



Denne guide er oprindeligt udgivet på Eksperten.dk

## MySQL C API

Denne artikel beskriver hvordan man bruger MySQL C API.

Der er beskrivelse af build med forskellige compilere.

Den forudsætter lidt kendskab til C og SQL.

Skrevet den **05. Feb 2009** af **arne\_v** | kategorien **Programmering / C/C++** | ★★☆☆☆

Historie:

V1.0 - 07/03/2004 - original

V1.1 - 01/04/2004 - korrekt håndtering af NULL

V1.2 - 20/08/2005 - ny procedure for GCC/MinGW32

V1.3 - 26/12/2008 - små ændringer

### MySQL C API

MySQL kommer med mange forskellige client API'er.

Men et af de mest low level er C API.

Denne artikel vil beskrive hvordan man lave simple forespørgsler og opdateringer i en MySQL database via MySQL C API.

Dokumentationen til C API'et findes her:

<a href="http://dev.mysql.com/doc/refman/5.1/en/c.html">http://dev.mysql.com/doc/refman/5.1/en/c.html</a>  
≥

Eksemplerne vil bruge denne meget simple database:

```
CREATE TABLE t1 (  
  f1 INTEGER;  
  f2 VARCHAR(50),  
  PRIMARY KEY(f1)  
);
```

Eksemplerne vil være i C d.v.s. kunne bruges både i C og C++.

Eksemplerne er meget simple, men bør nemt kunne udvides til noget mere avanceret udfra dokumentationen på ovennævnte link.

Den PHP kyndige vil notere at PHP mysql extension og MySQL C API ligner hinanden en del.

## Forespørgsel

Eksempel:

```
#include <stdio.h>
#include <stdlib.h>

#include "mysql.h"

int main()
{
    /* connection */
    MYSQL *handle;
    /* query result */
    MYSQL_RES *result;
    /* række i query result */
    MYSQL_ROW row;
    /* antal felter i query result */
    int nfields;
    /* pointer til array med felt længder i række i query result */
    int *l;
    /* counter */
    int i;
    /*
     * åben connection til:
     *   server = "localhost"
     *   username = "root"
     *   password = ""
     *   database = "Test"
     *   port = 0 (bliver opfattet som default 3306)
     */
    handle= mysql_init(NULL);
    if(handle == NULL)
    {
        printf("MySQL error: %s", mysql_error(handle));
        exit(1);
    }
    if(!mysql_real_connect(handle, "localhost", "root", "", "Test", 0, NULL,
0))
    {
        printf("MySQL error: %s", mysql_error(handle));
        exit(1);
    }
    /* udfør query */
    mysql_query(handle, "SELECT * FROM t1");
    result = mysql_store_result(handle);
    /* print resultat af query */
    nfields = mysql_num_fields(result);
    while ((row = mysql_fetch_row(result))) {
        l = (int *)mysql_fetch_lengths(result);
        for (i=0; i<nfields; i++) {
            printf(" %.*s", row[i] ? l[i] : 4, row[i] ? row[i] : "NULL");
        }
    }
}
```

```

        printf("\n");
    }
    /* luk query */
    mysql_free_result(result);
    /* luk connection */
    mysql_close(handle);
    return 0;
}

```

## Opdatering

Eksempel:

```

#include <stdio.h>
#include <stdlib.h>

#include "mysql.h"

int main()
{
    /* connection */
    MYSQL *handle;
    /* SQL kommando */
    char sqlcmd[200];
    /* counter */
    int i;
    /*
    * åben connection til:
    * server = "localhost"
    * username = "root"
    * password = ""
    * database = "Test"
    * port = 0 (bliver opfattet som default 3306)
    */
    handle= mysql_init(NULL);
    if(handle == NULL)
    {
        printf("MySQL error: %s", mysql_error(handle));
        exit(1);
    }
    if(!mysql_real_connect(handle, "localhost", "root", "", "Test", 0, NULL,
0))
    {
        printf("MySQL error: %s", mysql_error(handle));
        exit(1);
    }
    /* indsæt 10 rækker */
    for(i=0; i<10; i++)
    {
        sprintf(sqlcmd, "INSERT INTO t1 VALUES (%d,'%s')", i, "Dette er en
test");
    }
}

```

```
        mysql_query(handle, sqlcmd);
    }
    /* luk connection */
    mysql_close(handle);
    return 0;
}
```

## Build

Nedenfor angives hvordan man builder command line med forskellige compilere.

Hvis man bruger en IDE skal man lave noget tilsvarende.

Microsoft VC++:

- \* include from MySQL include directory
- \* link med dynamisk library
- \* define SOCKET til at være int
- \* sæt PATH til directory med DLL

```
cl /DSOCKET=int /I\mysql\include query.c \mysql\lib\opt\libmysql.lib
cl /DSOCKET=int /I\mysql\include insert.c \mysql\lib\opt\libmysql.lib
PATH=\mysql\lib\opt;%PATH%
```

Borland C++ Builder:

- \* convert dynamisk library fra COFF til OMF
- \* include from MySQL include directory
- \* link med dynamisk library
- \* define SOCKET til at være int
- \* sæt PATH til directory med DLL

```
coff2omf \mysql\lib\opt\libmysql.lib .\libmysql.lib
bcc32 -DSOCKET=int -I\mysql\include query.c libmysql.lib
bcc32 -DSOCKET=int -I\mysql\include insert.c libmysql.lib
PATH=\mysql\lib\opt;%PATH%
```

GCC/MinGW32 (nye versioner):

- \* converter dynamisk library fra .LIB til .A
- \* include from MySQL include directory
- \* link med dynamisk library
- \* define SOCKET til at være int
- \* sæt PATH til directory med DLL

```
copy \mysql\lib\opt\libmysql.lib *.*
reimp -d libmysql.lib
dlltool -k -d libmysql.def -l libmysql.a
gcc -DSOCKET=int -I\mysql\include query.c libmysql.a -o query.exe
gcc -DSOCKET=int -I\mysql\include insert.c libmysql.a -o insert.exe
PATH=\mysql\lib\opt;%PATH%
```

GCC/MinGW32 (gamle versioner):

- \* generer special library for at få \_chkstk funktionen
- \* include from MySQL include directory
- \* link med statisk library (dynamisk library har nogle calling convention problemer)
- \* link med winsock2 og MS VC++ runtime library
- \* link med special library
- \* define SOCKET til at være int

```
dlltool -D/WinNT/System32/ntdll.dll -dlibntdll.def -llibntdll.a
gcc -DSOCKET=int -I\mysql\include query.c \mysql\lib\opt\mysqlclient.lib
libntdll.a -lws2_32 -lmsvcrt40 -o query.exe
gcc -DSOCKET=int -I\mysql\include insert.c \mysql\lib\opt\mysqlclient.lib
libntdll.a -lws2_32 -lmsvcrt40 -o insert.exe
```

libntdll.def:

```
LIBRARY ntdll.dll
EXPORTS
_chkstk
```

GCC/Linux:

- \* include from MySQL include directory
- \* link med statisk library
- \* link med compress library

```
gcc -I/usr/include/mysql query.c /usr/lib/mysql/libmysqlclient.a -lz -o query
gcc -I/usr/include/mysql insert.c /usr/lib/mysql/libmysqlclient.a -lz -o insert
```

**C++**

For løsninger der udnytter OOP mulighederne i C++ se:

<a href="http://www.eksperten.dk/artikler/244">Brug af MySQL i C++</a>

**Kommentar af danielhep d. 19. Mar 2004 | 1**

En rigtig god artikel. :)

**Kommentar af htm d. 07. Mar 2004 | 2**

Endnu en god arktikel fra arne\_v. Giver en god forståelse til hvordan man starter med MySQL C API.

**Kommentar af mathiash d. 03. Oct 2005 | 3**

Rigtigt god artikel