dotclear to dokuwiki

php script for migrating posts

```
#!/usr/bin/php
<?php
// dc2dw.php x

// Use at your own risk! No warranty implied!
// Before using you need to remove the space from line '< /code>'
// it is only included to be able to show the source here in DokuWiki

//source: https://www.dokuwiki.org/tips:ewiki2doku
//adapted: fradeff@akademia.ch

// Open connection
try
{
    $pdo = new PDO('mysql:host=localhost;dbname=dc', 'dc', 'dc');

}
catch (PDOException $e)
{
    echo 'Error: ' . $e->getMessage();
    exit();
}

// Run Query
$sql    = 'SELECT
post_id AS id,
post_dt AS date,
post_title AS titre,
post_excerpt_xhtml AS chapo,
post_content_xhtml AS contenu,
post_meta AS tag
FROM dc_post
WHERE blog_id="default"
AND post_meta IS NOT NULL
ORDER BY post_id

';
//LIMIT 900,100
$stmt  = $pdo->prepare($sql); // Prevent MySQl injection. $stmt means
statement
$stmt->execute();

// Turn off all error reporting
error_reporting(0);
```

```
while ($row = $stmt->fetch())
{
//a:1:{s:3:"tag";a:3:{i:0;s:11:"complotisme";i:1;s:11:"consternant";i:2;s:9:
"politique";}}
  //a:1:{s:3:"tag";a:1:{i:0;s:9:"politique";}}
$tag=  $row[tag];
//$tag=preg_replace("/^(.*)\"tag\";[a-z]:[0-9]\{/", "", $tag);
  $tag=preg_replace("/^(.*)\"tag/", "", $tag);
  $tag=preg_replace("/\";[a-z]:[0-9]:\{[a-z]:[0-9];/", "", $tag);
  $tag=preg_replace("/;\}\}$/", "", $tag);
  $tag=preg_replace("/s:[0-9]:/", "", $tag);
  $tag=preg_replace("/^s:[0-9]*:/", "", $tag);
  $tag=preg_replace("/;[a-z]*:[0-9]*;/", "", $tag);
  $tag=preg_replace("/s:[0-9]*/", "", $tag);
  $tag=preg_replace("\":\"", " ", $tag);
  $tag=str_replace('"', " ", $tag);
  $tag=preg_replace("/  */", " ", $tag);
  $tag=str_replace("a 0 {}", "", $tag);
    $tag=preg_replace("/<.*/", "", $tag);
    $tag=str_replace("geekeries - linuxeries", "geekeries", $tag);
    $tag=str_replace("big brother", "bigBrother", $tag);
    $tag=str_replace("souveraineté alimentaire", "souveraineteAlimentaire",
$tag);
    $tag=str_replace("tag> ", "tag>", $tag);
    $tag=str_replace(" }}", "}}", $tag);
$tag=trim($tag);
  //$tag=preg_replace("\" \"", " ", $tag);

$billet="====== " .$row['titre'] ." ======
~~META:
date created = " .$row['date'] ."
~~
{{tag>" .$tag ."}}

<html>"
. $row['chapo']
." "
.$row ['contenu']
."
</html>

 ";
    echo $billet;
  //echo "$tag";
  echo "\n";
$file="file".$row['id'].".txt";
$fw = fopen($file, "w");
  if ($fw) {
    fwrite($fw, $billet);
  }
```

```
  fclose($fw);

}

// Close connection
$pdo = null;
exit;

//check command line parameters
if ($argc != 3 || in_array($argv[1], array('--help', '-help', '-h', '-?')))
{
  echo "\n  Converts all files from given directory\n";
  echo "  from DotClear to DokuWiki syntax. NOT RECURSIV\n\n";
  echo "  Usage:\n";
  echo "  ".$argv[0]." <dotclearDbName> <output dir>\n\n";
}
else {
  //get input and output directories
  $inDir = realpath($argv[1]) or die("input dir error");
  $outDir = realpath($argv[2]) or die("output dir error");
  //just print information
  echo "\nInput Directory: ".$inDir."\n";
  echo "Output Directory: ".$outDir."\n\n";

  //get all files from directory
  if (is_dir($inDir)) {
    $files = filesFromDir($inDir);
  }

  //migrate each file
  foreach ($files As $file) {
    //convert filename
    $ofile = convFileNames($file);
    //just print information
    echo "Migrating from ".$inDir."/".$file." to ".$outDir."/".$ofile."\n";

    //read input file
    $text = readFl($inDir."/".$file);

    //convert content
    $text = ewiki2doku($text);

    //encode in utf8
    $text = utf8_encode($text);

    //write output file
    writeFl($outDir."/".$ofile, $text);
  }
}

function ewiki2doku($text) {
```

```
  //line by line
  $lines = explode("\n", $text);
  foreach($lines As $line) {
    //start converting
    $find = Array(
       '/\[notify: ?[^ ]*\]/',         //remove [notify:...]
       '/\[jump:([^]]+)\]/',           //[jump:...]
       '/<\?plugin *settitle(.*)\?>/i', //sort of a heading 1
       '/^    *([^ ])/',              //indented paragraphs (we always used
4 spaces but also [tab] is allowed
       '/%%%/',                        //newline
       '/([^!~=|[])(\b[A-Z]+[a-z]+[A-Z][A-Za-z]*\b):(\b[A-Z]+[a-z]+[A-Z][A-
Za-z]*\b)(([^]|#])|$)/',
                                       //CamelCase InterWiki link
       '/([^-!~=|>&[])(\b[A-Z]+[a-z]+[A-Z][A-Za-z]*\b)(([^]|#>])|$)/',
//CamelCase, dont change if CamelCase is in InternalLink
       '/([^!~]|^)\[([^] |[]+)\]/',    //internal link
       '/\[([^]|[]+)\|([^]|[]+)\]/',   //external links and links with |
       '/\["([^"]+)" ([^ ]+)\]/',      //Ewiki ["..." ...] style links ([...
"..."] not recognized)
       '/\[\[([^ :]+):([^]\/@]+)\]\]/', //InterWiki link (the /@ tries to
exclude http:// and mailto:)
       '/\[\[([^] |[]+)\.(png|jpe?g|gif))\]\]/', //image link (only some)
       '/<pre>/',                      //pre open
       '/<\/pre>/',                    //pre close
       '/^\* /',                       //lists 1
       '/^\*\* /',                     //lists 2
       '/^\*\*\* /',                   //lists 3
       '/^# /',                        //ordered lists 1
       '/^## /',                       //ordered lists 2
       '/^### /',                      //ordered lists 3
       '/^!{3} ?(.*)$/',               //heading 1
       '/^!{2} ?(.*)$/',               //heading 2
       '/^!{1} ?(.*)$/',               //heading 3
       '/__([^_]+)__/',                //bold 1
       '/\*\*([^*]+)\*\*/',            //bold 2
       '/\'\'([^\']+)\'\'/',           //italic (emphasize)
       '/==([^= ][^=]+)|[^=])==/',     //monospaced (also taking care of
==X==)
       '/<tt>(.+)<\/tt>/',             //teletype
       '/##([^#]+)##/',                //big text
       '/µµ([^µ]+)µµ/',                //small text
       '/[!~](\b[A-Z]+[a-z]+[A-Z][A-Za-z]*\b)/', //~CamelCase + !CamelCase
       '/[!~](\[[^][]+\])/',           //~[text] + !text (just remove ~ and
!)
       '/<cc>([A-Z]+[a-z]+[A-Z][A-Za-z>]*)<\/cc>/', //CamelCase, dont change
if CamelCase is in InternalLink
       '/^(=+ .*)\[\[(.*)\]\](.* =+)$/',   //remove links in headlines
       '/<([-A-Za-z0-9+_.]+@[-A-Za-z0-9_]+\.[-A-Za-z0-9_.]+[A-Za-z])>/',
//<email> addresses
```

```
        '/([^<:!~]|^)(\b[-A-Za-z0-9+_.]+@[-A-Za-z0-9_]+\.[-A-Za-z0-9_.]+[A-
Za-z]\b)([^>]|$)/', //email addresses
        '/^keywords: /',                //misc1
        '/\[\[ManPages>/',              //misc2
        '/\[\[WikiPedia>/',             //misc3
        '/\[\[FooBarWiki>/'             //misc4
        );
    $replace = Array(
        '',                             //remove [notify:...]
        'Please go to [${1}]',          //[jump:...]
        '====== ${1} ======',           //heading 1 (from plugin settitle)
        '> ${1}',                       //indented paragraphs
        '\\\\\\ ',                      //newline
        '${1}<cc>${2}>${3}</cc>${4}',   //CamelCase InterWiki link
        '${1}<cc>${2}</cc>${3}',        //CamelCase (preparation, see below
for finish)
        '${1}[[${2}]]',                 //internal link
        '[[${2}|${1}]]',                //external link and links with |
        '[[${2}|${1}]]',                //Ewiki ["..." ...] style links
        '[[${1}>${2}]]',                //InterWiki link
        '{{${1}}}',                     //images link
        '<code>',                       //(<pre>) code open
        '< /code>',                     //(</pre>)code close - remove space
between < and /, it is included for viewing in dokuwiki
        '  * ',                         //lists 1
        '    * ',                       //lists 2
        '      * ',                     //lists 3
        '  - ',                         //ordered lists 1
        '    - ',                       //ordered lists 2
        '      - ',                     //ordered lists 3
        '====== ${1} ======',           //heading 1
        '===== ${1} =====',             //heading 2
        '==== ${1} ====',               //heading 3
        '**${1}**',                     //bold 1
        '**${1}**',                     //bold 2
        '//${1}//',                     //italic (emphasize)
        '\'\'${1}\'\'',                 //monospaced
        '\'\'${1}\'\'',                 //teletype
        '**${1}**',                     //big text -- no markup in dokuwiki
        '${1}',                         //small text -- no markup in dokuwiki
        '${1}',                         //~CamelCase + !CamelCase
        '${1}',                         //~[text] + !text (just remove ~ and
!)
        '[[${1}]]',                     //CamelCase, finish
<cc>CamelCase</cc>
        '${1}${2}${3}',                 //remove links in headlines
        '${1}',                         //<email> addresses
        '${1}<${2}>${3}',               //email addresses
        '**keywords:** ',               //misc1
        '[[man>',                       //misc2
        '[[wp>',                        //misc3
```

```
        '[[FooBarWiki>'                 //misc4
        );
    $line = preg_replace($find,$replace,$line);

    $ret = $ret.$line."\n";
  }
  return $ret;
}

function convFileNames($name) {
  /* ö,ä,ü, ,. and more
  */
  $find = Array('/_20/',
                '/_5f/',
                '/_2e/',
                '/_c4/',
                '/_f6/',
                '/_fc/',
                '/_26/',
                '/_2d/'
                );
  $replace = Array('_',
                   '_',
                   '_',
                   'ae',
                   'oe',
                   'ue',
                   '_',
                   '-'
                   );
  $name = preg_replace($find,$replace,$name);
  $name = strtolower($name);
  return $name.".txt";
}


function filesFromDir($dir) {
  $files = Array();
  $handle=opendir($dir);
  while ($file = readdir ($handle)) {
     if ($file != "." && $file != ".." && !is_dir($dir."/".$file)) {
         array_push($files, $file);
     }
  }
  closedir($handle);
  return $files;
}

function readFl($file) {
  $fr = fopen($file,"r");
  if ($fr) {
```

```
    while(!feof($fr)) {
      $text = $text.fgets($fr);
    }
    fclose($fr);
  }
  return $text;
}

function writeFl($file, $text) {
  $fw = fopen($file, "w");
  if ($fw) {
    fwrite($fw, $text);
  }
  fclose($fw);
}

?>
```

hack on dw to integrate dotclear categories

edit doku.php and add at the beginning of the file:

```
/*
hack radeff for redirecting old dotclear links
ex.:
http://radeff.red/cms/dotclear/index.php?category/v%C3%A9lo
->
https://radeff.red/blog/tag/velo?do=showtag&tag=v%C3%A9lo
*/
//echo phpinfo(); exit;
if(preg_match("/index\.php\?category/",$_SERVER['REQUEST_URI'])){
  $url=$_SERVER['REQUEST_URI'];
  $url=preg_replace("/.*category\//","",$url);
  $url="/blog/tag/velo?do=showtag&tag=".$url;
  header("Location: ".$url);
  exit;
}
```



caution: make a backup, otherwise on next upgrade you'll loose your hack!

search engine

add this in your .htaccess

```
#redirection ancien moteur de recherche dotclear
```

```
RewriteRule ^index.php?(.*)   doku.php?id=start&do=search&$1 [QSA,L]
```

From:
https://radeff.red/dokuwiki/ - **Radeff's Wiki**

Permanent link:
https://radeff.red/dokuwiki/info:dc2dw

Last update: **2019/02/01 08:04**