Also see: Cookbook:Farm Setup By Example, Cookbook:Wiki Farm Alternative

A WikiFarm is a collection of two or more wikis running on the same web server and sharing a set of common components. The t is based on the computing phrase "server farm".

This page provides some background information about WikiFarms and describes how to turn a "normal" configuration into a far by adding a wiki. There are many ways to configure wiki farms; this page describes only one, in an effort to make it as simple as possible for the administrator who is creating a farm for the first time.

Why use a farm?

The primary motivation for using a wiki farm is to reduce the amount of administrative work involved in managing several wikis. In farm, most of the PmWiki code is stored in one place and is shared by all the wikis. An administrator can (for example) upgrade t new version of PmWiki on every wiki in the farm by simply updating the shared components in a single location.

From a reader's point of view, each wiki in a farm is completely independent, and appears as a separate web site. Each wiki in a farm:

- has its own URL, and the URLs can be in different domains
- can have its own look and feel by using different skins
- can have its own add-ons or "recipes" from the Cookbook
- can have its own administrator responsible for local configuration

Why not to use a farm

Because the wikis in a farm are all independent, it is difficult (but not impossible) to provide services that require access to more than one wiki. For example, the PmWiki search function can only search within one wiki. Using a farm as a way of subdividing related content is generally a bad idea. A much better way to subdivide content is to use WikiGroups.

I still can't decide if I need a farm ...

The good news is that you don't have to decide in advance. In fact, the recommended procedure is to first do a "normal" or single installation of PmWiki. Use it for a while. Create pages and edit them. Get to know how to add recipes. Be sure to try out WikiGro (they may be all you need).

Once you have decided that you need another wiki, you have two basic choices:

1. Do a complete installation of PmWiki in a new directory. This gives you two totally independent wikis that are completely self-contained. This is **not** a wiki farm.
2. Create a wiki farm using your existing wiki as the "home wiki" where most of the shared PmWiki components will live.

Choice number 1 can be a good choice for several reasons:

- it is not a wiki farm, and requires no additional administrative knowledge - it's just two installations
- if you decide to move one of the wikis to another server, you can simply copy the wiki directory structure to the second serv and it will work (assuming there is a web server and PHP in place).
- you can run different versions of PmWiki on each wiki (good for testing new versions)
- no matter how badly you mess up one installation, it doesn't affect the other

If you choose to create a wiki farm, then read on ...

Prerequisites

Before you create a farm, make sure that:

- you have a working installation of PmWiki ready to become the home wiki for your farm
- all of the wikis in your farm will be on the same web server
- each wiki will have a unique URL, such as http://www.example.com/wiki1/, http://www.example.com/wiki2/, http://another.example.com/wiki1/ and so on.

Creating the home wiki

You do have a working installation of PmWiki at this point, don't you? That's good, because your existing wiki is about to become the home wiki of your farm.

In the directory that contains your existing wiki, create the file *local/farmconfig.php*. This file is used to hold any local customizations that apply across the whole farm. For example, you could assign an admin password in *farmconfig.php* that will be used by all of the wikis in your farm.

If the URL used to access your existing wiki is http://www.example.com/pmwiki/ then a minimal *farmconfig.php* file would look like this:

```
<?php if (!defined('PmWiki')) exit();
$FarmPubDirUrl = 'http://www.example.com/pmwiki/pub';
```

This loads the variable `$FarmPubDirUrl` with the URL location of your home wiki's *pub/* directory. All of the wikis in your farm share this *pub/* directory. The *pub/* directory holds skin definitions and GUI-edit buttons to be shared by all the wikis in the farm.

Amazing as it may sound, this completes all of the changes you need to make in order to turn your existing wiki into the home wiki of your farm.

Creating an additional wiki in your farm

1. Create a directory to hold the new wiki. This directory must be web-accessible, just like the directory that holds your home wiki
2. Create a file called *index.php* in the directory with the following contents:

```
    <?php include('path/to/pmwiki.php');
```

This allows your new wiki to share the PmWiki code stored in your home wiki. The *`path/to/pmwiki.php`* is the file path to *pmwiki.php* in your home wiki. Use an absolute file path (`/home/username/pmwiki/pmwiki.php`) or a relative file path (`../pmwiki/pmwiki.php`). Do not use a url path - there should not be an '`http://`' in it anywhere. For a web server running under Windows, you need to use a complete file path as in `C:/Apache Group/Apache2 /www/mynewwiki/`.

3. Open a web browser and browse the URL of the new wiki. This will be a web address starting with '`http://`'. PmWiki will atte to automatically create a writable *wiki.d/* directory where the wiki's pages will be stored. If you see an error message, follow the instructions. If you choose the option for a "slightly more secure installation" be sure to execute both commands.

Your new wiki is now set up, and your farm now contains 2 wikis. To add more wikis, just repeat these 3 steps.

Customization

Each wiki in a farm inherits the settings stored in *farmconfig.php*. Do any customization that you want to apply farm-wide (to all the wikis) in *farmconfig.php*.

Create a *local/* directory within each wiki's directory to hold local customizations that apply only to that wiki. You should at least create the local/config.php file with a new title, like so :

```
<?php if (!defined('PmWiki')) exit();
  ## Title of your farmed wiki
  $WikiTitle = 'New Wiki';
?>
```

Farm-wide customizations are processed before the individual wiki local customizations.

The PmWiki variable *`$FarmD`* points to the directory in which pmwiki.php is installed, and your home wiki, and it is used as a prefix to allow the other wikis to share PmWiki components. For example:

- *`$FarmD`/scripts/* points to the shared *scripts/* directory
- *`$FarmD`/pub/* points to the shared *pub/* directory
- *`$FarmD`/cookbook/* points to the shared *cookbook/* directory

Any Cookbook scripts you include in farmconfig.php must be included with a line such as:

```
include_once("$FarmD/cookbook/scriptfile.php");
```

Note the double quotes - single quotes may work for per farm inclusions, but they will not work for `$FarmD`.

Notes

- The terminology used to describe wiki farms is not used consistently. See WikiFarmTerminology for more info.
- It is important to remember that not all of the recipes in the Cookbook have been written for or tested with farms. Be sure to look for instructions on how to use a recipe on a farm.
- There are many, many more things you can do with farms. Some are described on PmWiki:WikiFarmsAdvanced which also contains links to step-by-step examples of setting up a farm.

Password use/authorization on farm wikis:

How come when I switch to another wiki within a farm, I keep my same authorization?

PmWiki uses PHP sessions to keep track of authentication/authorization information, and by default PHP sets things up such that all interactions with the same server are considered part of the same session.

An easy way to fix this is to make sure each wiki is using a different cookie name for its session identifier. Near the top of one of the wiki's local/config.php files, before calling authuser or any other recipes, add a line like:

```
    session_name('XYZSESSID');
```

You can pick any alphanumeric name for XYZSESSID; for example, for the cs559-1 wiki you might choose

```
    session_name('CS559SESSID');
```

This will keep the two wikis' sessions independent of each other.

Categories: WikiFarms

Copyright: Petko - 04/10/2009