

From EULER Project

PmWiki: Debug Variables

\$EnableDiag

The following actions are available only if you set \$EnableDiag = 1; in your configuration file. They can be used for debugging and should not be set in a production environment.

?action=**ruleset**

displays a list of all markups in 3 columns:

- ◊ column 1 = markup-name (1. parameter of markup())
- ◊ column 2 = when will rule apply (2. parameter of markup())
- ◊ column 3 = PmWiki's internal sort key (derived from #2)

(see Custom Markup Using the Markup() function for custom wiki syntax).

To see more than what ?action=ruleset gives you, apply the Cookbook:MarkupRulesetDebugging recipe: it can also show the pattern and the replacement strings.

- doesn't make use of PmWiki's authorization mechanisms.

?action=**phpinfo**

displays the output of phpinfo() and exits. no page will be processed

- ◊ doesn't make use of PmWiki's authorization mechanisms.

?action=**diag**

displays a dump of all global vars and exits. no page will be processed

- ◊ doesn't make use of PmWiki's authorization mechanisms.

\$EnableIMSCaching

A variable which, when set equal to 1, recognizes the "If-Modified-Since" header coming from browsers and allows browsers to use locally cached pages. Disabled by default to help the administrator customize its page without needing permanent reloading.

\$EnableStopWatch

This activates an internal stopwatch that shows how long it takes to render a page. (If you have a wiki that composes a HTML page from multiple pages, such as a normal layout with a sidebar, you'll get separate timings for each subpage and for the total page.)

The timings can be displayed by adding <!--function:StopWatchHTML 1--> in the wiki template.

Valid values are:

```
$EnableStopWatch = 0; # No timings (the default). No HTML will be generated.
```

```
$EnableStopWatch = 1; # Wall-clock timings only.
```

```
$EnableStopWatch = 2; # Wall-clock and CPU usage timings. Won't work on Windows.
```

See Stopwatch for more details.

See also:

- scripts/refcount.php is useful for debugging
-

Is it possible for someone with admin privileges to always have access to debugging tools, without letting everyone else access them?

You can easily & automatically allow debugging for anyone with admin privileges (meanwhile leaving it off for everyone else) by including this line in config.php - *just be sure that 1) \$EnableDiag is either null or set to 0, and 2) to include it near the end of config.php, AFTER declaring your passwords, and after any AuthUser or other privilege settings:*

```
if (CondAuth($pagename, 'admin')) $EnableDiag = 1; # allows admin to
always call phpinfo, etc
```

Retrieved from <https://www-sop.inria.fr/mascotte/EULER/wiki/pmwiki.php/PmWiki/DebugVariables>
Page last modified on February 14, 2009, at 03:44 PM