

Phorum Support Forums / 5.1 Phorum Hacks

Alternatives to hacking format_messages()

Ambush Commander / June 27, 2007 05:40PM

[Alternatives to hacking format_messages\(\)](#)

Hi, I would like to write a module that enables [HTML Purifier](#) on Phorum posts. After doing some initial work, I have come up with two primary problems with enabling HTML in Phorum posts:

* Phorum performs too much data cleaning in phorum_format_messages() that cannot be turned off. It does me no good if I get fully escaped input when my format hook finally gets to take a whack at it. I need some way to stop this from happening without editing the file.

* I need to add a second, "message_cache" field to the database, because it is prohibitively expensive to be running HTML Purifier continually on post data. While the form of the \$messages array makes it easy to piggy-back another table field off of, during the installation process how would an extra table row be configured?

The second question is simply a matter of convention: does Phorum have any facilities in place for mods that need to make DB schema changes, and are there any recommendations on how to make sure said changes are as compatible as possible.

The first question, I have a few ideas on how this might be worked around. Because I'm caching the fully filtered/formatted output, and that will be in the \$messages array, I could cunningly swap in the cached data into ['body'] when my format hook gets called, and hope hard as hell that the post-cleanup format operations don't do anything too silly to my HTML (phorum break stuff seems innocuous enough). There is no post-format hook, which really ties my hands. Also, I'm not sure this will work very well for preview (although the posting_custom_action hook might save the day), since there will be no cached data to cunningly swap in to begin with! (And there might be other actions that also use format_messages in this manner, it seems that list.php, pm.php, report.php and search.php are subject: is the \$messages array coming from the database or the user? Questions, questions!)

P.S. If this module is successfully written, it means that WYSIWYG editors can be fully integrated with Phorum with no problem at all. :-)

Edited 1 time(s). Last edit at 06/27/2007 06:13PM by Ambush Commander.

Maurice Makaay / June 27, 2007 06:21PM

[Re: Alternatives to hacking format_messages\(\)](#)

There is no facility at all for adding tables to the database tables that Phorum uses. If you want to add fields, it's fully up to you to come up with the db management code and SQL layering. I did some mods where I created extra tables (look at spam hurdles for example), but there it's really a separate data store and not a hacked Phorum table.

If you need to store the data, then you might want to take a look at the "meta" field. This field contains serialized array data and can be used to store any kind of data. Take a look at the topic poll module for example, which uses the meta data for storing poll results. This might be an easier and more compliant way of storing extra data.

Quote

Phorum performs too much data cleaning in phorum_format_messages() that cannot be turned off

There's no such thing as too much in this IMO. Phorum just takes care of safely formatting the code. Although I see how this bites whatever you are doing with HTML code. HTML modules need to convert the formatted code back to HTML to work.

You seem to run into troubles with hook ordering. In 5.1 there is no module / hook ordering system, so you cannot really take care of making your module the very last formatter of the message. In 5.2, this is possible (but of course that version is still under heavy development). One thing you could do is take a look at [this code](#), which I wrote for 5.1 to make bbcode the last formatter in all cases. Maybe you could use some ideas from this to trick Phorum into running your module as the last formatter in the chain? This is currently the best you can do to mimic post formatting processing.

Further note that Phorum 5.2 comes with a caching system for messages. So when moving your module to 5.2, it might no longer be necessary to do your own caching system to make things fast. Keep that in mind for the future.

I hope these remarks helped. Good luck!

Maurice Makaay

Phorum Development Team

[my blog](#) [linkedin profile](#) [secret sauce](#)

Ambush Commander / June 27, 2007 11:41PM

[Re: Alternatives to hacking format_messages\(\)](#)

Thank you for your prompt and informative response.

Quote

If you need to store the data, then you might want to take a look at the "meta" field. This field contains serialized array data and can be used to store any kind of data. Take a look at the topic poll module for example, which uses the meta data for storing poll results. This might be an easier and more compliant way of storing extra data.

Sounds like a deal. serialize/unserialize is quite fast, and there is little to no reason why the HTML needs to be directly accessed by the database (i.e. sorting or fulltext search). I'll have to take care that I can easily invalidate the cache, but that sounds like a plan for now.

Quote

There's no such thing as too much in this IMO. Phorum just takes care of safely formatting the code. Although I see how this bites whatever you are doing with HTML code. HTML modules need to convert the formatted code back to HTML to work.

Taking another look at the format_functions.php code, it seems that the process is quite reversible. De-escape the angled brackets/ampersands and remove the Phorum breaks. While this is not exactly the cleanest way of doing things, it should suffice.

Quote

You seem to run into troubles with hook ordering. In 5.1 there is no module / hook ordering system, so you cannot really take care of making your module the very last formatter of the message. In 5.2, this is possible

(but of course that version is still under heavy development). One thing you could do is take a look at this code, which I wrote for 5.1 to make bbcode the last formatter in all cases. Maybe you could use some ideas from this to trick Phorum into running your module as the last formatter in the chain? This is currently the best you can do to mimic post formatting processing.

Ah, that's quite tricky of you. I would have implemented it slightly differently (instead of swapping the last mod and the bbcode mod, I would have used `array_splice` to extract the bbcode mod and then putting on the end. As far as I can tell, `$idx` is an integer, right? But it certainly suggests a usable technique for bubbling the mod to the end. In fact, we could probably genericize this and have a "sorting mod" for mods, but given that 5.2 will have this functionality I won't bother. ;-)

Quote

Further note that Phorum 5.2 comes with a caching system for messages. So when moving your module to 5.2, it might no longer be necessary to do your own caching system to make things fast. Keep that in mind for the future.

Sounds fantastic. I'll be sure to stay updated.
