**1. /usr/include/mysql/mysql\_com.h**

typedef struct st\_net {

#if !defined(CHECK\_EMBEDDED\_DIFFERENCES) || !defined(EMBEDDED\_LIBRARY)

Vio \*vio;

unsigned char \*buff,\*buff\_end,\*write\_pos,\*read\_pos;

my\_socket fd; /\* For Perl DBI/dbd \*/

/\* The following variable is set if we are doing several queries in one command ( as in LOAD TABLE ... FROM MASTER ),

\* and do not want to confuse the client with OK at the wrong time \*/

unsigned long remain\_in\_buf,length, buf\_length, where\_b;

unsigned long max\_packet,max\_packet\_size;

unsigned int pkt\_nr,compress\_pkt\_nr;

unsigned int write\_timeout, read\_timeout, retry\_count;

int fcntl;

unsigned int \*return\_status;

unsigned char reading\_or\_writing;

char save\_char;

my\_bool unused0; /\* Please remove with the next incompatible ABI change. \*/

my\_bool unused; /\* Please remove with the next incompatible ABI change \*/

my\_bool compress;

my\_bool unused1; /\* Please remove with the next incompatible ABI change. \*/

/\* Pointer to query object in query cache, do not equal NULL (0) for queries in cache that have not stored its results yet \*/

#endif

/\* 'query\_cache\_query' should be accessed only via query cache functions and methods to maintain proper locking. \*/

unsigned char \*query\_cache\_query;

unsigned int last\_errno;

unsigned char error;

my\_bool unused2; /\* Please remove with the next incompatible ABI change. \*/

my\_bool return\_errno; /\*\* Client library error message buffer. Actually belongs to struct MYSQL. \*/

char last\_error[MYSQL\_ERRMSG\_SIZE]; /\*\* Client library sqlstate buffer. Set along with the error message. \*/

char sqlstate[SQLSTATE\_LENGTH+1];

void \*extension;

#if defined(MYSQL\_SERVER) && !defined(EMBEDDED\_LIBRARY)

/\* Controls whether a big packet should be skipped. Initially set to FALSE by default. Unauthenticated sessions must have this set to

\*FALSE so that the server can't be tricked to read packets indefinitely. \*/

my\_bool skip\_big\_packet;

#endif

} NET;

enum enum\_field\_types { MYSQL\_TYPE\_DECIMAL, MYSQL\_TYPE\_TINY,

MYSQL\_TYPE\_SHORT, MYSQL\_TYPE\_LONG,

MYSQL\_TYPE\_FLOAT, MYSQL\_TYPE\_DOUBLE,

MYSQL\_TYPE\_NULL, MYSQL\_TYPE\_TIMESTAMP,

MYSQL\_TYPE\_LONGLONG,MYSQL\_TYPE\_INT24,

MYSQL\_TYPE\_DATE, MYSQL\_TYPE\_TIME,

MYSQL\_TYPE\_DATETIME, MYSQL\_TYPE\_YEAR,

MYSQL\_TYPE\_NEWDATE, MYSQL\_TYPE\_VARCHAR,

MYSQL\_TYPE\_BIT,

MYSQL\_TYPE\_NEWDECIMAL=246,

MYSQL\_TYPE\_ENUM=247,

MYSQL\_TYPE\_SET=248,

MYSQL\_TYPE\_TINY\_BLOB=249,

MYSQL\_TYPE\_MEDIUM\_BLOB=250,

MYSQL\_TYPE\_LONG\_BLOB=251,

MYSQL\_TYPE\_BLOB=252,

MYSQL\_TYPE\_VAR\_STRING=253,

MYSQL\_TYPE\_STRING=254,

MYSQL\_TYPE\_GEOMETRY=255

};

/\* For backward compatibility \*/

#define CLIENT\_MULTI\_QUERIES CLIENT\_MULTI\_STATEMENTS

#define FIELD\_TYPE\_DECIMAL MYSQL\_TYPE\_DECIMAL

#define FIELD\_TYPE\_NEWDECIMAL MYSQL\_TYPE\_NEWDECIMAL

#define FIELD\_TYPE\_TINY MYSQL\_TYPE\_TINY

#define FIELD\_TYPE\_SHORT MYSQL\_TYPE\_SHORT

#define FIELD\_TYPE\_LONG MYSQL\_TYPE\_LONG

#define FIELD\_TYPE\_FLOAT MYSQL\_TYPE\_FLOAT

#define FIELD\_TYPE\_DOUBLE MYSQL\_TYPE\_DOUBLE

#define FIELD\_TYPE\_NULL MYSQL\_TYPE\_NULL

#define FIELD\_TYPE\_TIMESTAMP MYSQL\_TYPE\_TIMESTAMP

#define FIELD\_TYPE\_LONGLONG MYSQL\_TYPE\_LONGLONG

#define FIELD\_TYPE\_INT24 MYSQL\_TYPE\_INT24

#define FIELD\_TYPE\_DATE MYSQL\_TYPE\_DATE

#define FIELD\_TYPE\_TIME MYSQL\_TYPE\_TIME

#define FIELD\_TYPE\_DATETIME MYSQL\_TYPE\_DATETIME

#define FIELD\_TYPE\_YEAR MYSQL\_TYPE\_YEAR

#define FIELD\_TYPE\_NEWDATE MYSQL\_TYPE\_NEWDATE

#define FIELD\_TYPE\_ENUM MYSQL\_TYPE\_ENUM

#define FIELD\_TYPE\_SET MYSQL\_TYPE\_SET

#define FIELD\_TYPE\_TINY\_BLOB MYSQL\_TYPE\_TINY\_BLOB

#define FIELD\_TYPE\_MEDIUM\_BLOB MYSQL\_TYPE\_MEDIUM\_BLOB

#define FIELD\_TYPE\_LONG\_BLOB MYSQL\_TYPE\_LONG\_BLOB

#define FIELD\_TYPE\_BLOB MYSQL\_TYPE\_BLOB

#define FIELD\_TYPE\_VAR\_STRING MYSQL\_TYPE\_VAR\_STRING

#define FIELD\_TYPE\_STRING MYSQL\_TYPE\_STRING

#define FIELD\_TYPE\_CHAR MYSQL\_TYPE\_TINY

#define FIELD\_TYPE\_INTERVAL MYSQL\_TYPE\_ENUM

#define FIELD\_TYPE\_GEOMETRY MYSQL\_TYPE\_GEOMETRY

#define FIELD\_TYPE\_BIT MYSQL\_TYPE\_BIT

**2. /usr/include/mysql/mysql.h**

extern unsigned int mysql\_port;

extern char \*mysql\_unix\_port;

#define CLIENT\_NET\_READ\_TIMEOUT 365\*24\*3600 /\* Timeout on read \*/

#define CLIENT\_NET\_WRITE\_TIMEOUT 365\*24\*3600 /\* Timeout on write \*/

typedef struct st\_mysql\_field {

char \*name; /\* Name of column \*/

char \*org\_name; /\* Original column name, if an alias \*/

char \*table; /\* Table of column if column was a field \*/

char \*org\_table; /\* Org table name, if table was an alias \*/

char \*db; /\* Database for table \*/

char \*catalog; /\* Catalog for table \*/

char \*def; /\* Default value (set by mysql\_list\_fields) \*/

unsigned long length; /\* Width of column (create length) \*/

unsigned long max\_length; /\* Max width for selected set \*/

unsigned int name\_length;

unsigned int org\_name\_length;

unsigned int table\_length;

unsigned int org\_table\_length;

unsigned int db\_length;

unsigned int catalog\_length;

unsigned int def\_length;

unsigned int flags; /\* Div flags \*/

unsigned int decimals; /\* Number of decimals in field \*/

unsigned int charsetnr; /\* Character set \*/

enum enum\_field\_types type; /\* Type of field. See mysql\_com.h for types \*/

void \*extension;

**} MYSQL\_FIELD;**

**typedef char \*\*MYSQL\_ROW; /\* return data as array of strings \*/**

typedef unsigned int MYSQL\_FIELD\_OFFSET; /\* offset to current field \*/

typedef struct st\_mysql\_rows {

struct st\_mysql\_rows \*next; /\* list of rows \*/

MYSQL\_ROW data;

unsigned long length;

**} MYSQL\_ROWS;**

typedef struct embedded\_query\_result EMBEDDED\_QUERY\_RESULT;

typedef struct st\_mysql\_data {

MYSQL\_ROWS \*data;

struct embedded\_query\_result \*embedded\_info;

MEM\_ROOT alloc;

my\_ulonglong rows;

unsigned int fields;

void \*extension; /\* extra info for embedded library \*/

**} MYSQL\_DATA;**

struct st\_mysql\_options {

unsigned int connect\_timeout, read\_timeout, write\_timeout;

unsigned int port, protocol;

unsigned long client\_flag;

char \*host,\*user,\*password,\*unix\_socket,\*db;

struct st\_dynamic\_array \*init\_commands;

char \*my\_cnf\_file,\*my\_cnf\_group, \*charset\_dir, \*charset\_name;

char \*ssl\_key; /\* PEM key file \*/

char \*ssl\_cert; /\* PEM cert file \*/

char \*ssl\_ca; /\* PEM CA file \*/

char \*ssl\_capath; /\* PEM directory of CA-s? \*/

char \*ssl\_cipher; /\* cipher to use \*/

char \*shared\_memory\_base\_name;

unsigned long max\_allowed\_packet;

my\_bool use\_ssl; /\* if to use SSL or not \*/

my\_bool compress,named\_pipe;

/\*On connect, find out the replication role of the server, and establish connections to all the peers \*/

my\_bool rpl\_probe;

/\*Each call to mysql\_real\_query() will parse it to tell if it is a read or a write, and direct it to the slave or the master \*/

my\_bool rpl\_parse;

/\*If set, never read from a master, only from slave, when doing a read that is replication-aware \*/

my\_bool no\_master\_reads;

#if !defined(CHECK\_EMBEDDED\_DIFFERENCES) || defined(EMBEDDED\_LIBRARY)

my\_bool separate\_thread;

#endif

enum mysql\_option methods\_to\_use;

char \*client\_ip;

/\* Refuse client connecting to server if it uses old (pre-4.1.1) protocol \*/

my\_bool secure\_auth;

/\* 0 - never report, 1 - always report (default) \*/

my\_bool report\_data\_truncation;

/\* function pointers for local infile support \*/

int (\*local\_infile\_init)(void \*\*, const char \*, void \*);

int (\*local\_infile\_read)(void \*, char \*, unsigned int);

void (\*local\_infile\_end)(void \*);

int (\*local\_infile\_error)(void \*, char \*, unsigned int);

void \*local\_infile\_userdata;

void \*extension;

};

enum mysql\_status

{

MYSQL\_STATUS\_READY, MYSQL\_STATUS\_GET\_RESULT, MYSQL\_STATUS\_USE\_RESULT,

MYSQL\_STATUS\_STATEMENT\_GET\_RESULT

};

struct st\_mysql\_methods;

struct st\_mysql\_stmt;

**typedef struct st\_mysql**

**{**

**NET net; /\* Communication parameters \*/**

**unsigned char \*connector\_fd; /\* ConnectorFd for SSL \*/**

**char \*host,\*user,\*passwd,\*unix\_socket,\*server\_version,\*host\_info;**

**char \*info, \*db;**

**struct charset\_info\_st \*charset;**

**MYSQL\_FIELD \*fields;**

**MEM\_ROOT field\_alloc;**

**my\_ulonglong affected\_rows;**

**my\_ulonglong insert\_id; /\* id if insert on table with NEXTNR \*/**

**my\_ulonglong extra\_info; /\* Not used \*/**

**unsigned long thread\_id; /\* Id for connection in server \*/**

**unsigned long packet\_length;**

**unsigned int port;**

**unsigned long client\_flag,server\_capabilities;**

**unsigned int protocol\_version;**

**unsigned int field\_count;**

**unsigned int server\_status;**

**unsigned int server\_language;**

**unsigned int warning\_count;**

**struct st\_mysql\_options options;**

**enum mysql\_status status;**

**my\_bool free\_me; /\* If free in mysql\_close \*/**

**my\_bool reconnect; /\* set to 1 if automatic reconnect \*/**

**char scramble[SCRAMBLE\_LENGTH+1]; /\* session-wide random string \*/**

**/\*Set if this is the original connection, not a master or a slave we have**

**\*added though mysql\_rpl\_probe() or mysql\_set\_master()/ mysql\_add\_slave() \*/**

**my\_bool rpl\_pivot;**

**/\* Pointers to the master, and the next slave connections, points to itself if lone connection. \*/**

**struct st\_mysql\* master, \*next\_slave;**

**struct st\_mysql\* last\_used\_slave; /\* needed for round-robin slave pick \*/**

**/\* needed for send/read/store/use result to work correctly with replication \*/**

**struct st\_mysql\* last\_used\_con;**

**LIST \*stmts; /\* list of all statements \*/**

**const struct st\_mysql\_methods \*methods;**

**void \*thd;**

**/\* Points to boolean flag in MYSQL\_RES or MYSQL\_STMT. We set this flag**

**\* from mysql\_stmt\_close if close had to cancel result set of this object. \*/**

**my\_bool \*unbuffered\_fetch\_owner;**

**/\* needed for embedded server - no net buffer to store the 'info' \*/**

**char \*info\_buffer;**

**void \*extension;**

**} MYSQL;**

**typedef struct st\_mysql\_res {**

**my\_ulonglong row\_count;**

**MYSQL\_FIELD \*fields;**

**MYSQL\_DATA \*data;**

**MYSQL\_ROWS \*data\_cursor;**

**unsigned long \*lengths; /\* column lengths of current row \*/**

**MYSQL \*handle; /\* for unbuffered reads \*/**

**const struct st\_mysql\_methods \*methods;**

**MYSQL\_ROW row; /\* If unbuffered read \*/**

**MYSQL\_ROW current\_row; /\* buffer to current row \*/**

**MEM\_ROOT field\_alloc;**

**unsigned int field\_count, current\_field;**

**my\_bool eof; /\* Used by mysql\_fetch\_row \*/**

**/\* mysql\_stmt\_close() had to cancel this result \*/**

**my\_bool unbuffered\_fetch\_cancelled;**

**void \*extension;**

**} MYSQL\_RES;**