

Setting up Alien's DokuWiki

Here's what I had to do to get the Wiki up and running. Remember that the DokuWiki has no need for a database server like MySQL. The Wiki pages, indexes and archives are stored as files below the DokuWiki root (or anyplace else if you want to separate the content from the code). This makes it ideal to install in your own home directory. You do not need any additional permissions on the webserver.

I upgraded to dokuwiki-2008-05-05 and decided to re-visit this page as well. Some of the instructions were outdated, and the section about the *blog* plugin has been removed, now that I use WordPress software for [my new blog](#). The [old revisions of this page](#) still contains everything I edited away.

The DokuWiki comes with a default template which is very functional but rather bland-looking. The DokuWiki web page lists a lot of user-contributed templates, some of which are visually more appealing or functionally better. I opted for the *Monobook* template. This is a port from the default [MediaWiki](#) template (as seen on [WikiPedia](#)) which I find very nice looking. This template is actively developed and has many nice features, which make it even better to customize.

Installing the dokuwiki tarball

- Download the latest and greatest DokuWiki archive at the [download page](#). Dokuwiki comes with a clear [installation](#) and [configuration](#) pages. This paragraph will recap some of that information.
- Unpack the tarball into your Apache DocumentRoot, change to the DocumentRoot directory (by default that is `/var/www/htdocs`) and create a symbolic link called `dokuwiki` that points to the fresh DokuWiki directory tree:

```
ln -s dokuwiki-YYYY-MM-DD dokuwiki
```

. If you already have a DokuWiki running, better wait with changing the symlink until you're sure you've configured and upgraded everything right.

- Change into the DokuWiki directory and have a look around.
The `conf` directory is where we will be customizing the new Wiki soon. The `data` directory is where all the Wiki content (pages, indexes, archives, media, cache) will be stored. If anything, your Apache webserver *must* be able to write there. The `conf` and `data` directories are the most important to back up regularly!
- Create an empty logfile

```
touch ./data/changes.log
```

- Configuration.
You can customize the Wiki by overriding any of the values that are found in `./conf/dokuwiki.php`. Just run

```
cd conf
```

```
cp -a local.php.dist local.php
```

and add any line from `dokuwiki.php` that you want to give a custom value. When you want to add user accounts and access control lists, create the two needed configuration files and edit them (or use the Admin interface of your wiki to set this up):

```
cd conf
cp -a users.auth.php.dist users.auth.php
cp -a acl.auth.php.dist acl.auth.php
```

- Finally, make sure that the webserver can read and *write* to your DokuWiki:

```
chown -R apache:wheel .
```

In this example command, I assumed that you are running your Apache as the user “apache” which is the default for Slackware, and that you have root access to the server.

In my case on the Slackware server, I have no privileged account, so I cannot change the ownership of any files. In order to make the Wiki writable by the Apache process, I had to make the `./data` directory writable for “others”:

```
chmod -R o+w /path/to/dokuwiki/data
```

There is also a problem with Dokuwiki's configuration manager. When you change configuration through the admin interface, the file `./conf/local.php` is (re-)written, actually the current file is moved to `local.php.bak` and a *new* file is written. In this process the `umask` settings are not honoured and you end up with a file owned by the web server and read-only to yourself. A small patch is enough to fix this, in the file

`./lib/plugins/config/settings/config.class.php` at the start of function `save_settings()`.

This is my diff:

```
--- ./lib/plugins/config/settings/config.class.php.org 2006-03-23
08:22:32.000000000 -0800
+++ ./lib/plugins/config/settings/config.class.php      2006-03-23
08:23:37.000000000 -0800
@@ -80,6 +80,9 @@
 
         function save_settings($id, $header='', $backup=true) {
 
+            global $conf;
+            umask($conf['umask']);
+
             if ($this->locked) return false;
 
             $file = eval('return '.$this->local_file.';');
```

Now, with a customized umask like `$conf['umask'] = '0111'`; the `local.php` file will have mode 666, so that even while it is owned by the webserver, you can still edit it. These are

the settings I use in my Wiki (you can change these using the configuration manager on the [admin](#) page)

```
$conf['dmode'] = 0777;
$conf['fmode'] = 0666;
$conf['umask'] = '0111';
$conf['dmask'] = '0000';
```



You have to take care with the default umask and dmask settings because you will end up with lots of other files and directories that are owned by the webserver and read-only to you. This has the effect for instance, that you cannot delete the ./data/cache directories and files any longer!

- DokuWiki comes with a plugin manager that according to the documentation is disabled by default - to enable the plugin manager, add the following line to ./conf/local.php

```
$conf['pluginmanager'] = 1; //enable automated plugin management
```

For this to work, the plugin manager (i.e. effectively the web server process) must have write access to the ./lib/plugins directory, so I also did

```
chmod -R a+w ./lib/plugins
```

Template and plugins

Template: MonoBook

- Get the archive file for the monobook template from [Tatewake's web site](#). The template should match with the release of DokuWiki you've just installed. This is only an issue when there has been a major new release of DokuWiki. Usually you would just grab the most recent monobook release.
- Unzip the archive into the DokuWiki template directory:

```
unzip monobook-ddmmyyyy.zip -d /path/to/dokuwiki/lib/tpl/
```

- Fix the permissions and ownership of the monobook directory tree, so that the webserver can at least read them.

Plugin: note

I use the “note” plugin for DokuWiki. This plugin displays nice informational icons in the margin, like “tip” “warning” “information” and “note”. The plugin source is available at [the DokuWiki site](#). The initial version did not do what I wanted from it, plus it required you to copy the images into your

template directory, and add the CSS to the template's stylesheet. So I hacked it and my changes were accepted by the plugin author, so that now the plugin is fully contained into the DokuWiki's `./lib/plugins/note/` directory.

An example of this plugin is used a bit higher up in the page (in the *Installing the dokumwiki tarball* section).

Plugin: keyboard

The keyboard plugin enables you to add visual clues to keypress examples. Like, Windows users often need `Ctrl+Alt+Del`!

This plugin automatically adds a button to the button bar of the Wiki editor. If the plugin does not show up after you reload the editor, you may have to purge the wiki's cached javascript code by visiting this link:

```
http://your.domain.com/path/to/dokumwiki/lib/exe/js.php?edit=1&write=1&purge=1
```

Plugin: displaywikipage

This plugin enables you to embed a Wiki page into any other Wiki page. It is used by the Monobook template and can be downloaded from that same website.

Plugin: referrers

The `referrers` plugin (also found on tatewake.com) keeps a list of visitors of the Wiki and particularly the `HTTP_REFERER` value the browser sends to Apache. This is what (part of) the page would look like:

hostname	ip address	referrer
some.hostname.net	xxx.xxx.xxx.xxx	http://www.linuxquestions.org/questions/slackware-14/qemu-and-kqemu-on-slackware-465123/
some.other.hostname.net	yyy.yyy.yyy.yyy	http://www.google.com/search?hl=en&q=boot+usb+stick+fdisk&btnG=Google+Search&aq=f&oq=
another.hostname.net	zzz.zzz.zzz.zzz	http://www.linuxquestions.org/questions/slackware-14/mounting-nfs-482516/

Note: I needed to manually create the `local_prefs.php` file and fill it with defaults, or else the configuration manager would give an error when trying to save the preferences for this plugin:

```
touch ./lib/plugins/referrers/local_pref.php
chmod 666 ./lib/plugins/referrers/local_pref.php
```

and this is what the file looks like (the value for `referrer-ignore` was set through the configuration manager)

```
<?php
// This file is automatically generated
$re_settings['referrer-ignore'] = 'alien.slackbook.org/dokumwiki';
$re_settings['dontcountadmin'] = '';
$re_settings['dontcountusers'] = '';
```

It is required for the plugin to work as intended, to have a non-empty value for `referrer-ignore`!

Plugin: captcha

I needed this plugin to get rid of Wiki spammers. Before I had installed this, spammers would attack my discussion pages on a daily basis. Get it at <http://www.dokuwiki.org/plugin:captcha>

Plugin: odt

This plugin enables you to export any wiki page as an OpenOffice.org text document. Get it at <http://www.dokuwiki.org/plugin:odt> - you may have to watch out for plugins that expand the wiki syntax - such a plugin needs to be able to generate OpenOffice code. For instance, I use the *note* plugin, and the odt plugin editor wrote a patch for the *note* plugin to make it support *odt*. In the [next section](#) I have included a patch for the Monobook template's toolbox to make it contain an “OpenOffice export” link. I also patched the plugin so that you can define an “*odt template*” document which will be used as the basis for the generated .ODT document. My own template is here: [wiki-template.odt](#). The patch I used is

```
diff -uarN odt.orig/conf/default.php odt/conf/default.php
--- odt.orig/conf/default.php 2008-05-07 16:20:34.000000000 +0000
+++ odt/conf/default.php 2008-11-19 16:12:34.657733978 +0000
@@ -5,3 +5,7 @@
// Directory of the templates in the media manager
$conf['tpl_dir'] = 'odt';
+
+// Leave empty in order to not use any template
+$conf['tpl_template'] = '';
+
diff -uarN odt.orig/conf/metadata.php odt/conf/metadata.php
--- odt.orig/conf/metadata.php 2008-05-07 16:20:34.000000000 +0000
+++ odt/conf/metadata.php 2008-11-19 16:06:09.217839938 +0000
@@ -6,3 +6,4 @@
 */
$meta['tpl_dir'] = array('string');
+$meta['tpl_template'] = array('string');
diff -uarN odt.orig/renderer.php odt/renderer.php
--- odt.orig/renderer.php 2008-05-07 16:20:34.000000000 +0000
+++ odt/renderer.php 2008-11-19 16:12:15.285718568 +0000
@@ -247,7 +247,11 @@
    * Closes the document
    */
    function document_end(){
-        if ($this->template) { // template chosen
+        if (empty($this->template)) { // if no template chosen in current
page, get default template from config
+            $this->template=$this->getConf("tpl_template");
+        }
+
+        if (!empty($this->template)) { // template chosen

```

```

    if
(file_exists(DOKU_INC.'data/media/'.$this->getConf("tpl_dir")."/".$this->tem
plate)) { //template found
    $this->document_end_template();
} else { // template chosen but not found : warn the user and
use the default template

```

and then you can define a template in the wiki's configuration manager.

Tuning the look and feel

- The default bitmap that comes with DokuWiki (the one you see in the top left of every page) did not fit with the look and feel I wanted for my site. So I used the “blue orb” version of the Slackware “S” logo I found at [KDE-look](#) because that one is transparent and the blue “S” logo on the [Slackware web site](#) was not (and the orb is *much* nicer to look at).

I ran these commands to download it and convert it to a 128×128 pixel bitmap:

```

wget http://www.kde-look.org/content/pre1/11377-1.png
convert 11377-1.png -resize 128x128 blueorb_128.png
cat blueorb_128.png > /path/to/dokuwiki/lib/tpl/monobook-yyyy-dmm-
dd/user/logo.png

```

Note that the monobook template redefines the location of the site's logo bitmap. The default bitmap for dokuwiki is ./data/media/wiki/dokuwiki-128.png

- I added my own links to the “navigation” and “toolbox” boxes on the left. The “navigation” box actually is a separate wiki page, it's location is defined in the monobook section in the wiki's configuration manager. I liked the old php array better, but o well... The “toolbox” is actually hard-coded into lib/tpl/monobook-yyyy-mm-dd/context.php and still uses the php array. I added a few items to the default toolbox:

```

--- context.php.orig      2008-07-30 03:49:22.000000000 +0000
+++ context.php 2008-09-26 14:40:39.377450051 +0000
@@ -357,6 +357,14 @@
    $monobook['defaulttoolbox']['whatlinkshere']['text'] =
$lang['monobook_whatlinkshere'];
    $monobook['defaulttoolbox']['whatlinkshere']['rel'] = "nofollow";

+$monobook['defaulttoolbox']['related']['href'] =
DOKU_BASE."doku.php?id=".$ID."&do=recent";
+$monobook['defaulttoolbox']['related']['text'] = "Related changes";
+$monobook['defaulttoolbox']['related']['rel'] = "nofollow";
+
+$monobook['defaulttoolbox']['wikisyn']['href'] =
DOKU_BASE."doku.php?id=wiki:syntax";
+$monobook['defaulttoolbox']['wikisyn']['text'] = "Wiki syntax";
+$monobook['defaulttoolbox']['wikisyn']['rel'] = "nofollow";
+
    $monobook['defaulttoolbox']['upload']['href'] =
DOKU_BASE."lib/exe/mediamanager.php?ns=".getNS($ID);

```

```

$monobook['defaulttoolbox']['upload']['text'] =
$lang['monobook_upload'];
$monobook['defaulttoolbox']['upload']['rel'] = "nofollow";
@@ -391,4 +399,12 @@
$monobook['defaulttoolbox']['cite']['text'] = $lang['monobook_cite'];
$monobook['defaulttoolbox']['cite']['rel'] = "nofollow";

-?>
\ No newline at end of file
+if (isset($_REQUEST['rev'])) {
+  $monobook['defaulttoolbox']['odt']['href'] =
exportlink($ID."&rev=".requestREV, 'odt');
+else
+  $monobook['defaulttoolbox']['odt']['href'] = exportlink($ID,
'odt');
+
+$monobook['defaulttoolbox']['odt']['text'] = "OpenOffice export";
+$monobook['defaulttoolbox']['odt']['rel'] = "nofollow";
+
+?>

```

- To silence Apache in it's error_log, I also copied a favicon.ico file to the DokuWiki image library:

```
cp /path/to/documentroot/favicon.ico /path/to/dokuwiki/lib/images/
```

Adding an editor toolbar button

The *editor toolbar* is the array of buttons that is displayed after you open a Wiki page in edit mode ("edit this page"). It can help you inserting markup tags like italic, bold, hyperlinks, images etc... comfortably.

I missed a toolbar button for quickly inserting `<code></code>` tags. This is what is needed to add such a button to the toolbar:

- Create a toolbar button (dimension 16×16 pixels) and call it "code.png" This is the resulting bitmap: [#]
- Copy the newly created "code.png" button image to `./lib/images/toolbar/`
- Add this array element for the new button to the file `./inc/toolbar.php`:

```

array(
    'type'    => 'format',
    'title'   => $lang['qb_syntax'],
    'icon'    => 'code.png',
    'key'     => 's',
    'open'    => '<code>',
    'close'   => '</code>',
),

```

- Add this text to the english language file `inc/lang/en/lang.php` (for other languages the corresponding texts have not yet been updated):

```
$lang['qb_syntax'] = 'Syntax Code';
```

And voilá!

From:
<https://wiki.alienbase.nl/> - Alien's Wiki

Permanent link:
<https://wiki.alienbase.nl/doku.php?id=slackware:wikinotes>



Last update: **2009/06/19 19:40**