* [Welcome to CodeIgniter](http://docs.google.com/general/welcome.html)
* [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/installation/downloads.html)
  + [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/installation/upgrading.html)
  + [Troubleshooting](http://docs.google.com/installation/troubleshooting.html)
* [CodeIgniter Overview](http://docs.google.com/overview/index.html)
  + [Getting Started](http://docs.google.com/overview/getting_started.html)
  + [CodeIgniter at a Glance](http://docs.google.com/overview/at_a_glance.html)
  + [Supported Features](http://docs.google.com/overview/features.html)
  + [Application Flow Chart](http://docs.google.com/overview/appflow.html)
  + [Model-View-Controller](http://docs.google.com/overview/mvc.html)
  + [Architectural Goals](http://docs.google.com/overview/goals.html)
* [Tutorial](http://docs.google.com/tutorial/index.html)
  + [Static pages](http://docs.google.com/tutorial/static_pages.html)
  + [News section](http://docs.google.com/tutorial/news_section.html)
  + [Create news items](http://docs.google.com/tutorial/create_news_items.html)
  + [Conclusion](http://docs.google.com/tutorial/conclusion.html)
* [Contributing to CodeIgniter](http://docs.google.com/contributing/index.html)
  + [Writing CodeIgniter Documentation](http://docs.google.com/documentation/index.html)
  + [Developer’s Certificate of Origin 1.1](http://docs.google.com/DCO.html)
* [General Topics](http://docs.google.com/general/index.html)
  + [CodeIgniter URLs](http://docs.google.com/general/urls.html)
  + [Controllers](http://docs.google.com/general/controllers.html)
  + [Reserved Names](http://docs.google.com/general/reserved_names.html)
  + [Views](http://docs.google.com/general/views.html)
  + [Models](http://docs.google.com/general/models.html)
  + [Helpers](http://docs.google.com/general/helpers.html)
  + [Using CodeIgniter Libraries](http://docs.google.com/general/libraries.html)
  + [Creating Libraries](http://docs.google.com/general/creating_libraries.html)
  + [Using CodeIgniter Drivers](http://docs.google.com/general/drivers.html)
  + [Creating Drivers](http://docs.google.com/general/creating_drivers.html)
  + [Creating Core System Classes](http://docs.google.com/general/core_classes.html)
  + [Creating Ancillary Classes](http://docs.google.com/general/ancillary_classes.html)
  + [Hooks - Extending the Framework Core](http://docs.google.com/general/hooks.html)
  + [Auto-loading Resources](http://docs.google.com/general/autoloader.html)
  + [Common Functions](http://docs.google.com/general/common_functions.html)
  + [Compatibility Functions](http://docs.google.com/general/compatibility_functions.html)
  + [URI Routing](http://docs.google.com/general/routing.html)
  + [Error Handling](http://docs.google.com/general/errors.html)
  + [Caching](http://docs.google.com/general/caching.html)
  + [Profiling Your Application](http://docs.google.com/general/profiling.html)
  + [Running via the CLI](http://docs.google.com/general/cli.html)
  + [Managing your Applications](http://docs.google.com/general/managing_apps.html)
  + [Handling Multiple Environments](http://docs.google.com/general/environments.html)
  + [Alternate PHP Syntax for View Files](http://docs.google.com/general/alternative_php.html)
  + [Security](http://docs.google.com/general/security.html)
  + [PHP Style Guide](http://docs.google.com/general/styleguide.html)
* [Libraries](http://docs.google.com/libraries/index.html)
  + [Benchmarking Class](http://docs.google.com/libraries/benchmark.html)
  + [Caching Driver](http://docs.google.com/libraries/caching.html)
  + [Calendaring Class](http://docs.google.com/libraries/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/libraries/cart.html)
  + [Config Class](http://docs.google.com/libraries/config.html)
  + [Email Class](http://docs.google.com/libraries/email.html)
  + [Encrypt Class](http://docs.google.com/libraries/encrypt.html)
  + [Encryption Library](http://docs.google.com/libraries/encryption.html)
  + [File Uploading Class](http://docs.google.com/libraries/file_uploading.html)
  + [Form Validation](http://docs.google.com/libraries/form_validation.html)
  + [FTP Class](http://docs.google.com/libraries/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/libraries/image_lib.html)
  + [Input Class](http://docs.google.com/libraries/input.html)
  + [Javascript Class](http://docs.google.com/libraries/javascript.html)
  + [Language Class](http://docs.google.com/libraries/language.html)
  + [Loader Class](http://docs.google.com/libraries/loader.html)
  + [Migrations Class](http://docs.google.com/libraries/migration.html)
  + [Output Class](http://docs.google.com/libraries/output.html)
  + [Pagination Class](http://docs.google.com/libraries/pagination.html)
  + [Template Parser Class](http://docs.google.com/libraries/parser.html)
  + [Security Class](http://docs.google.com/libraries/security.html)
  + [Session Library](http://docs.google.com/libraries/sessions.html)
  + [HTML Table Class](http://docs.google.com/libraries/table.html)
  + [Trackback Class](http://docs.google.com/libraries/trackback.html)
  + [Typography Class](http://docs.google.com/libraries/typography.html)
  + [Unit Testing Class](http://docs.google.com/libraries/unit_testing.html)
  + [URI Class](http://docs.google.com/libraries/uri.html)
  + [User Agent Class](http://docs.google.com/libraries/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/libraries/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/libraries/zip.html)
* [Database Reference](http://docs.google.com/index.html)
  + [Quick Start: Usage Examples](http://docs.google.com/examples.html)
  + [Database Configuration](http://docs.google.com/configuration.html)
  + [Connecting to a Database](http://docs.google.com/connecting.html)
  + [Running Queries](http://docs.google.com/queries.html)
  + [Generating Query Results](http://docs.google.com/results.html)
  + [Query Helper Functions](http://docs.google.com/helpers.html)
  + [Query Builder Class](http://docs.google.com/query_builder.html)
  + [Transactions](http://docs.google.com/transactions.html)
  + [Getting MetaData](http://docs.google.com/metadata.html)
  + [Custom Function Calls](http://docs.google.com/call_function.html)
  + [Query Caching](http://docs.google.com/caching.html)
  + [Database Manipulation with Database Forge](http://docs.google.com/forge.html)
  + Database Utilities Class
  + [Database Driver Reference](http://docs.google.com/db_driver_reference.html)
* [Helpers](http://docs.google.com/helpers/index.html)
  + [Array Helper](http://docs.google.com/helpers/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/helpers/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/helpers/cookie_helper.html)
  + [Date Helper](http://docs.google.com/helpers/date_helper.html)
  + [Directory Helper](http://docs.google.com/helpers/directory_helper.html)
  + [Download Helper](http://docs.google.com/helpers/download_helper.html)
  + [Email Helper](http://docs.google.com/helpers/email_helper.html)
  + [File Helper](http://docs.google.com/helpers/file_helper.html)
  + [Form Helper](http://docs.google.com/helpers/form_helper.html)
  + [HTML Helper](http://docs.google.com/helpers/html_helper.html)
  + [Inflector Helper](http://docs.google.com/helpers/inflector_helper.html)
  + [Language Helper](http://docs.google.com/helpers/language_helper.html)
  + [Number Helper](http://docs.google.com/helpers/number_helper.html)
  + [Path Helper](http://docs.google.com/helpers/path_helper.html)
  + [Security Helper](http://docs.google.com/helpers/security_helper.html)
  + [Smiley Helper](http://docs.google.com/helpers/smiley_helper.html)
  + [String Helper](http://docs.google.com/helpers/string_helper.html)
  + [Text Helper](http://docs.google.com/helpers/text_helper.html)
  + [Typography Helper](http://docs.google.com/helpers/typography_helper.html)
  + [URL Helper](http://docs.google.com/helpers/url_helper.html)
  + [XML Helper](http://docs.google.com/helpers/xml_helper.html)

Toggle Table of Contents

[CodeIgniter](http://docs.google.com/index.html)

* [Welcome to CodeIgniter](http://docs.google.com/general/welcome.html)
* [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/installation/downloads.html)
  + [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/installation/upgrading.html)
  + [Troubleshooting](http://docs.google.com/installation/troubleshooting.html)
* [CodeIgniter Overview](http://docs.google.com/overview/index.html)
  + [Getting Started](http://docs.google.com/overview/getting_started.html)
  + [CodeIgniter at a Glance](http://docs.google.com/overview/at_a_glance.html)
  + [Supported Features](http://docs.google.com/overview/features.html)
  + [Application Flow Chart](http://docs.google.com/overview/appflow.html)
  + [Model-View-Controller](http://docs.google.com/overview/mvc.html)
  + [Architectural Goals](http://docs.google.com/overview/goals.html)
* [Tutorial](http://docs.google.com/tutorial/index.html)
  + [Static pages](http://docs.google.com/tutorial/static_pages.html)
  + [News section](http://docs.google.com/tutorial/news_section.html)
  + [Create news items](http://docs.google.com/tutorial/create_news_items.html)
  + [Conclusion](http://docs.google.com/tutorial/conclusion.html)
* [Contributing to CodeIgniter](http://docs.google.com/contributing/index.html)
  + [Writing CodeIgniter Documentation](http://docs.google.com/documentation/index.html)
  + [Developer’s Certificate of Origin 1.1](http://docs.google.com/DCO.html)
* [General Topics](http://docs.google.com/general/index.html)
  + [CodeIgniter URLs](http://docs.google.com/general/urls.html)
  + [Controllers](http://docs.google.com/general/controllers.html)
  + [Reserved Names](http://docs.google.com/general/reserved_names.html)
  + [Views](http://docs.google.com/general/views.html)
  + [Models](http://docs.google.com/general/models.html)
  + [Helpers](http://docs.google.com/general/helpers.html)
  + [Using CodeIgniter Libraries](http://docs.google.com/general/libraries.html)
  + [Creating Libraries](http://docs.google.com/general/creating_libraries.html)
  + [Using CodeIgniter Drivers](http://docs.google.com/general/drivers.html)
  + [Creating Drivers](http://docs.google.com/general/creating_drivers.html)
  + [Creating Core System Classes](http://docs.google.com/general/core_classes.html)
  + [Creating Ancillary Classes](http://docs.google.com/general/ancillary_classes.html)
  + [Hooks - Extending the Framework Core](http://docs.google.com/general/hooks.html)
  + [Auto-loading Resources](http://docs.google.com/general/autoloader.html)
  + [Common Functions](http://docs.google.com/general/common_functions.html)
  + [Compatibility Functions](http://docs.google.com/general/compatibility_functions.html)
  + [URI Routing](http://docs.google.com/general/routing.html)
  + [Error Handling](http://docs.google.com/general/errors.html)
  + [Caching](http://docs.google.com/general/caching.html)
  + [Profiling Your Application](http://docs.google.com/general/profiling.html)
  + [Running via the CLI](http://docs.google.com/general/cli.html)
  + [Managing your Applications](http://docs.google.com/general/managing_apps.html)
  + [Handling Multiple Environments](http://docs.google.com/general/environments.html)
  + [Alternate PHP Syntax for View Files](http://docs.google.com/general/alternative_php.html)
  + [Security](http://docs.google.com/general/security.html)
  + [PHP Style Guide](http://docs.google.com/general/styleguide.html)
* [Libraries](http://docs.google.com/libraries/index.html)
  + [Benchmarking Class](http://docs.google.com/libraries/benchmark.html)
  + [Caching Driver](http://docs.google.com/libraries/caching.html)
  + [Calendaring Class](http://docs.google.com/libraries/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/libraries/cart.html)
  + [Config Class](http://docs.google.com/libraries/config.html)
  + [Email Class](http://docs.google.com/libraries/email.html)
  + [Encrypt Class](http://docs.google.com/libraries/encrypt.html)
  + [Encryption Library](http://docs.google.com/libraries/encryption.html)
  + [File Uploading Class](http://docs.google.com/libraries/file_uploading.html)
  + [Form Validation](http://docs.google.com/libraries/form_validation.html)
  + [FTP Class](http://docs.google.com/libraries/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/libraries/image_lib.html)
  + [Input Class](http://docs.google.com/libraries/input.html)
  + [Javascript Class](http://docs.google.com/libraries/javascript.html)
  + [Language Class](http://docs.google.com/libraries/language.html)
  + [Loader Class](http://docs.google.com/libraries/loader.html)
  + [Migrations Class](http://docs.google.com/libraries/migration.html)
  + [Output Class](http://docs.google.com/libraries/output.html)
  + [Pagination Class](http://docs.google.com/libraries/pagination.html)
  + [Template Parser Class](http://docs.google.com/libraries/parser.html)
  + [Security Class](http://docs.google.com/libraries/security.html)
  + [Session Library](http://docs.google.com/libraries/sessions.html)
  + [HTML Table Class](http://docs.google.com/libraries/table.html)
  + [Trackback Class](http://docs.google.com/libraries/trackback.html)
  + [Typography Class](http://docs.google.com/libraries/typography.html)
  + [Unit Testing Class](http://docs.google.com/libraries/unit_testing.html)
  + [URI Class](http://docs.google.com/libraries/uri.html)
  + [User Agent Class](http://docs.google.com/libraries/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/libraries/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/libraries/zip.html)
* [Database Reference](http://docs.google.com/index.html)
  + [Quick Start: Usage Examples](http://docs.google.com/examples.html)
  + [Database Configuration](http://docs.google.com/configuration.html)
  + [Connecting to a Database](http://docs.google.com/connecting.html)
  + [Running Queries](http://docs.google.com/queries.html)
  + [Generating Query Results](http://docs.google.com/results.html)
  + [Query Helper Functions](http://docs.google.com/helpers.html)
  + [Query Builder Class](http://docs.google.com/query_builder.html)
  + [Transactions](http://docs.google.com/transactions.html)
  + [Getting MetaData](http://docs.google.com/metadata.html)
  + [Custom Function Calls](http://docs.google.com/call_function.html)
  + [Query Caching](http://docs.google.com/caching.html)
  + [Database Manipulation with Database Forge](http://docs.google.com/forge.html)
  + Database Utilities Class
  + [Database Driver Reference](http://docs.google.com/db_driver_reference.html)
* [Helpers](http://docs.google.com/helpers/index.html)
  + [Array Helper](http://docs.google.com/helpers/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/helpers/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/helpers/cookie_helper.html)
  + [Date Helper](http://docs.google.com/helpers/date_helper.html)
  + [Directory Helper](http://docs.google.com/helpers/directory_helper.html)
  + [Download Helper](http://docs.google.com/helpers/download_helper.html)
  + [Email Helper](http://docs.google.com/helpers/email_helper.html)
  + [File Helper](http://docs.google.com/helpers/file_helper.html)
  + [Form Helper](http://docs.google.com/helpers/form_helper.html)
  + [HTML Helper](http://docs.google.com/helpers/html_helper.html)
  + [Inflector Helper](http://docs.google.com/helpers/inflector_helper.html)
  + [Language Helper](http://docs.google.com/helpers/language_helper.html)
  + [Number Helper](http://docs.google.com/helpers/number_helper.html)
  + [Path Helper](http://docs.google.com/helpers/path_helper.html)
  + [Security Helper](http://docs.google.com/helpers/security_helper.html)
  + [Smiley Helper](http://docs.google.com/helpers/smiley_helper.html)
  + [String Helper](http://docs.google.com/helpers/string_helper.html)
  + [Text Helper](http://docs.google.com/helpers/text_helper.html)
  + [Typography Helper](http://docs.google.com/helpers/typography_helper.html)
  + [URL Helper](http://docs.google.com/helpers/url_helper.html)
  + [XML Helper](http://docs.google.com/helpers/xml_helper.html)

[CodeIgniter](http://docs.google.com/index.html)

* [Docs](http://docs.google.com/index.html) »
* [Database Reference](http://docs.google.com/index.html) »
* Database Utility Class

classic layout

Database Utility Class[**¶**](#gjdgxs)

The Database Utility Class contains methods that help you manage your database.

* [Initializing the Utility Class](#30j0zll)
* [Using the Database Utilities](#1fob9te)
  + [Retrieve list of database names](#3znysh7)
  + [Determine If a Database Exists](#2et92p0)
  + [Optimize a Table](#tyjcwt)
  + [Repair a Table](#3dy6vkm)
  + [Optimize a Database](#1t3h5sf)
  + [Export a Query Result as a CSV File](#4d34og8)
  + [Export a Query Result as an XML Document](#2s8eyo1)
* [Backup Your Database](#17dp8vu)
  + [Database Backup Notes](#3rdcrjn)
  + [Usage Example](#26in1rg)
  + [Setting Backup Preferences](#lnxbz9)
  + [Description of Backup Preferences](#35nkun2)
* [Class Reference](#1ksv4uv)

[**Initializing the Utility Class**](#44sinio)[**¶**](#30j0zll)

Important

In order to initialize the Utility class, your database driver must already be running, since the utilities class relies on it.

Load the Utility Class as follows:

$this->load->dbutil();

You can also pass another database object to the DB Utility loader, in case the database you want to manage isn’t the default one:

$this->myutil = $this->load->dbutil($this->other\_db, TRUE);

In the above example, we’re passing a custom database object as the first parameter and then tell it to return the dbutil object, instead of assigning it directly to $this->dbutil.

Note

Both of the parameters can be used individually, just pass an empty value as the first one if you wish to skip it.

Once initialized you will access the methods using the $this->dbutil object:

$this->dbutil->some\_method();

[**Using the Database Utilities**](#2jxsxqh)[**¶**](#1fob9te)

[**Retrieve list of database names**](#z337ya)[**¶**](#3znysh7)

Returns an array of database names:

$dbs = $this->dbutil->list\_databases();  
  
foreach ($dbs as $db)  
{  
 echo $db;  
}

[**Determine If a Database Exists**](#3j2qqm3)[**¶**](#2et92p0)

Sometimes it’s helpful to know whether a particular database exists. Returns a boolean TRUE/FALSE. Usage example:

if ($this->dbutil->database\_exists('database\_name'))  
{  
 // some code...  
}

Note

Replace *database\_name* with the name of the database you are looking for. This method is case sensitive.

[**Optimize a Table**](#1y810tw)[**¶**](#tyjcwt)

Permits you to optimize a table using the table name specified in the first parameter. Returns TRUE/FALSE based on success or failure:

if ($this->dbutil->optimize\_table('table\_name'))  
{  
 echo 'Success!';  
}

Note

Not all database platforms support table optimization. It is mostly for use with MySQL.

[**Repair a Table**](#4i7ojhp)[**¶**](#3dy6vkm)

Permits you to repair a table using the table name specified in the first parameter. Returns TRUE/FALSE based on success or failure:

if ($this->dbutil->repair\_table('table\_name'))  
{  
 echo 'Success!';  
}

Note

Not all database platforms support table repairs.

[**Optimize a Database**](#2xcytpi)[**¶**](#1t3h5sf)

Permits you to optimize the database your DB class is currently connected to. Returns an array containing the DB status messages or FALSE on failure.

$result = $this->dbutil->optimize\_database();  
  
if ($result !== FALSE)  
{  
 print\_r($result);  
}

Note

Not all database platforms support database optimization. It it is mostly for use with MySQL.

[**Export a Query Result as a CSV File**](#1ci93xb)[**¶**](#4d34og8)

Permits you to generate a CSV file from a query result. The first parameter of the method must contain the result object from your query. Example:

$this->load->dbutil();  
  
$query = $this->db->query("SELECT \* FROM mytable");  
  
echo $this->dbutil->csv\_from\_result($query);

The second, third, and fourth parameters allow you to set the delimiter newline, and enclosure characters respectively. By default commas are used as the delimiter, “n” is used as a new line, and a double-quote is used as the enclosure. Example:

$delimiter = ",";  
$newline = "\r\n";  
$enclosure = '"';  
  
echo $this->dbutil->csv\_from\_result($query, $delimiter, $newline, $enclosure);

Important

This method will NOT write the CSV file for you. It simply creates the CSV layout. If you need to write the file use the [*File Helper*](http://docs.google.com/helpers/file_helper.html).

[**Export a Query Result as an XML Document**](#3whwml4)[**¶**](#2s8eyo1)

Permits you to generate an XML file from a query result. The first parameter expects a query result object, the second may contain an optional array of config parameters. Example:

$this->load->dbutil();  
  
$query = $this->db->query("SELECT \* FROM mytable");  
  
$config = array (  
 'root' => 'root',  
 'element' => 'element',  
 'newline' => "\n",  
 'tab' => "\t"  
);  
  
echo $this->dbutil->xml\_from\_result($query, $config);

Important

This method will NOT write the XML file for you. It simply creates the XML layout. If you need to write the file use the [*File Helper*](http://docs.google.com/helpers/file_helper.html).

[**Backup Your Database**](#2bn6wsx)[**¶**](#17dp8vu)

[**Database Backup Notes**](#qsh70q)[**¶**](#3rdcrjn)

Permits you to backup your full database or individual tables. The backup data can be compressed in either Zip or Gzip format.

Note

This feature is only available for MySQL and Interbase/Firebird databases.

Note

For Interbase/Firebird databases, the backup file name is the only parameter.

$this->dbutil->backup(‘db\_backup\_filename’);

Note

Due to the limited execution time and memory available to PHP, backing up very large databases may not be possible. If your database is very large you might need to backup directly from your SQL server via the command line, or have your server admin do it for you if you do not have root privileges.

[**Usage Example**](#3as4poj)[**¶**](#26in1rg)

// Load the DB utility class  
$this->load->dbutil();  
  
// Backup your entire database and assign it to a variable  
$backup = $this->dbutil->backup();  
  
// Load the file helper and write the file to your server  
$this->load->helper('file');  
write\_file('/path/to/mybackup.gz', $backup);  
  
// Load the download helper and send the file to your desktop  
$this->load->helper('download');  
force\_download('mybackup.gz', $backup);

[**Setting Backup Preferences**](#1pxezwc)[**¶**](#lnxbz9)

Backup preferences are set by submitting an array of values to the first parameter of the backup() method. Example:

$prefs = array(  
 'tables' => array('table1', 'table2'), // Array of tables to backup.  
 'ignore' => array(), // List of tables to omit from the backup  
 'format' => 'txt', // gzip, zip, txt  
 'filename' => 'mybackup.sql', // File name - NEEDED ONLY WITH ZIP FILES  
 'add\_drop' => TRUE, // Whether to add DROP TABLE statements to backup file  
 'add\_insert' => TRUE, // Whether to add INSERT data to backup file  
 'newline' => "\n" // Newline character used in backup file  
);  
  
$this->dbutil->backup($prefs);

[**Description of Backup Preferences**](#49x2ik5)[**¶**](#35nkun2)

| Preference | Default Value | Options | Description |
| --- | --- | --- | --- |
| **tables** | empty array | None | An array of tables you want backed up. If left blank all tables will be exported. |
| **ignore** | empty array | None | An array of tables you want the backup routine to ignore. |
| **format** | gzip | gzip, zip, txt | The file format of the export file. |
| **filename** | the current date/time | None | The name of the backed-up file. The name is needed only if you are using zip compression. |
| **add\_drop** | TRUE | TRUE/FALSE | Whether to include DROP TABLE statements in your SQL export file. |
| **add\_insert** | TRUE | TRUE/FALSE | Whether to include INSERT statements in your SQL export file. |
| **newline** | “\n” | “\n”, “\r”, “\r\n” | Type of newline to use in your SQL export file. |
| **foreign\_key\_checks** | TRUE | TRUE/FALSE | Whether output should keep foreign key checks enabled. |

[**Class Reference**](#2p2csry)[**¶**](#1ksv4uv)

*class* CI\_DB\_utility[¶](#147n2zr) backup([*$params = array()*])[¶](#3o7alnk)

| Parameters: | * **$params** (*array*) – An associative array of options |
| --- | --- |
| Returns: | raw/(g)zipped SQL query string |
| Return type: | string |

Perform a database backup, per user preferences.

database\_exists(*$database\_name*)[¶](#23ckvvd)

| Parameters: | * **$database\_name** (*string*) – Database name |
| --- | --- |
| Returns: | TRUE if the database exists, FALSE otherwise |
| Return type: | bool |

Check for the existence of a database.

list\_databases()[¶](#ihv636)

| Returns: | Array of database names found |
| --- | --- |
| Return type: | array |

Retrieve a list of all the database names.

optimize\_database()[¶](#32hioqz)

| Returns: | Array of optimization messages or FALSE on failure |
| --- | --- |
| Return type: | array |

Optimizes the database.

optimize\_table(*$table\_name*)[¶](#1hmsyys)

| Parameters: | * **$table\_name** (*string*) – Name of the table to optimize |
| --- | --- |
| Returns: | Array of optimization messages or FALSE on failure |
| Return type: | array |

Optimizes a database table.

repair\_table(*$table\_name*)[¶](#41mghml)

| Parameters: | * **$table\_name** (*string*) – Name of the table to repair |
| --- | --- |
| Returns: | Array of repair messages or FALSE on failure |
| Return type: | array |

Repairs a database table.

csv\_from\_result(*$query*[, *$delim = '*, *'*[, *$newline = "n"*[, *$enclosure = '"'*]]])[¶](#2grqrue)

| Parameters: | * **$query** (*object*) – A database result object * **$delim** (*string*) – The CSV field delimiter to use * **$newline** (*string*) – The newline character to use * **$enclosure** (*string*) – The enclosure delimiter to use |
| --- | --- |
| Returns: | The generated CSV file as a string |
| Return type: | string |

Translates a database result object into a CSV document.

xml\_from\_result(*$query*[, *$params = array()*])[¶](#vx1227)

| Parameters: | * **$query** (*object*) – A database result object * **$params** (*array*) – An associative array of preferences |
| --- | --- |
| Returns: | The generated XML document as a string |
| Return type: | string |

Translates a database result object into an XML document.

[Next](http://docs.google.com/db_driver_reference.html)   [Previous](http://docs.google.com/forge.html)

© Copyright 2014 - 2016, British Columbia Institute of Technology. Last updated on Oct 28, 2016.

Built with [Sphinx](http://sphinx-doc.org/) using a [theme](https://github.com/snide/sphinx_rtd_theme) provided by [Read the Docs](https://readthedocs.org).