

Advanced Farm Setups and Tips



Most of the [old farm documentation](#) which isn't covered in the [new farm documentation](#) should be added here.

The following sections assume a setup similar to the one described on [farms](#).

CLI tools

If you want to use the [command line tools](#) you only need to set the animal name in the `animal` environment variable. For example if you want to execute `bin/indexer.php` for the «foo» animal, you can use `animal=foo bin/indexer.php`.

Setting defaults and restricting options

You can add additional configuration files for the default values that should be used in animals and for restricting certain configuration options. If you add the following code to the end of `inc/preload.php`, the main configuration file of the farmer will be used as default for all animals

and the protected file of the farmer will be used as the last protected file for all animals which means that animals then can't change these options. This could be used in order to disable certain options like [phpok](#) if you don't trust the animal admin.

```
$config_cascade['main']['default'][] = DOKU_INC.'conf/local.php';
$config_cascade['main']['protected'][] =
DOKU_INC.'conf/local.protected.php';
```

Note that these changes also affect the farmer if it can be accessed. This will break the configuration manager in the farmer as it assumes that all changed values are defaults and will remove them from the configuration.

You can use similar changes for the plugins cascade in order to define a set of plugins that are enabled by default or a set of plugins that must or must not be enabled.

You can of course also completely change the configuration cascade, you can find the default configuration cascade that is used in farms at the end of `inc/farm.php`.

Disabling access to the farmer

If you want to disable access to the farmer you can add something like

```
if (DOKU_FARM == false) { nice_die('Access to the farmer denied'); }
```

at the end of `inc/preload.php` (after the inclusion of `farm.php`).

Central user management

If some or all of your animals shall share the users you can do that not just by using the same external authentication but also with the default user management of DokuWiki by setting the path to the user list to the same location at the end of `inc/preload.php`:

```
$config_cascade['plainauth.users'] = array(
    'default' => '/your/path/to/conf/users.auth.php',
);
```

If you want to do that just for some animals you can of course simply add a condition based on the value of `DOKU_CONF`.

Flexible .htaccess-based setups

Actually `.htaccess`-based setups can be a lot more flexible than the setups that is described on [farms](#). For these advanced setups your farm directory can and even should be outside the document root

(the directory that is accessible from the web). If your farm directory is inside the document root you need to make sure that you properly protect each data directory (for example using .htaccess as it is done in a normal DokuWiki setup) as in these advanced setups they might not be automatically protected (depends on the rewrite conditions you use).

In order to make the animals accessible as in the standard setup you can create a second farm directory inside the document root with a .htaccess file and no animal directories. You can also add RewriteCond statements as in the example below in order to prevent redirect loops. Then your animals can be in the domain's root directory, i.e. you can have example.com/animal1/ for animal «animal1» and example.com/dokuwiki/ for the farmer.

.htaccess

```
RewriteEngine On

# Prevent rewrite loops by making sure that the farmer directory won't
# be matched
# If you need to enable access to other directories either add them
# here or try something like
# RewriteCond %{REQUEST_FILENAME}          !-f
# RewriteCond %{REQUEST_FILENAME}          !-d
# but then also the files of your animals (if they should be in this
# directory) will be accessible
# so make sure you have properly protected them

RewriteCond %{REQUEST_URI} !^/?dokuwiki/
RewriteRule ^/?([^/]+)/(.*) /dokuwiki/$2?animal=$1 [QSA]
RewriteCond %{REQUEST_URI} !^/?dokuwiki$
# RewriteCond %{REQUEST_FILENAME}          !-f # allow files in the root
# directory (like some index file)
RewriteRule ^/?([^/]+)$ /dokuwiki/?animal=$1 [QSA]
Options +FollowSymLinks
```

More than one domain

If you can use [symbolic links](#) you can easily link a directory in each domain's document root to the farmer and specify this directory in the rewrite rules.

Custom paths

If you don't want to have all animals in one directory you can simply create a .htaccess file for each animal. In this .htaccess file you can specify the animal name.

.htaccess

```
RewriteEngine On
```

```
# Prevent rewrite loops by making sure that the farmer directory won't
be matched
# If you need to enable access to other directories either add them
here or try something like
# RewriteCond %{REQUEST_FILENAME}          !-f
# RewriteCond %{REQUEST_FILENAME}          !-d
# but then also the files of your animal (if they should be in this
directory) will be accessible
# so make sure you have properly protected them
RewriteCond %{REQUEST_URI} !^/?dokuwiki/
RewriteRule ^/?(.*) /dokuwiki/$1?animal=ANIMAL [QSA]
RewriteRule ^/?$ /dokuwiki/?animal=ANIMAL [QSA]

Options +FollowSymLinks
```

The RewriteCond statements prevent redirect loops when your farmer directory is inside the animal directory. That way you can also put an animal in the root directory of a domain.

Running a farm behind a reverse proxy

This is when you set up your farm on a backend server (e.g. <http://11.22.33.44>) and present your wikis to the world via a front-end Apache server (e.g. <http://www.example.com>).

Firstly set up the farm on the backend as per the [standard docs](#).

In my case I've decided to put all my DokuWiki files into one location to keep them separate from any other sites running on the backend server:

`/var/www/wikis/` - The root of my DokuWiki installation.

Therefore my farm setup is:

- `/var/www/wikis/dokuwiki` - the DokuWiki engine, this needs to be accessible from the web
- `/var/www/wikis/farm` - the DokuWiki farm directory which contains all the animals
- `/var/www/wikis/farm/bear` - my first wiki which is about bears

On the backend server my wiki about bears (once fully setup) will be available at the following address: <http://11.22.33.44/wikis/farm/bear>

However, on the frontend server I want to present my wiki with the following address (notice I've chosen to drop the 's' from 'wikis' in this URL): <http://www.example.com/wiki/bear>

Using the `.htaccess` type of setup, I set up `/var/www/wikis/farm/.htaccess` with the following code:

[.htaccess](#)

```
RewriteEngine On

RewriteRule ^/?([^/]+)/(.*) /wikis/dokuwiki/$2?animal=$1 [QSA]
RewriteRule ^/?([^/]+)$ /wikis/dokuwiki/?animal=$1 [QSA]
Options +FollowSymLinks
```

Under `/var/www/wikis/farm/bear/conf` I then edit **local.protected.php** like so (note: this uses the front-end address):

[local.protected.php](#)

```
<?php
/**
 * These settings are "protected" and cannot be overwritten
 * by the configuration manager, but need to be edited manually.
 */

$config['savedir'] = DOKU_CONF.'../data';
$config['updatecheck'] = 0;
// if you're using the .htaccess base setup, set this to your animal's
base directory:
$config['basedir'] = '/wiki/bear/';
```

Therefore I use [Apache's mod_proxy](#) and set up the config in **httpd.conf** on the front-end server:

[httpd.conf](#)

```
<Location /wiki>
    ProxyPass http://11.22.33.44/wikis/farm
    ProxyPassReverse http://11.22.33.44/wikis/farm
</Location>
```

This means that any animals in my farm will always be reverse proxied from the desired front-end address.

This works very well until you perform certain actions (e.g. login, logout, saving edits etc.) Due to the fact that the DokuWiki engine does a redirect at these points, the end-user's browser is sent a HTTP 302 header containing the following:

Location: <http://11.22.33.44/wiki/bear/doku.php?id=blahblahblah>

Of course there is no page at this location on the backend server and the result is a HTTP 404 Page Not Found error message in the user's browser.

The remedy is to use [Apache's mod_headers](#). We amend our code block in **httpd.conf** like so:

[httpd.conf](#)

```
<Location /wiki>  
ProxyPass http://11.22.33.44/wikis/farm  
ProxyPassReverse http://11.22.33.44/wikis/farm  
Header edit Location ^http://11.22.33.44 http://www.example.com  
</Location>
```

The 'Location' response header is now re-written to the correct address and everything works fine.

From: <http://synoinstall-2pkhywzfvulqafp3.direct.quickconnect.to/> - worldwide open-source software

Permanent link: <http://synoinstall-2pkhywzfvulqafp3.direct.quickconnect.to/doku.php?id=wiki:farms:advanced&rev=1736961086>

Last update: **2025/01/15 20:11**

