You may have many documents that you would like to use a local program to format in a format PmWiki can display.

You could open each document and copy/paste the content to new pmwiki pages or you could format the document in advance a upload it using a FTP client.

Only two lines are necessary in a PmWiki page file:

```
version=pmwiki-2.1.0 urlencoded=1
text=Markup text
```

"version=" tells PmWiki that the values are urlencoded. The actual value doesn't matter, as long as "urlencoded=1" appears somewhere in the line.

"text=" *needs* to have the markup text with newlines converted to "%0a" and percent signs converted to "%25".

In addition, *PmWiki writes* pages with '<' encoded as "%3c" (to help with security), but it doesn't require that <'s be encoded that in order to be able to read the page. More conversions are possible to be added in the future.

In order to let the `(:pagelist :)` markup work, make sure the filename begins with an uppercase letter.

In order to have the `(:pagelist link= ... :)` markup on other pages list this page, a third attribute is required:

```
targets=GroupName1.Pagename1,GroupName2.Pagename2,...
```

"targets=" is a comma delimited list of all links from the current page.

Keys you could see in a raw PmWiki file:

version

Version of PmWiki used to create the file More??? (ordered, urlencoded)

agent

Author's browser when saving the page

author

Last author to save page

charset

The character encoding of the page text, may be used by future upgrades

csum

Change summary

ctime

Page creation time

host

Host created this page

name

Name of the page (e.g., `Main.WikiSandbox`)

rev

Number of times the page has been edited

targets

Targets for links in the page

text

The page's wiki markup

time

Time the page was last saved (seconds since 1 Jan 1970 00:00 UTC)

title

Page title set via `(:title The Page Title:)`.

newline

Character used for newlines (deprecated)

passwdedit

Encrypted version of the password
Below these you will see information used to keep track of the page's revision history.

updatedto

Creating a Page for Distribution
The version to which PmWiki has been updated to by `upgrades.php` (only on <u>SiteAdmin.Status</u>)

A simple way to create a wikipage file to use for distribution (for example with a recipe or a skin) is to create the page with PmWiki and then use a text editor to delete all lines but *version*, *text*, and *ctime*. Example:

```
version=pmwiki-2.1.0 ordered=1 urlencoded=1
text=This is a line.%0aThis is another.
ctime=1142030000
```

Keeping track of page history

Inside of a page file, PmWiki stores the latest version of the markup text, and uses this to render the page. The page history is kept as a sequence of differences between the latest version of the page and each previous version.

PmWiki normally puts the page history at the end of each page file in reverse chronological sequence, and sets the "ordered=1" items in the header. If an operation needs only the most recent version of a page, then PmWiki will stop reading and processing a page file at the point where the history begins, potentially saving a lot of time and memory. If the "ordered=1" flag isn't present, PmWiki makes no assumptions about the ordering of items in the pagefile and processes the entire file.

Load pages from text files

See [Cookbook: Import text](#).

Unix utility to extract wiki text

The following unix script (tested on MacOSX) will extract and decode the current text from a wiki file:

```
#!/bin/tcsh
# wtext - extract wiki text
#
# wtext wikifile > output

set fn = "$1"
if ("$fn" == "") then
  echo "need input file parameter"
  exit 999
endif
if (! -f $fn) then
  echo "$fn does not exist"
  exit 999
endif
rm sedin.$$ >& /dev/null
set ch = `grep ^newline= $fn | cut -d= -f2`
if ("$ch" == "") set ch = "%0a"
cat <<eof > sedin.$$
s/^text=//
s/$ch/\
/g
s/%3c/</g
s/%25/%/g
eof
grep "^text=" "$1" | sed -f sedin.$$
rm sedin.$$ >& /dev/null
```

See also

- Cookbook:AdminByShell A collection of ways to assist sysadmin of pmwiki using shell tools
- Cookbook:PageTopStore A PageStore alternative which doesn't mangle page contents when viewed outside PmWiki

Copyright: Petko - 26/08/2009