Note that this page is intended for installations where you have access to a shell in the server to run commands. Alternatively, you might want to follow the steps of the Installation instructions if you only have ftp access to your host.

You can do this "manually", even for MultiTiki installations.

Note: For Tiki5.1 and onwards (5.3, 6.0, ...), you need to enforce the charset of the connection to utf-8 (unless you know very well what you are doing). Therefore, you need to add in your custom local.php files this new line:

```
$client_charset='utf8';
```

1. MultiTikis using SubDomains

This section describes a sub-domain MultiTiki setup, where a subdomain points to a directory immediately below your www directory, e.g.:

- site1.mydomain.org (resolves to www.mydomain.org/site1)
- site2.mydomain.org (resolves to www.mydomain.org/site2)

1.1. Create subdirectories

If you can't use setup.sh (See MultiTiki) and/or tiki-install.php, you have to create your subfolders manually for your multitiki installation to work, but you'll need to make symlinks (or have them made for you by an admin), as many as sites you want to install with the same multitiki setup.

You could run ./setup.sh and provide the two names (**site1** and **site2** afterwards, when requested after the **[multi]:** option), or you need to manually create your subdirectories **site1** and **site2** inside each of the following folders (list of folders are defined as the `DIR_LIST_WRITABLE` variable in `setup.sh`):

```
db/site1 db/site2 dump/site1 dump/site2 files/site1 files/site2 img/wiki/site1 img/wiki/site2
img/wiki_up/site1 img/wiki_up/site2 img/trackers/site1 img/trackers/site2 maps/site1 maps/site2
mods/site1 mods/site2 modules/cache/site1 modules/cache/site2 storage/site1 storage/site2
storage/public/site1 storage/public/site2 temp/site1 temp/site2 temp/cache/site1 temp/cache/site2
temp/public/site1 temp/public/site2 temp/templates_c/site1 temp/templates_c/site2 temp/unified-
index/site1 temp/unified-index/site2 templates/site1 templates/site2 themes/site1 themes/site2
tiki_tests/tests/site1 tiki_tests/tests/site2 whelp/site1 whelp/site2
```

1.2. Add a single customized local.php for all

You need to add a manually modified ./db/local.php with the following syntax:

db/local.php

```
<?php $db_tiki='mysqli'; $dbversion_tiki='3.0'; $host_tiki='localhost'; if ($_SERVER["HTTP_HOST"]
== "site1.mydomain.org" || $_SERVER["HTTP_HOST"] == "www.site1.mydomain.org") { $user_tiki =
'user1'; $pass_tiki = 'pass1'; $tikidomain = "site1"; $dbs_tiki = "tiki30_site1"; } elseif
($_SERVER["HTTP_HOST"] == "site2.mydomain.org" || $_SERVER["HTTP_HOST"] ==
"www.site2.mydomain.org") { $user_tiki = 'user2'; $pass_tiki = 'pass2'; $tikidomain = "site2"; $dbs_tiki
= "tiki30_site2"; } else { $user_tiki = 'user1'; $pass_tiki = 'pass1'; $tikidomain = "site1";
$dbs_tiki='tiki30_site1'; }
```

1.3. Add local.php for each site

Proceed as usual with tiki-install.php, or if you can't for whatever reason, then, add a standard local.php file inside each of each multitiki subfolders of db, i.e.:
./db/site1/local.php

./db/site2/local.php

```
<?php $db_tiki='mysql'; $dbversion_tiki='3.0'; $host_tiki='localhost'; $user_tiki='user1';
$pass_tiki='pass1'; $dbs_tiki='tiki30_site1';
```

and

db/site2/local.php

```
<?php $db_tiki='mysql'; $dbversion_tiki='3.0'; $host_tiki='localhost'; $user_tiki='user2';
$pass_tiki='pass2'; $dbs_tiki='tiki30_site2';
```

1.4. Create symlinks for the multitiki to work

You need to create symbolic links (symlinks) in order to make your multitiki in subdirectories work.
Example 1
Imagine your www root is at:
/var/www

and you want to setup your multitiki out of the apache public directory, but here instead, for example:
/var/tiki

Then you need to create symlinks (or request them to be created by an admin for you), with instructions like these (prepending "sudo" to the commands or switching to the root account):

```
ln -s /var/tiki /var/www/site1 ln -s /var/tiki /var/www/site2
```

Note: Make sure the site1 and site2 directories do not exist before executing the above commands.

1.5. Running tiki-install

You are done with the manual setup. One by one, browse to each of your MultiTiki sites (e.g. http://localhost/site1/ and http://localhost/site2/) and start configuring them individually.

If you setup a common username and password for all your multi-tiki's, then the tiki-install.php script (Tiki 3.0) will recognize the MultiTiki installation as you proceed.

If you manually coded the db/local.php files and the /db/siteX/local.php files correctly, the tiki-install script sees all the databases as connected. On the left hand side of the image below, all but one database is showing connectivity. If you are happy, you can just use the existing connection to proceed.

NOTE (by little_papillon): in my installation (multi-tiki with multiple domains), I had to update information in /db/virtuals.inc before tiki-install.php would list the multitiki-sites

2. MultiTikis in subdirectories (Directory Structure Setup)

Suppose you want to avoid subdomains or would instead like your tiki structure to be directly accessed and located within a single directory (e.g. under a /var/tiki/, out of the public htdocs folder of your server, for instance, and symlinked from /var/www/site1 and /var/www/site2)

- *http://www.mydomain.com/tiki1*
- *http://www.mydomain.com/tiki2*

The setup is similar to a subdomain MultiTiki setup (see above). By way of example, we will be setting up **site1 and site2** with the URLs shown at the top of this section. This example assumes a directory **/var/www/** where your sub-tiki's will be "located". For sake, if someone does enter *www.mydomain.com/* we have setup to redirect them to the main (site1).

We begin by loading all the tiki files into the /var/tiki/ directory.

2.1. Create Subdirectories

Per instructions *above*, either run setup.sh or manually create the required subdirectories for your various tikis (tiki1, tiki2)

2.2. Add a single customized local.php for all

This is where we will diverge somewhat slightly from the subdomain strategy. Namely, we need to test the REQUEST_URI to determine where to funnel the visitor.

Once again, with our specific example values shown, you need to add a manually modified ./db/local.php with the following syntax:

db/local.php

```
<?php $db_tiki='mysqli'; $dbversion_tiki='3.0'; $host_tiki='localhost';
$myurl=$_SERVER["REQUEST_URI"]; $url_array=explode("/",$myurl); //for www.mydomain.com/site1
//the array[0] is empty, array[1] == site1 $requestedtiki = $url_array[1]; if ($requestedtiki == "site1")
{ $tikidomain = 'site1'; include "site1/local.php"; } elseif ($requestedtiki == "site2") { $tikidomain =
'site2'; include "site2/local.php"; } else { $tikidomain = 'site1'; include "site1/local.php"; header(
'Location: http://www.mydomain.com/site1'); }
```

Note how if all else fails, we'll send the user directly to our tiki of choice (site1).

2.3. Add local.php for each sub-tiki

Again, we'll branch a little from the prior setup by populating out the local.php files for each tiki.

At this point, it helps to take a side step and set up your databases (see documentation elsewhere). For simplicity you may want your MultiTikis to share a common username/password, but this is not necessary (or particularly secure).

Below is an example for our **site1**. This file would get saved under ./db/site1/local.php

./db/site1/local.php

```
<?php $db_tiki='mysql'; $dbversion_tiki='3.0'; $host_tiki='localhost'; $user_tiki='site1db_username';
$pass_tiki='site1db_password'; $dbs_tiki='site1_db';
```

2.4. Create Symlinks

You need to create symlinks (or request them to be created by an admin for you), with instructions like these:

```
cd /var/tiki/ //cd into the main tiki directory ln -s . /var/www/site1 ln -s . /var/www/site2
```

This will effectively redirect a visitor headed to *www.mydomain.com/site2* to *www.mydomain.com/site2* where our local.php script will parse the REQUESTED_URL, realize that the user wants *site2* and proceed accordingly. If the visitor attempts to visit *www.mydomain.com/site3*, he/she will be redirected to *www.mydomain.com/site1* instead.

2.5. Run tiki-install.php

Now, for each site, you will browse to it and run tiki-install.php (e.g. www.mydomain.com/site1/tiki-install.php)

A few notes:

- Since we went ahead and setup our ./db/site1/local.php files, you will see that the installer recognizes the MultiTiki's during the setup.
  - ''Not sure if this is required, but during installation it seemed to be important to click on the MultiTiki (selecting it and turning it **bold**) that you are installing.
  - Given the above, since the database is setup, you can use the Existing Connection you selected.
- If you are setting up multiple tiki's, use the Enter Without Locking The Installer until you get to the last tiki....to keep from having to constantly delete the ./db/lock file.

3. Extra: Add .htaccess for all (optional)

You can also add a single customized .htaccess for them all at the MultiTiki root
See Rewrite rules | Apache Clean URLs

See also

- Manager - Tiki Manager