage()

**function description**

Use this function to retrieve the last server message, after a successful SQL query. You can use this function for example after INSERT or UPDATE queries. The server message has a format like (for example): Rows matched: 1 Changed: 1 Warnings: 0  
This message directly comes from the mySQL server.

**parameters**

n/a

**return values**

the server message as String

**exceptions**

A MySQLMobileException will be thrown when an internal error has occurred. The exception.Message will contain a detailed description of the problem.

**VB.NET example of usage**

See example of ExecuteQuery()

**function**

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 as Integer

**exceptions**

A MySQLMobileException will be thrown when an internal error has occurred. The exception.Message will contain a detailed description of the problem.

## function

GetSingleField(index) - to get single fields at a time  
GetFields() - to get all fields in one call

## function description

You can use this function to get the field names, after you've successfully fired a query, for example a SELECT query.

Note that you can also get (more) information about fields when you fire the query  
SHOW COLUMNS FROM <tablename>

This is useful when you want to know the default field value, for example.

## parameters

[in] index (Integer/long)      0-based index of field (get number of fields with  
GetFieldCount())

## return values

the field name as String

## exceptions

A MySQLMobileException will be thrown when an internal error has occurred or when an invalid index is specified.

The exception.Message will contain a detailed description of the problem.

## VB.NET example of usage

```
Dim fieldName As String
Dim str As String
Dim i As Integer

Try
    ' Execute a SELECT query
    mySQLMobile.ExecuteQuery("SELECT * FROM tbl1 LIMIT 20")
    ' Get field names
    str = ""
    For i = 0 To mySQLMobile.GetFieldCount() - 1
        fieldName = mySQLMobile.GetSingleField(i)
        str = str + "[" + fieldName + " ] "
    Next i
    fnAddMessage("fields: " + str)
Catch ex As MySQLMobileException
    fnAddMessage(ex.Message)
End Try
```

As an alternative, you can also call

```
Dim fields As List(Of String) = mySQLMobile.GetFields()
to get all the fields at once.
```

**function**

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 as Integer

**exceptions**

A MySQLMobileException will be thrown when an internal error has occurred.

The exception.Message will contain a detailed description of the problem.

**VB.NET example of usage**

See example of GetRowData()

## function

GetSingleRecord(rowIndex, columnIndex) - get a single record at a time  
GetRowData(rowIndex) - get the whole row at once

## function description

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

Note that the internal buffer to hold the result row data is limited to a length of 8192, which should be enough in most cases.

## parameters

[in] rowIndex (Integer/long)            row index (0 based)  
[in] columnIndex (Integer/long)        column index (0 based)

## return values

row element, as String

## exceptions

A MySQLMobileException will be thrown when an internal error has occurred or when an invalid index is specified.

The exception.Message will contain a detailed description of the problem.

## VB.NET example of usage

```
Try
    ' Execute a SELECT query
    mySQLMobile.ExecuteNonQuery("SELECT * FROM tbl1 LIMIT 20")
    Dim temp As String = ""
    For i As Integer = 0 To mySQLMobile.GetRowCount() - 1
        Dim rowData As List(Of String) = mySQLMobile.GetRowData(i)
        temp = ""
        For Each record As String In rowData
            temp += record + Chr(9) + Chr(9)
        Next
        fnAddMessage("record: " + temp)
    Next
Catch ex As MySQLMobileException
    fnAddMessage(ex.Message)
End Try
```

## function

SetBufferSize(size)

## function description

Use this function to set the buffer size for data fields. You can set this size whenever you want, for example before a call to `GetData()`. The default buffer size is 8192 bytes (8K). You should try to keep the buffer size as small as possible to avoid memory shortage.

## parameters

[in] size (int/long)      Buffer size

## exceptions

This function will throw an exception when the `MySQLMobile.dll` is not present, or could not be found.

## VB.NET example of usage

```
' Set buffer size to 64 KB  
mySQLMobile.SetBufferSize(64*1024)
```

## Using binary data

If you want to use binary data, you should make sure it's base64 encoded in a text field (for example a *mediumtext* field) in MySQL. After getting the field data with `MySQLMobile`, you can decode the base64 encoded string back to binary data.

→ Note that you should disable compression in `MySQLMobile` or you will get empty results. Use the `EnableCompression()` function with its argument set to `false` to do this.

→ Make sure you set the buffer to be large enough to hold the data. Use the `SetBufferSize()` function to do this. The default buffer size is only 8192 bytes.

→ Query 1 data record at a time to minimize memory requirements.