# Backup

Summary: Backup in 3 simple steps
Two types of things in a standard Tiki installation must be saved for a backup of your specific site:

1. Backup your MySQL database. (see below for examples)
2. Backup your files.
   Copy via FTP your entire folder where Tiki is installed (and all subfolders)
3. Make sure to also backup your files that are outside the web directory (if you set up your Tiki this way)

## Backup the files

If you configured the file gallery (or any feature) to store files elsewhere than where Tiki is installed (ex: outside the web directory), you need copy these files too. (see tiki-admin_system.php)

## Backup the database

Your web host will usually provide a tool to backup your database. Here are various options.

### A. Backing up through Web interface (Adminer)

Adminer (formerly phpMinAdmin) is a full-featured MySQL management tool written in PHP. Conversely to phpMyAdmin, it consists of a **single file** ready to deploy to the target server.

### B. Backing up through Web interface (phpMyAdmin)

If needed, ask your system admin if they provide phpMyAdmin in that hosting, and where is it located for you to access it.

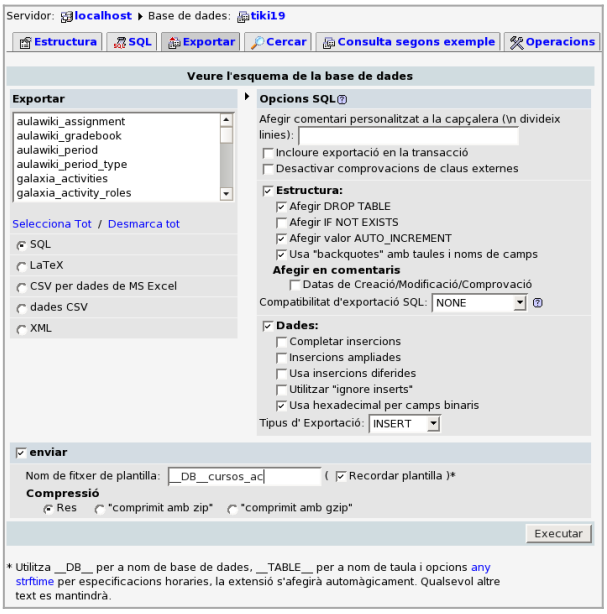
Eventually, you could get phpMyAdmin software to install it yourself: http://www.phpMyAdmin.net/

The main screen in phpMyAdmin looks like:
*(screenshots are taken using the TikiLiveCD: http://tiki.org/TikiLiveCD)*

Click to expand

Select your database from the list (tiki19 in this example; tiki_1 in screenshot below), at the drop down from the left column of the screen.

Click on the tab that says **SQL** (or **Export**, in newer versions of phpMyAdmin):

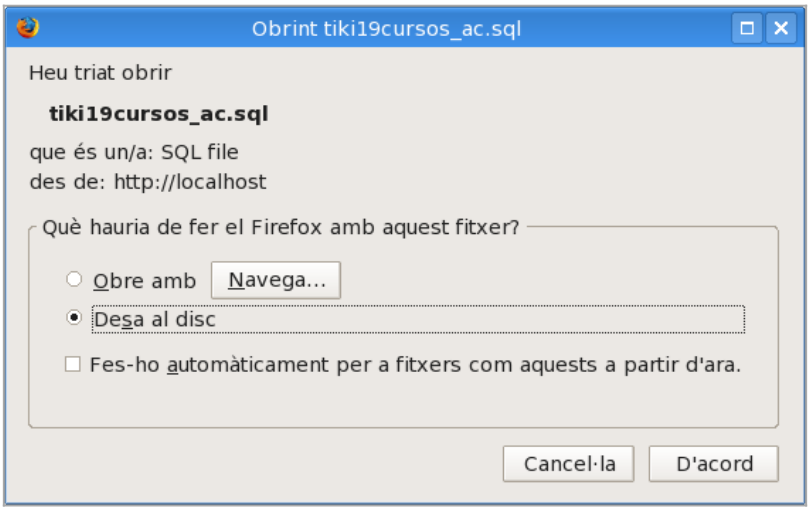



Click to expand

At the export screen, make sure that you check the box from the section "**Structure**", that says "**Add DROP TABLE**", since it will be much easier the process to re-insert a backup later on from this same database that you are now exporting.

The rest of options are ok as seen by default.

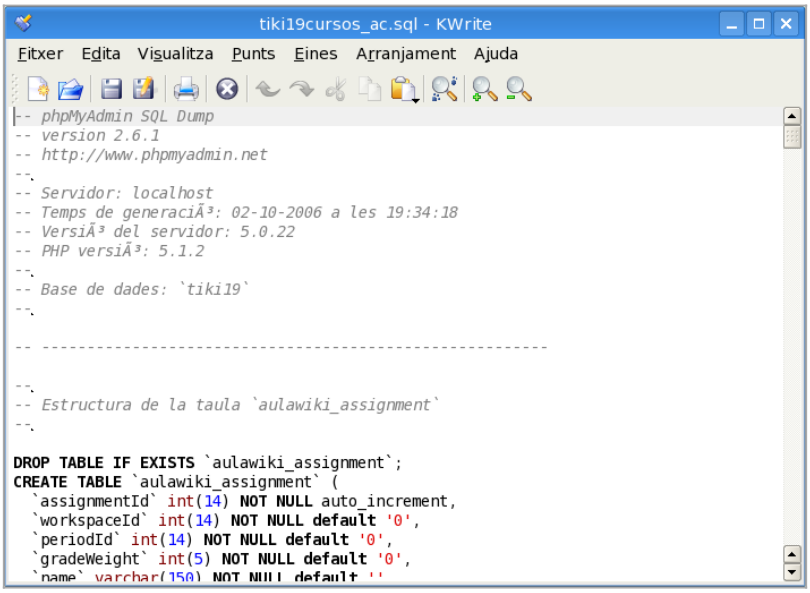
At last, select the box "**send**", and click on the button "**Execute**". You will see a dialog menu like this one:




Click to expand

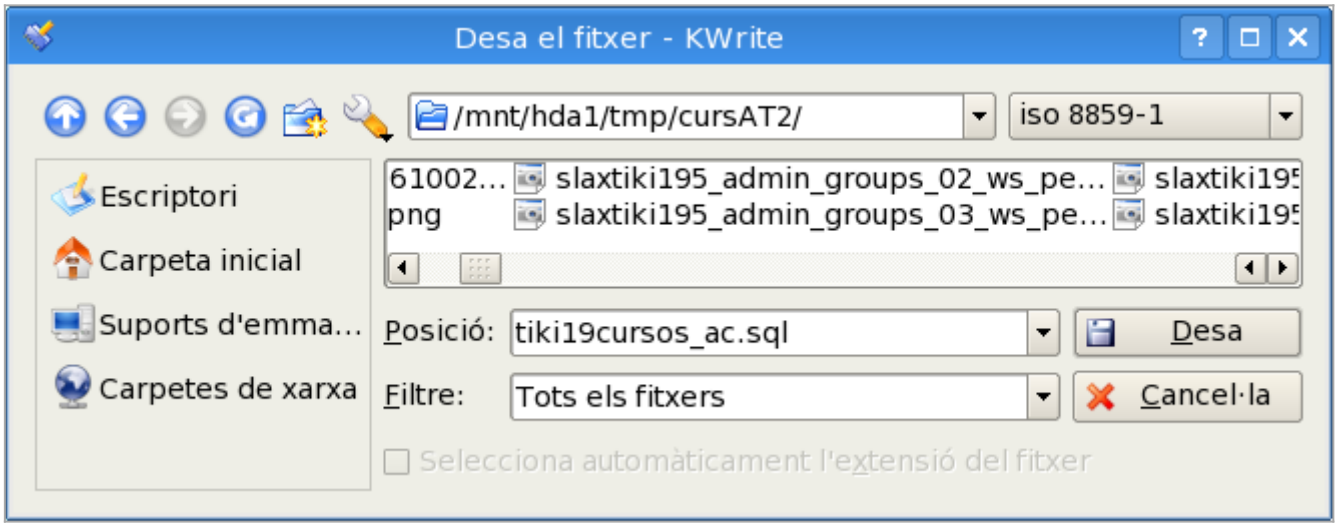
Save it on disc at your preferred location.

We will now open it to check that everything is all right (and to let you show the type of file saved, if this is new to you):



Click to expand

The file has to be encoded in **utf-8**. If we want to change the character set to, for instance, iso-8859-1 (to be able to edit it with most MS Windows applications that didn't cope well with other charsets different from that default), we can do it direcetly from true "utf-8 enabled" editors (see ToolBox pages for links on common applications to do that depending on your operating system). For instance, using **KWrite**, on GNU/Linux, it would look like:



Click to expand

## C. Backing up with console.php at a terminal screen

Basic command (this will make a date and timestamped gzipped dump of the database in */path/to/backups/*)

```
php console.php database:backup /path/to/backups/
```

Format the backup file name, this example will only use the year, month and day in the output file name.

```
php console.php database:backup /path/to/backups/ Y-m-d
```

Add a cron command to do this to create an hourly backup which will overwrite yesterday's ones 24 hours later

```
cd /home/tiki/public_html; php console.php database:backup /home/backups/hourly/ H00 >/dev/null
```

### D. Backing up through commands at a terminal screen (mysqldump)

Write a command like:

```
mysqldump --opt -u user -p tiki12 > tiki12_backup_yyyymmdd.sql
```

In your case, change in the example above:

- --opt is to use common options for mysqldump
- *user* for your username at the MySQL server
- -p will prompt you for the password
- *tiki12* for your database name, and
- *yyyymmdd* for the four digits of the year, month and day, respectively, for instance, to have your backups easily sorted by name and date of creation also.

Then, save your backup copy in a safe place. It's recommended to send by **ftp** or **sftp** that file to your local computer to prevent problems if server harddisk and backups fail at any time.

To restore, please see: Import Database

### E. phpMyBackupPro

I installed **phpMyBackupPro** v.1.8 http://www.phpmybackuppro.net and modded 1 line of tiki-login: including a call of the scheduled backup script of pMBP. Now I get my daily gzipped backup of the Database delivered right into my mailbox.

phpMyBackupPro has also a feature for directory backup per FTP; I've not yet tried it for attachment and gallery folders located outside of www root.

I included the backup feature into the Login script because only logged users can post on my site - thus the script is only called once for every user, and performance is only slightly affected by the first user each new day who triggers the daily backup (has to wait some seconds until the backup is finished). If nobody logs in there's no need to update the last backup.

### F. MySQLDumper

**MySQLDumper** is a PHP and Perl based tool for backing up MySQL databases. You can easily dump your data into a backup file and - if needed - restore it. It is especially suited for shared hosting webspaces, where you don't have shell access. MySQLDumper is an open source project and released under the GNU-license.
The problem: A PHP script has a maximum execution time that is usually set to 30 seconds on most server installations. A script running longer than this limit will simply stop working. This behavior makes backing up large databases impossible. Maybe you already had this specific problem when using

other tools. MySQLDumper fills a gap ÃƒÆ'Ã‚Â¢ÃƒÂ¢Ã¢â‚¬Ã¢â€žÃ‚Â¬Ã‚Â¦

MySQLDumper uses a proprietary technique to avoid this problem. It only reads and saves a certain amount of data, then calls itself recursively via JavaScript and remembers how far in the backup process it was. The script then resumes backing up from that point.

The restore process is similar. Unlike other tools, splitting and splicing of large backup files is no longer necessary.

MySQLDumper can write the data directly into a compressed .gz file. The restore script is able to read this file directly without unpacking it. You can also use the script without compression, but using Gzip saves a lot of bandwidth. You can even configure the script to automatically send the backup file to an FTP account or your email address.
Get it on http://www.mysqldumper.net/.

### G. HeidiSQL

**HeidiSQL** is a graphical and easy to use interface to your SQL database. You can define connections in the server manager. Dump selected databases and data into a single dump file, one file per table, directly to another host, to clipboard or to another database on the same server.Get it on http://www.heidisql.com.

## Related links

### Importing the database again

In order to restore the database you have in a backup, you have to import it to you MySQL server. See Import database for more details.

### Backup via the Tiki Remote Instance Manager (TRIM)

Please see TRIM, which backs up both files and the database.