

This should merge with [Migrate a Tiki to a new server with SSH](#)

Move Tiki using SSH

WARNING: INCOMPLETE/INACCURATE - this is a new page not yet complete or even correct.

Initial content adapted from [Domain Migration to a Unix Server](#) by Dirk Brockhausen

To switch Web hosts or install tiki on a new server entails the migration of all your tiki files, image uploads, attachments and databases. This page will focus on how to transfer the files and database from your previous server to the new one.

Migration often isn't as Simple as it Looks

At first glance, this task may seem straightforward enough: in some cases you can use FTP to [move tiki to a new server](#), but sometimes there are problems.

However, this approach has several serious drawbacks:

- your database may be too big to import by phpMyAdmin
- file permissions will not be preserved
- file ownership will change
- ASCII files carrying an unknown extension may be treated as binaries
- the files will be down- and uploaded in uncompressed form, which wastes time, bandwidth and system resources.

If you proceed in this manner, you'll be in for a lot of adjustment work to make everything run as it did before. This applies especially to CGI scripts where directory and file permissions are critical to ensure functionality.

Migration Made Easy

Here's a method that will simplify your migration and slash unnecessary work in the process.

Log in to your old server

You can do this by Telnet or PuTTY as user which here we call *domainowner*.

Find your site directory

Navigate to your old domain directory using a command like this.

```
:cd /usr/www/htdocs/yourdomain).(substitute your info for italicized text)
```

Create a zip file with everything in it.

You can compress all the files into an archive by entering the following on the command line:

```
tar -cpz --same-owner -f yourdomain.tar.gz * .htaccess
```

Understanding tar

The "tar" command requires the following syntax:

tar -function -option file
function:

- *c* is "create archive"
- *p* extracts all protection information
- *z* compresses data with gzip

- *same-owner* create extracted files with the same ownership
- *f* use archive file
- the archive file "yourdomain.tar.gz" will be generated.

You may choose any file name you prefer, as long as you stick to the extension ".tar.gz".

The wildcard character "*" (second-last in the command above) specifies that all files, including those that reside in subdirectories, will be included in the archive. However, because this won't include files that begin with a period, you will have to list those files explicitly. In our example one file like this is file ".htaccess".

Create a user account on the new server

Now, on the new server, you will have to configure user account "domainowner" and the directory "/usr/www/htdocs/yourdomain". Note that if the directory names on your previous server are not identical to those on your new server, you may have to modify the path variables in your tiki and theme files. Hence, it's strongly recommended that you simply clone directory names.

Upload the archive.

Upload the file "yourdomain.tar.gz" to your new system. There are two possible ways to do this:

* directly, by logging in to your new server and downloading the file from there using whatever FTP client is available, e.g. WU-FTP. Consult the program's documentation to learn how to do this as we cannot cover it here. or

* by downloading the file to your local system first, and then uploading it to the new server from there.

Log into the new system

log in via Telnet e.g. as user "domainowner".

Locate your new directory

Switch into your domain directory e.g. by typing: `cd /usr/www/htdocs/yourdomain`

Unpack the archive

Next, you'll need to unpack the archive file. Enter the command:

```
tar -xz -f yourdomain.tar.gz
```

function: *x* extract files from an archive

option: *z* uncompress data with gzip

f use archive file

There - you have now successfully duplicated the previous, familiar environment on your new system!

However, I'd recommend that you test the new domain setup thoroughly, especially your CGI scripts, before you launch it officially.

Exporting the database

Restoring the database