

FrontBase

Introduction

These functions allow you to access FrontBase database servers. More information about FrontBase can be found at » <http://www.frontbase.com/>.

Documentation for FrontBase can be found at » <http://www.frontbase.com/cgi-bin/WebObjects/FrontBase.woa/wa/productsPage?currentPage=Documentation>.

Frontbase support has been added to PHP 4.0.6.

Installing/Configuring

Requirements

You must install the FrontBase database server or at least the fbsql client libraries to use this functions. You can get FrontBase from » <http://www.frontbase.com/>.

Installation

In order to have these functions available, you must compile PHP with fbsql support by using the `--with-fbsql[=DIR]` option. If you use this option without specifying the path to fbsql, PHP will search for the fbsql client libraries in the default installation location for the platform. Users who installed FrontBase in a non standard directory should always specify the path to fbsql: `--with-fbsql=/path/to/fbsql`. This will force PHP to use the client libraries installed by FrontBase, avoiding any conflicts.

Runtime Configuration

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

FrontBase configuration options

Name	Default	Changeable	Changelog
fbsql.allow_persistent	"1"	PHP_INI_SYSTEM	Available since PHP 4.2.0.
fbsql.generate_warnings	"0"	PHP_INI_SYSTEM	Available since PHP 4.0.6.
fbsql.autocommit	"1"	PHP_INI_SYSTEM	Available since PHP 4.0.6.
fbsql.max_persistent	"-1"	PHP_INI_SYSTEM	Available since PHP 4.0.6.
fbsql.max_links	"128"	PHP_INI_SYSTEM	Available since PHP 4.0.6.
fbsql.max_connections	"128"	PHP_INI_SYSTEM	Available since PHP 4.0.6.
fbsql.max_results	"128"	PHP_INI_SYSTEM	Available since PHP 4.0.6.

fbsql.batchSize	"1000"	PHP_INI_SYSTEM	Available since PHP 4.2.0. Removed in PHP 5.1.0.
fbsql.default_host	NULL	PHP_INI_SYSTEM	Available since PHP 4.0.6.
fbsql.default_user	"_SYSTEM"	PHP_INI_SYSTEM	Available since PHP 4.0.6.
fbsql.default_password	""	PHP_INI_SYSTEM	Available since PHP 4.0.6.
fbsql.default_database	""	PHP_INI_SYSTEM	Available since PHP 4.0.6.
fbsql.default_database_password	""	PHP_INI_SYSTEM	Available since PHP 4.0.6.

For further details and definitions of the PHP_INI_* constants, see the [php.ini directives](#).

Resource Types

This extension has no resource types defined.

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.

FBSQL_ASSOC ([integer](#))

FBSQL_NUM ([integer](#))

FBSQL_BOTH ([integer](#))

FBSQL_LOCK_DEFERRED ([integer](#))

FBSQL_LOCK_OPTIMISTIC ([integer](#))

FBSQL_LOCK_PESSIMISTIC ([integer](#))

FBSQL_ISO_READ_UNCOMMITTED ([integer](#))

FBSQL_ISO_READ_COMMITTED ([integer](#))

FBSQL_ISO_REPEATABLE_READ ([integer](#))

FBSQL_ISO_SERIALIZABLE ([integer](#))

FBSQL_ISO_VERSIONED ([integer](#))

FBSQL_UNKNOWN ([integer](#))

FBSQL_STOPPED ([integer](#))

FBSQL_STARTING ([integer](#))

FBSQL_RUNNING ([integer](#))

FBSQL_STOPPING ([integer](#))

FBSQL_NOEXEC ([integer](#))

FBSQL_LOB_DIRECT ([integer](#))

FBSQL_LOB_HANDLE ([integer](#))

FrontBase Functions

fbsql_affected_rows

fbsql_affected_rows -- Get number of affected rows in previous FrontBase operation

Description

int **fbsql_affected_rows** ([resource *\$link_identifier*])

[fbsql_affected_rows\(\)](#) returns the number of rows affected by the last INSERT, UPDATE or DELETE query associated with *link_identifier*.

Note
If you are using transactions, you need to call fbsql_affected_rows() after your INSERT, UPDATE, or DELETE query, not after the commit.

If the last query was a DELETE query with no WHERE clause, all of the records will have been deleted from the table but this function will return zero.

Note
When using UPDATE, FrontBase will not update columns where the new value is the same as the old value. This creates the possibility that fbsql_affected_rows() may not actually equal the number of rows matched, only the number of rows that were literally affected by the query.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

If the last query failed, this function will return -1.

See Also

- [fbsql_num_rows\(\)](#)

fbsql_autocommit

fbsql_autocommit -- Enable or disable autocommit

Description

bool **fbsql_autocommit** (resource \$link_identifier [, bool \$OnOff])

Returns the current autocommit status.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

OnOff

If this optional parameter is given the auto commit status will be changed. With *OnOff* set to **TRUE** each statement will be committed automatically, if no errors was found. With *OnOff* set to **FALSE** the user must commit or rollback the transaction using either [fbsql_commit\(\)](#) or [fbsql_rollback\(\)](#).

Return Values

Returns the current autocommit status, as a boolean.

See Also

- [fbsql_commit\(\)](#)
- [fbsql_rollback\(\)](#)

fbsql_blob_size

fbsql_blob_size -- Get the size of a BLOB

Description

int **fbsql_blob_size** (string \$blob_handle [, resource \$link_identifier])

Returns the size of the given BLOB.

Parameters

blob_handle

A BLOB handle, returned by [fbsql_create_blob\(\)](#).

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns the BLOB size as an integer, or **FALSE** on error.

See Also

- [fbsql_clob_size\(\)](#)

fbsql_change_user

fbsql_change_user -- Change logged in user of the active connection

Description

```
bool fbsql_change_user ( string $user, string $password [, string $database [, resource $link_identifier ] ] )
```

[fbsql_change_user\(\)](#) changes the logged in user of the specified connection. If the new user and password authorization fails, the current connected user stays active.

Parameters

user

The new user name.

password

The new user password.

database

If specified, this will be the default or current database after the user has been changed.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

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

fbsql_clob_size

fbsql_clob_size -- Get the size of a CLOB

Description

int **fbsql_clob_size** (string \$clob_handle [, resource \$link_identifier])

Returns the size of the given CLOB.

Parameters

clob_handle

A CLOB handle, returned by [fbsql_create_clob\(\)](#).

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns the CLOB size as an integer, or **FALSE** on error.

See Also

- [fbsql_blob_size\(\)](#)

fbsql_close

fbsql_close -- Close FrontBase connection

Description

bool **fbsql_close** ([resource \$link_identifier])

Closes the connection to the FrontBase server that's associated with the specified link identifier.

Using [fbsql_close\(\)](#) isn't usually necessary, as non-persistent open links are automatically closed at the end of the script's execution.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

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

Examples

Example #1 - [fbsql_close\(\)](#) example

```
<?php
$link = fbsql_connect("localhost", "_SYSTEM", "secret")
    or die("Could not connect");
echo "Connected successfully";
fbsql_close($link);
?>
```

See Also

- [fbsql_connect\(\)](#)
- [fbsql_pconnect\(\)](#)

fbsql_commit

fbsql_commit -- Commits a transaction to the database

Description

bool **fbsql_commit** ([resource \$link_identifier])

Ends the current transaction by writing all inserts, updates and deletes to the disk and unlocking all row and table locks held by the transaction. This command is only needed if autocommit is set to false.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

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

See Also

- [fbsql_autocommit\(\)](#)
- [fbsql_rollback\(\)](#)

fbsql_connect

fbsql_connect -- Open a connection to a FrontBase Server

Description

resource **fbsql_connect** ([string \$hostname [, string \$username [, string \$password]]])

[fbsql_connect\(\)](#) establishes a connection to a FrontBase server.

If a second call is made to [fbsql_connect\(\)](#) with the same arguments, no new link will be established, but instead, the link identifier of the already opened link will be returned.

The link to the server will be closed as soon as the execution of the script ends, unless it's closed earlier by explicitly calling [fbsql_close\(\)](#).

Parameters

hostname

The server host name. Defaults to ' **NULL** '.

username

The user name for the connection. Defaults to `_SYSTEM`.

password

The password for the connection. Defaults to the empty string.

Return Values

Returns a positive FrontBase link identifier on success, or **FALSE** on errors.

Examples

Example #2 - [fbsql_connect\(\)](#) example

```
<?php

$link = fbsql_connect("localhost", "_SYSTEM", "secret")
    or die("Could not connect");
echo "Connected successfully";
fbsql_close($link);

?>
```


See Also

- [fbsql_pconnect\(\)](#)
- [fbsql_close\(\)](#)

fbsql_create_blob

fbsql_create_blob -- Create a BLOB

Description

string **fbsql_create_blob** (string \$blob_data [, resource \$link_identifier])

Creates a BLOB from the given data.

Parameters

blob_data

The BLOB data.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns a resource handle to the newly created BLOB, which can be used with insert and update commands to store the BLOB in the database.

Examples

Example #3 - [fbsql_create_blob\(\)](#) example

```
<?php
$link = fbsql_pconnect("localhost", "_SYSTEM", "secret")
    or die("Could not connect");
$filename = "blobfile.bin";
$fp = fopen($filename, "rb");
$blobdata = fread($fp, filesize($filename));
fclose($fp);

$blobHandle = fbsql_create_blob($blobdata, $link);

$sql = "INSERT INTO BLOB_TABLE (BLOB_COLUMN) VALUES ($blobHandle)";
$rs = fbsql_query($sql, $link);
?>
```

See Also

- [fbsql_create_clob\(\)](#)
- [fbsql_read_blob\(\)](#)
- [fbsql_read_clob\(\)](#)
- [fbsql_set_lob_mode\(\)](#)

fbsql_create_clob

fbsql_create_clob -- Create a CLOB

Description

string **fbsql_create_clob** (string \$clob_data [, resource \$link_identifier])

Creates a CLOB from the given data.

Parameters

clob_data

The CLOB data.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns a resource handle to the newly created CLOB, which can be used with insert and update commands to store the CLOB in the database.

Examples

Example #4 - [fbsql_create_clob\(\)](#) example

```
<?php
$link = fbsql_pconnect("localhost", "_SYSTEM", "secret")
    or die("Could not connect");
$filename = "clob_file.txt";
$fp = fopen($filename, "rb");
$clobdata = fread($fp, filesize($filename));
fclose($fp);

$clobHandle = fbsql_create_clob($clobdata, $link);

$sql = "INSERT INTO CLOB_TABLE (CLOB_COLUMN) VALUES ($clobHandle)";
$rs = fbsql_query($sql, $link);
?>
```

See Also

- [fbsql_create_blob\(\)](#)
- [fbsql_read_blob\(\)](#)
- [fbsql_read_clob\(\)](#)
- [fbsql_set_lob_mode\(\)](#)

fbsql_create_db

fbsql_create_db -- Create a FrontBase database

Description

```
bool fbsql_create_db ( string $database_name [, resource $link_identifier [, string $database_options ] ] )
```

Attempts to create a new database on the specied server.

Parameters

database_name

The database name, as a string.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

database_options

Return Values

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

Examples

Example #5 - [fbsql_create_db\(\)](#) example

```
<?php
$link = fbsql_pconnect("localhost", "_SYSTEM", "secret")
    or die("Could not connect");
if (fbsql_create_db("my_db")) {
    echo "Database created successfully\n";
} else {
    printf("Error creating database: %s\n", fbsql_error());
}
?>
```

See Also

- [fbsql_drop_db\(\)](#)

fbsql_data_seek

fbsql_data_seek -- Move internal result pointer

Description

bool **fbsql_data_seek** (resource \$result, int \$row_number)

Moves the internal row pointer of the FrontBase result associated with the specified result identifier to point to the specified row number.

The next call to [fbsql_fetch_row\(\)](#) would return that row.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

row_number

The row number. Starts at 0.

Return Values

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

Examples

Example #6 - [fbsql_data_seek\(\)](#) example

```
<?php
$link = fbsql_pconnect("localhost", "_SYSTEM", "secret")
    or die("Could not connect");

fbsql_select_db("samp_db")
    or die("Could not select database");

$query = "SELECT last_name, first_name FROM friends;";
$result = fbsql_query($query)
    or die("Query failed");

// fetch rows in reverse order

for ($i = fbsql_num_rows($result) - 1; $i >=0; $i--) {
    if (!fbsql_data_seek($result, $i)) {
        printf("Cannot seek to row %d\n", $i);
        continue;
    }
}
```



```
    if (!($row = fbsql_fetch_object($result)))
        continue;

    echo $row->last_name . $row->first_name . "<br />\n";
}

fbsql_free_result($result);
?>
```

fbsql_database_password

fbsql_database_password -- Sets or retrieves the password for a FrontBase database

Description

string **fbsql_database_password** (resource \$link_identifier [, string \$database_password])

Sets and retrieves the database password used by the connection. If a database is protected by a database password, the user must call this function before calling [fbsql_select_db\(\)](#).

If no link is open, the function will try to establish a link as if [fbsql_connect\(\)](#) was called, and use it.

This function does not change the database password in the database nor can it be used to retrieve the database password for a database.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

database_password

The database password, as a string. If given, the function sets the database password for the specified link identifier.

Return Values

Returns the database password associated with the link identifier.

Examples

Example #7 - [fbsql_create_clob\(\)](#) example

```
<?php
$link = fbsql_pconnect("localhost", "_SYSTEM", "secret")
    or die("Could not connect");
fbsql_database_password($link, "secret db password");
fbsql_select_db($database, $link);
?>
```

See Also

- [fbsql_connect\(\)](#)
- [fbsql_pconnect\(\)](#)
- [fbsql_select_db\(\)](#)

fbsql_database

fbsql_database -- Get or set the database name used with a connection

Description

string **fbsql_database** (resource \$link_identifier [, string \$database])

Get or set the database name used with the connection.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

database

The database name. If given, the default database of the connexion will be changed to *database*.

Return Values

Returns the name of the database used with this connection.

fbsql_db_query

fbsql_db_query -- Send a FrontBase query

Description

resource **fbsql_db_query** (string \$database, string \$query [, resource \$link_identifier])

Selects a database and executes a query on it.

Parameters

database

The database to be selected.

query

The SQL query to be executed.

Note
The query string shall always end with a semicolon.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns a positive FrontBase result identifier to the query result, or **FALSE** on error.

See Also

- [fbsql_query\(\)](#)
- [fbsql_connect\(\)](#)

fbsql_db_status

fbsql_db_status -- Get the status for a given database

Description

```
int fbsql_db_status ( string $database_name [, resource $link_identifier ] )
```

Gets the current status of the specified database.

Parameters

database_name

The database name.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns an integer value with the current status. This can be one of the following constants:

- **FALSE** - The exec handler for the host was invalid. This error will occur when the *link_identifier* connects directly to a database by using a port number. FBExec can be available on the server but no connection has been made for it.
- **FBSQL_UNKNOWN** - The Status is unknown.
- **FBSQL_STOPPED** - The database is not running. Use [fbsql_start_db\(\)](#) to start the database.
- **FBSQL_STARTING** - The database is starting.
- **FBSQL_RUNNING** - The database is running and can be used to perform SQL operations.
- **FBSQL_STOPPING** - The database is stopping.
- **FBSQL_NOEXEC** - FBExec is not running on the server and it is not possible to get the status of the database.

See Also

- [fbsql_start_db\(\)](#)
- [fbsql_stop_db\(\)](#)

fbsql_drop_db

fbsql_drop_db -- Drop (delete) a FrontBase database

Description

bool **fbsql_drop_db** (string \$database_name [, resource \$link_identifier])

[fbsql_drop_db\(\)](#) attempts to drop (remove) an entire database from the server associated with the specified link identifier.

Parameters

database_name

The database name, as a string.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

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

See Also

- [fbsql_create_db\(\)](#)

fbsql_errno

fbsql_errno -- Returns the error number from previous operation

Description

int **fbsql_errno** ([resource \$link_identifier])

Returns the numerical value of the error message from previous FrontBase operation.

Errors coming back from the fbsql database backend don't issue warnings. Instead, use [fbsql_errno\(\)](#) to retrieve the error code. Note that this function only returns the error code from the most recently executed fbsql function (not including [fbsql_error\(\)](#) and [fbsql_errno\(\)](#)), so if you want to use it, make sure you check the value before calling another fbsql function.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns the error number from the last fbsql function, or 0 (zero) if no error occurred.

Examples

Example #8 - [fbsql_errno\(\)](#) Example

```
<?php
fbsql_connect("marliesle");
echo fbsql_errno() . ": " . fbsql_error() . "<br />";
fbsql_select_db("nonexistentdb");
echo fbsql_errno() . ": " . fbsql_error() . "<br />";
$conn = fbsql_query("SELECT * FROM nonexistenttable;");
echo fbsql_errno() . ": " . fbsql_error() . "<br />";
?>
```

See Also

- [fbsql_error\(\)](#)
- [fbsql_warnings\(\)](#)

fbsql_error

fbsql_error -- Returns the error message from previous operation

Description

string **fbsql_error** ([resource \$link_identifier])

Returns the error message from previous FrontBase operation.

Errors coming back from the fbsql database backend don't issue warnings. Instead, use [fbsql_error\(\)](#) to retrieve the error text. Note that this function only returns the error code from the most recently executed fbsql function (not including [fbsql_error\(\)](#) and [fbsql_errno\(\)](#)), so if you want to use it, make sure you check the value before calling another fbsql function.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns the error text from the last fbsql function, or "" (the empty string) if no error occurred.

Examples

Example #9 - [fbsql_error\(\)](#) Example

```
<?php
fbsql_connect("marliesle");
echo fbsql_errno() . ": " . fbsql_error() . "<br />";
fbsql_select_db("nonexistentdb");
echo fbsql_errno() . ": " . fbsql_error() . "<br />";
$conn = fbsql_query("SELECT * FROM nonexistenttable;");
echo fbsql_errno() . ": " . fbsql_error() . "<br />";
?>
```

See Also

- [fbsql_errno\(\)](#)
- [fbsql_warnings\(\)](#)

fbsql_fetch_array

fbsql_fetch_array -- Fetch a result row as an associative array, a numeric array, or both

Description

array **fbsql_fetch_array** (resource \$result [, int \$result_type])

[fbsql_fetch_array\(\)](#) is a combination of [fbsql_fetch_row\(\)](#) and [fbsql_fetch_assoc\(\)](#).

An important thing to note is that using [fbsql_fetch_array\(\)](#) is NOT significantly slower than using [fbsql_fetch_row\(\)](#), while it provides a significant added value.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

result_type

A constant and can take the following values: **FBSQL_ASSOC**, **FBSQL_NUM**, or **FBSQL_BOTH**. When using **FBSQL_BOTH**, in addition to storing the data in the numeric indices of the result array, it also stores the data in associative indices, using the field names as keys.

Return Values

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

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 must the numeric index of the column or make an alias for the column.

```
select t1.fl as foo t2.fl as bar from t1, t2
```

Examples

Example #10 - [fbsql_fetch_array\(\)](#) example

```
<?php
fbsql_connect($host, $user, $password);
$result = fbsql_db_query("database", "select user_id, fullname from table");
while ($row = fbsql_fetch_array($result)) {
    echo "user_id: " . $row["user_id"] . "<br />\n";
    echo "user_id: " . $row[0] . "<br />\n";
}
```

```
    echo "fullname: " . $row["fullname"] . "<br />\n";  
    echo "fullname: " . $row[1] . "<br />\n";  
}  
fbsql_free_result($result);  
?>
```

See Also

- [fbsql_fetch_row\(\)](#)
- [fbsql_fetch_assoc\(\)](#)
- [fbsql_fetch_object\(\)](#)

fbsql_fetch_assoc

fbsql_fetch_assoc -- Fetch a result row as an associative array

Description

array **fbsql_fetch_assoc** (resource \$result)

Calling [fbsql_fetch_assoc\(\)](#) is equivalent to calling [fbsql_fetch_array\(\)](#) with **FBSQL_ASSOC** as second parameter. It only returns an associative array.

This is the way [fbsql_fetch_array\(\)](#) originally worked. If you need the numeric indices as well as the associative, use [fbsql_fetch_array\(\)](#).

An important thing to note is that using [fbsql_fetch_assoc\(\)](#) is NOT significantly slower than using [fbsql_fetch_row\(\)](#), while it provides a significant added value.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

Return Values

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

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 must use [fbsql_fetch_array\(\)](#) and have it return the numeric indices as well.

Examples

Example #11 - [fbsql_fetch_assoc\(\)](#) example

```
<?php
fbsql_connect($host, $user, $password);
$result = fbsql_db_query("database", "select * from table");
while ($row = fbsql_fetch_assoc($result)) {
    echo $row["user_id"];
    echo $row["fullname"];
}
fbsql_free_result($result);
?>
```

See Also

- [fbsql_fetch_row\(\)](#)
- [fbsql_fetch_array\(\)](#)
- [fbsql_fetch_object\(\)](#)

fbsql_fetch_field

fbsql_fetch_field -- Get column information from a result and return as an object

Description

object **fbsql_fetch_field** (resource \$result [, int \$field_offset])

Used in order to obtain information about fields in a certain query result.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

field_offset

The numerical offset of the field. The field index starts at 0. If not specified, the next field that wasn't yet retrieved by [fbsql_fetch_field\(\)](#) is retrieved.

Return Values

Returns an object containing field information, or **FALSE** on errors.

The properties of the object are:

- name - column name
- table - name of the table the column belongs to
- max_length - maximum length of the column
- not_null - 1 if the column cannot be **NULL**
- type - the type of the column

Examples

Example #12 - [fbsql_fetch_field\(\)](#) example

```
<?php
fbsql_connect($host, $user, $password)
    or die("Could not connect");
$result = fbsql_db_query("database", "select * from table")
    or die("Query failed");
# get column metadata
$i = 0;
```

```
while ($i < fbsql_num_fields($result)) {
    echo "Information for column $i:<br />\n";
    $meta = fbsql_fetch_field($result);
    if (!$meta) {
        echo "No information available<br />\n";
    }
    echo "<pre>
max_length:    $meta->max_length
name:          $meta->name
not_null:      $meta->not_null
table:         $meta->table
type:          $meta->type
</pre>";
    $i++;
}
fbsql_free_result($result);
?>
```

See Also

- [fbsql_field_seek\(\)](#)

fbsql_fetch_lengths

fbsql_fetch_lengths -- Get the length of each output in a result

Description

array **fbsql_fetch_lengths** (resource \$result)

Stores the lengths of each result column in the last row returned by [fbsql_fetch_row\(\)](#), [fbsql_fetch_array\(\)](#) and [fbsql_fetch_object\(\)](#) in an array.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

Return Values

Returns an array, starting at offset 0, that corresponds to the lengths of each field in the last row fetched by [fbsql_fetch_row\(\)](#), or **FALSE** on error.

See Also

- [fbsql_fetch_row\(\)](#)

fbsql_fetch_object

fbsql_fetch_object -- Fetch a result row as an object

Description

object **fbsql_fetch_object** (resource \$result)

[fbsql_fetch_object\(\)](#) is similar to [fbsql_fetch_array\(\)](#), with one difference - an object is returned, instead of an array. Indirectly, that means that you can only access the data by the field names, and not by their offsets (numbers are illegal property names).

Speed-wise, the function is identical to [fbsql_fetch_array\(\)](#), and almost as quick as [fbsql_fetch_row\(\)](#) (the difference is insignificant).

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

Return Values

Returns an object with properties that correspond to the fetched row, or **FALSE** if there are no more rows.

Examples

Example #13 - [fbsql_fetch_object\(\)](#) example

```
<?php
fbsql_connect($host, $user, $password);
$result = fbsql_db_query("database", "select * from table");
while ($row = fbsql_fetch_object($result)) {
    echo $row->user_id;
    echo $row->fullname;
}
fbsql_free_result($result);
?>
```

See Also

- [fbsql_fetch_array\(\)](#)

- [fbsql_fetch_row\(\)](#)
- [fbsql_fetch_assoc\(\)](#)

fbsql_fetch_row

fbsql_fetch_row -- Get a result row as an enumerated array

Description

array **fbsql_fetch_row** (resource \$result)

[fbsql_fetch_row\(\)](#) fetches one row of data from the result associated with the specified result identifier.

Subsequent call to [fbsql_fetch_row\(\)](#) would return the next row in the result set, or **FALSE** if there are no more rows.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

Return Values

Returns an array that corresponds to the fetched row where each result column is stored in an offset, starting at offset 0, or **FALSE** if there are no more rows.

See Also

- [fbsql_fetch_array\(\)](#)
- [fbsql_fetch_assoc\(\)](#)
- [fbsql_fetch_object\(\)](#)
- [fbsql_data_seek\(\)](#)
- [fbsql_fetch_lengths\(\)](#)
- [fbsql_result\(\)](#)

fbsql_field_flags

fbsql_field_flags -- Get the flags associated with the specified field in a result

Description

string **fbsql_field_flags** (resource \$result [, int \$field_offset])

Gets the flags associated with the specified field in a result.

Parameters

result

A result pointer returned by [fbsql_list_fields\(\)](#).

field_offset

The numerical offset of the field. The field index starts at 0.

Return Values

Returns the field flags of the specified field as a single word per flag separated by a single space, so that you can split the returned value using [explode\(\)](#).

fbsql_field_len

fbsql_field_len -- Returns the length of the specified field

Description

int **fbsql_field_len** (resource \$result [, int \$field_offset])

Returns the length of the specified field.

Parameters

result

A result pointer returned by [fbsql_list_fields\(\)](#).

field_offset

The numerical offset of the field. The field index starts at 0.

Return Values

Returns the length of the specified field.

fbsql_field_name

fbsql_field_name -- Get the name of the specified field in a result

Description

string **fbsql_field_name** (resource \$result [, int \$field_index])

Returns the name of the specified field index.

Parameters

result

A result pointer returned by [fbsql_list_fields\(\)](#).

field_index

The numerical offset of the field. The field index starts at 0.

Return Values

Returns the name as a string, or **FALSE** if the field doesn't exist.

Examples

Example #14 - [fbsql_field_name\(\)](#) example

```
<?php
// The users table consists of three fields:
//   user_id
//   username
//   password.

$res = fbsql_db_query("users", "select * from users", $link);

echo fbsql_field_name($res, 0) . "\n";
echo fbsql_field_name($res, 2);
?>
```

The above example will output:

```
user_id
password
```

See Also

- [fbsql_field_type\(\)](#)

fbsql_field_seek

fbsql_field_seek -- Set result pointer to a specified field offset

Description

bool **fbsql_field_seek** (resource \$result [, int \$field_offset])

Seeks to the specified field offset. If the next call to [fbsql_fetch_field\(\)](#) doesn't include a field offset, the field offset specified in [fbsql_field_seek\(\)](#) will be returned.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

field_offset

The numerical offset of the field. The field index starts at 0.

Return Values

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

See Also

- [fbsql_fetch_field\(\)](#)

fbsql_field_table

fbsql_field_table -- Get name of the table the specified field is in

Description

string **fbsql_field_table** (resource \$result [, int \$field_offset])

Returns the name of the table that the specified field is in.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

field_offset

The numerical offset of the field. The field index starts at 0.

Return Values

Returns the name of the table, as a string.

fbsql_field_type

fbsql_field_type -- Get the type of the specified field in a result

Description

string **fbsql_field_type** (resource \$result [, int \$field_offset])

[fbsql_field_type\(\)](#) is similar to the [fbsql_field_name\(\)](#) function, but the field type is returned instead.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

field_offset

The numerical offset of the field. The field index starts at 0.

Return Values

Returns the field type, as a string.

This can be one of *int*, *real*, *string*, *blob*, and others as detailed in the [» FrontBase documentation](#).

Examples

Example #15 - [fbsql_field_type\(\)](#) example

```
<?php

fbsql_connect("localhost", "_SYSTEM", "");
fbsql_select_db("wisconsin");
$result = fbsql_query("SELECT * FROM onek;");
$fields = fbsql_num_fields($result);
$rows   = fbsql_num_rows($result);
$i = 0;
$table = fbsql_field_table($result, $i);
echo "Your '" . $table . "' table has " . $fields . " fields and " . $rows .
" records <br />";
echo "The table has the following fields <br />";
while ($i < $fields) {
    $type = fbsql_field_type($result, $i);
    $name = fbsql_field_name($result, $i);
    $len  = fbsql_field_len($result, $i);
    $flags = fbsql_field_flags($result, $i);
```

```
        echo $type . " " . $name . " " . $len . " " . $flags . "<br />";  
        $i++;  
    }  
    fbsql_close();  
  
?>
```

See Also

- [fbsql_field_name\(\)](#)

fbsql_free_result

fbsql_free_result -- Free result memory

Description

bool **fbsql_free_result** (resource \$result)

Frees all memory associated with the given *result* identifier.

[fbsql_free_result\(\)](#) only needs to be called if you are concerned about how much memory is being used for queries that return large result sets. All associated result memory is automatically freed at the end of the script's execution.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

Return Values

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

fbsql_get_autostart_info

fbsql_get_autostart_info --

Description

array **fbsql_get_autostart_info** ([resource \$link_identifier])

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

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

fbsql_hostname

fbsql_hostname -- Get or set the host name used with a connection

Description

string **fbsql_hostname** (resource *\$link_identifier* [, string *\$host_name*])

Gets or sets the host name used with a connection.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

host_name

If provided, this will be the new connection host name.

Return Values

Returns the current host name used for the connection.

See Also

- [fbsql_username\(\)](#)
- [fbsql_password\(\)](#)

fbsql_insert_id

fbsql_insert_id -- Get the id generated from the previous INSERT operation

Description

int **fbsql_insert_id** ([resource \$link_identifier])

Gets the id generated from the previous INSERT operation which created a DEFAULT UNIQUE value.

Note
The value of the FrontBase SQL function fbsql_insert_id() always contains the most recently generated DEFAULT UNIQUE value, and is not reset between queries.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns the ID generated from the previous INSERT query, or 0 if the previous query does not generate an DEFAULT UNIQUE value.

If you need to save the value for later, be sure to call this function immediately after the query that generates the value.

See Also

- [fbsql_affected_rows\(\)](#)

fbsql_list_dbs

fbsql_list_dbs -- List databases available on a FrontBase server

Description

resource **fbsql_list_dbs** ([resource *\$link_identifier*])

Return a result pointer containing the databases available from the current fbsql daemon. Use the [fbsql_tablename\(\)](#) to traverse this result pointer.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns a result pointer or **FALSE** on error.

Examples

Example #16 - [fbsql_list_dbs\(\)](#) example

```
$link = fbsql_connect('localhost', 'myname', 'secret');
$db_list = fbsql_list_dbs($link);

while ($row = fbsql_fetch_object($db_list)) {
    echo $row->Database . "\n";
}
```

The above example will output something similar to:

```
database1
database2
database3
...
```

Note

The above code would just as easily work with [fbsql_fetch_row\(\)](#) or other similar functions.

See Also

- [fbsql_list_fields\(\)](#)
- [fbsql_list_tables\(\)](#)

fbsql_list_fields

fbsql_list_fields -- List FrontBase result fields

Description

resource **fbsql_list_fields** (string \$database_name, string \$table_name [, resource \$link_identifier])

Retrieves information about the given table.

Parameters

database_name

The database name.

table_name

The table name.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns a result pointer which can be used with the *fbsql_field_xxx* functions, or **FALSE** on error.

Errors/Exceptions

A string describing the error will be placed in *\$phperrormsg*, and unless the function was called as *@fbsql()* then this error string will also be printed out.

Examples

Example #17 - [fbsql_list_fields\(\)](#) example

```
<?php
$link = fbsql_connect('localhost', 'myname', 'secret');

$fields = fbsql_list_fields("database1", "table1", $link);
$columns = fbsql_num_fields($fields);
```

```
for ($i = 0; $i < $columns; $i++) {  
    echo fbsql_field_name($fields, $i) . "\n";  
}  
?>
```

The above example will output something similar to:

```
field1  
field2  
field3  
...
```

See Also

- [fbsql_field_len\(\)](#)
- [fbsql_field_name\(\)](#)
- [fbsql_field_type\(\)](#)
- [fbsql_field_flags\(\)](#)

fbsql_list_tables

fbsql_list_tables -- List tables in a FrontBase database

Description

resource **fbsql_list_tables** (string \$database [, resource \$link_identifier])

Returns a result pointer describing the *database*.

Parameters

database

The database name.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns a result pointer which can be used with the [fbsql_tablename\(\)](#) function to read the actual table names, or **FALSE** on error.

See Also

- [fbsql_list_fields\(\)](#)
- [fbsql_list_dbs\(\)](#)

fbsql_next_result

fbsql_next_result -- Move the internal result pointer to the next result

Description

bool **fbsql_next_result** (resource \$result)

When sending more than one SQL statement to the server or executing a stored procedure with multiple results will cause the server to return multiple result sets. This function will test for additional results available from the server. If an additional result set exists it will free the existing result set and prepare to fetch the words from the new result set.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

Return Values

Returns **TRUE** if an additional result set was available or **FALSE** otherwise.

Examples

Example #18 - [fbsql_next_result\(\)](#) example

```
<?php
$link = fbsql_connect("localhost", "_SYSTEM", "secret");
fbsql_select_db("MyDB", $link);
$SQL = "Select * from table1; select * from table2;";
$rs = fbsql_query($SQL, $link);
do {
    while ($row = fbsql_fetch_row($rs)) {
    }
} while (fbsql_next_result($rs));
fbsql_free_result($rs);
fbsql_close($link);
?>
```


fbsql_num_fields

fbsql_num_fields -- Get number of fields in result

Description

int **fbsql_num_fields** (resource *\$result*)

Returns the number of fields in the given *result* set.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

Return Values

Returns the number of fields, as an integer.

See Also

- [fbsql_db_query\(\)](#)
- [fbsql_query\(\)](#)
- [fbsql_fetch_field\(\)](#)
- [fbsql_num_rows\(\)](#)

fbsql_num_rows

fbsql_num_rows -- Get number of rows in result

Description

int **fbsql_num_rows** (resource \$result)

Gets the number of rows in the given *result* set.

This function is only valid for SELECT statements. To retrieve the number of rows returned from a INSERT, UPDATE or DELETE query, use [fbsql_affected_rows\(\)](#).

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

Return Values

Returns the number of rows returned by the last SELECT statement.

Examples

Example #19 - [fbsql_num_rows\(\)](#) example

```
<?php

$link = fbsql_connect("localhost", "username", "password");
fbsql_select_db("database", $link);

$result = fbsql_query("SELECT * FROM table1;", $link);
$num_rows = fbsql_num_rows($result);

echo "$num_rows Rows\n";

?>
```

See Also

- [fbsql_affected_rows\(\)](#)
- [fbsql_connect\(\)](#)

- [fbsql_select_db\(\)](#)
- [fbsql_query\(\)](#)

fbsql_password

fbsql_password -- Get or set the user password used with a connection

Description

string **fbsql_password** (resource \$link_identifier [, string \$password])

Get or set the user password used with a connection.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

password

If provided, this will be the new connection password.

Return Values

Returns the current password used for the connection.

See Also

- [fbsql_username\(\)](#)
- [fbsql_hostname\(\)](#)

fbsql_pconnect

fbsql_pconnect -- Open a persistent connection to a FrontBase Server

Description

```
resource fbsql_pconnect ( [ string $hostname [, string $username [, string $password ] ] ] )
```

Establishes a persistent connection to a FrontBase server.

To set the server port number, use [fbsql_select_db\(\)](#).

[fbsql_pconnect\(\)](#) acts very much like [fbsql_connect\(\)](#) with two major differences:

First, when connecting, the function would first try to find a (persistent) link that's already open with the same host, username and password. If one is found, an identifier for it will be returned instead of opening a new connection.

Second, the connection to the SQL server will not be closed when the execution of the script ends. Instead, the link will remain open for future use.

This type of links is therefore called 'persistent'.

Parameters

hostname

The server host name. Defaults to *localhost*.

username

The user name for the connection. Defaults to *_SYSTEM*.

password

The password for the connection. Defaults to the empty string.

Return Values

Returns a positive FrontBase persistent link identifier on success, or **FALSE** on error.

See Also

- [fbsql_connect\(\)](#)

fbsql_query

fbsql_query -- Send a FrontBase query

Description

resource **fbsql_query** (string \$query [, resource \$link_identifier [, int \$batch_size]])

Sends a *query* to the currently active database on the server.

If the query succeeds, you can call [fbsql_num_rows\(\)](#) to find out how many rows were returned for a SELECT statement or [fbsql_affected_rows\(\)](#) to find out how many rows were affected by a DELETE, INSERT, REPLACE, or UPDATE statement.

Parameters

query

The SQL query to be executed.

Note
The query string shall always end with a semicolon.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

batch_size

Return Values

[fbsql_query\(\)](#) returns **TRUE** (non-zero) or **FALSE** to indicate whether or not the query succeeded. A return value of **TRUE** means that the query was legal and could be executed by the server. It does not indicate anything about the number of rows affected or returned. It is perfectly possible for a query to succeed but affect no rows or return no rows.

For SELECT statements, [fbsql_query\(\)](#) returns a new result identifier that you can pass to [fbsql_result\(\)](#).

[fbsql_query\(\)](#) will also fail and return **FALSE** if you don't have permission to access the table(s) referenced by the query.

Examples

The following query is syntactically invalid, so [fbsql_query\(\)](#) fails and returns **FALSE**:

Example #20 - [fbsql_query\(\)](#) example

```
<?php
$result = fbsql_query("SELECT * WHERE 1=1")
    or die ("Invalid query");
?>
```

The following query is semantically invalid if *my_col* is not a column in the table *my_tbl*, so [fbsql_query\(\)](#) fails and returns **FALSE**:

Example #21 - [fbsql_query\(\)](#) example

```
<?php
$result = fbsql_query ("SELECT my_col FROM my_tbl;")
    or die ("Invalid query");
?>
```

See Also

- [fbsql_affected_rows\(\)](#)
- [fbsql_db_query\(\)](#)
- [fbsql_free_result\(\)](#)
- [fbsql_result\(\)](#)
- [fbsql_select_db\(\)](#)
- [fbsql_connect\(\)](#)

fbsql_read_blob

fbsql_read_blob -- Read a BLOB from the database

Description

string **fbsql_read_blob** (string \$blob_handle [, resource \$link_identifier])

Reads BLOB data from the database.

If a select statement contains BLOB and/or CLOB columns FrontBase will return the data directly when data is fetched. This default behavior can be changed with [fbsql_set_lob_mode\(\)](#) so the fetch functions will return handles to BLOB and CLOB data. If a handle is fetched a user must call [fbsql_read_blob\(\)](#) to get the actual BLOB data from the database.

Parameters

blob_handle

A BLOB handle, returned by [fbsql_create_blob\(\)](#).

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns a string containing the specified BLOB data.

Examples

Example #22 - [fbsql_read_blob\(\)](#) example

```
<?php
$link = fbsql_pconnect("localhost", "_SYSTEM", "secret")
    or die("Could not connect");
$sql = "SELECT BLOB_COLUMN FROM BLOB_TABLE;";
$rs = fbsql_query($sql, $link);
$row_data = fbsql_fetch_row($rs);
// $row_data[0] will now contain the blob data for the first row
fbsql_free_result($rs);

$rs = fbsql_query($sql, $link);
fbsql_set_lob_mode($rs, FBSQL_LOB_HANDLE);
```



```
$row_data = fbsql_fetch_row($rs);  
// $row_data[0] will now contain a handle to the BLOB data in the first row  
$blob_data = fbsql_read_blob($row_data[0]);  
fbsql_free_result($rs);  
  
?>
```

See Also

- [fbsql_create_blob\(\)](#)
- [fbsql_read_clob\(\)](#)
- [fbsql_set_lob_mode\(\)](#)

fbsql_read_clob

fbsql_read_clob -- Read a CLOB from the database

Description

string **fbsql_read_clob** (string \$clob_handle [, resource \$link_identifier])

Reads CLOB data from the database.

If a select statement contains BLOB and/or CLOB columns FrontBase will return the data directly when data is fetched. This default behavior can be changed with [fbsql_set_lob_mode\(\)](#) so the fetch functions will return handles to BLOB and CLOB data. If a handle is fetched a user must call [fbsql_read_clob\(\)](#) to get the actual CLOB data from the database.

Parameters

clob_handle

A CLOB handle, returned by [fbsql_create_clob\(\)](#).

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

Returns a string containing the specified CLOB data.

Examples

Example #23 - [fbsql_read_clob\(\)](#) example

```
<?php
$link = fbsql_pconnect("localhost", "_SYSTEM", "secret")
    or die("Could not connect");
$sql = "SELECT CLOB_COLUMN FROM CLOB_TABLE;";
$rs = fbsql_query($sql, $link);
$row_data = fbsql_fetch_row($rs);
// $row_data[0] will now contain the clob data for the first row
fbsql_free_result($rs);

$rs = fbsql_query($sql, $link);
fbsql_set_lob_mode($rs, FBSQL_LOB_HANDLE);
```

```
$row_data = fbsql_fetch_row($rs);  
// $row_data[0] will now contain a handle to the CLOB data in the first row  
$clob_data = fbsql_read_clob($row_data[0]);  
fbsql_free_result($rs);  
  
?>
```

See Also

- [fbsql_create_clob\(\)](#)
- [fbsql_read_blob\(\)](#)
- [fbsql_set_lob_mode\(\)](#)

fbsql_result

fbsql_result -- Get result data

Description

mixed **fbsql_result** (resource \$result [, int \$row [, **mixed** \$field]])

Returns the contents of one cell from a FrontBase *result* set.

When working on large result sets, you should consider using one of the functions that fetch an entire row (specified below). As these functions return the contents of multiple cells in one function call, they're MUCH quicker than [fbsql_result\(\)](#).

Calls to [fbsql_result\(\)](#) should not be mixed with calls to other functions that deal with the result set.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

row

field

Can be the field's offset, or the field's name, or the field's table dot field's name (tablename.fieldname). If the column name has been aliased ('select foo as bar from...'), use the alias instead of the column name.

Note
Specifying a numeric offset for the field argument is much quicker than specifying a fieldname or tablename.fieldname argument.

Return Values

See Also

Recommended high-performance alternatives:

- [fbsql_fetch_row\(\)](#)

- [fbsql_fetch_array\(\)](#)
- [fbsql_fetch_assoc\(\)](#)
- [fbsql_fetch_object\(\)](#)

fbsql_rollback

fbsql_rollback -- Rollback a transaction to the database

Description

bool **fbsql_rollback** ([resource \$link_identifier])

Ends the current transaction by rolling back all statements issued since last commit.

This command is only needed if autocommit is set to false.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

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

See Also

- [fbsql_autocommit\(\)](#)
- [fbsql_commit\(\)](#)

fbsql_rows_fetched

fbsql_rows_fetched -- Get the number of rows affected by the last statement

Description

```
int fbsql_rows_fetched ( resource $result )
```

Gets the number of rows affected by the last statement.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

Return Values

Returns the number of rows, as an integer.

fbsql_select_db

fbsql_select_db -- Select a FrontBase database

Description

```
bool fbsql_select_db ( [ string $database_name [, resource $link_identifier ] ] )
```

Sets the current active database on the given link identifier.

The client contacts FBExec to obtain the port number to use for the connection to the database. If the database name is a number the system will use that as a port number and it will not ask FBExec for the port number. The FrontBase server can be started as FRontBase -FBExec=No -port=<port number> <database name>.

Every subsequent call to [fbsql_query\(\)](#) will be made on the active database.

Parameters

database_name

The name of the database to be selected. If the database is protected with a database password, the you must call [fbsql_database_password\(\)](#) before selecting the database.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

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

See Also

- [fbsql_connect\(\)](#)
- [fbsql_pconnect\(\)](#)
- [fbsql_database_password\(\)](#)
- [fbsql_query\(\)](#)

fbsql_set_charset

fbsql_set_charset -- Change input/output character set

Description

```
void fbsql_set_charset ( resource $link_identifier, int $charset [, int $
in_out_both ] )
```

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

Return Values

No value is returned.

fbsql_set_lob_mode

fbsql_set_lob_mode -- Set the LOB retrieve mode for a FrontBase result set

Description

bool **fbsql_set_lob_mode** (resource \$result, int \$lob_mode)

Sets the mode for retrieving LOB data from the database.

When BLOB and CLOB data is retrieved in FrontBase it can be retrieved direct or indirect. Direct retrieved LOB data will always be fetched no matter the setting of the lob mode. If the LOB data is less than 512 bytes it will always be retrieved directly.

Parameters

result

A result identifier returned by [fbsql_query\(\)](#) or [fbsql_db_query\(\)](#).

lob_mode

Can be one of:

- **FBSQL_LOB_DIRECT** - LOB data is retrieved directly. When data is fetched from the database with [fbsql_fetch_row\(\)](#), and other fetch functions, all CLOB and BLOB columns will be returned as ordinary columns. This is the default value on a new FrontBase result.
- **FBSQL_LOB_HANDLE** - LOB data is retrieved as handles to the data. When data is fetched from the database with [fbsql_fetch_row\(\)](#), and other fetch functions, LOB data will be returned as a handle to the data if the data is stored indirect or the data if it is stored direct. If a handle is returned it will be a 27 byte string formatted as '@00000000000000000000000000000000'.

Return Values

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

See Also

- [fbsql_create_blob\(\)](#)
- [fbsql_create_clob\(\)](#)
- [fbsql_read_blob\(\)](#)
- [fbsql_read_clob\(\)](#)

fbsql_set_password

fbsql_set_password -- Change the password for a given user

Description

```
bool fbsql_set_password ( resource $link_identifier, string $user, string $password,  
string $old_password )
```

Changes the password for the given *user*.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

user

The user name.

password

The new password to be set.

old_password

The old password to be replaced.

Return Values

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

fbsql_set_transaction

fbsql_set_transaction -- Set the transaction locking and isolation

Description

void fbsql_set_transaction (resource *\$link_identifier*, int *\$locking*, int *\$isolation*)

Sets the transaction *locking* and *isolation*.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

locking

The type of locking to be set. It can be one of the following constants:

FBSQL_LOCK_DEFERRED, **FBSQL_LOCK_OPTIMISTIC**, or
FBSQL_LOCK_PESSIMISTIC.

isolation

The type of isolation to be set. It can be one of the following constants:

FBSQL_ISO_READ_UNCOMMITTED, **FBSQL_ISO_READ_COMMITTED**,
FBSQL_ISO_REPEATABLE_READ, **FBSQL_ISO_SERIALIZABLE**, or
FBSQL_ISO_VERSIONED.

Return Values

No value is returned.

fbsql_start_db

fbsql_start_db -- Start a database on local or remote server

Description

```
bool fbsql_start_db ( string $database_name [, resource $link_identifier [, string $database_options ] ] )
```

Start a database on local or remote server.

Parameters

database_name

The database name.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

database_options

Return Values

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

See Also

- [fbsql_db_status\(\)](#)
- [fbsql_stop_db\(\)](#)

fbsql_stop_db

fbsql_stop_db -- Stop a database on local or remote server

Description

bool **fbsql_stop_db** (string \$database_name [, resource \$link_identifier])

Stops a database on local or remote server.

Parameters

database_name

The database name.

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

Return Values

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

See Also

- [fbsql_db_status\(\)](#)
- [fbsql_start_db\(\)](#)

fbsql_table_name

fbsql_table_name -- Get table name of field

Description

string **fbsql_table_name** (resource \$result, int \$index)

[fbsql_table_name\(\)](#) gets the name of the current table in the given *result* set.

The [fbsql_num_rows\(\)](#) function may be used to determine the number of tables in the result pointer.

Parameters

result

A result pointer returned by [fbsql_list_tables\(\)](#).

index

Integer index for the current table.

Return Values

Returns the name of the table, as a string.

Examples

Example #24 - [fbsql_table_name\(\)](#) example

```
<?php
fbsql_connect("localhost", "_SYSTEM", "");
$result = fbsql_list_tables("wisconsin");
$i = 0;
while ($i < fbsql_num_rows($result)) {
    $tb_names[$i] = fbsql_table_name($result, $i);
    echo $tb_names[$i] . "<br />";
    $i++;
}
?>
```

fbsql_tablename

fbsql_tablename -- Alias of of [fbsql_table_name\(\)](#)

Description

This function is an alias of: [fbsql_table_name\(\)](#).

fbsql_username

fbsql_username -- Get or set the username for the connection

Description

string **fbsql_username** (resource \$link_identifier [, string \$username])

Get or set the username used for the connection.

Parameters

link_identifier

A FrontBase link identifier returned by [fbsql_connect\(\)](#) or [fbsql_pconnect\(\)](#). If optional and not specified, the function will try to find an open link to the FrontBase server and if no such link is found it will try to create one as if [fbsql_connect\(\)](#) was called with no arguments

username

If provided, this is the new username to set.

Return Values

Returns the current username used with the connection, as a string.

See Also

- [fbsql_password\(\)](#)
- [fbsql_hostname\(\)](#)

fbsql_warnings

fbsql_warnings -- Enable or disable FrontBase warnings

Description

bool **fbsql_warnings** ([bool *\$OnOff*])

Enables or disables FrontBase warnings.

Parameters

OnOff

Whether to enable warnings or no.

Return Values

Returns **TRUE** if warnings is turned on, **FALSE** otherwise.

See Also

- [fbsql_errno\(\)](#)
- [fbsql_error\(\)](#)