

UserPageclbustos

Guys:

I want to port some plugins from phpWiki to Tiki.

You can find the examples on <http://www.apsique.virtuabyte.cl/tiki-index.php?page=NewPluginTests>

Changes in tikilib

First, I find that TikiLib::list_pages() doesn't accept an array of pages, so I do this modification at line 3422. This modification doesn't break the implementation, only expand it.

```
if (is_array($find)) { $mid = " where `pageName` IN ('".implode("'", $find). "')"; $bindvars = array(); }
elseif (is_string($find)) { $mid = " where `pageName` like ? "; $bindvars = array('%' . $find . '%'); }
else { $mid = ""; $bindvars = array(); }
```

wikiplugin_backlinks.php

```
<?php /** * Backlinks plugin * List all pages which link to specific pages (same as tiki-backlinks.php) *
*/ function wikiplugin_backlinks_help() { return tra("List all pages which link to specific pages").":<br
/>{BACKLINKS(info=>hits+user,exclude=>HomePage+SandBox,include_self=>1,noheader=>0,page
=>HomePage)}{BACKLINKS}"; } /** * Create a list of backlinks to a page * @param string not used *
@param array // info (allows multiple columns, joined by '+') | *
info=hits,lastModif,user,ip,len,comment, creator, version, * flag, versions,links,backlinks * // exclude
(allows multiple pagenames)|exclude=HomePage+RecentChanges * // include_self = by default, false *
// noheader = by default, false * // page = by default, the current page. * @return string */ function
wikiplugin_backlinks($data, $params) { global $wikilib; $aInfoPresetNames = array( "hits" => "Hits",
"lastModif" => "Last mod", "user" => "Last author", "len" => "Size", "comment" => "Com", "creator"
=> "Creator", "version" => "Last ver", "flag" => "Status", "versions" => "Vers", "links" => "Links",
"backlinks" => "Backlinks"); $aInfoPreset = array_keys($aInfoPresetNames); extract ($params);
//////////////////// // Default values ////////////////////// // $page = isset($page)? $page :
$_REQUEST["page"]; // The page to request $include_self = isset($include_self)?(bool)$include_self :
false; // Include $page in the Backlinks $noheader = isset($noheader)?(bool)$noheader : 0; // Include a
header, with the name of the page and numbers of Backlinks $exclude = isset($exclude)?explode("+",
$exclude) : array(); // List of pages to exclude, separated by '+' // ////////////////////// // Create a valid
list for $info ////////////////////// // if (isset($info)) { $info = explode("+", $info); $info_temp =
array(); foreach($info as $sInfo) { if (in_array(trim($sInfo), $aInfoPreset)) { $info_temp[] =
trim($sInfo); } $info = $info_temp?$info_temp: false; } } else { $info = false; } // //////////////////////
// Process backlinks ////////////////////// // $backlinks = $wikilib->get_backlinks($page); if
(!isset($include_self)) { $exclude[] = $page; } foreach($backlinks as $backlink) { if
(!array_search($backlink["fromPage"], $exclude)) { $aBackRequest[] = $backlink["fromPage"]; } } if
($include_self) { $aBackRequest[] = $page; } $sOutput = ""; $aInfo = $wikilib->list_pages(0, -1,
'pageName_desc', $aBackRequest); // ////////////////////// // Start of Output ////////////////////// //
if (!$noheader) { // Create header $count = $aInfo["cant"]; if (!$count) { $sOutput .= "No pages link to
(($page))"; } elseif ($count == 1) { $sOutput .= "One page links to (($page))"; } else { $sOutput =
"$count pages links to (($page))"; } $sOutput .= "\n"; } if ($info) { // Header for info $sOutput .=
"<table class='normal'><tr><td class='heading'>".tra("Page")."</td>"; foreach($info as $iInfo =>
$sHeader) { $sOutput .= "<td class='heading'>".tra($sHeader)."</td>"; } $sOutput .= "</tr>"; }
foreach($aInfo["data"] as $aPage) { // Loop of Backlinks if (!$info) { $sOutput .=
"*((".$aPage["pageName"]."))\n"; } else { $sOutput .= "<tr><td>((".$aPage["pageName"]."))</td>";
foreach($info as $sInfo) { if (isset($aPage[trim($sInfo)])) { $sOutput .=
"<td>".$aPage[trim($sInfo)]."</td>"; } } } if ($info) { if ($info) { $sOutput .= "</table>"; } } return
$sOutput; } ?>
```

```

<?php /** * Title Search Plugin * Search the titles of all pages in this wiki */ function
wikiplugin_titlesearch_help() { return tra("Search the titles of all pages in this wiki").":<br
/>{TITLESEARCH(search=>Admin,info=>hits+user,exclude=>HomePage+SandBox,noheader=>0,)}
{TITLESEARCH}"; } function wikiplugin_titlesearch($data, $params) { global $wikilib;
$aInfoPresetNames = array( "hits" => "Hits", "lastModif" => "Last mod", "user" => "Last author",
"len" => "Size", "comment" => "Com", "creator" => "Creator", "version" => "Last ver", "flag" =>
"Status", "versions" => "Vers", "links" => "Links", "backlinks" => "Backlinks"); $aInfoPreset =
array_keys($aInfoPresetNames); extract ($params); // Default values
// if (!isset($search)) { return ("You have to define a search"); } $noheader =
isset($noheader)?(bool)$noheader : 0; // Include a header, with the search and number of pages
$exclude = isset($exclude)?explode("+", $exclude) : array(); // List of pages to exclude, separated by
'+' // Create a valid list for $info // if (isset($info)) { $info
= explode("+", $info); $info_temp = array(); foreach($info as $sInfo) { if (in_array(trim($sInfo),
$aInfoPreset)) { $info_temp[] = trim($sInfo); } $info = $info_temp?$info_temp: false; } } else { $info =
false; } // Process pages // $sOutput = ""; $aInfo =
$wikilib->list_pages(0, -1, 'pageName_desc', $search); foreach($aInfo["data"] as $idPage=>$aPage) {
if(in_array($aPage["pageName"],$exclude)) { unset($aInfo["data"][$idPage]); $aInfo["cant"]--; } //
// Start of Output // if (!$noheader) { // Create header
$count = $aInfo["cant"]; if (!$count) { $sOutput .= tra("No pages found for title search")."
'__$search.__"; } elseif ($count == 1) { $sOutput .= tra("One page found for title search")."
'__$search.__"; } else { $sOutput = "$count".tra(" pages found for title search")." '__$search.__"; }
$sOutput .= "\n"; } if ($info) { // Header for info $sOutput .= "<table class='normal'><tr><td
class='heading'>".tra("Page")."</td>"; foreach($info as $iInfo => $sHeader) { $sOutput .= "<td
class='heading'>".tra($sHeader)."</td>"; } $sOutput .= "</tr>"; } foreach($aInfo["data"] as $aPage) {
// Loop of Backlinks if (!$info) { $sOutput .= "*"($.aPage["pageName"].")\n"; } else { $sOutput .=
"<tr><td>($.aPage["pageName"].")</td>"; foreach($info as $sInfo) { if (isset($aPage[trim($sInfo)]))
{ $sOutput .= "<td>$.aPage[trim($sInfo)].</td>"; } } } if ($info) { if ($info) { $sOutput .=
"</table>"; } } return $sOutput; } ?>

```