MaxDB

Introduction

The MaxDB PHP extension allows you to access the functionality provided by MaxDB 7.5.0 and above. More information about the MaxDB Database server can be found at » http://www.sdn.sap.com/irj/sdn/maxdb.

The MaxDB PHP extension is compatible to the MySQL mysqli extension. There are only minor differences in the behaviour of some functions due to the differences of the underlying database servers, MaxDB and MySQL.

The main differences to mysqli are in the following functions:

- maxdb_character_set_name() - Returns only ascii or unicode.
- maxdb_get_client_info() - Returns a different version string.
- maxdb_get_client_version() - Returns a different version string.
- maxdb_get_host_info() - Returns localhost or hostname.
- maxdb_get_server_info() - Returns a different version string.
- maxdb_get_server_version() - Returns a different version string.
- maxdb_kill() - Only disconnects the session.
- maxdb_multi_query() - Can not handle multiple SQL statements.
- maxdb_next_result() - Function returns always false.
- maxdb_options() - Supports a different set of options.
- maxdb_report() - Supports a different report mode.
- maxdb_stat() - Returns a different system status string.
- maxdb_stmt_store_result() - Is not necessary for MaxDB.
- maxdb_store_result() - Is not necessary for MaxDB.

Documentation for MaxDB can be found at » http://maxdb.sap.com/documentation/.

Installing/Configuring

Requirements

In order to have these functions available, you must compile PHP with MaxDB support. Additionally, you must have the MaxDB SQLDBC runtime library available to access the MaxDB server.

Documentation for MaxDB SQLDBC can be found at
» http://maxdb.sap.com/documentation/.

Download the MaxDB SQLDBC package from
» http://www.sdn.sap.com/irj/sdn/maxdb-downloads.

Installation

By using the *--with-maxdb[=DIR]* configuration option you enable PHP to access MaxDB databases. *[DIR]* points to the directory that contains the installed MaxDB SQLDBC package.

Windows users will need to enable *php_maxdb.dll* inside of *php.ini*.

Runtime Configuration

The behaviour of these functions is affected by settings in *php.ini*.

MaxDB Configuration Options

Name	Default	Changeable	Changelog
maxdb.default_host	NULL	PHP_INI_ALL	
maxdb.default_db	NULL	PHP_INI_ALL	
maxdb.default_user	NULL	PHP_INI_ALL	
maxdb.default_pw	NULL	PHP_INI_ALL	
maxdb.long_readlen	"200"	PHP_INI_ALL	

For further details and definitions of the PHP_INI_* constants, see the php.ini directives.

Here's a short explanation of the configuration directives.

maxdb.default_host string
: The default server host to use when connecting to the database server if no other host is specified.

maxdb.default_db string
: The default server database to use when connecting if no other database is specified.

maxdb.default_user string
: The default user name to use when connecting to the database server if no other name is specified.

maxdb.default_pw string
: The default password to use when connecting to the database server if no other password is specified.

maxdb.long_readlen integer
: The default maximum length of bytes that is transferred to the client if long data is retrieved from the MaxDB database server.

Resource Types

This extension defines resources.

Predefined Constants

The constants below are defined by this extension, and will only be available when the extension has either been compiled into PHP or dynamically loaded at runtime.

The following constants to use with maxdb_options() are defined. For further description of these constants see » http://maxdb.sap.com/documentation/.

MaxDB PHP client constants

Constant	Description
MAXDB_COMPNAME	The component name used to initialise the SQLDBC runtime environment.
MAXDB_APPLICATION	The application to be connected to the database.
MAXDB_APPVERSION	The version of the application.
MAXDB_SQLMODE	The SQL mode.
MAXDB_UNICODE	TRUE, if the connection is an unicode (UCS2) client or FALSE, if not.
MAXDB_TIMEOUT	The maximum allowed time of inactivity after which the connection to the database is closed by the system.
MAXDB_ISOLATIONLEVEL	Specifies whether and how shared locks and exclusive locks are implicitly requested or released.
MAXDB_PACKETCOUNT	The number of different request packets used for the connection.
MAXDB_STATEMENTCACHESIZE	The number of prepared statements to be cached for the connection for re-use.
MAXDB_CURSORPREFIX	The prefix to use for result tables that are automatically named.

The function maxdb_fetch_array() uses a constant for the different types of result arrays. The following constants are defined:

MaxDB fetch constants

Constant	Description

MAXDB_ASSOC	Columns are returned into the array having the fieldname as the array index.
MAXDB_ASSOC_UPPER	Columns are returned into the array having the upper case fieldname as the array index.
MAXDB_ASSOC_LOWER	Columns are returned into the array having the lower case fieldname as the array index.
MAXDB_BOTH	Columns are returned into the array having both a numerical index and the fieldname as the array index.
MAXDB_NUM	Columns are returned into the array having a numerical index to the fields. This index starts with 0, the first field in the result.

Examples

All examples in the MaxDB PHP documentation use the HOTELDB demo database from MaxDB. More about this database can be found at » http://maxdb.sap.com/doc/7_7/44/d8c25164bb38d0e10000000a1553f7/content.htm.

To use the examples in the MaxDB PHP documentation, you have to load the tutorial data into your database. Then you have to set maxdb.default_db in *php.ini* to the database that contains the tutorial data.

This simple example shows how to connect, execute a query, print resulting rows and disconnect from a MaxDB database.

Example #1 - MaxDB extension overview example

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* Performing SQL query */
$query = "SELECT * FROM hotel.city";
$result = maxdb_query($link, $query) or die("Query failed : " .
maxdb_error());

/* Printing results in HTML */
echo "<table>\n";
while ($line = maxdb_fetch_array($result, MAXDB_ASSOC)) {
   echo "  <tr>\n";
   foreach ($line as $col_value) {
       echo "    <td>$col_value</td>\n";
   }
   echo "  </tr>\n";
}
echo "</table>\n";

/* Free resultset */
maxdb_free_result($result);

/* Closing connection */
maxdb_close($link);
?>
```

The following example shows how to bind variables to a SELECT INTO statement.

Example #2 - Example for use of SELECT INTO statements

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* Performing SQL query */
$stmt = maxdb_prepare ($link, "SELECT percentage INTO ? FROM
hotel.countrylanguage where language = ?");
if (!$stmt) {
 printf ("Prepare failed: %s\n", maxdb_error($link));
}

$name = "Mbundu";

maxdb_stmt_bind_param($stmt, 'ds', $percentage, $name);
maxdb_stmt_execute($stmt);

printf ("%f\n", $percentage);

maxdb_stmt_close ($stmt);
?>
```

The following example shows how to use MaxDB database procedures.

Example #3 - Example fore using database procedures

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_report (MAXDB_REPORT_OFF);
maxdb_query($link,"DROP DBPROC test_proc");
maxdb_report (MAXDB_REPORT_ERROR);

$query = "create dbproc test_proc (INOUT e_text char(72)) AS select * from
SYSDBA.DUAL; fetch into :e_text;";

maxdb_query($link, $query);

/* Performing SQL query */
$stmt = maxdb_prepare ($link, "CALL test_proc (?)");
if (!$stmt) {
 printf ("Prepare failed: %s\n", maxdb_error($link));
}
```

```
maxdb_stmt_bind_param($stmt, 's', $result);
maxdb_stmt_execute($stmt);

printf ("%s\n", $result);

maxdb_stmt_close ($stmt);
?>
```

MaxDB Functions

Predefined Classes

maxdb

Represents a connection between PHP and a MaxDB database.

Constructor

- maxdb - construct a new maxdb object

Methods

- autocommit - turns on or off auto-commiting database modifications
- change_user - changes the user of the specified database connection
- character_set_name - returns the default character set for the database connection
- close - closes a previously opened connection
- commit - commits the current transaction
- connect - opens a new connection to MaxDB database server
- debug - performs debugging operations
- dump_debug_info - dumps debug information
- get_client_info - returns client version
- get_host_info - returns type of connection used
- get_server_info - returns version of the MaxDB server
- get_server_version - returns version of the MaxDB server
- init - initializes maxdb object
- info - retrieves information about the most recently executed query
- kill - asks the server to kill a MaxDB thread
- multi_query - performs multiple queries
- more_results - check if more results exist from currently executed multi-query
- next_result - reads next result from currently executed multi-query

- options - set options
- ping - pings a server connection or reconnects if there is no connection
- prepare - prepares a SQL query
- query - performs a query
- real_connect - attempts to open a connection to MaxDB database server
- escape_string - escapes special characters in a string for use in a SQL statement, taking into account the current charset of the connection
- rollback - rolls back the current transaction
- select_db - selects the default database
- ssl_set - sets ssl parameters
- stat - gets the current system status
- stmt_init - initializes a statement for use with maxdb_stmt_prepare
- store_result - transfers a resultset from last query
- use_result - transfers an unbuffered resultset from last query
- thread-safe - returns whether thread safety is given or not

Properties

- affected_rows - gets the number of affected rows in a previous MaxDB operation
- client_info - returns the MaxDB client version as a string
- client_version - returns the MaxDB client version as an integer
- errno - returns the error code for the most recent function call
- error - returns the error string for the most recent function call
- field_count - returns the number of columns for the most recent query
- host_info - returns a string representing the type of connection used
- info - retrieves information about the most recently executed query
- insert_id - returns the auto generated id used in the last query
- protocol_version - returns the version of the MaxDB protocol used
- sqlstate - returns a string containing the SQLSTATE error code for the last error
- thread_id - returns the thread ID for the current connection
- warning_count - returns the number of warnings generated during execution of the previous SQL statement

maxdb_stmt

Represents a prepared statement.

Methods

- bind_param - binds variables to a prepared statement
- bind_result - binds variables to a prepared statement for result storage
- close - closes a prepared statement
- data-seek - seeks to an arbitrary row in a statement result set
- execute - executes a prepared statement
- fetch - fetches result from a prepared statement into bound variables
- free_result - frees stored result memory for the given statement handle
- result_metadata - retrieves a resultset from a prepared statement for metadata information
- prepare - prepares a SQL query
- send_long_data - sends data in chunks
- close_long_data - end sending long data
- reset - resets a prepared statement
- store_result - buffers complete resultset from a prepared statement

Properties

- affected_rows - returns affected rows from last statement execution
- errno - returns errorcode for last statement function
- errno - returns errormessage for last statement function
- param_count - returns number of parameter for a given prepare statement
- sqlstate - returns a string containing the SQLSTATE error code for the last statement function

maxdb_result

Represents the result set obtained from a query against the database.

Methods

- close - closes resultset

- data_seek - moves internal result pointer
- fetch_field - gets column information from a resultset
- fetch_fields - gets information for all columns from a resulset
- fetch_field_direct - gets column information for specified column
- fetch_array - fetches a result row as an associative array, a numeric array, or both.
- fetch_assoc - fetches a result row as an associative array
- fetch_object - fetches a result row as an object
- fetch_row - gets a result row as an enumerated array
- close - frees result memory
- field_seek - set result pointer to a specified field offset

Properties

- current_field - returns offset of current fieldpointer
- field_count - returns number of fields in resultset
- lengths - returns an array of columnlengths
- num_rows - returns number of rows in resultset

maxdb_affected_rows

maxdb->affected_rows

maxdb_affected_rows -- maxdb->affected_rows -- Gets the number of affected rows in a previous MaxDB operation

Description

Procedural style:

int **maxdb_affected_rows** (resource `$link`)

Object oriented style (property):

maxdb

int *affected_rows*;

maxdb_affected_rows() returns the number of rows affected by the last INSERT, UPDATE, or DELETE query associated with the provided *`link`* parameter. If this number cannot be determined, this function will return -1.

Note
For SELECT statements maxdb_affected_rows() works like maxdb_num_rows().

The maxdb_affected_rows() function only works with queries which modify a table. In order to return the number of rows from a SELECT query, use the maxdb_num_rows() function instead.

Return Values

An integer greater than zero indicates the number of rows affected or retrieved. Zero indicates that no records where updated for an UPDATE statement, no rows matched the WHERE clause in the query or that no query has yet been executed. -1 indicates that the number of rows affected can not be determined.

Examples

Example #4 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_report (MAXDB_REPORT_OFF);
$maxdb->query("DROP TABLE mycustomer");
maxdb_report (MAXDB_REPORT_ERROR);

/* Insert rows */
$maxdb->query("CREATE TABLE mycustomer AS SELECT * from hotel.customer");
printf("Affected rows (INSERT): %d\n", $maxdb->affected_rows);

$maxdb->query("ALTER TABLE mycustomer ADD Status int default 0");

/* update rows */
$maxdb->query("UPDATE mycustomer SET Status=1 WHERE cno > 50");
printf("Affected rows (UPDATE): %d\n", $maxdb->affected_rows);

/* delete rows */
$maxdb->query("DELETE FROM mycustomer WHERE cno < 50");
printf("Affected rows (DELETE): %d\n", $maxdb->affected_rows);

/* select all rows */
$result = $maxdb->query("SELECT title FROM mycustomer");
printf("Affected rows (SELECT): %d\n", $maxdb->affected_rows);

$result->close();

/* Delete table Language */
$maxdb->query("DROP TABLE mycustomer");

/* close connection */
$maxdb->close();
?>
```

Example #5 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

if (!$link) {
  printf("Can't connect to localhost. Error: %s\n", maxdb_connect_error());
  exit();
}

maxdb_report (MAXDB_REPORT_OFF);
maxdb_query($link,"DROP TABLE mycustomer");
maxdb_report (MAXDB_REPORT_ERROR);
```

```
/* Insert rows */
maxdb_query($link, "CREATE TABLE mycustomer AS SELECT * from
hotel.customer");
printf("Affected rows (INSERT): %d\n", maxdb_affected_rows($link));

maxdb_query($link, "ALTER TABLE mycustomer ADD Status int default 0");

/* update rows */
maxdb_query($link, "UPDATE mycustomer SET Status=1 WHERE cno > 50");
printf("Affected rows (UPDATE): %d\n", maxdb_affected_rows($link));

/* delete rows */
maxdb_query($link, "DELETE FROM mycustomer WHERE cno < 50");
printf("Affected rows (DELETE): %d\n", maxdb_affected_rows($link));

/* select all rows */
$result = maxdb_query($link, "SELECT title FROM mycustomer");
printf("Affected rows (SELECT): %d\n", maxdb_affected_rows($link));

maxdb_free_result($result);

/* Delete table Language */
maxdb_query($link, "DROP TABLE mycustomer");

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Affected rows (INSERT): 15
Affected rows (UPDATE): 15
Affected rows (DELETE): 0
Affected rows (SELECT): 15
```

See Also

- maxdb_num_rows()
- maxdb_info()

maxdb_autocommit

maxdb->auto_commit

maxdb_autocommit -- maxdb->auto_commit -- Turns on or off auto-commiting database modifications

Description

Procedural style:

bool **maxdb_autocommit** (resource `$link`, bool `$mode`)

Object oriented style (method)

maxdb

bool **auto_commit** (bool `$mode`)

maxdb_autocommit() is used to turn on or off auto-commit mode on queries for the database connection represented by the *`link`* resource.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #6 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* turn autocommit on */
$maxdb->autocommit(TRUE);

/* close connection */
$maxdb->close();
?>
```

Example #7 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

if (!$link) {
  printf("Can't connect to localhost. Error: %s\n", maxdb_connect_error());
  exit();
}

/* turn autocommit on */
maxdb_autocommit($link, TRUE);

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

See Also

- maxdb_commit()
- maxdb_rollback()

maxdb_bind_param

maxdb_bind_param -- Alias of maxdb_stmt_bind_param()

Description

This function is an alias of: maxdb_stmt_bind_param()

This function alias is deprecated and only exists for backwards compatibility reasons. The use of this function is not recommended, as it may be removed from PHP in the future.

maxdb_bind_result

maxdb_bind_result -- Alias of maxdb_stmt_bind_result()

Description

This function is an alias of: maxdb_stmt_bind_result().

This function alias is deprecated and only exists for backwards compatibility reasons. The use of this function is not recommended, as it may be removed from PHP in the future.

maxdb_change_user

maxdb->change_user

maxdb_change_user -- maxdb->change_user -- Changes the user of the specified database connection

Description

Procedural style:

bool **maxdb_change_user** (resource `$link`, string `$user`, string `$password`, string `$database`)

Object oriented style (method):

maxdb

bool **change_user** (string `$user`, string `$password`, string `$database`)

maxdb_change_user() is used to change the user of the specified database connection as given by the *`link`* parameter and to set the current database to that specified by the *`database`* parameter.

In order to successfully change users a valid *`username`* and *`password`* parameters must be provided and that user must have sufficient permissions to access the desired database. If for any reason authorization fails, the current user authentication will remain.

Note
Using this command will always cause the current database connection to behave as if was a completely new database connection, regardless of if the operation was completed successfully. This reset includes performing a rollback on any active transactions, closing all temporary tables, and unlocking all locked tables.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #8 - Object oriented style

```
<?php

/* connect database test */
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

if ($result = $maxdb->query("SELECT * FROM dual")) {
  $row = $result->fetch_row();
  printf("Result: %s\n", $row[0]);
  $result->free();
}

/* reset all and select a new database */
if (!$maxdb->change_user("DBADMIN", "SECRET", "DEMODB")) {
 printf("Database not running\n");
} else {
 printf("Database running\n");
}

/* close connection */
$maxdb->close();
?>
```

Example #9 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

if ($result = maxdb_query($link, "SELECT * FROM dual")) {
  $row = maxdb_fetch_row($result);
  printf("Result: %s\n", $row[0]);
  maxdb_free_result($result);
}

/* reset all and select a new database */
if (!maxdb_change_user($link, "DBADMIN", "SECRET", "DEMODB")) {
 printf("Database not running\n");
} else {
 printf("Database running\n");
}

/* close connection */
maxdb_close($link);
```

```
?>
```

The above example will output something similar to:

```
Result: a
Database running
```

See Also

- maxdb_connect()
- maxdb_select_db()

maxdb_character_set_name

maxdb->character_set_name

maxdb_character_set_name -- maxdb->character_set_name -- Returns the default character set for the database connection

Description

Procedural style:

string **maxdb_character_set_name** (resource `$link`)

Object oriented style (method):

maxdb

string **character_set_name** (void)

Returns the current character set for the database connection specified by the *`link`* parameter.

Return Values

The default character set for the current connection, either ascii or unicode.

Examples

Example #10 - Object oriented style

```
<?php
/* Open a connection */
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* Print current character set */
$charset = $maxdb->character_set_name();
printf ("Current character set is %s\n", $charset);

$maxdb->close();
```

```
?>
```

Example #11 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* Print current character set */
$charset = maxdb_character_set_name($link);
printf ("Current character set is %s\n",$charset);

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Current character set is ascii
```

See Also

- maxdb_client_encoding()
- maxdb_real_escape_string()

maxdb_client_encoding

maxdb_client_encoding -- Alias of maxdb_character_set_name()

Description

This function is an alias of: maxdb_character_set_name().

This function alias is deprecated and only exists for backwards compatibility reasons. The use of this function is not recommended, as it may be removed from PHP in the future.

maxdb_close_long_data

maxdb->close_long_data

maxdb_close_long_data -- maxdb->close_long_data -- Alias of maxdb_stmt_close_long_data()

Description

This function is an alias of: maxdb_stmt_close_long_data().

This function alias is deprecated and only exists for backwards compatibility reasons. The use of this function is not recommended, as it may be removed from PHP in the future.

maxdb_close

maxdb->close

maxdb_close -- maxdb->close -- Closes a previously opened database connection

Description

Procedural style:

bool **maxdb_close** (resource `$link`)

Object oriented style (method):

maxdb

bool **close** (void)

The maxdb_close() function closes a previously opened database connection specified by the *`link`* parameter.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- maxdb_connect()
- maxdb_init()
- maxdb_real_connect()

maxdb_commit

maxdb->commit

maxdb_commit -- maxdb->commit -- Commits the current transaction

Description

Procedural style:

bool **maxdb_commit** (resource `$link`)

Object oriented style (method)

maxdb

bool **commit** (void)

Commits the current transaction for the database connection specified by the *`link`* parameter.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #12 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* set autocommit to off */
$maxdb->autocommit(FALSE);

maxdb_report (MAXDB_REPORT_OFF);
$maxdb->query("DROP TABLE mycustomer");
maxdb_report (MAXDB_REPORT_ERROR);
```

```
$maxdb->query("CREATE TABLE mycustomer LIKE hotel.customer");

/* Insert some values */
$maxdb->query("INSERT INTO mycustomer VALUES
(3000,'Mrs','Jenny','Porter','10580','1340 N.Ash Street, #3')");
$maxdb->query("INSERT INTO mycustomer VALUES
(3100,'Mr','Peter','Brown','48226','1001 34th Str., APT.3')");

/* commit transaction */
$maxdb->commit();

/* close connection */
$maxdb->close();
?>
```

Example #13 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* set autocommit to off */
maxdb_autocommit($link, FALSE);

maxdb_report (MAXDB_REPORT_OFF);
maxdb_query($link,"DROP TABLE mycustomer");
maxdb_report (MAXDB_REPORT_ERROR);

maxdb_query($link, "CREATE TABLE mycustomer LIKE hotel.customer");

/* Insert some values */
maxdb_query($link, "INSERT INTO mycustomer VALUES
(3000,'Mrs','Jenny','Porter','10580','1340 N.Ash Street, #3')");
maxdb_query($link, "INSERT INTO mycustomer VALUES
(3100,'Mr','Peter','Brown','48226','1001 34th Str., APT.3')");

/* commit transaction */
maxdb_commit($link);

/* close connection */
maxdb_close($link);
?>
```

The above examples produces no output.

See Also

- maxdb_autocommit()

- maxdb_rollback()

maxdb_connect_errno

maxdb_connect_errno -- Returns the error code from last connect call

Description

int **maxdb_connect_errno** (void)

The maxdb_connect_errno() function will return the last error code number for last call to maxdb_connect(). If no errors have occured, this function will return zero.

Return Values

An error code value for the last call to maxdb_connect(), if it failed. zero means no error occurred.

Examples

Example #14 - maxdb_connect_errno sample

```
<?php
$link = maxdb_connect("localhost", "XXXXXXXX", "YYYYYYYYY");

if (!$link) {
  printf("Can't connect to localhost. Errorcode: %d\n",
maxdb_connect_errno());
}
?>
```

The above example will output something similar to:

```
PHP Warning:  maxdb_connect(): -4008 POS(1) Unknown user name/password
combination [08004] <...>
Can't connect to localhost. Errorcode: -4008
```

See Also

- maxdb_connect()
- maxdb_connect_error()
- maxdb_errno()
- maxdb_error()
- maxdb_sqlstate()

maxdb_connect_error

maxdb_connect_error -- Returns a string description of the last connect error

Description

string **maxdb_connect_error** (void)

The maxdb_connect_error() function is identical to the corresponding maxdb_connect_errno() function in every way, except instead of returning an integer error code the maxdb_connect_error() function will return a string representation of the last error to occur for the last maxdb_connect() call. If no error has occured, this function will return an empty string.

Return Values

A string that describes the error. An empty string if no error occurred.

Examples

Example #15 - maxdb_connect_error sample

```
<?php

$link = maxdb_connect("localhost", "nonexisting_user", "");

if (!$link) {
  printf("Can't connect to localhost. Error: %s\n", maxdb_connect_error());
}
?>
```

The above example will output something similar to:

```
PHP Warning:  maxdb_connect(): -4008 POS(1) Unknown user name/password
combination <...>
Can't connect to localhost. Error: POS(1) Unknown user name/password combination
```

See Also

- maxdb_connect()
- maxdb_connect_errno()
- maxdb_errno()
- maxdb_error()
- maxdb_sqlstate()

maxdb_connect

maxdb()

maxdb_connect -- maxdb() -- Open a new connection to the MaxDB server

Description

Procedural style

resource **maxdb_connect** ([string `$host` [, string `$username` [, string `$passwd` [, string `$dbname` [, int `$port` [, string `$socket`]]]]]])

Object oriented style (constructor):

maxdb

__construct ([string `$host` [, string `$username` [, string `$passwd` [, string `$dbname` [, int `$port` [, string `$socket`]]]]]])

The maxdb_connect() function attempts to open a connection to the MaxDB Server running on *`host`* which can be either a host name or an IP address. Passing the string "localhost" to this parameter, the local host is assumed. If successful, the maxdb_connect() will return an resource representing the connection to the database, or **FALSE** on failure.

The *`username`* and *`password`* parameters specify the username and password under which to connect to the MaxDB server. If the password is not provided (the **NULL** value is passed), the MaxDB server will attempt to authenticate the user against the *`maxdb.default_pw`* in *php.ini*.

The *`dbname`* parameter if provided will specify the default database to be used when performing queries. If not provied, the entry *`maxdb.default_db`* in *php.ini* is used.

The *`port`* and *`socket`* parameters are ignored for the MaxDB server.

Return Values

Returns a resource which represents the connection to a MaxDB Server or **FALSE** if the connection failed.

Examples

Example #16 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

printf("Host information: %s\n", $maxdb->host_info);

/* close connection */
$maxdb->close();
?>
```

Example #17 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

printf("Host information: %s\n", maxdb_get_host_info($link));

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Host information: localhost
```

maxdb_data_seek

result->data_seek

maxdb_data_seek -- result->data_seek -- Adjusts the result pointer to an arbitary row in the result

Description

Procedural style:

bool **maxdb_data_seek** (resource `$result`, int `$offset`)

Object oriented style (method):

result

bool **data_seek** (int `$offset`)

The maxdb_data_seek() function seeks to an arbitrary result pointer specified by the *`offset`* in the result set represented by *`result`*. The *`offset`* parameter must be between zero and the total number of rows minus one (0.. maxdb_num_rows() - 1).

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #18 - Object oriented style

```
<?php
/* Open a connection */
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, state FROM hotel.city ORDER BY name";
if ($result = $maxdb->query( $query)) {

  /* seek to row no. 10 */
```

```
  $result->data_seek(10);

  /* fetch row */
  $row = $result->fetch_row();

  printf ("City: %s  State: %s\n", $row[0], $row[1]);

  /* free result set*/
  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #19 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, state FROM hotel.city ORDER BY name";

if ($result = maxdb_query($link, $query)) {

  /* seek to row no. 400 */
  maxdb_data_seek($result, 10);

  /* fetch row */
  $row = maxdb_fetch_row($result);

  printf ("City: %s  State: %s\n", $row[0], $row[1]);

  /* free result set*/
  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
City: Irvine  State: CA
```

See Also

- maxdb_store_result()

- maxdb_fetch_row()
- maxdb_num_rows()

maxdb_debug

maxdb_debug -- Performs debugging operations

Description

void **maxdb_debug** (string `$debug`)

The maxdb_debug() can be used to trace the SQLDBC communication. The following strings can be used as a parameter to maxdb_debug():

- TRACE SHORT ON|OFF - Enables/disables method call trace.
- TRACE LONG ON|OFF - Enables/disables method argument and detail debug trace.
- TRACE PACKET ON|OFF|<size> - Enables/disables packet trace, limiting the size of the traced object to the specified number of bytes, or 1000 if no size is specified.
- TRACE SQL ON|OFF - Enables/disables high level api trace.
- TRACE TIMESTAMP ON|OFF - Enables/disables a timestamp prefix on each line that is traced.
- TRACE SIZE <size> - Limits the size of the trace file to <size> bytes, at least 8192 bytes are required.

Return Values

maxdb_debug() doesn't return any value.

Examples

Example #20 - Procedural style

```
<?php

maxdb_debug("trace packet 200");

$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* close connection */
maxdb_close($link);
?>
```

The above example produces no output.

maxdb_disable_reads_from_master

maxdb->disable_reads_from_master

maxdb_disable_reads_from_master -- maxdb->disable_reads_from_master -- Disable reads from master

Description

Procedural style:

bool **maxdb_disable_reads_from_master** (resource `$link`)

Object oriented style (method):

maxdb

void **disable_reads_from_master** (void)

Warning
This function is currently not documented; only its argument list is available.

maxdb_disable_rpl_parse

maxdb_disable_rpl_parse -- Disable RPL parse

Description

bool **maxdb_disable_rpl_parse** (resource `$link`)

Warning
This function is currently not documented; only its argument list is available.

maxdb_dump_debug_info

maxdb_dump_debug_info -- Dump debugging information into the log

Description

bool **maxdb_dump_debug_info** (resource `$link`)

Warning
This function is currently not documented; only its argument list is available.

maxdb_embedded_connect

maxdb_embedded_connect -- Open a connection to an embedded MaxDB server

Description

resource **maxdb_embedded_connect** ([string `$dbname`])

Warning
This function is currently not documented; only its argument list is available.

maxdb_enable_reads_from_master

maxdb_enable_reads_from_master -- Enable reads from master

Description

bool **maxdb_enable_reads_from_master** (resource `$link`)

Warning
This function is currently not documented; only its argument list is available.

maxdb_enable_rpl_parse

maxdb_enable_rpl_parse -- Enable RPL parse

Description

bool **maxdb_enable_rpl_parse** (resource `$link`)

Warning
This function is currently not documented; only its argument list is available.

maxdb_errno

maxdb->errno

maxdb_errno -- maxdb->errno -- Returns the error code for the most recent function call

Description

Procedural style:

int **maxdb_errno** (resource `$link`)

Object oriented style (property):

maxdb

int *errno*;

The maxdb_errno() function will return the last error code for the most recent MaxDB function call that can succeed or fail with respect to the database link defined by the *`link`* parameter. If no errors have occured, this function will return zero.

Return Values

An error code value for the last call, if it failed. zero means no error occurred.

Examples

Example #21 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

if (!$maxdb->query("SELECT xxx FROM hotel.city")) {
  printf("Errorcode: %d\n", $maxdb->errno);
}

/* close connection */
$maxdb->close();
```

```
?>
```

Example #22 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

if (!maxdb_query($link, "SELECT xxx FROM hotel.city")) {
  printf("Errorcode: %d\n", maxdb_errno($link));
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
PHP Warning:  maxdb_query(): -4005 POS(8) Unknown column name:XXX [42000] <...>
Errorcode: -4005
```

See Also

- maxdb_connect_errno()
- maxdb_connect_error()
- maxdb_error()
- maxdb_sqlstate()

maxdb_error

maxdb_error -- Returns a string description of the last error

Description

Procedural style:

string **maxdb_error** (resource `$link`)

Object oriented style (property)

maxdb

string *error*;

The maxdb_error() function is identical to the corresponding maxdb_errno() function in every way, except instead of returning an integer error code the maxdb_error() function will return a string representation of the last error to occur for the database connection represented by the *link* parameter. If no error has occured, this function will return an empty string.

Return Values

A string that describes the error. An empty string if no error occurred.

Examples

Example #23 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

if (!$maxdb->query("SELECT xxx FROM hotel.city")) {
  printf("Errormessage: %s\n", $maxdb->error);
}

/* close connection */
$maxdb->close();
?>
```

Example #24 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

if (!maxdb_query($link, "SELECT xxx FROM hotel.city")) {
  printf("Errormessgae: %s\n", maxdb_error($link));
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
PHP Warning:  maxdb_query(): -4005 POS(8) Unknown column name:XXX [42000]
Errormessgae: POS(8) Unknown column name:XXX
```

See Also

- maxdb_connect_errno()
- maxdb_connect_error()
- maxdb_errno()
- maxdb_sqlstate()

maxdb_escape_string

maxdb_escape_string -- Alias of maxdb_real_escape_string()

Description

This function is an alias of: maxdb_real_escape_string().

maxdb_execute

maxdb_execute -- Alias of maxdb_stmt_execute()

Description

This function is an alias of: maxdb_stmt_execute().

This function alias is deprecated and only exists for backwards compatibility reasons. The use of this function is not recommended, as it may be removed from PHP in the future.

maxdb_fetch_array

result->fetch_array

maxdb_fetch_array -- result->fetch_array -- Fetch a result row as an associative, a numeric array, or both

Description

Procedural style:

mixed **maxdb_fetch_array** (resource `$result` [, int `$resulttype`])

Object oriented style (method):

result

mixed **fetch_array** ([int `$resulttype`])

Returns an array that corresponds to the fetched row or **NULL** if there are no more rows for the resultset represented by the *`result`* parameter.

maxdb_fetch_array() is an extended version of the maxdb_fetch_row() function. In addition to storing the data in the numeric indices of the result array, the maxdb_fetch_array() function can also store the data in associative indices, using the field names of the result set as keys.

Note
Field names returned by this function are *case-sensitive*.

Note
This function sets NULL fields to the PHP **NULL** value.

If two or more columns of the result have the same field names, the last column will take precedence and overwrite the earlier data. In order to access multiple columns with the same name, the numerically indexed version of the row must be used.

The optional second argument *`resulttype`* is a constant indicating what type of array

should be produced from the current row data. The possible values for this parameter are the constants MAXDB_ASSOC, MAXDB_ASSOC_UPPER, MAXDB_ASSOC_LOWER, MAXDB_NUM, or MAXDB_BOTH. By default the maxdb_fetch_array() function will assume MAXDB_BOTH, which is a combination of MAXDB_NUM and MAXDB_ASSOC for this parameter.

By using the MAXDB_ASSOC constant this function will behave identically to the maxdb_fetch_assoc(), while MAXDB_NUM will behave identically to the maxdb_fetch_row() function. The final option MAXDB_BOTH will create a single array with the attributes of both.

By using the MAXDB_ASSOC_UPPER constant, the behaviour of this function is identical to the use of MAXDB_ASSOC except the array index of a column is the fieldname in upper case.

By using the MAXDB_ASSOC_LOWER constant, the behaviour of this function is identical to the use of MAXDB_ASSOC except the array index of a column is the fieldname in lower case.

Return Values

Returns an array that corresponds to the fetched row or **NULL** if there are no more rows in resultset.

Examples

Example #25 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, state FROM hotel.city ORDER by zip";
$result = $maxdb->query($query);

/* numeric array */
$row = $result->fetch_array(MAXDB_NUM);
printf ("%s (%s)\n", $row[0], $row[1]);

/* associative array */
$row = $result->fetch_array(MAXDB_ASSOC);
printf ("%s (%s)\n", $row["NAME"], $row["STATE"]);

/* associative and numeric array */
$row = $result->fetch_array(MAXDB_BOTH);
printf ("%s (%s)\n", $row[0], $row["STATE"]);

/* free result set */
$result->close();
```

```
/* close connection */
$maxdb->close();
?>
```

Example #26 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, state FROM hotel.city ORDER by zip";
$result = maxdb_query($link, $query);

/* numeric array */
$row = maxdb_fetch_array($result, MAXDB_NUM);
printf ("%s (%s)\n", $row[0], $row[1]);

/* associative array */
$row = maxdb_fetch_array($result, MAXDB_ASSOC);
printf ("%s (%s)\n", $row["NAME"], $row["STATE"]);

/* associative and numeric array */
$row = maxdb_fetch_array($result, MAXDB_BOTH);
printf ("%s (%s)\n", $row[0], $row["STATE"]);

/* free result set */
maxdb_free_result($result);

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
New York (NY)
New York (NY)
Long Island (NY)
```

See Also

- maxdb_fetch_assoc()
- maxdb_fetch_row()
- **maxdb_fetch_resource()**

maxdb_fetch_assoc

maxdb->fetch_assoc

maxdb_fetch_assoc -- maxdb->fetch_assoc -- Fetch a result row as an associative array

Description

Procedural style:

array **maxdb_fetch_assoc** (resource `$result`)

Object oriented style (method):

result

array **fetch_assoc** (void)

Returns an associative array that corresponds to the fetched row or **NULL** if there are no more rows.

The maxdb_fetch_assoc() function is used to return an associative array representing the next row in the result set for the result represented by the *`result`* parameter, where each key in the array represents the name of one of the result set's columns.

If two or more columns of the result have the same field names, the last column will take precedence. To access the other column(s) of the same name, you either need to access the result with numeric indices by using maxdb_fetch_row() or add alias names.

Note
Field names returned by this function are *case-sensitive*.

Note
This function sets NULL fields to the PHP **NULL** value.

Return Values

Returns an array that corresponds to the fetched row or **NULL** if there are no more rows in

resultset.

Examples

Example #27 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, state FROM hotel.city ORDER by zip";

if ($result = $maxdb->query($query)) {

  /* fetch associative array */
  while ($row = $result->fetch_assoc()) {
      printf ("%s (%s)\n", $row["NAME"], $row["STATE"]);
  }

  /* free result set */
  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #28 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, state FROM hotel.city ORDER by zip";

if ($result = maxdb_query($link, $query)) {

  /* fetch associative array */
  while ($row = maxdb_fetch_assoc($result)) {
      printf ("%s (%s)\n", $row["NAME"], $row["STATE"]);
  }

  /* free result set */
  maxdb_free_result($result);
}
```

```
/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
New York (NY)
New York (NY)
Long Island (NY)
Albany (NY)
Washington (DC)
Washington (DC)
Washington (DC)
Silver Spring (MD)
Daytona Beach (FL)
Deerfield Beach (FL)
Clearwater (FL)
Cincinnati (OH)
Detroit (MI)
Rosemont (IL)
Chicago (IL)
Chicago (IL)
New Orleans (LA)
Dallas (TX)
Los Angeles (CA)
Hollywood (CA)
Long Beach (CA)
Palm Springs (CA)
Irvine (CA)
Santa Clara (CA)
Portland (OR)
```

See Also

- maxdb_fetch_array()
- maxdb_fetch_row()
- **maxdb_fetch_resource()**

maxdb_fetch_field_direct

result->fetch_field_direct

maxdb_fetch_field_direct -- result->fetch_field_direct -- Fetch meta-data for a single field

Description

Procedural style:

mixed **maxdb_fetch_field_direct** (resource `$result`, int `$fieldnr`)

Object oriented style (method):

result

mixed **fetch_field_direct** (int `$fieldnr`)

maxdb_fetch_field_direct() returns an resource which contains field definition informations from specified resultset. The value of fieldnr must be in the range from *0* to *number of fields - 1*.

Return Values

Returns an resource which contains field definition informations or **FALSE** if no field information for specified *fieldnr* is available.

Object attributes

Attribute	Description
name	The name of the column
max_length	The maximum width of the field for the result set.
type	The data type used for this field
decimals	The number of decimals used (for integer fields)

Examples

Example #29 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, cno from hotel.customer ORDER BY name";

if ($result = $maxdb->query($query)) {

  /* Get field information for column 'SurfaceArea' */
  $finfo = $result->fetch_field_direct(1);

  printf("Name:     %s\n", $finfo->name);
  printf("Table:    %s\n", $finfo->table);
  printf("max. Len: %d\n", $finfo->max_length);
  printf("Flags:    %d\n", $finfo->flags);
  printf("Type:     %d\n", $finfo->type);

  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #30 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, cno from hotel.customer ORDER BY name";

if ($result = maxdb_query($link, $query)) {

  /* Get field information for column 'cno' */
  $finfo = maxdb_fetch_field_direct($result, 1);

  printf("Name:     %s\n", $finfo->name);
  printf("Table:    %s\n", $finfo->table);
  printf("max. Len: %d\n", $finfo->max_length);
  printf("Flags:    %d\n", $finfo->flags);
  printf("Type:     %d\n", $finfo->type);
```

```
  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Name:     CNO
Table:
max. Len: 4
Flags:    -1
Type:     0
```

See Also

- maxdb_num_fields()
- maxdb_fetch_field()
- maxdb_fetch_fields()

maxdb_fetch_field

result->fetch_field

maxdb_fetch_field -- result->fetch_field -- Returns the next field in the result set

Description

Procedural style:

mixed **maxdb_fetch_field** (resource `$result`)

Object oriented style (method):

result

mixed **fetch_field** (void)

The maxdb_fetch_field() returns the definition of one column of a result set as an resource. Call this function repeatedly to retrieve information about all columns in the result set. maxdb_fetch_field() returns **FALSE** when no more fields are left.

Return Values

Returns an resource which contains field definition informations or **FALSE** if no field information is available.

Object properties

Property	**Description**
name	The name of the column
max_length	The maximum width of the field for the result set.
type	The data type used for this field
decimals	The number of decimals used (for integer fields)

Examples

Example #31 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, cno from hotel.customer ORDER BY cno";

if ($result = $maxdb->query($query)) {

  /* Get field information for all columns */
  while ($finfo = $result->fetch_field()) {

      printf("Name:     %s\n", $finfo->name);
      printf("Table:    %s\n", $finfo->table);
      printf("max. Len: %d\n", $finfo->max_length);
      printf("Flags:    %d\n", $finfo->flags);
      printf("Type:     %d\n\n", $finfo->type);
  }
  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #32 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, cno from hotel.customer ORDER BY cno";

if ($result = maxdb_query($link, $query)) {

  /* Get field information for all fields */
  while ($finfo = maxdb_fetch_field($result)) {

      printf("Name:     %s\n", $finfo->name);
      printf("Table:    %s\n", $finfo->table);
      printf("max. Len: %d\n", $finfo->max_length);
      printf("Flags:    %d\n", $finfo->flags);
      printf("Type:     %d\n\n", $finfo->type);
```

```
  }
  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Name:     NAME
Table:
max. Len: 10
Flags:    -1
Type:     2

Name:     CNO
Table:
max. Len: 4
Flags:    -1
Type:     0
```

See Also

- maxdb_num_fields()
- maxdb_fetch_field_direct()
- maxdb_fetch_fields()
- maxdb_field_seek()

maxdb_fetch_fields

result->fetch_fields

maxdb_fetch_fields -- result->fetch_fields -- Returns an array of resources representing the fields in a result set

Description

Procedural Style:

mixed **maxdb_fetch_fields** (resource `$result`)

Object oriented style (method):

result

mixed **fetch_fields** (void)

This function serves an identical purpose to the maxdb_fetch_field() function with the single difference that, instead of returning one resource at a time for each field, the columns are returned as an array of resources.

Return Values

Returns an array of resources which contains field definition informations or **FALSE** if no field information is available.

Object properties

Property	Description
name	The name of the column
max_length	The maximum width of the field for the result set.
type	The data type used for this field
decimals	The number of decimals used (for integer fields)

Examples

Example #33 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, cno from hotel.customer ORDER BY cno";

if ($result = $maxdb->query($query)) {

  /* Get field information for all columns */
  $finfo = $result->fetch_fields();

  foreach ($finfo as $val) {
      printf("Name:     %s\n", $val->name);
      printf("Table:    %s\n", $val->table);
      printf("max. Len: %d\n", $val->max_length);
      printf("Flags:    %d\n", $val->flags);
      printf("Type:     %d\n\n", $val->type);
  }
  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #34 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, cno from hotel.customer ORDER BY cno";

if ($result = maxdb_query($link, $query)) {

  /* Get field information for all columns */
  $finfo = maxdb_fetch_fields($result);

  foreach ($finfo as $val) {
      printf("Name:     %s\n", $val->name);
      printf("Table:    %s\n", $val->table);
      printf("max. Len: %d\n", $val->max_length);
```

```
      printf("Flags:    %d\n", $val->flags);
      printf("Type:     %d\n\n", $val->type);
  }
  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Name:     NAME
Table:
max. Len: 10
Flags:    -1
Type:     2

Name:     CNO
Table:
max. Len: 4
Flags:    -1
Type:     0
```

See Also

- maxdb_num_fields()
- maxdb_fetch_field()
- maxdb_fetch_field_direct()

maxdb_fetch_lengths

result->lengths

maxdb_fetch_lengths -- result->lengths -- Returns the lengths of the columns of the current row in the result set

Description

Procedural style:

array **maxdb_fetch_lengths** (resource `$result`)

Object oriented style (property):

result

array *lengths*;

The maxdb_fetch_lengths() function returns an array containing the lengths of every column of the current row within the result set represented by the `result` parameter. If successful, a numerically indexed array representing the lengths of each column is returned or **FALSE** on failure.

Return Values

An array of integers representing the size of each column (not including any terminating null characters). **FALSE** if an error occurred.

maxdb_fetch_lengths() is valid only for the current row of the result set. It returns **FALSE** if you call it before calling maxdb_fetch_row/array/resource or after retrieving all rows in the result.

Examples

Example #35 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
```

```
}

$query = "SELECT * from hotel.customer WHERE cno = 3000";

if ($result = $maxdb->query($query)) {

  $row = $result->fetch_row();

  /* display column lengths */
  foreach ($result->lengths as $i => $val) {
      printf("Field %2d has Length %2d\n", $i+1, $val);
  }
  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #36 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT * from hotel.customer WHERE cno = 3000";

if ($result = maxdb_query($link, $query)) {

  $row = maxdb_fetch_row($result);

  /* display column lengths */
  foreach (maxdb_fetch_lengths($result) as $i => $val) {
      printf("Field %2d has Length %2d\n", $i+1, $val);
  }
  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Field  1 has Length  4
Field  2 has Length  3
Field  3 has Length  5
Field  4 has Length  6
Field  5 has Length  5
Field  6 has Length 21
```

maxdb_fetch_object

result->fetch_object

maxdb_fetch_object -- result->fetch_object -- Returns the current row of a result set as an object

Description

Procedural style:

object **maxdb_fetch_object** (object `$result`)

Object oriented style (method):

result

object **fetch_object** (void)

The maxdb_fetch_object() will return the current row result set as an object where the attributes of the object represent the names of the fields found within the result set. If no more rows exist in the current result set, **NULL** is returned.

Return Values

Returns an object that corresponds to the fetched row or **NULL** if there are no more rows in resultset.

Note
Field names returned by this function are *case-sensitive*.

Note
This function sets NULL fields to the PHP **NULL** value.

Examples

Example #37 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, state FROM hotel.city ORDER by zip";

if ($result = $maxdb->query($query)) {

  /* fetch object array */
  while ($obj = $result->fetch_object()) {
      printf ("%s (%s)\n", $obj->NAME, $obj->STATE);
  }

  /* free result set */
  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #38 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, state FROM hotel.city ORDER by zip";

if ($result = maxdb_query($link, $query)) {

  /* fetch object array */
  while ($obj = maxdb_fetch_object($result)) {
      printf ("%s (%s)\n", $obj->NAME, $obj->STATE);
  }

  /* free result set */
  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
New York (NY)
New York (NY)
Long Island (NY)
Albany (NY)
Washington (DC)
Washington (DC)
Washington (DC)
Silver Spring (MD)
Daytona Beach (FL)
Deerfield Beach (FL)
Clearwater (FL)
Cincinnati (OH)
Detroit (MI)
Rosemont (IL)
Chicago (IL)
Chicago (IL)
New Orleans (LA)
Dallas (TX)
Los Angeles (CA)
Hollywood (CA)
Long Beach (CA)
Palm Springs (CA)
Irvine (CA)
Santa Clara (CA)
Portland (OR)
```

See Also

- maxdb_fetch_array()
- maxdb_fetch_assoc()
- maxdb_fetch_row()

maxdb_fetch_row

result->fetch_row

maxdb_fetch_row -- result->fetch_row -- Get a result row as an enumerated array

Description

Procedural style:

mixed **maxdb_fetch_row** (resource `$result`)

Object oriented style (method):

result

mixed **fetch_row** (void)

Returns an array that corresponds to the fetched row, or **NULL** if there are no more rows.

maxdb_fetch_row() fetches one row of data from the result set represented by *`result`* and returns it as an enumerated array, where each column is stored in an array offset starting from 0 (zero). Each subsequent call to the maxdb_fetch_row() function will return the next row within the result set, or **FALSE** if there are no more rows.

Return Values

maxdb_fetch_row() returns an array that corresponds to the fetched row or **NULL** if there are no more rows in result set.

Note
This function sets NULL fields to the PHP **NULL** value.

Examples

Example #39 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");
```

```
/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, state FROM hotel.city ORDER by zip";

if ($result = $maxdb->query($query)) {

  /* fetch enumerated array */
  while ($row = $result->fetch_row()) {
      printf ("%s (%s)\n", $row[0], $row[1]);
  }

  /* free result set */
  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #40 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, state FROM hotel.city ORDER by zip";

if ($result = maxdb_query($link, $query)) {

  /* fetch enumerated array */
  while ($row = maxdb_fetch_row($result)) {
      printf ("%s (%s)\n", $row[0], $row[1]);
  }

  /* free result set */
  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
New York (NY)
New York (NY)
Long Island (NY)
```

```
Albany (NY)
Washington (DC)
Washington (DC)
Washington (DC)
Silver Spring (MD)
Daytona Beach (FL)
Deerfield Beach (FL)
Clearwater (FL)
Cincinnati (OH)
Detroit (MI)
Rosemont (IL)
Chicago (IL)
Chicago (IL)
New Orleans (LA)
Dallas (TX)
Los Angeles (CA)
Hollywood (CA)
Long Beach (CA)
Palm Springs (CA)
Irvine (CA)
Santa Clara (CA)
Portland (OR)
```

See Also

- maxdb_fetch_array()
- maxdb_fetch_assoc()
- **maxdb_fetch_resource()**

maxdb_fetch

maxdb_fetch -- Alias of maxdb_stmt_fetch()

Description

This function is an alias of: maxdb_stmt_fetch().

This function alias is deprecated and only exists for backwards compatibility reasons. The use of this function is not recommended, as it may be removed from PHP in the future.

maxdb_field_count

maxdb->field_count

maxdb_field_count -- maxdb->field_count -- Returns the number of columns for the most recent query

Description

Procedural style:

int **maxdb_field_count** (resource `$link`)

Object oriented style (method):

maxdb

int **field_count** (void)

Returns the number of columns for the most recent query on the connection represented by the *`link`* parameter. This function can be useful when using the maxdb_store_result() function to determine if the query should have produced a non-empty result set or not without knowing the nature of the query.

Return Values

An integer representing the number of fields in a result set.

Examples

Example #41 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

maxdb_report (MAXDB_REPORT_OFF);
$maxdb->query("DROP TABLE friends");
maxdb_report (MAXDB_REPORT_ERROR);

$maxdb->query( "CREATE TABLE friends (id int, name varchar(20))");

$maxdb->query( "INSERT INTO friends VALUES (1,'Hartmut')");
$maxdb->query( "INSERT INTO friends VALUES (2, 'Ulf')");
```

```
if ($maxdb->field_count()) {
  /* this was a select/show or describe query */
  $result = $maxdb->store_result();

  /* process resultset */
  $row = $result->fetch_row();

  /* free resultset */
  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #42 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

maxdb_report (MAXDB_REPORT_OFF);
maxdb_query($link,"DROP TABLE friends");
maxdb_report (MAXDB_REPORT_ERROR);

maxdb_query($link, "CREATE TABLE friends (id int, name varchar(20))");

maxdb_query($link, "INSERT INTO friends VALUES (1,'Hartmut')");
maxdb_query($link, "INSERT INTO friends VALUES (2, 'Ulf')");

if (maxdb_field_count($link)) {
  /* this was a select/show or describe query */
  $result = maxdb_store_result($link);

  /* process resultset */
  $row = maxdb_fetch_row($result);

  /* free resultset */
  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example produces no output.

maxdb_field_seek

result->field_seek

maxdb_field_seek -- result->field_seek -- Set result pointer to a specified field offset

Description

Procedural style:

bool **maxdb_field_seek** (resource `$result`, int `$fieldnr`)

Object oriented style (method):

result

bool **field_seek** (int `$fieldnr`)

Sets the field cursor to the given offset. The next call to maxdb_fetch_field() will retrieve the field definition of the column associated with that offset.

Note
To seek to the beginning of a row, pass an offset value of zero.

Return Values

maxdb_field_seek() returns previuos value of field cursor.

Examples

Example #43 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}
```

```
$query = "SELECT name, cno from hotel.customer ORDER BY cno";

if ($result = $maxdb->query($query)) {

  /* Get field information for 2nd column */
  $result->field_seek(1);
  $finfo = $result->fetch_field();

  printf("Name:     %s\n", $finfo->name);
  printf("Table:    %s\n", $finfo->table);
  printf("max. Len: %d\n", $finfo->max_length);
  printf("Flags:    %d\n", $finfo->flags);
  printf("Type:     %d\n\n", $finfo->type);

  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #44 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, cno from hotel.customer ORDER BY cno";

if ($result = maxdb_query($link, $query)) {

  /* Get field information for 2nd column */
  maxdb_field_seek($result, 1);
  $finfo = maxdb_fetch_field($result);

  printf("Name:     %s\n", $finfo->name);
  printf("Table:    %s\n", $finfo->table);
  printf("max. Len: %d\n", $finfo->max_length);
  printf("Flags:    %d\n", $finfo->flags);
  printf("Type:     %d\n\n", $finfo->type);

  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Name:     NAME
```

```
Table:
max. Len: 10
Flags:    -1
Type:     2
```

See Also

- maxdb_fetch_field()

maxdb_field_tell

result->current_field

maxdb_field_tell -- result->current_field -- Get current field offset of a result pointer

Description

Procedural style:

int **maxdb_field_tell** (resource `$result`)

Object oriented style (property):

result

int *current_field*;

Returns the position of the field cursor used for the last maxdb_fetch_field() call. This value can be used as an argument to maxdb_field_seek().

Return Values

Returns current offset of field cursor.

Examples

Example #45 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, cno from hotel.customer ORDER BY cno";

if ($result = $maxdb->query($query)) {

  /* Get field information for all columns */
  while ($finfo = $result->fetch_field()) {
```

```
      /* get fieldpointer offset */
      $currentfield = $result->current_field;

      printf("Column    %d:\n", $currentfield);
      printf("Name:     %s\n", $finfo->name);
      printf("Table:    %s\n", $finfo->table);
      printf("max. Len: %d\n", $finfo->max_length);
      printf("Flags:    %d\n", $finfo->flags);
      printf("Type:     %d\n\n", $finfo->type);
  }
  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #46 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, cno from hotel.customer ORDER BY cno";

if ($result = maxdb_query($link, $query)) {

  /* Get field information for all fields */
  while ($finfo = maxdb_fetch_field($result)) {

      /* get fieldpointer offset */
      $currentfield = maxdb_field_tell($result);

      printf("Column    %d:\n", $currentfield);
      printf("Name:     %s\n", $finfo->name);
      printf("Table:    %s\n", $finfo->table);
      printf("max. Len: %d\n", $finfo->max_length);
      printf("Flags:    %d\n", $finfo->flags);
      printf("Type:     %d\n\n", $finfo->type);
  }
  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Column    1:
Name:     NAME
```

```
Table:
max. Len: 10
Flags:    -1
Type:     2

Column    2:
Name:     CNO
Table:
max. Len: 4
Flags:    -1
Type:     0
```

See Also

- maxdb_fetch_field()
- maxdb_field_seek()

maxdb_free_result

result->free

maxdb_free_result -- result->free -- Frees the memory associated with a result

Description

Procedural style:

void **maxdb_free_result** (resource `$result`)

Object oriented style (method):

result

void **free** (void)

The maxdb_free_result() function frees the memory associated with the result represented by the *`result`* parameter, which was allocated by maxdb_query(), maxdb_store_result() or maxdb_use_result().

Note
You should always free your result with maxdb_free_result(), when your result resource is not needed anymore.

Return Values

This function doesn't return any value.

See Also

- maxdb_query()
- maxdb_stmt_store_result()
- maxdb_store_result()
- maxdb_use_result()

maxdb_get_client_info

maxdb_get_client_info -- Returns the MaxDB client version as a string

Description

string **maxdb_get_client_info** (void)

The maxdb_get_client_info() function is used to return a string representing the client version being used in the MaxDB extension.

Return Values

A string that represents the MaxDB client library version

Examples

Example #47 - maxdb_get_client_info

```
<?php

/* We don't need a connection to determine
  the version of MaxDB client library */

printf("Client library version: %s\n", maxdb_get_client_info());
?>
```

The above example will output something similar to:

```
Client library version: libSQLDBC <...>
```

See Also

- maxdb_get_client_version()
- maxdb_get_server_info()
- maxdb_get_server_version()

maxdb_get_client_version

maxdb_get_client_version -- Get MaxDB client info

Description

int **maxdb_get_client_version** (void)

Returns client version number as an integer.

Return Values

A number that represents the MaxDB client library version in format: *main_version*10000 + minor_version *100 + sub_version*. For example, 7.5.0 is returned as 70500.

This is useful to quickly determine the version of the client library to know if some capability exists.

Examples

Example #48 - maxdb_get_client_version

```
<?php

/* We don't need a connection to determine
  the version of MaxDB client library */

printf("Client library version: %d\n", maxdb_get_client_version());
?>
```

The above example will output something similar to:

```
Client library version: 7.<...>
```

See Also

- maxdb_get_client_info()
- maxdb_get_server_info()
- maxdb_get_server_version()

maxdb_get_host_info

maxdb->get_host_info

maxdb_get_host_info -- maxdb->get_host_info -- Returns a string representing the type of connection used

Description

Procdural style:

string **maxdb_get_host_info** (resource `$link`)

Object oriented style (property):

maxdb

string *host_info*;

The maxdb_get_host_info() function returns a string describing the connection represented by the *`link`* parameter is using.

Return Values

A character string representing the server hostname and the connection type.

Examples

Example #49 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* print host information */
printf("Host info: %s\n", $maxdb->host_info);

/* close connection */
$maxdb->close();
?>
```

Example #50 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* print host information */
printf("Host info: %s\n", maxdb_get_host_info($link));

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Host info: localhost
```

See Also

- maxdb_get_proto_info()

maxdb_get_metadata

maxdb_get_metadata -- Alias of maxdb_stmt_result_metadata()

Description

This function is an alias of: maxdb_stmt_result_metadata().

This function alias is deprecated and only exists for backwards compatibility reasons. The use of this function is not recommended, as it may be removed from PHP in the future.

maxdb_get_proto_info

maxdb->protocol_version

maxdb_get_proto_info -- maxdb->protocol_version -- Returns the version of the MaxDB protocol used

Description

Procedural style:

int **maxdb_get_proto_info** (resource `$link`)

Object oriented style (property):

maxdb

string *protocol_version*;

Returns an integer representing the MaxDB protocol version used by the connection represented by the *`link`* parameter.

Return Values

Returns an integer representing the protocol version (constant 10).

Examples

Example #51 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* print protocol version */
printf("Protocol version: %d\n", $maxdb->protocol_version);

/* close connection */
$maxdb->close();
?>
```

Example #52 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* print protocol version */
printf("Protocol version: %d\n", maxdb_get_proto_info($link));

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Protocol version: 10
```

See Also

- maxdb_get_host_info()

maxdb_get_server_info

maxdb->server_info

maxdb_get_server_info -- maxdb->server_info -- Returns the version of the MaxDB server

Description

Procedural style:

string **maxdb_get_server_info** (resource `$link`)

Object oriented style (property):

maxdb

string *server_info*;

Returns a string representing the version of the MaxDB server that the MaxDB extension is connected to (represented by the *`link`* parameter).

Return Values

A character string representing the server version.

Examples

Example #53 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* print server version */
printf("Server version: %s\n", $maxdb->server_info);

/* close connection */
$maxdb->close();
?>
```

Example #54 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* print server version */
printf("Server version: %s\n", maxdb_get_server_info($link));

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Server version: Kernel    7<...>
```

See Also

- maxdb_get_client_info()
- maxdb_get_client_version()
- maxdb_get_server_version()

maxdb_get_server_version

maxdb_get_server_version -- Returns the version of the MaxDB server as an integer

Description

Procedural style:

int **maxdb_get_server_version** (resource `$link`)

Object oriented style (property):

maxdb

int *server_version*;

The maxdb_get_server_version() function returns the version of the server connected to (represented by the *`link`* parameter) as an integer.

The form of this version number is *main_version * 10000 + minor_version * 100 + sub_version* (i.e. version 7.5.0 is 70500).

Return Values

An integer representing the server version.

Examples

Example #55 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* print server version */
printf("Server version: %d\n", $maxdb->server_version);

/* close connection */
$maxdb->close();
?>
```

Example #56 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* print server version */
printf("Server version: %d\n", maxdb_get_server_version($link));

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Server version: 7<...>
```

See Also

- maxdb_get_client_info()
- maxdb_get_client_version()
- maxdb_get_server_info()

maxdb_info

maxdb->info

maxdb_info -- maxdb->info -- Retrieves information about the most recently executed query

Description

Procedural style:

string **maxdb_info** (resource `$link`)

Object oriented style (property)

maxdb

string *info*;

The maxdb_info() function returns a string providing information about the last query executed. The nature of this string is provided below:

Possible maxdb_info return values

Query type	**Example result string**
INSERT INTO...SELECT...	Records: 100 Duplicates: 0 Warnings: 0
INSERT INTO...VALUES (...),(...),(...)	Records: 3 Duplicates: 0 Warnings: 0
LOAD DATA INFILE ...	Records: 1 Deleted: 0 Skipped: 0 Warnings: 0
ALTER TABLE ...	Records: 3 Duplicates: 0 Warnings: 0
UPDATE ...	Rows matched: 40 Changed: 40 Warnings: 0

Note

Queries which do not fall into one of the above formats are not supported. In these situations, maxdb_info() will return an empty string.

Return Values

A character string representing additional information about the most recently executed query.

Examples

Example #57 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$maxdb->query("CREATE TABLE temp.t1 LIKE hotel.city");

/* INSERT INTO .. SELECT */
$maxdb->query("INSERT INTO temp.t1 SELECT * FROM hotel.city");
printf("%s\n", $maxdb->info);

/* close connection */
$maxdb->close();
?>
```

Example #58 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_query($link, "CREATE TABLE temp.t1 LIKE hotel.city");

/* INSERT INTO .. SELECT */
maxdb_query($link, "INSERT INTO temp.t1 SELECT * FROM hotel.city");
printf("%s\n", maxdb_info($link));

/* close connection */
```

```
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Records: 25 Duplicates: 0 Warnings: 0
```

See Also

- maxdb_affected_rows()
- maxdb_warning_count()
- maxdb_num_rows()

maxdb_init

maxdb_init -- Initializes MaxDB and returns an resource for use with maxdb_real_connect

Description

resource **maxdb_init** (void)

Allocates or initializes a MaxDB resource suitable for maxdb_options() and maxdb_real_connect().

Note
Any subsequent calls to any maxdb function (except maxdb_options()) will fail until maxdb_real_connect() was called.

Return Values

Returns an resource.

See Also

- maxdb_options()
- maxdb_close()
- maxdb_real_connect()
- maxdb_connect()

maxdb_insert_id

maxdb->insert_id

maxdb_insert_id -- maxdb->insert_id -- Returns the auto generated id used in the last query

Description

Procedural style:

mixed **maxdb_insert_id** (resource `$link`)

Object oriented style (property):

maxdb

mixed *insert_id*;

The maxdb_insert_id() function returns the ID generated by a query on a table with a column having the DEFAULT SERIAL attribute. If the last query wasn't an INSERT or UPDATE statement or if the modified table does not have a column with the DEFAULT SERIAL attribute, this function will return zero.

Return Values

The value of the *DEFAULT SERIAL* field that was updated by the previous query. Returns zero if there was no previous query on the connection or if the query did not update an *DEFAULT_SERIAL* value.

Note
If the number is greater than maximal int value, maxdb_insert_id() will return a string.

Examples

Example #59 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");
```

```
/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_report (MAXDB_REPORT_OFF);
$maxdb->query("DROP TABLE mycity");
maxdb_report (MAXDB_REPORT_ERROR);

$maxdb->query("CREATE TABLE mycity LIKE hotel.city");
$maxdb->query("ALTER TABLE mycity ADD id FIXED(11) DEFAULT SERIAL");

$query = "INSERT INTO mycity (zip,name,state) VALUES ('12203','Albany'
,'NY')";
$maxdb->query($query);

printf ("New Record has id %d.\n", $maxdb->insert_id);

/* drop table */
$maxdb->query("DROP TABLE mycity");

/* close connection */
$maxdb->close();
?>
```

Example #60 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_report (MAXDB_REPORT_OFF);
maxdb_query($link,"DROP TABLE mycity");
maxdb_report (MAXDB_REPORT_ERROR);

maxdb_query($link, "CREATE TABLE mycity LIKE hotel.city");
maxdb_query($link, "ALTER TABLE mycity ADD id FIXED(11) DEFAULT SERIAL");

$query = "INSERT INTO mycity (zip,name,state) VALUES ('12203','Albany'
,'NY')";
maxdb_query($link, $query);

printf ("New Record has id %d.\n", maxdb_insert_id($link));

/* drop table */
maxdb_query($link, "DROP TABLE mycity");

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
New Record has id 1.
```

maxdb_kill

maxdb->kill

maxdb_kill -- maxdb->kill -- Disconnects from a MaxDB server

Description

Procedural style:

bool **maxdb_kill** (resource `$link`, int `$processid`)

Object oriented style (method)

maxdb

bool **kill** (int `$processid`)

This function is used to disconnect from a MaxDB server specified by the *`processid`* parameter.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #61 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* determine our thread id */
$thread_id = $maxdb->thread_id;

/* Kill connection */
$maxdb->kill($thread_id);

/* This should produce an error */
```

```
if (!$maxdb->query("CREATE TABLE myCity LIKE City")) {
  printf("Error: %s\n", $maxdb->error);
  exit;
}

/* close connection */
$maxdb->close();
?>
```

Example #62 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* determine our thread id */
$thread_id = maxdb_thread_id($link);

/* Kill connection */
maxdb_kill($link, $thread_id);

/* This should produce an error */
if (!maxdb_query($link, "CREATE TABLE myCity LIKE City")) {
  printf("Error: %s\n", maxdb_error($link));
  exit;
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Error: Session not connected
```

See Also

- maxdb_thread_id()

maxdb_master_query

maxdb_master_query -- Enforce execution of a query on the master in a master/slave setup

Description

bool **maxdb_master_query** (resource `$link`, string `$query`)

Warning
This function is currently not documented; only its argument list is available.

maxdb_more_results

maxdb->more_results

maxdb_more_results -- maxdb->more_results -- Check if there any more query results from a multi query

Description

bool **maxdb_more_results** (resource `$link`)

maxdb_more_results() indicates if one or more result sets are available from a previous call to maxdb_multi_query().

Return Values

Always **FALSE**.

Examples

See maxdb_multi_query().

See Also

- maxdb_multi_query()
- maxdb_next_result()
- maxdb_store_result()
- maxdb_use_result()

maxdb_multi_query

maxdb->multi_query

maxdb_multi_query -- maxdb->multi_query -- Performs a query on the database

Description

Procedural style:

bool **maxdb_multi_query** (resource `$link`, string `$query`)

Object oriented style (method):

maxdb

bool **multi_query** (string `$query`)

The maxdb_multi_query() works like the function maxdb_query(). Multiple queries are not yet supported.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #63 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query  = "SELECT * FROM dual";

/* execute multi query */
if ($maxdb->multi_query($query)) {
  do {
      /* store first result set */
      if ($result = $maxdb->store_result()) {
```

```
          while ($row = $result->fetch_row()) {
              printf("%s\n", $row[0]);
          }
          $result->close();
      }
      /* print divider */
      if ($maxdb->more_results()) {
          printf("-----------------\n");
      }
  } while ($maxdb->next_result());
}

/* close connection */
$maxdb->close();
?>
```

Example #64 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT * FROM dual";

/* execute multi query */
if (maxdb_multi_query($link, $query)) {
  do {
      /* store first result set */
      if ($result = maxdb_store_result($link)) {
          while ($row = maxdb_fetch_row($result)) {
              printf("%s\n", $row[0]);
          }
          maxdb_free_result($result);
      }
      /* print divider */
      if (maxdb_more_results($link)) {
          printf("-----------------\n");
      }
  } while (maxdb_next_result($link));
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

a

See Also

- maxdb_use_result()
- maxdb_store_result()
- maxdb_next_result()
- maxdb_more_results()

maxdb_next_result

maxdb->next_result

maxdb_next_result -- maxdb->next_result -- Prepare next result from multi_query

Description

bool **maxdb_next_result** (resource `$link`)

Since multiple queries are not yet supported, maxdb_next_result() returns always **FALSE**.

Return Values

Returns **FALSE**.

See Also

- maxdb_multi_query()
- maxdb_more_results()
- maxdb_store_result()
- maxdb_use_result()

maxdb_num_fields

result->field_count

maxdb_num_fields -- result->field_count -- Get the number of fields in a result

Description

Procedural style:

int **maxdb_num_fields** (resource `$result`)

Object oriented style (property):

result

int *field_count*;

maxdb_num_fields() returns the number of fields from specified result set.

Return Values

The number of fields from a result set

Examples

Example #65 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

if ($result = $maxdb->query("SELECT * FROM hotel.city ORDER BY zip")) {

  /* determine number of fields in result set */
  $field_cnt = $result->field_count;

  printf("Result set has %d fields.\n", $field_cnt);

  /* close result set */
```

```
  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #66 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

if ($result = maxdb_query($link, "SELECT * FROM hotel.city ORDER BY zip")) {

  /* determine number of fields in result set */
  $field_cnt = maxdb_num_fields($result);

  printf("Result set has %d fields.\n", $field_cnt);

  /* close result set */
  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Result set has 3 fields.
```

See Also

- maxdb_fetch_field()

maxdb_num_rows

maxdb_num_rows -- Gets the number of rows in a result

Description

Procedural style:

int **maxdb_num_rows** (resource `$result`)

Object oriented style (property):

maxdb

int *num_rows*;

Returns the number of rows in the result set.

The use of maxdb_num_rows() depends on whether you use buffered or unbuffered result sets. In case you use unbuffered resultsets maxdb_num_rows() will not correct the correct number of rows until all the rows in the result have been retrieved.

Return Values

Returns number of rows in the result set.

Note
If the number of rows is greater than maximal int value, the number will be returned as a string.

Examples

Example #67 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}
```

```
if ($result = $maxdb->query("SELECT cno, name FROM hotel.customer ORDER BY
name")) {

  /* determine number of rows result set */
  $row_cnt = $result->num_rows;

  printf("Result set has %d rows.\n", $row_cnt);

  /* close result set */
  $result->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #68 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

if ($result = maxdb_query($link, "SELECT cno, name FROM hotel.customer ORDER
BY name")) {

  /* determine number of rows result set */
  $row_cnt = maxdb_num_rows($result);

  printf("Result set has %d rows.\n", $row_cnt);

  /* close result set */
  maxdb_free_result($result);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Result set has 15 rows.
```

See Also

- maxdb_affected_rows()
- maxdb_store_result()
- maxdb_use_result()

- maxdb_query()

maxdb_options

maxdb->options

maxdb_options -- maxdb->options -- Set options

Description

Procedural style:

bool **maxdb_options** (resource `$link`, int `$option`, mixed `$value`)

Object oriented style (method)

maxdb

bool **options** (int `$option`, mixed `$value`)

maxdb_options() can be used to set extra connect options and affect behavior for a connection.

This function may be called multiple times to set several options.

maxdb_options() should be called after maxdb_init() and before maxdb_real_connect().

The parameter *`option`* is the option that you want to set, the *`value`* is the value for the option. For detailed description of the options see » http://maxdb.sap.com/documentation/
The parameter *`option`* can be one of the following values:

Valid options

Name	Description
MAXDB_COMPNAME	The component name used to initialise the SQLDBC runtime environment.
MAXDB_APPLICATION	The application to be connected to the database.
MAXDB_APPVERSION	The version of the application.
MAXDB_SQLMODE	The SQL mode.
MAXDB_UNICODE	TRUE, if the connection is an unicode

	(UCS2) client or FALSE, if not.
MAXDB_TIMEOUT	The maximum allowed time of inactivity after which the connection to the database is closed by the system.
MAXDB_ISOLATIONLEVEL	Specifies whether and how shared locks and exclusive locks are implicitly requested or released.
MAXDB_PACKETCOUNT	The number of different request packets used for the connection.
MAXDB_STATEMENTCACHESIZE	The number of prepared statements to be cached for the connection for re-use.
MAXDB_CURSORPREFIX	The prefix to use for result tables that are automatically named.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

See maxdb_real_connect().

See Also

- maxdb_init()
- maxdb_real_connect()

maxdb_param_count

maxdb_param_count -- Alias of maxdb_stmt_param_count()

Description

This function is an alias of: maxdb_stmt_param_count().

This function alias is deprecated and only exists for backwards compatibility reasons. The use of this function is not recommended, as it may be removed from PHP in the future.

maxdb_ping

maxdb->ping

maxdb_ping -- maxdb->ping -- Pings a server connection, or tries to reconnect if the connection has gone down

Description

Procedural style:

bool **maxdb_ping** (resource `$link`)

Object oriented style (method):

maxdb

bool **ping** (void)

Checks whether the connection to the server is working. If it has gone down, and global option *maxdb.reconnect* is enabled an automatic reconnection is attempted.

This function can be used by clients that remain idle for a long while, to check whether the server has closed the connection and reconnect if necessary.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #69 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* check if server is alive */
if ($maxdb->ping()) {
```

```
  printf ("Our connection is ok!\n");
} else {
  printf ("Error: %s\n", $maxdb->error);
}

/* close connection */
$maxdb->close();
?>
```

Example #70 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* check if server is alive */
if (maxdb_ping($link)) {
  printf ("Our connection is ok!\n");
} else {
  printf ("Error: %s\n", maxdb_error($link));
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Our connection is ok!
```

maxdb_prepare

maxdb->prepare

maxdb_prepare -- maxdb->prepare -- Prepare a SQL statement for execution

Description

Procedure style:

resource **maxdb_prepare** (resource `$link`, string `$query`)

Object oriented style (method)

stmt

resource **prepare** (string `$query`)

maxdb_prepare() prepares the SQL query pointed to by the null-terminated string query, and returns a statement handle to be used for further operations on the statement. The query must consist of a single SQL statement.

Note
You should not add a terminating semicolon or *\g* to the statement.

The parameter *`query`* can include one or more parameter markers in the SQL statement by embedding question mark (*?*) characters at the appropriate positions.

Note
The markers are legal only in certain places in SQL statements. For example, they are allowed in the VALUES() list of an INSERT statement (to specify column values for a row), or in a comparison with a column in a WHERE clause to specify a comparison value. However, they are not allowed for identifiers (such as table or column names), in the select list that names the columns to be returned by a SELECT statement), or to specify both operands of a binary operator such as the = equal sign. The latter restriction is necessary because it would be impossible to determine the parameter type. In general, parameters are legal only in Data Manipulation Languange (DML)

statements, and not in Data Defination Language (DDL) statements.

The parameter markers must be bound to application variables using maxdb_stmt_bind_param() and/or maxdb_stmt_bind_result() before executing the statement or fetching rows.

Return Values

maxdb_prepare() returns a statement resource or **FALSE** if an error occured.

Examples

Example #71 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$city = "Rosemont";

/* create a prepared statement */
if ($stmt = $maxdb->prepare("SELECT state FROM hotel.city WHERE name=?")) {

  /* bind parameters for markers */
  $stmt->bind_param("s", $city);

  /* execute query */
  $stmt->execute();

  /* bind result variables */
  $stmt->bind_result($district);

  /* fetch value */
  $stmt->fetch();

  printf("%s is in district %s\n", $city, $district);

  /* close statement */
  $stmt->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #72 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$city = "Rosemont";

/* create a prepared statement */
if ($stmt = maxdb_prepare($link, "SELECT state FROM hotel.city WHERE
name=?")) {

  /* bind parameters for markers */
  maxdb_stmt_bind_param($stmt, "s", $city);

  /* execute query */
  maxdb_stmt_execute($stmt);

 /* bind result variables */
  maxdb_stmt_bind_result($stmt, $district);

  /* fetch value */
  maxdb_stmt_fetch($stmt);

  printf("%s is in district %s\n", $city, $district);

  /* close statement */
  maxdb_stmt_close($stmt);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Rosemont is in district IL
```

See Also

- maxdb_stmt_execute()
- maxdb_stmt_fetch()
- maxdb_stmt_bind_param()
- maxdb_stmt_bind_result()
- maxdb_stmt_close()

maxdb_query

maxdb->query

maxdb_query -- maxdb->query -- Performs a query on the database

Description

Procedural style:

mixed **maxdb_query** (resource `$link`, string `$query` [, int `$resultmode`])

Object oriented style (method):

maxdb

mixed **query** (string `$query`)

The maxdb_query() function is used to simplify the act of performing a query against the database represented by the *`link`* parameter.

Return Values

Returns **TRUE** on success or **FALSE** on failure. For *SELECT, SHOW, DESCRIBE* or *EXPLAIN* maxdb_query() will return a result resource.

Examples

Example #73 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* Create table doesn't return a resultset */
if ($maxdb->query("CREATE TABLE temp.mycity LIKE hotel.city") === TRUE) {
  printf("Table mycity successfully created.\n");
}

/* Select queries return a resultset */
```

```
if ($result = $maxdb->query("SELECT name FROM hotel.city")) {
  printf("Select returned %d rows.\n", $result->num_rows);

  /* free result set */
  $result->close();
}

/* If we have to retrieve large amount of data we use MAXDB_USE_RESULT */
if ($result = $maxdb->query("SELECT * FROM hotel.city", MAXDB_USE_RESULT)) {
  $result->close();
}

$maxdb->close();
?>
```

Example #74 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* Create table doesn't return a resultset */
if (maxdb_query($link, "CREATE TABLE temp.mycity LIKE hotel.city") === TRUE)
{
  printf("Table mycity successfully created.\n");
}

/* Select queries return a resultset */
if ($result = maxdb_query($link, "SELECT name FROM hotel.city")) {
  printf("Select returned %d rows.\n", maxdb_num_rows($result));

  /* free result set */
  maxdb_free_result($result);
}

/* If we have to retrieve large amount of data we use MAXDB_USE_RESULT */
if ($result = maxdb_query($link, "SELECT * FROM hotel.city",
MAXDB_USE_RESULT)) {
  maxdb_free_result($result);
}

maxdb_close($link);
?>
```

The above example will output something similar to:

```
Table mycity successfully created.
Select returned 25 rows.
```

See Also

- maxdb_real_query()
- maxdb_multi_query()
- maxdb_free_result()

maxdb_real_connect

maxdb->real_connect

maxdb_real_connect -- maxdb->real_connect -- Opens a connection to a MaxDB server

Description

Procedural style

bool **maxdb_real_connect** (resource `$link` [, string `$hostname` [, string `$username` [, string `$passwd` [, string `$dbname` [, int `$port` [, string `$socket`]]]]]])

Object oriented style (method)

maxdb

bool **real_connect** ([string `$hostname` [, string `$username` [, string `$passwd` [, string `$dbname` [, int `$port` [, string `$socket`]]]]]])

maxdb_real_connect() attempts to establish a connection to a MaxDB database engine running on *`hostname`*.

This function differs from maxdb_connect():

- maxdb_real_connect() needs a valid resource which has to be created by function maxdb_init()
- With function maxdb_options() you can set various options for connection.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #75 - Object oriented style

```
<?php

/* create a connection object which is not connected */
$maxdb = maxdb_init();
```

```
/* set connection options */
$maxdb->options(MAXDB_UNICODE, "FALSE");
$maxdb->options(MAXDB_TIMEOUT, 5);

/* connect to server */
$maxdb->real_connect('localhost', 'MONA', 'RED', 'DEMODB');

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

printf ("Connection: %s\n.", $maxdb->host_info);

$maxdb->close();
?>
```

Example #76 - Procedural style

```
<?php

/* create a connection object which is not connected */
$link = maxdb_init();

/* set connection options */
maxdb_options($link, MAXDB_UNICODE, "FALSE");
maxdb_options($link, MAXDB_TIMEOUT, 5);

/* connect to server */
maxdb_real_connect($link, 'localhost', 'MONA', 'RED', 'DEMODB');

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

printf ("Connection: %s\n.", maxdb_get_host_info($link));

maxdb_close($link);
?>
```

The above example will output something similar to:

```
Connection: localhost <...>
```

See Also

- maxdb_connect()
- maxdb_init()
- maxdb_options()

- maxdb_ssl_set()
- maxdb_close()

maxdb_real_escape_string

maxdb->real_escape_string

maxdb_real_escape_string -- maxdb->real_escape_string -- Escapes special characters in a string for use in a SQL statement, taking into account the current charset of the connection

Description

Procedural style:

string **maxdb_real_escape_string** (resource `$link`, string `$escapestr`)

Object oriented style (method):

maxdb

string **real_escape_sring** (string `$escapestr`)

This function is used to create a legal SQL string that you can use in a SQL statement. The string *escapestr* is encoded to an escaped SQL string, taking into account the current character set of the connection.

Characters encoded are *', "*.

Return Values

Returns an escaped string.

Examples

Example #77 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$maxdb->query("CREATE TABLE temp.mycity LIKE hotel.city");
```

```
$city = "'s Hertogenbosch";

/* this query will fail, cause we didn't escape $city */
if (!$maxdb->query("INSERT into temp.mycity VALUES ('11111','$city','NY')"))
{
  printf("Error: %s\n", $maxdb->sqlstate);
}

$city = $maxdb->real_escape_string($city);

/* this query with escaped $city will work */
if ($maxdb->query("INSERT into temp.mycity VALUES ('22222','$city','NY')"))
{
  printf("%d Row inserted.\n", $maxdb->affected_rows);
}

$maxdb->close();
?>
```

Example #78 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_query($link, "CREATE TABLE temp.mycity LIKE hotel.city");

$city = "'s Hertogenbosch";

/* this query will fail, cause we didn't escape $city */
if (!maxdb_query($link, "INSERT into temp.mycity VALUES
('11111','$city','NY')")) {
  printf("Error: %s\n", maxdb_sqlstate($link));
}

$city = maxdb_real_escape_string($link, $city);

/* this query with escaped $city will work */
if (maxdb_query($link, "INSERT into temp.mycity VALUES
('22222','$city','NY')")) {
  printf("%d Row inserted.\n", maxdb_affected_rows($link));
}

maxdb_close($link);
?>
```

The above example will output something similar to:

```
Warning: maxdb_query(): -5016 POS(43) Missing delimiter: ) <...>
Error: 42000
```

```
1 Row inserted.
```

See Also

- maxdb_character_set_name()

maxdb_real_query

maxdb->real_query

maxdb_real_query -- maxdb->real_query -- Execute an SQL query

Description

Procedural style

bool **maxdb_real_query** (resource `$link`, string `$query`)

Object oriented style (method):

maxdb

bool **real_query** (string `$query`)

The maxdb_real_query() is functionally identical with the maxdb_query().

Note
In order to determine if a given query should return a result set or not, see maxdb_field_count().

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- maxdb_query()
- maxdb_store_result()
- maxdb_use_result()

maxdb_report

maxdb_report -- Enables or disables internal report functions

Description

bool **maxdb_report** (int `$flags`)

Examples

Example #79 - Procedural style

```
<?php
/* activate reporting */
maxdb_report(MAXDB_REPORT_ERROR);

$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* this query should report an error */
$result = maxdb_query($link,"SELECT Name FROM Nonexistingtable WHERE
population > 50000");

maxdb_close($link);
?>
```

The above example will output something similar to:

```
Warning: maxdb_query(): -4004 POS(18) Unknown table name:NONEXISTINGTABLE <...>
```

maxdb_rollback

maxdb->rollback

maxdb_rollback -- maxdb->rollback -- Rolls back current transaction

Description

bool **maxdb_rollback** (resource `$link`)

maxdb

bool **rollback** (void)

Rollbacks the current transaction for the database specified by the *`link`* parameter.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #80 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* disable autocommit */
$maxdb->autocommit(FALSE);

$maxdb->query("CREATE TABLE temp.mycity LIKE hotel.city");
$maxdb->query("INSERT INTO temp.mycity SELECT * FROM hotel.city");

/* commit insert */
$maxdb->commit();

/* delete all rows */
$maxdb->query("DELETE FROM temp.mycity");

if ($result = $maxdb->query("SELECT COUNT(*) FROM temp.mycity")) {
```

```
  $row = $result->fetch_row();
  printf("%d rows in table mycity.\n", $row[0]);
  /* Free result */
  $result->close();
}

/* Rollback */
$maxdb->rollback();

if ($result = $maxdb->query("SELECT COUNT(*) FROM temp.mycity")) {
  $row = $result->fetch_row();
  printf("%d rows in table mycity (after rollback).\n", $row[0]);
  /* Free result */
  $result->close();
}

/* Drop table myCity */
$maxdb->query("DROP TABLE temp.mycity");

$maxdb->close();
?>
```

Example #81 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* disable autocommit */
maxdb_autocommit($link, FALSE);

maxdb_query($link, "CREATE TABLE temp.mycity LIKE hotel.city");
maxdb_query($link, "INSERT INTO temp.mycity SELECT * FROM hotel.city");

/* commit insert */
maxdb_commit($link);

/* delete all rows */
maxdb_query($link, "DELETE FROM temp.mycity");

if ($result = maxdb_query($link, "SELECT COUNT(*) FROM temp.mycity")) {
  $row = maxdb_fetch_row($result);
  printf("%d rows in table mycity.\n", $row[0]);
  /* Free result */
  maxdb_free_result($result);
}

/* Rollback */
maxdb_rollback($link);

if ($result = maxdb_query($link, "SELECT COUNT(*) FROM temp.mycity")) {
  $row = maxdb_fetch_row($result);
  printf("%d rows in table mycity (after rollback).\n", $row[0]);
```

```
  /* Free result */
  maxdb_free_result($result);
}

/* Drop table myCity */
maxdb_query($link, "DROP TABLE temp.mycity");

maxdb_close($link);
?>
```

The above example will output something similar to:

```
0 rows in table mycity.
25 rows in table mycity (after rollback).
```

See Also

- maxdb_commit()
- maxdb_autocommit()

maxdb_rpl_parse_enabled

maxdb_rpl_parse_enabled -- Check if RPL parse is enabled

Description

int **maxdb_rpl_parse_enabled** (resource `$link`)

Warning
This function is currently not documented; only its argument list is available.

maxdb_rpl_probe

maxdb_rpl_probe -- RPL probe

Description

bool **maxdb_rpl_probe** (resource `$link`)

Warning
This function is currently not documented; only its argument list is available.

maxdb_rpl_query_type

maxdb->rpl_query_type

maxdb_rpl_query_type -- maxdb->rpl_query_type -- Returns RPL query type

Description

int **maxdb_rpl_query_type** (resource `$link`)

Object oriented style (method)

maxdb

int **rpl_query_type** (void)

Warning
This function is currently not documented; only its argument list is available.

maxdb_select_db

maxdb->select_db

maxdb_select_db -- maxdb->select_db -- Selects the default database for database queries

Description

bool **maxdb_select_db** (resource `$link`, string `$dbname`)

The maxdb_select_db() function selects the default database (specified by the *`dbname`* parameter) to be used when performing queries against the database connection represented by the *`link`* parameter.

Note
This function should only be used to change the default database for the connection. You can select the default database with 4th parameter in maxdb_connect().

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #82 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* return name of current default database */
if ($result = $maxdb->query("SELECT SERVERDB FROM USERS WHERE
USERNAME='MONA'")) {
  $row = $result->fetch_row();
  printf("Default database is %s.\n", $row[0]);
  $result->close();
}

/* change db to non existing db */
$maxdb->select_db("XXXXXXXX");
```

```
/* return name of current default database */
if ($result = $maxdb->query("SELECT SERVERDB FROM USERS WHERE
USERNAME='MONA'")) {
  $row = $result->fetch_row();
  printf("Default database is %s.\n", $row[0]);
  $result->close();
}

$maxdb->close();
?>
```

Example #83 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* return name of current default database */
if ($result = maxdb_query($link, "SELECT SERVERDB FROM USERS WHERE
USERNAME='MONA'")) {
  $row = maxdb_fetch_row($result);
  printf("Default database is %s.\n", $row[0]);
  maxdb_free_result($result);
}

/* change db to non existing db */
maxdb_select_db($link, "XXXXXXXX");

/* return name of current default database */
if ($result = maxdb_query($link, "SELECT SERVERDB FROM USERS WHERE
USERNAME='MONA'")) {
  $row = maxdb_fetch_row($result);
  printf("Default database is %s.\n", $row[0]);
  maxdb_free_result($result);
}

maxdb_close($link);
?>
```

The above example will output something similar to:

```
Default database is <...>.

Warning: maxdb_select_db(): -10709 Connection failed (RTE:database not running)
<...>

Warning: maxdb_query(): -10821 Session not connected [] <...>

Warning: maxdb_close(): -10821 Session not connected [] <...>
```

See Also

- maxdb_connect()
- maxdb_real_connect()

maxdb_send_long_data

maxdb_send_long_data -- Alias of maxdb_stmt_send_long_data()

Description

This function is an alias of: maxdb_stmt_send_long_data().

This function alias is deprecated and only exists for backwards compatibility reasons. The use of this function is not recommended, as it may be removed from PHP in the future.

maxdb_send_query

maxdb->send_query

maxdb_send_query -- maxdb->send_query -- Send the query and return

Description

bool **maxdb_send_query** (resource `$link`, string `$query`)

Object oriented style (method)

maxdb

bool **send_query** (string `$query`)

Warning

This function is currently not documented; only its argument list is available.

maxdb_server_end

maxdb_server_end -- Shut down the embedded server

Description

void **maxdb_server_end** (void)

Warning
This function is currently not documented; only its argument list is available.

maxdb_server_init

maxdb_server_init -- Initialize embedded server

Description

bool **maxdb_server_init** ([array `$server` [, array `$groups`]])

Warning
This function is currently not documented; only its argument list is available.

maxdb_set_opt

maxdb_set_opt -- Alias of maxdb_options()

Description

This function is an alias of: maxdb_options().

maxdb_sqlstate

maxdb->sqlstate

maxdb_sqlstate -- maxdb->sqlstate -- Returns the SQLSTATE error from previous MaxDB operation

Description

Procedural style:

string **maxdb_sqlstate** (resource `$link`)

Object oriented style (property):

maxdb

string *sqlstate*;

Returns a string containing the SQLSTATE error code for the last error. The error code consists of five characters. *'00000'* means no error. The values are specified by ANSI SQL and ODBC.

Note
Note that not all MaxDB errors are yet mapped to SQLSTATE's. The value *HY000* (general error) is used for unmapped errors.

Return Values

Returns a string containing the SQLSTATE error code for the last error. The error code consists of five characters. *'00000'* means no error.

Examples

Example #84 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
```

```
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* Table City already exists, so we should get an error */
if (!$maxdb->query("CREATE TABLE hotel.city (ID INT, Name VARCHAR(30))")) {
  printf("Error - SQLSTATE %s.\n", $maxdb->sqlstate);
}

$maxdb->close();
?>
```

Example #85 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* Table City already exists, so we should get an error */
if (!maxdb_query($link, "CREATE TABLE hotel.city (ID INT, Name
VARCHAR(30))")) {
  printf("Error - SQLSTATE %s.\n", maxdb_sqlstate($link));
}

maxdb_close($link);
?>
```

The above example will output something similar to:

```
Warning: maxdb_query(): -6000 POS(20) Duplicate table name:CITY [I6000] <...>
Error - SQLSTATE I6000.
```

See Also

- maxdb_errno()
- maxdb_error()

maxdb_ssl_set

maxdb->ssl_set

maxdb_ssl_set -- maxdb->ssl_set -- Used for establishing secure connections using SSL

Description

Procedural style:

bool **maxdb_ssl_set** (resource `$link`, string `$key`, string `$cert`, string `$ca`, string `$capath`, string `$cipher`)

Object oriented style (method):

maxdb

bool **ssl_set** (string `$key`, string `$cert`, string `$ca`, string `$capath`, string `$cipher`)

Warning
This function is currently not documented; only its argument list is available.

maxdb_stat

maxdb->stat

maxdb_stat -- maxdb->stat -- Gets the current system status

Description

Procedural style:

string **maxdb_stat** (resource `$link`)

Object oriented style (method):

maxdb

string **maxdb->stat** (void)

maxdb_stat() returns a string containing several information about the MaxDB server running.

Return Values

A string describing the server status. **FALSE** if an error occurred.

Examples

Example #86 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

printf ("System status: %s\n", $maxdb->stat());

$maxdb->close();
?>
```

Example #87 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

printf("System status: %s\n", maxdb_stat($link));

maxdb_close($link);
?>
```

The above example will output something similar to:

```
System status: Kernel    7<...>
```

See Also

- maxdb_get_server_info()

maxdb_stmt_affected_rows

maxdb_stmt->affected_rows

maxdb_stmt_affected_rows -- maxdb_stmt->affected_rows -- Returns the total number of rows changed, deleted, or inserted by the last executed statement

Description

Procedural style :

int **maxdb_stmt_affected_rows** (resource `$stmt`)

Object oriented style (property):

stmt

int *affected_rows*;

maxdb_stmt_affected_rows() returns the number of rows affected by INSERT, UPDATE, or DELETE query. If the last query was invalid or the number of rows can not determined, this function will return -1.

Return Values

An integer greater than zero indicates the number of rows affected or retrieved. Zero indicates that no records where updated for an UPDATE/DELETE statement, no rows matched the WHERE clause in the query or that no query has yet been executed. -1 indicates that the query has returned an error or the number of rows can not determined.

Examples

Example #88 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* create temp table */
```

```
$maxdb->query("CREATE TABLE temp.mycity LIKE hotel.city");

$query = "INSERT INTO temp.mycity SELECT * FROM hotel.city WHERE state LIKE
?";

/* prepare statement */
if ($stmt = $maxdb->prepare($query)) {

  /* Bind variable for placeholder */
  $code = 'N%';
  $stmt->bind_param("s", $code);

  /* execute statement */
  $stmt->execute();

  printf("rows inserted: %d\n", $stmt->affected_rows);

  /* close statement */
  $stmt->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #89 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* create temp table */
maxdb_query($link, "CREATE TABLE temp.mycity LIKE hotel.city");

$query = "INSERT INTO temp.mycity SELECT * FROM hotel.city WHERE state LIKE
?";

/* prepare statement */
if ($stmt = maxdb_prepare($link, $query)) {

  /* Bind variable for placeholder */
  $code = 'N%';
  maxdb_stmt_bind_param($stmt, "s", $code);

  /* execute statement */
  maxdb_stmt_execute($stmt);

  printf("rows inserted: %d\n", maxdb_stmt_affected_rows($stmt));

  /* close statement */
  maxdb_stmt_close($stmt);
}
```

```
/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
rows inserted: 4
```

See Also

- maxdb_stmt_num_rows()
- maxdb_prepare()

maxdb_stmt_bind_param

stmt->bind_param

maxdb_stmt_bind_param -- stmt->bind_param -- Binds variables to a prepared statement as parameters

Description

Procedural style:

bool **maxdb_stmt_bind_param** (resource `$stmt`, string `$types`, mixed `&$var1` [, mixed `&$...`])

Object oriented style (method):

stmt

bool **bind_param** (string `$types`, mixed `&$var1` [, mixed `&$...`])

Procedural style (extended syntax):

bool **maxdb_stmt_bind_param** (resource `$stmt`, string `$types`, array `&$var`)

Object oriented style (method) (extended syntax):

stmt

bool **bind_param** (string `$types`, array `&$var`)

maxdb_stmt_bind_param() is used to bind variables for the parameter markers in the SQL statement that was passed to maxdb_prepare(). The string *`types`* contains one or more characters which specify the types for the corresponding bind variables.

The extended syntax of maxdb_stmt_bind_param() allows to give the parameters as an array instead of a variable list of PHP variables to the function. If the array variable has not been used before calling maxdb_stmt_bind_param(), it has to be initialized as an emtpy array. See the examples how to use maxdb_stmt_bind_param() with extended syntax.

Variables for SELECT INTO SQL statements can also be bound using

maxdb_stmt_bind_param(). Parameters for database procedures can be bound using maxdb_stmt_bind_param(). See the examples how to use maxdb_stmt_bind_param() in this cases.

If a variable bound as INTO variable to a SQL statement was used before, the content of this variable is overwritten by the data of the SELECT INTO statement. A reference to this variable will be invalid after a call to maxdb_stmt_bind_param().

For INOUT parameters of database procedures the content of the bound INOUT variable is overwritten by the output value of the database procedure. A reference to this variable will be invalid after a call to maxdb_stmt_bind_param().

Type specification chars

Character	Description
i	corresponding variable has type integer
d	corresponding variable has type double
s	corresponding variable has type string
b	corresponding variable is a blob and will be sent in packages

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #90 - Object oriented style

```
<?php
$maxdb = new maxdb('localhost', 'MONA', 'RED', 'DEMODB');

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$maxdb->query ("CREATE TABLE temp.mycity LIKE hotel.city");
$maxdb->query ("INSERT INTO temp.mycity SELECT * FROM hotel.city");

$stmt = $maxdb->prepare("INSERT INTO temp.mycity VALUES (?, ?, ?)");
$stmt->bind_param('sss', $zip, $name, $state);
```

```
$zip = '11111';
$name = 'Georgetown';
$state = 'NY';

/* execute prepared statement */
$stmt->execute();

printf("%d Row inserted.\n", $stmt->affected_rows);

/* close statement and connection */
$stmt->close();

/* Clean up table CountryLanguage */
$maxdb->query("DELETE FROM temp.mycity WHERE name='Georgetown'");
printf("%d Row deleted.\n", $maxdb->affected_rows);

/* close connection */
$maxdb->close();
?>
```

Example #91 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_query ($link, "CREATE TABLE temp.mycity LIKE hotel.city");
maxdb_query ($link, "INSERT INTO temp.mycity SELECT * FROM hotel.city");

$stmt = maxdb_prepare($link, "INSERT INTO temp.mycity VALUES (?, ?, ?)");
maxdb_stmt_bind_param($stmt, 'sss', $zip, $name, $state);

$zip = '11111';
$name = 'Georgetown';
$state = 'NY';

/* execute prepared statement */
maxdb_stmt_execute($stmt);

printf("%d Row inserted.\n", maxdb_stmt_affected_rows($stmt));

/* close statement and connection */
maxdb_stmt_close($stmt);

/* Clean up table CountryLanguage */
maxdb_query($link, "DELETE FROM temp.mycity WHERE name='Georgetown'");
printf("%d Row deleted.\n", maxdb_affected_rows($link));

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
1 Row inserted.
1 Row deleted.
```

Example #92 - Procedural style (SELECT INTO)

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* Performing SQL query */
$stmt = maxdb_prepare ($link, "SELECT price INTO ? FROM hotel.room where hno
= ? and type = ?");
if (!$stmt) {
 printf ("Prepare failed: %s\n", maxdb_error($link));
}

$hno = "50";
$rtype = "suite";

maxdb_stmt_bind_param($stmt, 'dss', $price, $hno, $rtype);
maxdb_stmt_execute($stmt);

printf ("%f\n", $price);

maxdb_stmt_close ($stmt);
?>
```

The above example will output something similar to:

```
21.600000
```

Example #93 - Procedural style (DB procedure)

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_report (MAXDB_REPORT_OFF);
maxdb_query($link,"DROP DBPROC test_proc");
maxdb_report (MAXDB_REPORT_ERROR);

$query = "create dbproc test_proc (INOUT e_text char(72)) AS select * from
SYSDBA.DUAL; fetch into :e_text;";
```

```
maxdb_query($link, $query);

/* Performing SQL query */
$stmt = maxdb_prepare ($link, "CALL test_proc (?)");
if (!$stmt) {
 printf ("Prepare failed: %s\n", maxdb_error($link));
}

maxdb_stmt_bind_param($stmt, 's', $result);
maxdb_stmt_execute($stmt);

printf ("%s\n", $result);

maxdb_stmt_close ($stmt);
?>
```

The above example will output something similar to:

```
a
```

Example #94 - Object oriented style (extended syntax)

```
<?php
$maxdb = new maxdb('localhost', 'MONA', 'RED', 'DEMODB');

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$maxdb->query ("CREATE TABLE temp.mycity LIKE hotel.city");
$maxdb->query ("INSERT INTO temp.mycity SELECT * FROM hotel.city");

$stmt = $maxdb->prepare("INSERT INTO temp.mycity VALUES (?, ?, ?)");

$arr = array();

$stmt->bind_param('iss', $arr);

$arr[0] = 11111;
$arr[1] = 'Georgetown';
$arr[2] = 'NY';

/* execute prepared statement */
$stmt->execute();

printf("%d Row inserted.\n", maxdb_stmt_affected_rows($stmt));

$arr[0] = 22222;
$arr[1] = 'New Orleans';
$arr[2] = 'LA';

/* execute prepared statement */
$stmt->execute();

printf("%d Row inserted.\n", $stmt->affected_rows);
```

```
/* close statement and connection */
$stmt->close();

$result = $maxdb->query("SELECT * from temp.mycity WHERE zip = '11111' OR
zip = '22222'");
if ($result) {
 while ($row = $result->fetch_row()) {
   printf ("%s %s %s\n", $row[0], $row[1], $row[2]);
 }
}

/* Clean up table CountryLanguage */
$maxdb->query("DELETE FROM temp.mycity WHERE name='Georgetown'");
$maxdb->query("DELETE FROM temp.mycity WHERE name='New Orleans'");
printf("%d Rows deleted.\n", $maxdb->affected_rows);

/* close connection */
$maxdb->close();
?>
```

The above example will output something similar to:

```
1 Row inserted.
1 Row inserted.
11111 Georgetown NY
22222 New Orleans LA
2 Rows deleted.
```

See Also

- maxdb_stmt_bind_result()
- maxdb_stmt_execute()
- maxdb_stmt_fetch()
- maxdb_prepare()
- maxdb_stmt_send_long_data()
- maxdb_stmt_errno()
- maxdb_stmt_error()

maxdb_stmt_bind_result

stmt->bind_result

maxdb_stmt_bind_result -- stmt->bind_result -- Binds variables to a prepared statement for result storage

Description

Procedural style:

bool **maxdb_stmt_bind_result** (resource `$stmt`, mixed `&$var1` [, mixed `&$...`])

Object oriented style (method):

stmt

bool **bind_result** (mixed `&$var1` [, mixed `&$...`])

maxdb_stmt_bind_result() is used to associate (bind) columns in the result set to variables. When maxdb_stmt_fetch() is called to fetch data, the MaxDB client/server protocol places the data for the bound columns into the specified variables *`var1, ...`*.

Note
Note that all columns must be bound prior to calling maxdb_stmt_fetch(). Depending on column types bound variables can silently change to the corresponding PHP type. A column can be bound or rebound at any time, even after a result set has been partially retrieved. The new binding takes effect the next time maxdb_stmt_fetch() is called.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #95 - Object oriented style
`<?php`

```
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* prepare statement */
if ($stmt = $maxdb->prepare("SELECT zip, name FROM hotel.city ORDER BY
name")) {
  $stmt->execute();

  /* bind variables to prepared statement */
  $stmt->bind_result($col1, $col2);

  /* fetch values */
  while ($stmt->fetch()) {
      printf("%s %s\n", $col1, $col2);
  }

  /* close statement */
  $stmt->close();
}
/* close connection */
$maxdb->close();

?>
```

Example #96 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (!$link) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* prepare statement */
if ($stmt = maxdb_prepare($link, "SELECT zip, name FROM hotel.city ORDER BY
name")) {
  maxdb_stmt_execute($stmt);

  /* bind variables to prepared statement */
  maxdb_stmt_bind_result($stmt, $col1, $col2);

  /* fetch values */
  while (maxdb_stmt_fetch($stmt)) {
      printf("%s %s\n", $col1, $col2);
  }

  /* close statement */
  maxdb_stmt_close($stmt);
}

/* close connection */
maxdb_close($link);
```

```
?>
```

The above example will output something similar to:

```
12203 Albany
60601 Chicago
60615 Chicago
45211 Cincinnati
33575 Clearwater
75243 Dallas
32018 Daytona Beach
33441 Deerfield Beach
48226 Detroit
90029 Hollywood
92714 Irvine
90804 Long Beach
11788 Long Island
90018 Los Angeles
70112 New Orleans
10019 New York
10580 New York
92262 Palm Springs
97213 Portland
60018 Rosemont
95054 Santa Clara
20903 Silver Spring
20005 Washington
20019 Washington
20037 Washington
```

See Also

- maxdb_stmt_bind_param()
- maxdb_stmt_execute()
- maxdb_stmt_fetch()
- maxdb_prepare()
- maxdb_stmt_prepare()
- maxdb_stmt_init()
- maxdb_stmt_errno()
- maxdb_stmt_error()

maxdb_stmt_close_long_data

stmt->close_long_data

maxdb_stmt_close_long_data -- stmt->close_long_data -- Ends a sequence of maxdb_stmt_send_long_data()

Description

Procedural style:

bool **maxdb_stmt_close_long_data** (resource `$stmt`, int `$param_nr`)

Object oriented style (method):

maxdb_stmt

bool **maxdb_stmt->close_long_data** (void)

This function has to be called after a sequence of maxdb_stmt_send_long_data(), that was started after maxdb_execute().

`param_nr` indicates which parameter to associate the end of data with. Parameters are numbered beginning with 0.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- maxdb_prepare()
- maxdb_stmt_bind_param()

maxdb_stmt_close

maxdb_stmt->close

maxdb_stmt_close -- maxdb_stmt->close -- Closes a prepared statement

Description

Procedural style:

bool **maxdb_stmt_close** (resource `$stmt`)

Object oriented style (method):

maxdb_stmt

bool **maxdb_stmt->close** (void)

Closes a prepared statement. maxdb_stmt_close() also deallocates the statement handle pointed to by *`stmt`*. If the current statement has pending or unread results, this function cancels them so that the next query can be executed.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- maxdb_prepare()

maxdb_stmt_data_seek

stmt->data_seek

maxdb_stmt_data_seek -- stmt->data_seek -- Seeks to an arbitray row in statement result set

Description

Procedural style:

bool **maxdb_stmt_data_seek** (resource `$statement`, int `$offset`)

Object oriented style (method):

stmt

bool **data_seek** (int `$offset`)

The maxdb_stmt_data_seek() function seeks to an arbitrary result pointer specified by the *`offset`* in the statement result set represented by *`statement`*. The *`offset`* parameter must be between zero and the total number of rows minus one (0.. maxdb_stmt_num_rows() - 1).

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #97 - Object oriented style

```
<?php
/* Open a connection */
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, zip FROM hotel.city ORDER BY name";
if ($stmt = $maxdb->prepare($query)) {
```

```
  /* execute query */
  $stmt->execute();

  /* bind result variables */
  $stmt->bind_result($name, $code);

  /* store result */
  $stmt->store_result();

  /* seek to row no. 5 */
  $stmt->data_seek(5);

  /* fetch values */
  $stmt->fetch();

  printf ("City: %s  Zip: %s\n", $name, $code);

  /* close statement */
  $stmt->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #98 - Procedural style

```
<?php
/* Open a connection */
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, zip FROM hotel.city ORDER BY name";
if ($stmt = maxdb_prepare($link, $query)) {

  /* execute query */
  maxdb_stmt_execute($stmt);

  /* bind result variables */
  maxdb_stmt_bind_result($stmt, $name, $code);

  /* store result */
  maxdb_stmt_store_result($stmt);

  /* seek to row no. 5 */
  maxdb_stmt_data_seek($stmt, 5);

  /* fetch values */
  maxdb_stmt_fetch($stmt);

  printf ("City: %s  Zip: %s\n", $name, $code);

  /* close statement */
```

```
  maxdb_stmt_close($stmt);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
City: Dallas  Zip: 75243
```

See Also

- maxdb_prepare()

maxdb_stmt_errno

maxdb_stmt->errno

maxdb_stmt_errno -- maxdb_stmt->errno -- Returns the error code for the most recent statement call

Description

Procedural style :

int **maxdb_stmt_errno** (resource `$stmt`)

Object oriented style (property):

stmt

int *errno*;

For the statement specified by *stmt*, maxdb_stmt_errno() returns the error code for the most recently invoked statement function that can succeed or fail.

Note
For possible error codes see documentation of SQLDBC: » http://maxdb.sap.com/documentation/.

Return Values

An error code value. Zero means no error occurred.

Examples

Example #99 - Object oriented style

```
<?php
/* Open a connection */
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
   printf("Connect failed: %s\n", maxdb_connect_error());
```

```
  exit();
}

$maxdb->query("CREATE TABLE temp.mycity LIKE hotel.city");
$maxdb->query("INSERT INTO temp.mycity SELECT * FROM hotel.city");


$query = "SELECT name, zip FROM temp.mycity ORDER BY name";
if ($stmt = $maxdb->prepare($query)) {

  /* drop table */
  $maxdb->query("DROP TABLE temp.mycity");

  /* execute query */
  $stmt->execute();

  printf("Error: %d.\n", $stmt->errno);

  /* close statement */
  $stmt->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #100 - Procedural style

```
<?php
/* Open a connection */
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_query($link, "CREATE TABLE temp.mycity LIKE hotel.city");
maxdb_query($link, "INSERT INTO temp.mycity SELECT * FROM hotel.city");


$query = "SELECT name, zip FROM temp.mycity ORDER BY name";
if ($stmt = maxdb_prepare($link, $query)) {

  /* drop table */
  maxdb_query($link, "DROP TABLE temp.mycity");

  /* execute query */
  maxdb_stmt_execute($stmt);

  printf("Error: %d.\n", maxdb_stmt_errno($stmt));

  /* close statement */
  maxdb_stmt_close($stmt);
}

/* close connection */
```

```
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Warning: maxdb_stmt_execute(): -4004 POS(23) Unknown table name:MYCITY [42000]
<...>
Error: -4004.
```

See Also

- maxdb_stmt_error()
- maxdb_stmt_sqlstate()

maxdb_stmt_error

maxdb_stmt->error

maxdb_stmt_error -- maxdb_stmt->error -- Returns a string description for last statement error

Description

Procedural style:

string **maxdb_stmt_error** (resource `$stmt`)

Object oriented style (property):

stmt

string *error*;

For the statement specified by *stmt*, maxdb_stmt_error() returns a containing the error message for the most recently invoked statement function that can succeed or fail.

Return Values

A string that describes the error. An empty string if no error occurred.

Examples

Example #101 - Object oriented style

```
<?php
/* Open a connection */
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$maxdb->query("CREATE TABLE temp.mycity LIKE hotel.city");
$maxdb->query("INSERT INTO temp.mycity SELECT * FROM hotel.city");


$query = "SELECT name, zip FROM temp.mycity ORDER BY name";
```

```
if ($stmt = $maxdb->prepare($query)) {

  /* drop table */
  $maxdb->query("DROP TABLE temp.mycity");

  /* execute query */
  $stmt->execute();

  printf("Error: %s.\n", $stmt->error);

  /* close statement */
  $stmt->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #102 - Procedural style

```
<?php
/* Open a connection */
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_query($link, "CREATE TABLE temp.mycity LIKE hotel.city");
maxdb_query($link, "INSERT INTO temp.mycity SELECT * FROM hotel.city");


$query = "SELECT name, zip FROM temp.mycity ORDER BY name";
if ($stmt = maxdb_prepare($link, $query)) {

  /* drop table */
  maxdb_query($link, "DROP TABLE temp.mycity");

  /* execute query */
  maxdb_stmt_execute($stmt);

  printf("Error: %s.\n", maxdb_stmt_error($stmt));

  /* close statement */
  maxdb_stmt_close($stmt);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Warning: maxdb_stmt_execute(): -4004 POS(23) Unknown table name:MYCITY [42000]
```

```
<...>
Error: POS(23) Unknown table name:MYCITY.
```

See Also

- maxdb_stmt_errno()
- maxdb_stmt_sqlstate()

maxdb_stmt_execute

stmt->execute

maxdb_stmt_execute -- stmt->execute -- Executes a prepared Query

Description

Procedural style:

bool **maxdb_stmt_execute** (resource `$stmt`)

Object oriented style (method):

stmt

bool **execute** (void)

The maxdb_stmt_execute() function executes a query that has been previously prepared using the maxdb_prepare() function represented by the *`stmt`* resource. When executed any parameter markers which exist will automatically be replaced with the appropiate data.

If the statement is UPDATE, DELETE, or INSERT, the total number of affected rows can be determined by using the maxdb_stmt_affected_rows() function. Likewise, if the query yields a result set the maxdb_fetch() function is used.

Note
When using maxdb_stmt_execute(), the maxdb_fetch() function must be used to fetch the data prior to preforming any additional queries.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #103 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");
```

```
/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$maxdb->query("CREATE TABLE temp.mycity LIKE hotel.city");

/* Prepare an insert statement */
$query = "INSERT INTO temp.mycity (zip, name, state) VALUES (?,?,?)";
$stmt = $maxdb->prepare($query);

$stmt->bind_param("sss", $val1, $val2, $val3);

$val1 = '11111';
$val2 = 'Georgetown';
$val3 = 'NY';

/* Execute the statement */
$stmt->execute();

$val1 = '22222';
$val2 = 'Hubbatown';
$val3 = 'CA';

/* Execute the statement */
$stmt->execute();

/* close statement */
$stmt->close();

/* retrieve all rows from myCity */
$query = "SELECT zip, name, state FROM temp.mycity";
if ($result = $maxdb->query($query)) {
  while ($row = $result->fetch_row()) {
      printf("%s (%s,%s)\n", $row[0], $row[1], $row[2]);
  }
  /* free result set */
  $result->close();
}

/* remove table */
$maxdb->query("DROP TABLE temp.mycity");

/* close connection */
$maxdb->close();
?>
```

Example #104 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
```

```
}

maxdb_query($link, "CREATE TABLE temp.mycity LIKE hotel.city");

/* Prepare an insert statement */
$query = "INSERT INTO temp.mycity (zip, name, state) VALUES (?,?,?)";
$stmt = maxdb_prepare($link, $query);

maxdb_stmt_bind_param($stmt, "sss", $val1, $val2, $val3);

$val1 = '11111';
$val2 = 'Georgetown';
$val3 = 'NY';

/* Execute the statement */
maxdb_stmt_execute($stmt);

$val1 = '22222';
$val2 = 'Hubbatown';
$val3 = 'CA';

/* Execute the statement */
maxdb_stmt_execute($stmt);

/* close statement */
maxdb_stmt_close($stmt);

/* retrieve all rows from myCity */
$query = "SELECT zip, name, state FROM temp.mycity";
if ($result = maxdb_query($link, $query)) {
  while ($row = maxdb_fetch_row($result)) {
      printf("%s (%s,%s)\n", $row[0], $row[1], $row[2]);
  }
  /* free result set */
  maxdb_free_result($result);
}

/* remove table */
maxdb_query($link, "DROP TABLE temp.mycity");

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
11111 (Georgetown,NY)
22222 (Hubbatown,CA)
```

See Also

- maxdb_prepare()
- maxdb_stmt_bind_param()

maxdb_stmt_fetch

stmt->fetch

maxdb_stmt_fetch -- stmt->fetch -- Fetch results from a prepared statement into the bound variables

Description

Procedural style:

bool **maxdb_stmt_fetch** (resource `$stmt`)

Object oriented style (method):

stmt

bool **fetch** (void)

maxdb_stmt_fetch() returns row data using the variables bound by maxdb_stmt_bind_result().

Note
Note that all columns must be bound by the application before calling maxdb_stmt_fetch().

Return Values

Return values

Value	Description
TRUE	Success. Data has been fetched
FALSE	Error occured
NULL	No more rows/data exists

Examples

Example #105 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT zip, name FROM hotel.city ORDER by name";

if ($stmt = $maxdb->prepare($query)) {

  /* execute statement */
  $stmt->execute();

  /* bind result variables */
  $stmt->bind_result($name, $code);

  /* fetch values */
  while ($stmt->fetch()) {
      printf ("%s (%s)\n", $name, $code);
  }

  /* close statement */
  $stmt->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #106 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT zip, name FROM hotel.city ORDER by name";

if ($stmt = maxdb_prepare($link, $query)) {

  /* execute statement */
  maxdb_stmt_execute($stmt);

  /* bind result variables */
  maxdb_stmt_bind_result($stmt, $name, $code);

  /* fetch values */
```

```
  while (maxdb_stmt_fetch($stmt)) {
      printf ("%s (%s)\n", $name, $code);
  }

  /* close statement */
  maxdb_stmt_close($stmt);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
12203 (Albany)
60601 (Chicago)
60615 (Chicago)
45211 (Cincinnati)
33575 (Clearwater)
75243 (Dallas)
32018 (Daytona Beach)
33441 (Deerfield Beach)
48226 (Detroit)
90029 (Hollywood)
92714 (Irvine)
90804 (Long Beach)
11788 (Long Island)
90018 (Los Angeles)
70112 (New Orleans)
10019 (New York)
10580 (New York)
92262 (Palm Springs)
97213 (Portland)
60018 (Rosemont)
95054 (Santa Clara)
20903 (Silver Spring)
20005 (Washington)
20019 (Washington)
20037 (Washington)
```

See Also

- maxdb_prepare()
- maxdb_stmt_errno()
- maxdb_stmt_error()
- maxdb_stmt_bind_result()

maxdb_stmt_free_result

stmt->free_result

maxdb_stmt_free_result -- stmt->free_result -- Frees stored result memory for the given statement handle

Description

Procedural style:

void **maxdb_stmt_free_result** (resource `$stmt`)

Object oriented style (method):

stmt

void **free_result** (void)

The maxdb_stmt_free_result() function frees the result memory associated with the statement represented by the *`stmt`* parameter, which was allocated by maxdb_stmt_store_result().

Return Values

This function doesn't return any value.

See Also

- maxdb_stmt_store_result()

maxdb_stmt_init

maxdb->stmt_init

maxdb_stmt_init -- maxdb->stmt_init -- Initializes a statement and returns an resource for use with maxdb_stmt_prepare

Description

Procedural style :

resource **maxdb_stmt_init** (resource `$link`)

Object oriented style (property):

maxdb

object **stmt_init** (void)

Allocates and initializes a statement resource suitable for maxdb_stmt_prepare().

Note
Any subsequent calls to any maxdb_stmt function will fail until maxdb_stmt_prepare() was called.

Return Values

Returns an resource.

See Also

- maxdb_stmt_prepare()

maxdb_stmt_num_rows

stmt->num_rows

maxdb_stmt_num_rows -- stmt->num_rows -- Return the number of rows in statements result set

Description

Procedural style :

int **maxdb_stmt_num_rows** (resource `$stmt`)

Object oriented style (property):

stmt

int *num_rows*;

Returns the number of rows in the result set.

Return Values

An integer representing the number of rows in result set.

Examples

Example #107 - Object oriented style

```
<?php
/* Open a connection */
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT zip, name FROM hotel.city ORDER BY name";
if ($stmt = $maxdb->prepare($query)) {

  /* execute query */
  $stmt->execute();
```

```
  /* store result */
  $stmt->store_result();

  printf("Number of rows: %d.\n", $stmt->num_rows);

  /* close statement */
  $stmt->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #108 - Procedural style

```
<?php
/* Open a connection */
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT zip, name FROM hotel.city ORDER BY name";
if ($stmt = maxdb_prepare($link, $query)) {

  /* execute query */
  maxdb_stmt_execute($stmt);

  /* store result */
  maxdb_stmt_store_result($stmt);

  printf("Number of rows: %d.\n", maxdb_stmt_num_rows($stmt));

  /* close statement */
  maxdb_stmt_close($stmt);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Number of rows: 25.
```

See Also

- maxdb_stmt_affected_rows()
- maxdb_prepare()
- maxdb_stmt_store_result()

maxdb_stmt_param_count

stmt->param_count

maxdb_stmt_param_count -- stmt->param_count -- Returns the number of parameter for the given statement

Description

Procedural style:

int **maxdb_stmt_param_count** (resource `$stmt`)

Object oriented style (property):

stmt

int *param_count*;

maxdb_stmt_param_count() returns the number of parameter markers present in the prepared statement.

Return Values

returns an integer representing the number of parameters.

Examples

Example #109 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

if ($stmt = $maxdb->prepare("SELECT name FROM hotel.city WHERE name=? OR
state=?")) {

  $marker = $stmt->param_count;
  printf("Statement has %d markers.\n", $marker);
```

```
  /* close statement */
  $stmt->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #110 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

if ($stmt = maxdb_prepare($link, "SELECT name FROM hotel.city WHERE name=?
OR state=?")) {

  $marker = maxdb_stmt_param_count($stmt);
  printf("Statement has %d markers.\n", $marker);

  /* close statement */
  maxdb_stmt_close($stmt);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Statement has 2 markers.
```

See Also

- maxdb_prepare()

maxdb_stmt_prepare

stmt->prepare

maxdb_stmt_prepare -- stmt->prepare -- Prepare a SQL statement for execution

Description

Procedure style:

bool **maxdb_stmt_prepare** (resource `$stmt`, string `$query`)

Object oriented style (method)

stmt

mixed **prepare** (string `$query`)

maxdb_stmt_prepare() prepares the SQL query pointed to by the null-terminated string query. The statement resource has to be allocated by maxdb_stmt_init(). The query must consist of a single SQL statement.

Note
You should not add a terminating semicolon or \g to the statement.

The parameter *`query`* can include one or more parameter markers in the SQL statement by embedding question mark (*?*) characters at the appropriate positions.

Note
The markers are legal only in certain places in SQL statements. For example, they are allowed in the VALUES() list of an INSERT statement (to specify column values for a row), or in a comparison with a column in a WHERE clause to specify a comparison value. However, they are not allowed for identifiers (such as table or column names), in the select list that names the columns to be returned by a SELECT statement), or to specify both operands of a binary operator such as the = equal sign. The latter restriction is necessary because it would be impossible to determine the parameter type. In general, parameters are legal only in Data Manipulation Languange (DML)

statements, and not in Data Defination Language (DDL) statements.

The parameter markers must be bound to application variables using maxdb_stmt_bind_param() and/or maxdb_stmt_bind_result() before executing the statement or fetching rows.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #111 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$city = "Portland";

/* create a prepared statement */
$stmt =  $maxdb->stmt_init();
if ($stmt->prepare("SELECT state FROM hotel.city WHERE name=?")) {

  /* bind parameters for markers */
  $stmt->bind_param("s", $city);

  /* execute query */
  $stmt->execute();

  /* bind result variables */
  $stmt->bind_result($district);

  /* fetch value */
  $stmt->fetch();

  printf("%s is in district %s\n", $city, $district);

  /* close statement */
  $stmt->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #112 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$city = "Portland";

/* create a prepared statement */
$stmt = maxdb_stmt_init($link);
if (maxdb_stmt_prepare($stmt, "SELECT state FROM hotel.city WHERE name=?"))
{

  /* bind parameters for markers */
  maxdb_stmt_bind_param($stmt, "s", $city);

  /* execute query */
  maxdb_stmt_execute($stmt);

  /* bind result variables */
  maxdb_stmt_bind_result($stmt, $district);

  /* fetch value */
  maxdb_stmt_fetch($stmt);

  printf("%s is in district %s\n", $city, $district);

  /* close statement */
  maxdb_stmt_close($stmt);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Portland is in district OR
```

See Also

- maxdb_stmt_init()
- maxdb_stmt_execute()
- maxdb_stmt_fetch()
- maxdb_stmt_bind_param()
- maxdb_stmt_bind_result()
- maxdb_stmt_close()

maxdb_stmt_reset

stmt->reset

maxdb_stmt_reset -- stmt->reset -- Resets a prepared statement

Description

Procedural style:

bool **maxdb_stmt_reset** (resource `$stmt`)

Object oriented style (method):

stmt

bool **reset** (void)

Warning
This function is currently not documented; only its argument list is available.

maxdb_stmt_result_metadata

maxdb_stmt_result_metadata -- Returns result set metadata from a prepared statement

Description

Procedural style:

resource **maxdb_stmt_result_metadata** (resource `$stmt`)

Object oriented style (method):

stmt

resource **result_metadata** (void)

If a statement passed to maxdb_prepare() is one that produces a result set, maxdb_stmt_result_metadata() returns the result resource that can be used to process the meta information such as total number of fields and individual field information.

Note
This result set pointer can be passed as an argument to any of the field-based functions that process result set metadata, such as: • maxdb_num_fields() • maxdb_fetch_field() • maxdb_fetch_field_direct() • maxdb_fetch_fields() • maxdb_field_count() • maxdb_field_seek() • maxdb_field_tell() • maxdb_free_result()

The result set structure should be freed when you are done with it, which you can do by passing it to maxdb_free_result()

Note

The result set returned by maxdb_stmt_result_metadata() contains only metadata. It does not contain any row results. The rows are obtained by using the statement handle with maxdb_fetch().

Return Values

maxdb_stmt_result_metadata() returns a result resource or **FALSE** if an error occured.

Examples

Example #113 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

$maxdb->query("CREATE TABLE temp.friends (id int, name varchar(20))");

$maxdb->query("INSERT INTO temp.friends VALUES (1,'Hartmut')");
$maxdb->query("INSERT INTO temp.friends VALUES (2, 'Ulf')");

$stmt = $maxdb->prepare("SELECT id, name FROM temp.friends");
$stmt->execute();

/* get resultset for metadata */
$result = $stmt->result_metadata();

/* retrieve field information from metadata result set */
$field = $result->fetch_field();

printf("Fieldname: %s\n", $field->name);

/* close resultset */
$result->close();

/* close connection */
$maxdb->close();
?>
```

Example #114 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

maxdb_query($link, "CREATE TABLE temp.friends (id int, name varchar(20))");

maxdb_query($link, "INSERT INTO temp.friends VALUES (1,'Hartmut')");
maxdb_query($link, "INSERT INTO temp.friends VALUES (2, 'Ulf')");
```

```
$stmt = maxdb_prepare($link, "SELECT id, name FROM temp.friends");
maxdb_stmt_execute($stmt);

/* get resultset for metadata */
$result = maxdb_stmt_result_metadata($stmt);

/* retrieve field information from metadata result set */
$field = maxdb_fetch_field($result);

printf("Fieldname: %s\n", $field->name);

/* close resultset */
maxdb_free_result($result);

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Fieldname: ID
```

See Also

- maxdb_prepare()
- maxdb_free_result()

maxdb_stmt_send_long_data

stmt->send_long_data

maxdb_stmt_send_long_data -- stmt->send_long_data -- Send data in blocks

Description

Procedural style:

bool **maxdb_stmt_send_long_data** (resource `$stmt`, int `$param_nr`, string `$data`)

Object oriented style (method)

stmt

bool **stmt_send_long_data** (int `$param_nr`, string `$data`)

Allows to send parameter data to the server in pieces (or chunks). This function can be called multiple times to send the parts of a character or binary data value for a column, which must be one of the TEXT or BLOB datatypes.

`param_nr` indicates which parameter to associate the data with. Parameters are numbered beginning with 0. *`data`* is a string containing data to be sent.

Note
For efficiency reasons, this function should be used after calling maxdb_execute(). In this case, the data is not stored on the client side. The end of the sequence must end with a call to maxdb_stmt_close_long_data().

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- maxdb_prepare()
- maxdb_stmt_bind_param()

maxdb_stmt_sqlstate

maxdb_stmt_sqlstate -- Returns SQLSTATE error from previous statement operation

Description

string **maxdb_stmt_sqlstate** (resource `$stmt`)

Returns a string containing the SQLSTATE error code for the most recently invoked prepared statement function that can succeed or fail. The error code consists of five characters. *'00000'* means no error. The values are specified by ANSI SQL and ODBC.

Note
Note that not all MaxDB errors are yet mapped to SQLSTATE's. The value *HY000* (general error) is used for unmapped errors.

Return Values

Returns a string containing the SQLSTATE error code for the last error. The error code consists of five characters. *'00000'* means no error.

Examples

Example #115 - Object oriented style

```
<?php
/* Open a connection */
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$maxdb->query("CREATE TABLE temp.mycity LIKE hotel.city");
$maxdb->query("INSERT INTO temp.mycity SELECT * FROM hotel.city");

$query = "SELECT name, zip FROM temp.mycity ORDER BY name";
if ($stmt = $maxdb->prepare($query)) {

  /* drop table */
  $maxdb->query("DROP TABLE temp.mycity");

  /* execute query */
  $stmt->execute();

  printf("Error: %s.\n", $stmt->sqlstate);
```

```
  /* close statement */
  $stmt->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #116 - Procedural style

```
<?php
/* Open a connection */
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_query($link, "CREATE TABLE temp.mycity LIKE hotel.city");
maxdb_query($link, "INSERT INTO temp.mycity SELECT * FROM hotel.city");

$query = "SELECT name, zip FROM temp.mycity ORDER BY name";
if ($stmt = maxdb_prepare($link, $query)) {

  /* drop table */
  maxdb_query($link, "DROP TABLE temp.mycity");

  /* execute query */
  maxdb_stmt_execute($stmt);

  printf("Error: %s.\n", maxdb_stmt_sqlstate($stmt));

  /* close statement */
  maxdb_stmt_close($stmt);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Warning: maxdb_stmt_execute(): -4004 POS(23) Unknown table name:MYCITY [42000]
<...>
Error: 42000.
```

See Also

- maxdb_stmt_errno()
- maxdb_stmt_error()

maxdb_stmt_store_result

maxdb->store_result

maxdb_stmt_store_result -- maxdb->store_result -- Transfers a result set from a prepared statement

Description

Procedural style:

bool **maxdb_stmt_store_result** (resource `$stmt`)

Object oriented style (method):

maxdb

object **store_result** (void)

maxdb_stmt_store_result() has no functionally effect and should not be used for retrieving data from MaxDB server.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #117 - Object oriented style

```
<?php
/* Open a connection */
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, zip FROM hotel.city ORDER BY name";
if ($stmt = $maxdb->prepare($query)) {

  /* execute query */
  $stmt->execute();
```

```
  /* store result */
  $stmt->store_result();

  printf("Number of rows: %d.\n", $stmt->num_rows);

  /* free result */
  $stmt->free_result();

  /* close statement */
  $stmt->close();
}

/* close connection */
$maxdb->close();
?>
```

Example #118 - Procedural style

```
<?php
/* Open a connection */
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query = "SELECT name, zip FROM hotel.city ORDER BY name";
if ($stmt = maxdb_prepare($link, $query)) {

  /* execute query */
  maxdb_stmt_execute($stmt);

  /* store result */
  maxdb_stmt_store_result($stmt);

  printf("Number of rows: %d.\n", maxdb_stmt_num_rows($stmt));

  /* free result */
  maxdb_stmt_free_result($stmt);

  /* close statement */
  maxdb_stmt_close($stmt);
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Number of rows: 25.
```

See Also

- maxdb_prepare()
- maxdb_stmt_result_metadata()
- maxdb_fetch()

maxdb_store_result

maxdb->store_result

maxdb_store_result -- maxdb->store_result -- Transfers a result set from the last query

Description

Procedural style:

resource **maxdb_store_result** (resource `$link`)

Object oriented style (method):

maxdb

object **store_result** (void)

This function has no functionally effect.

Return Values

Returns a result resource or **FALSE** if an error occurred.

Examples

See maxdb_multi_query().

See Also

- maxdb_real_query()
- maxdb_use_result()

maxdb_thread_id

maxdb->thread_id

maxdb_thread_id -- maxdb->thread_id -- Returns the thread ID for the current connection

Description

Procedural style:

int **maxdb_thread_id** (resource `$link`)

Object oriented style (property):

maxdb

int *thread_id*;

The maxdb_thread_id() function returns the thread ID for the current connection which can then be killed using the maxdb_kill() function. If the connection is lost and you reconnect with maxdb_ping(), the thread ID will be other. Therefore you should get the thread ID only when you need it.

Note
The thread ID is assigned on a connection-by-connection basis. Hence, if the connection is broken and then re-established a new thread ID will be assigned.

Return Values

maxdb_thread_id() returns the Thread ID for the current connection.

Examples

Example #119 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
```

```
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* determine our thread id */
$thread_id = $maxdb->thread_id;

/* Kill connection */
$maxdb->kill($thread_id);

/* This should produce an error */
if (!$maxdb->query("CREATE TABLE mycity LIKE hotel.city")) {
  printf("Error: %s\n", $maxdb->error);
  exit;
}

/* close connection */
$maxdb->close();
?>
```

Example #120 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

/* determine our thread id */
$thread_id = maxdb_thread_id($link);

/* Kill connection */
maxdb_kill($link, $thread_id);

/* This should produce an error */
if (!maxdb_query($link, "CREATE TABLE mycity LIKE hotel.city")) {
  printf("Error: %s\n", maxdb_error($link));
  exit;
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Warning: maxdb_query(): -10821 Session not connected <...>
Error: Session not connected
```

See Also

- maxdb_kill()

maxdb_thread_safe

maxdb_thread_safe -- Returns whether thread safety is given or not

Description

Procedural style:

bool **maxdb_thread_safe** (void)

maxdb_thread_safe() indicates whether the client library is compiled as thread-safe.

Return Values

TRUE if the client library is thread-safe, otherwise **FALSE**.

maxdb_use_result

maxdb->use_result

maxdb_use_result -- maxdb->use_result -- Initiate a result set retrieval

Description

Procedural style:

resource **maxdb_use_result** (resource `$link`)

Object oriented style (method):

maxdb

resource **use_result** (void)

maxdb_use_result() has no effect.

Return Values

Returns result or **FALSE** in case of error.

Examples

Example #121 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query  = "SELECT * FROM DUAL";

/* execute multi query */
if ($maxdb->multi_query($query)) {
  do {
      /* store first result set */
      if ($result = $maxdb->use_result()) {
          while ($row = $result->fetch_row()) {
```

```
            printf("%s\n", $row[0]);
        }
        $result->close();
    }
    /* print divider */
    if ($maxdb->more_results()) {
        printf("-----------------\n");
    }
  } while ($maxdb->next_result());
}

/* close connection */
$maxdb->close();
?>
```

Example #122 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$query  = "SELECT * FROM DUAL";

/* execute multi query */
if (maxdb_multi_query($link, $query)) {
  do {
      /* store first result set */
      if ($result = maxdb_use_result($link)) {
          while ($row = maxdb_fetch_row($result)) {
              printf("%s\n", $row[0]);
          }
          maxdb_free_result($result);
      }
      /* print divider */
      if (maxdb_more_results($link)) {
          printf("-----------------\n");
      }
  } while (maxdb_next_result($link));
}

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
a
```

See Also

- maxdb_real_query()
- maxdb_store_result()

maxdb_warning_count

maxdb->warning_count

maxdb_warning_count -- maxdb->warning_count -- Returns the number of warnings from the last query for the given link

Description

Procedural style:

int **maxdb_warning_count** (resource `$link`)

Object oriented style (property):

maxdb

int *warning_count*;

maxdb_warning_count() returns the number of warnings from the last query in the connection represented by the *`link`* parameter.

Return Values

Number of warnings or zero if there are no warnings.

Examples

Example #123 - Object oriented style

```
<?php
$maxdb = new maxdb("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

$maxdb->query("CREATE TABLE temp.mycity LIKE hotel.city");

/* a remarkable city in Wales */
$query = "INSERT INTO temp.mycity (zip, name) VALUES('11111',
      'Llanfairpwllgwyngyllgogerychwyrndrobwllllantysiliogogogoch')";
```

```
$maxdb->query($query);

printf ("Number of warning: %d\n", $maxdb->warning_count);

/* close connection */
$maxdb->close();
?>
```

Example #124 - Procedural style

```
<?php
$link = maxdb_connect("localhost", "MONA", "RED", "DEMODB");

/* check connection */
if (maxdb_connect_errno()) {
  printf("Connect failed: %s\n", maxdb_connect_error());
  exit();
}

maxdb_query($link, "CREATE TABLE temp.mycity LIKE hotel.city");

/* a remarkable long city name in Wales */
$query = "INSERT INTO temp.mycity (zip, name) VALUES('11111',
      'Llanfairpwllgwyngyllgogerychwyrndrobwllllantysiliogogogoch')";

maxdb_query($link, $query);

printf ("Number of warning: %d\n", maxdb_warning_count($link));

/* close connection */
maxdb_close($link);
?>
```

The above example will output something similar to:

```
Warning: maxdb_query(): -8004 POS(62) Constant must be compatible with column
type and length <...>
Number of warning: 0
```

See Also

- maxdb_errno()
- maxdb_error()
- maxdb_sqlstate()