

Backing Up Your Database

Languages: [English](#) • [日本語](#) • [Português do Brasil](#) • [Русский](#) • [ไทย](#) • [中文\(简体\)](#) • [\(Add your language\)](#)

It is strongly recommended that you backup your database at regular intervals and before an upgrade.

[Restoring your database from backup](#) is then possible if something goes wrong.

Backup using cPanel X

cPanel is a popular control panel used by many web host. The backup feature can be used to backup your MySql database. Do not generate a full backup, as these are strictly for archival purposes and cannot be restored via cPanel. Look for 'Download a MySQL Database Backup' and click the name of the database. A *.gz file will be downloaded to your local drive.

There is no need to unzip this file to restore it. Using the same cPanel program, browse to the gz file and upload it. Once the upload is complete, the bottom of the browser will indicate dump complete. If you are uploading to a new host, you will need to recreate the database user along with the matching password. If you change the password, make the corresponding change in the wp-config.php file.

Using phpMyAdmin

[phpMyAdmin](#) is the name of the program used to manipulate your database.

Information below has been tried and tested using phpMyAdmin versions 2.5.3, 2.5.7-pl1, and 2.6.1-pl3 running on Unix.

Detailed options are listed below, though in most cases you will be fine with the default export options.

Backup Process with phpMyAdmin

- Log into phpMyAdmin on your server
- From the main login screen, select 'Databases'
(You may not need to do this step)

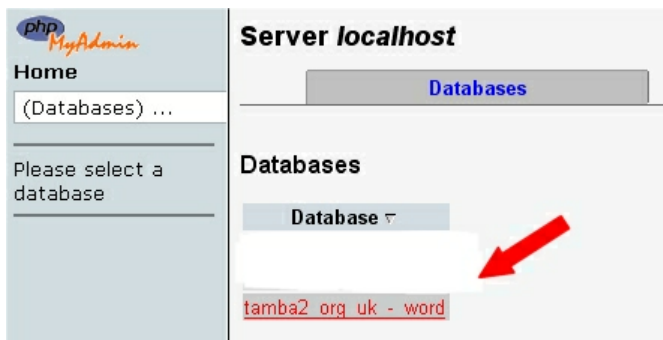


- Now click the name of your database - or your WordPress database if you have several databases.
(Your screen may look slightly different, depending on the version.)

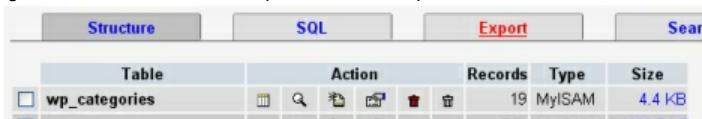
Contents

- [hide]
- 1 Backup using cPanel X
 - 2 Using phpMyAdmin
 - 2.1 Backup Process with phpMyAdmin
 - 3 Using Straight MySQL Commands
 - 4 Using MySQL Workbench
 - 4.1 Backing Up the Database
 - 4.2 Restoring From a Backup
 - 5 MySQL GUI Tools
 - 6 Using WordPress Database Backup Plugin
 - 6.1 Installation
 - 6.2 Backing up
 - 6.3 Restoring the Data
 - 7 Resources and Backup Plugins

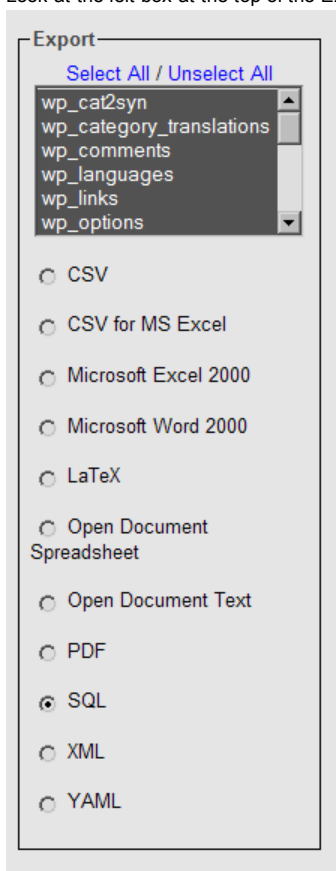
- Home Page
- WordPress Lessons
- Getting Started
- Working with WordPress
- Design and Layout
- Advanced Topics
- Troubleshooting
- Developer Docs
- About WordPress
- Codex Resources
- Community portal
- Current events
- Recent changes
- Random page
- Help



4. The next screen will show you all the tables inside your WordPress database. Ignore those, and click the 'Export' tab on the top set of tabs.



5. Look at the left box at the top of the Export section. All the tables in the database you selected are in that box.



- If you have other programs that use the database, then choose only those tables that correspond to your WordPress install. They will be the ones that start with "wp_" or whatever 'table_prefix' you specified in your 'wp-config.php' file.
- If you only have your WordPress blog installed, leave it as is (or click 'Select All' if you changed the selection)
- Ensure that the SQL radio button is selected.

☒ **Structure**

- ☒ Add DROP TABLE / VIEW / PROCEDURE / FUNCTION
- ☒ Add IF NOT EXISTS
- ☒ Add AUTO_INCREMENT value
- ☒ Enclose table and field names with backquotes
- ☐ Add CREATE PROCEDURE / FUNCTION

Add into comments

- ☐ Creation/Update/Check dates

6. The SQL section

Tick the following boxes:

- 'Structure'
- 'Add DROP TABLE / VIEW / PROCEDURE / FUNCTION'
- 'Add IF NOT EXISTS'
- 'Add AUTO_INCREMENT' and
- 'Enclose table and field names with backquotes'

7. The DATA section

It doesn't matter too much what you check here. So long as you check the DATA box itself, the Complete inserts and Extended Inserts are mostly cosmetic changes to the output style, though Extended Inserts can help if you are having issues with the file size.

Export

wp_links
wp_options
wp_post2cat
wp_postmeta
wp_posts
wp_users

1 Select All / Unselect All

☒ SQL
☐ LaTeX
☐ CSV for MS Excel
☐ CSV
☐ XML

SQL options

Add custom comment into header (\n splits lines):

☐ Enclose export in a transaction
☐ Disable foreign key checks

☒ **Structure:**

- ☒ Add DROP TABLE
- ☐ Add IF NOT EXISTS
- ☒ Add AUTO_INCREMENT value
- ☒ Enclose table and field names with backquotes

Add into comments

☐ Creation/Update/Check dates

☒ **Data:**

- ☒ Complete inserts
- ☐ Extended inserts
- ☐ Use delayed inserts
- ☐ Use ignore inserts
- ☒ Use hexadecimal for binary fields

Export type: INSERT

4 Save as file

File name template: (☒ remember template)*

Compression

☒ None ☐ "zipped" ☐ "gzipped"

6 Go

8. Tick the 'Save as file' option, and leave the template name as is.

9. Now click 'Go' and you should be prompted for a file to download. Save the file to your computer. Depending on the database size, this may take a few moments.

10. You have now backed up your database.

If you wanted, you could download a backup in each of the compression formats. Your choice. For example: **None** and **"zipped"**:



tamba2_orq_uk - word-1.sql
SQL File
2,247 KB



tamba2_orq_uk - word.sql.zip
469 KB

Remember - you have NOT backed up the files and folders - such as images - but all your posts and comments are now safe.

Using Straight MySQL Commands

phpMyAdmin cannot handle large databases so using straight MySQL code will help.

1. Change your directory to the directory you want to dump things to:

```
user@linux:~> cd files/blog
```

2. Use `mysqldump` to dump all database tables. To dump only certain tables from the database, give their names at the place shown by (tablename tablename tablename), and omit the parentheses () in any case. (For help, try: `man mysqldump`.):

```
user@linux:~/files/blog> mysqldump --add-drop-table -h mysqlhostserver
-u mysqlusername -p databasename (tablename tablename tablename) | bzip2
-c > blog.bak.sql.bz2
```

```
Enter password: (enter your mysql password)
user@linux~/files/blog>
```

Example:
`mysqldump --add-drop-table -h db01.example.net -u dbocodex -p dbwp | bzip2 -c > blog.bak.sql.bz2`

```
Enter password: my-password
user@linux~/files/blog>
```

The `bzip2 -c` after the `|` (pipe) means the backup is compressed on the fly, and the `> blog.bak.sql.bz2` sends the `bzip2` output to a file named `blog.bak.sql.bz2`. It does in one line the same thing that these two commands do:

```
mysqldump --add-drop-table -h db01.example.net -u dbocodex -p dbwp > blog.bak.sql
bzip2 blog.bak.sql
```

Despite `bzip2` being able to compress most files more effectively than the older compression algorithms (`.Z`, `.zip`, `.gz`), it is [considerably slower](#) (compression and decompression). If you have a large database to dump, `gzip` is a faster option to use.

```
mysqldump --add-drop-table -h db01.example.net -u dbocodex -p dbwp | gzip > blog.bak.sql.gz
```

Using MySQL Workbench

[MySQL Workbench](#) (formerly known as MySQL Administrator) is a program for performing administrative operations, such as configuring your MySQL server, monitoring its status and performance, starting and stopping it, managing users and connections, performing backups, restoring backups and a number of other administrative tasks.

You can perform most of those tasks using a command line interface such as that provided by [mysqladmin](#) or [mysql](#), but MySQL Workbench is advantageous in the following respects:

- Its graphical user interface makes it more intuitive to use.
- It provides a better overview of the settings that are crucial for the performance, reliability, and security of your MySQL servers.
- It displays performance indicators graphically, thus making it easier to determine and tune server settings.
- It is available for Linux, Windows and MacOS X, and allows a remote client to backup the database across platforms. As long as you have access to the MySQL databases on the remote server, you can backup your data to wherever you have write access.
- There is no limit to the size of the database to be backed up as there is with phpMyAdmin.

Note: The instruction below was written for older version (MySQL Administrator).

Backing Up the Database

This assumes you have already installed MySQL Admin and set it up so that you can login to the MySQL Database Server either locally or remotely. Refer to the documentation that comes with the installation package of MySQL Admin for your platform for

installation instructions.

1. Open the MySQL Admin client and login as you had previously set up to do.
2. From the icon menu on the left hand side of the client window select Backup.
3. If you have not already created a Backup Project, do this now by clicking on the "New Project" button at the lower part of the window and type in a name for the Backup Project where prompted.
4. Select one or more databases that you want to Backup (in the MySQL Admin client these are called a "Schema" (pl. "Schemata")). Add them to the Backup Content window on the right using the right-pointing arrow button.
5. When you have selected the Schema(ta), you can save the Backup Project. Or you may simply choose to Backup Now using the button on the lower right of the window.
6. A dialogue will come up asking you where to put the Backup. Enter the pathname or browse to the location using the dialogue.
7. Assuming all is correct (and you have write permissions in the directory to which you are writing the Backup), the backup will complete shortly.

Restoring From a Backup

1. Open the MySQL Admin client and login as you had previously set up to do.
2. From the icon menu on the left hand side of the client window select Restore.
3. Click the "Open Backup File" button on the lower right of the window.
4. Type in or browse to the Schema(ta) backup file and select. Click "Open".
5. The Target Schema(ta) will most likely be the "Original Location", or you may choose an alternate location using the drop-down menu.
6. Click the "Start Restore" button on the lower right of the window. The database restore will commence.

MySQL GUI Tools

In addition to MySQL Workbench, there are many GUI tools that let you backup (export) your database.

Name	OS (Paid edition)	OS (Free edition)	Notes
MySQL Workbench	Windows/Mac/Linux	Windows/Mac/Linux	See above
EMS SQL Management Studio for MySQL	Windows	-	
Aqua Data Studio	-	Windows/Mac/Linux	Available in 21 languages
Navicat for MySQL	Windows/Mac/Linux	Windows/Mac/Linux (Lite)	Available in 8 languages
SQLyog	Windows	-	
Toad for MySQL	-	Windows	
HeidiSQL	-	Windows	
Sequel Pro	-	Mac	CocoaMySQL successor
Querious	-	Mac	

Using WordPress Database Backup Plugin

[Austin Matzko](#) maintains a [WordPress plugin](#) originally created by [Skippy](#) called [WordPress Database Backup \(WP-DB-Backup\)](#). It was bundled with WordPress 2.0 but is no longer included with WordPress 2.1.

Installation

1. Search "WP-DB-Backup" on [Administration](#) > [Plugins Panel](#) > [Add New](#).
2. Activate the plugin.
3. The plugin will attempt to create a directory `/wp-content/backup-*/` inside your WordPress directory. You may need to make `/wp-content` [writable](#) (at least temporarily) for it to create this directory.

Backing up

1. Navigate to [Administration](#) > [Tools](#) > [Backup Panel](#)
2. Select any tables, in addition to the core WordPress tables, that should be backed-up.
3. Select the Backup Options; the backup can be saved on the server, downloaded, or emailed.
4. Finally, click on the Backup button to actually perform the backup. You can also schedule regular backups.

Restoring the Data

The file created is a standard SQL file. If you want information about how to upload that file, look at [Restoring Your Database From](#)

[Backup.](#)

Resources and Backup Plugins

- [Backup at its elementary level](#)
- [Backup Plugins](#)
- [Database Backup Plugin](#)
- [XCloner Backup Plugin](#)
- [myRepono WordPress Backup Plugin](#)

Categories: [Getting Started](#) | [Advanced Topics](#) | [Troubleshooting](#)

[Privacy](#) | [License / GPLv2](#) See also: [WordPress.com](#) | [WordPress.TV](#) | [WordCamp](#) | [WP Jobs](#) | [Matt](#) | [Blog RSS](#)

Me gusta

397000

16k