From EULER Project

PmWiki: Fmt Page Name

This page describes an internal function in PmWiki's engine called FmtPageName(). The contents are not intended for those with a weak heart ;-)

Also see: <u>PmWiki.Functions</u>

`FmtPageName($fmt, $pagename)`

Returns `$fmt`, with $variable and internationalisation substitutions performed, under the assumption that the current page is `pagename`. As a rule is used to pre-process all variables which by convention have a "Fmt" suffix (like <u>$GroupFooterFmt</u>), but also other strings that require <u>interpolation</u>, notably the page template (.tmpl) file. See <u>PmWiki.Variables</u> for an (incomplete) list of available variables, <u>PmWiki.Internationalizations</u> for internationalisation.

The function `FmtPageName()` applies internationalization-substitutions and $Variable-substitions to the string $fmt under the assumption that the current page is $pagename.

The substitutions go as follows:

1. Replace any sequences of the form `$XyzFmt` with value of any corresponding global variable.
2. Process the string for any `$[...]` phrases (internationalized phrase), using the currently loaded translation tables.
3. Replace any instances of `{$ScriptUrl}` with `$ScriptUrl` (to defer processing to the URI processing phase)
4. Replace any instances of standard <u>Page Variables</u> (beginning with an upper case letter, followed by at least one word character) with their values. If there are no more $-sequences, then return the formatted string and exit the function
5. Perform any pattern replacements from the array <u>$FmtP</u>. Typically this is used to handle things like $Name and $Group etc that are specific to the name of the current page. *?? Appears to be used in robots.php to hide actions from robots.*
6. Replace any remaining instances of Page Variables with their values.
7. If <u>$EnablePathInfo</u> isn't set, convert `URIs` to use the syntax <u>$ScriptUrl</u>?n=<Group>.<Name> instead of <u>$ScriptUrl</u>/<Group>/<Name>. In any case, replace <u>$ScriptUrl</u> with its value. If there are no more $-sequences, then return the formatted string and exit the function
8. Replace any $-sequences with global variables (caching as needed) of the same name (in reverse alphabetical order, and filtering out any unsafe globals) *
9. Replace any $-sequences with values out of the array <u>$FmtV</u>.

Note that `FmtPageName()` is automatically aware of any global variables. However, since modifying global variables may be expensive, the array <u>$FmtV</u> exists as a way to avoid rebuilding the variable cache for values that change frequently.

Security

According to PM, as a general rule it's unwise to be calling FmtPageName() on strings that are coming from page markup, as this exposes the ability for people to view the values of variables that perhaps they shouldn't see. This is also why page variables (which come from markup) use PageVar() and PageTextVar() and don't go through FmtPageName().

Availability of Variables in FmtPageName

To be very specific, here's what Pm wrote regarding different ways of defining a variable that can be used by FmtPageName (when it is formatting a string):

- Set a global variable. FmtPageName() automatically performs substitution on all global variables that aren't arrays. If the variable is going to change value over repeated calls to FmtPageName, it's probably better to use $FmtV as in the next item.
- Set a value in the $FmtV array. $FmtV['$MyVariable']='something' means to replace instances of '$MyVariable' with 'something'. Use this for variables that change value frequently over multiple calls to FmtPageName.
- Set a pattern/replacement in the $FmtP array. This is normally done for substitutions that have to be dynamic somehow based on the pagename being referenced, such as '$Title', '$Group', '$Name', '$PageUrl', etc.

Also see: Cookbook:Functions#FmtPageName

Finally, here's something else Pm wrote that is related and explains why we have this function:

In order to produce its output, PmWiki has to do a variety of string substitutions:

1. Generating the full name, group, title, or url of a page (other than the currently displayed page)
2. Substituting the values of global variables
3. Performing internationalization substitutions
4. Converting $ScriptUrl/$Group/$Name to $ScriptUrl?n=$Group.$Name for sites that cannot handle PATH_INFO urls
5. Other substitutions needed by specific functions

PmWiki centralizes all of that substitute-a-dynamic-value-in-a-string into the FmtPageName() subroutine. Because some things are extremely dynamic, such as the url or group for an arbitrary page that is not the current one, those things cannot be simple global PHP variables. Or, if they do become global variables, they're variables that cannot be trusted to hold a value for very long because some other routine (that may happen to be formatting a string for a different page) will come along and change that global variable for whatever it happens to be doing.

A limited set of $-substitutions -- basically anything that corresponds to a page attribute -- are not PHP variables and are only available through the FmtPageName() subroutine. The complete set of these special substitutions is $Group, $Name, $FullName, $PageUrl, $Title, $Titlespaced, $Namespaced, $Groupspaced, $LastModifiedBy, $LastModifiedHost, and $LastModified. These items cannot just be standard PHP variables because often PmWiki needs to obtain the url, name, group, title, etc. of a page other than the one currently being viewed by a browser.

At the moment, $Title, $LastModfied, $LastModifiedBy, and $LastModifiedHost can only work if the page's attributes have been loaded and cached using the PCache function. So, to get at these values one must typically do:

```
$page = ReadPage($pagename);
PCache($pagename, $page);
$ptitle = FmtPageName('$Title', $pagename);
$pauthor = FmtPageName('$LastModifiedBy', $pagename);
```

Retrieved from https://www-sop.inria.fr/mascotte/EULER/wiki/pmwiki.php/PmWiki/FmtPageName
Page last modified on February 14, 2009, at 08:58 PM