

This is an [installation](#) guide.

Multitiki appeared in tiki 1.7, it was one of my first contributions in june 2003. After some time and evolution this special tiki setup is now robust and well-tested. The current documentation talks about the 1.9 multitiki version. There is an alternative proposed by [ggeller](#) on [RecipeMultiTiki](#) that is more adapted to 1.8 version.

Multiki under Windows XP

For those who want to setup Multitiki under Windows XP, please check [MultitikiWindowsXP](#)

Following steps only apply to GNU/Linux environment.

Simple steps to add one virtual tiki to an existing one

Setup (perms and dir creation)

Run setup.sh with an extra parameter that will be the name of your virtual tiki. Either TIKI_VIRTUAL, SERVER_NAME, or HTTP_HOST can be used to detect in which multitiki to go depending the apache configuration. Here is what I typed to add <http://lir.be> test website to my multitiki collection.

```
su ./setup.sh mose www-data 02775 lir.be
```

The script will :

- verify that web-writable dirs exist, as well as virtual subdir
- change perms to those dirs to make apache can write in it
- update the file db/virtuals.inc that is in 1.9 the central file where all virtuals are listed

Install (db feeding)

Re-activate your web-install file if your neutralized it by a chmod. for example

```
chmod 444 tiki-install.php
```

Point your browser on <http://yourtiki/tiki-install.php> (if it doesn't work, try to add "lang=en" to the URL: <http://yourtiki/tiki-install.php?lang=en>). There should be a new item in the menu on the topleft, indicating the state of the db connection (actually the existence and validity of the local.php file for that virtual). That menu is based on db/virtuals.inc file so you can want to change things manually there for tuning. Clicking on the link activates the virtual web-install page like if it was a lone tiki. Then you can follow the install steps :

- give connection settings, that step will create a file named db/lir.be/local.php in my case
- select a db operation, either feeding a created but empty db, or upgrading a 1.8.3 version.

Secure (neutralize tiki-install.php)

The best known way is to change perms on the tiki-install, if you use cvs it won't break all things

```
chmod 500 tiki-install.php
```

The fast way (without using tiki-install.php)

On already setup multitiki, you can perform the required operation with command line. For example in case of the lir.be that we saw above :

mysqladmin -uroot -p create tiki_dbname mysql -uroot -p tiki_dbname < db/tiki.sql # optionally, if you want some prepared set of tiki : " # mysql -uroot -p tiki_base < db/profiles/BasicEnabled.prf mysql -uroot -p -e "grant usage on *.* to tiki_user@localhost identified by 'tiki_pass';" tiki_dbname mysql -uroot -p -e "grant all on tiki_dbname.* to tiki_user@localhost identified by 'tiki_pass';" tiki_dbname

then copy the local.php from another instance of the multitiki, edit it and change the dbname, user and password accordingly to the above creation.

Here is the console shoot of a 5-minutes install of a new multitiki :

```
15:41,mose@one:/home/multitiki sudo zsh
Password:
15:42,root@one:/home/multitiki ./setup.sh mose www-data 02777 brightgreen.lir.be
backups db dump img/wiki img/wiki_up modules/cache temp temp/cache templates_c templates styles maps whelp lib/Galaxia/processes
Creating directory backups/brightgreen.lir.be
Creating directory db/brightgreen.lir.be
Creating directory dump/brightgreen.lir.be
Creating directory img/wiki/brightgreen.lir.be
Creating directory img/wiki_up/brightgreen.lir.be
Creating directory modules/cache/brightgreen.lir.be
Creating directory temp/brightgreen.lir.be
Creating directory temp/cache/brightgreen.lir.be
Creating directory templates_c/brightgreen.lir.be
Creating directory templates/brightgreen.lir.be
Creating directory styles/brightgreen.lir.be
Creating directory maps/brightgreen.lir.be
Creating directory whelp/brightgreen.lir.be
Creating directory lib/Galaxia/processes/brightgreen.lir.be
15:43,root@one:/home/multitiki vi /etc/apache/conf.d/multitiki.conf
15:45,root@one:/home/multitiki apachectl configtest
Syntax OK
15:45,root@one:/home/multitiki /etc/init.d/apache restart
Restarting apache.
15:45,root@one:/home/multitiki mysqladmin -p create tiki_bg
Enter password:
15:45,root@one:/home/multitiki mysql -p tiki_bg < db/tiki.sql
Enter password:
15:46,root@one:/home/multitiki mysql -p tiki_bg < db/profiles/BasicEnabled.prf
Enter password:
15:46,root@one:/home/multitiki cp db/hornia.org/local.php db/brightgreen.lir.be
15:46,root@one:/home/multitiki vi db/brightgreen.lir.be
15:47,root@one:/home/multitiki Read from remote host one: Connection reset by peer
Connection to one closed.
zsh: 9563 exit 255 ssh one
16:38,mose@one:/home/mose
```

How it works ?

Tikiwiki uses his database a lot for storing most content. Therefore, some temporary or volatile content is written to disk, in some directories

- templates_c (smarty pre-compiled templates)
- temp (general temporary dir for tiki)
- temp/cache (centralized cache dir)
- modules/cache (used by smarty)
- img/wiki_up (where are stored wiki images)
- img/wiki (where are stored images generated by jgraphpad)
- maps/ (if maps feature is used)

Multitiki uses a subdir in each of these directories so each virtual tiki has his own temporary stuff and there is no collision risk. In more than temporary and cache directories, 1.9 adds a subdir in other places that become specific for each virtual

- db/lir.be/ (where there is a simple local.php file for db connection)
- templates/lir.be/ (for templates)
- styles/lir.be/ (for css and images linked to css)

About templates and styles, the virtual tiki uses its subdir by cascading. the subdir can be empty, then the

'root' items (templates or css) are used. Then the non-subdired templates and styles are what is common to all virtual tikis on this system. If there is an equivalent file name in a subdir for a virtual tiki, that local file will be preferred, in same order than in main tiki in case of styled templates.

Ok. I lost you. Here is an example :

- templates/lir.be/styles/moreneat/tiki-showpage is used if exists, else
- templates/lir.be/tiki-showpage is used if exists, else
- templates/styles/moreneat/tiki-showpage is used if exists, else
- templates/tiki-showpage is used (because it exists)

About how the hell tiki knows what virtual to use, it's located in db/tiki-db.php. That file tests the hostname and attribute it to the tikidomain variable that is the name of the subdirs used. Nothing very complicated. If you have multi-named hosts you can change that code to fit your needs. That's also where can be solved the case of subdirs (rather than virtual domains) by testing on url path.

[mose](#)

OPTIONAL

MULTITIKI ON SUBDOMAIN ONLY

To do multitiki according to the subdomain name (www, subdomain1, subdomain2...) whatever the main domain name is, add the following lines to tiki-db.php after line 55. That will select the database according to subdomain only. You must then have run the script to create the directory for subdomain.

```
$pattern="(.*)\.(.*)\.([[:alpha:]]{3})"; eregi($pattern,$_SERVER['HTTP_HOST'],$results);  
$subdomain=$results[1];
```

then this line after conditions on HTTP_HOST

```
elseif (isset($_SERVER['HTTP_HOST']) and is_file('db/'.$subdomain.'/local.php')) { $multi =  
$subdomain;
```

END MULTITIKI ON SUBDOMAIN ONLY

Note

Before 1.9rc4, make sure you have **\$tikidomain_multi = true;** in db/tiki-db.php
Since 1.9rc4 that is not necessary anymore.

More

[InstallVirtualHosting](#)

feedback is welcome, feel free to edit that page for adding or changing

Related pages

- [Multitiki](#)
- [Multitiki Akira](#) : several instances of tiki in subdirectories, based on a single tiki installation on the server
- [MultiTiki 3.0 Installation for Windows XP](#)