

Phorum Support Forums / 5.1 Phorum Hacks

Tagger Hack: Retrieving threads from db

Ambush Commander / September 13, 2007 01:38AM

[Tagger Hack: Retrieving threads from db](#)

I am currently working on a "tagger" mod/hack for Phorum that will enable users/admins to "tag" posts, search for posts with certain tags, and generate "tag clouds", ala del.icio.us. It's a project primarily for my own needs, but if someone else finds it useful, good for them.

I have managed to bang out a working (albeit private) prototype, but I'm not happy with some of the hacks/patches I've needed to make to the source code. The biggest problem involves alterations to the DB handling functions.

In order to allow users to filter threads by tags, I'd like to overload behavior on the List page. However, the list hook gets to the threads too late: I need to modify the SQL itself that is being used to retrieve the threads (I'm doing some fancy table joins and other things in order to retrieve the tags I want). As of right now, I've added in code that injects this extra SQL into the statement generated at `phorum_db_get_thread_list()`, detecting a "tag" get parameter.

It is, consequently, extremely messy, and not very forwards compatible. Is there a better way of doing this, or should I just bite my lip and press forward? Also, why do functions defined by modules not seem to be callable from the DB processor?

[HTML Purifier](#), standards-compliant HTML filtering

Maurice Makaay / September 13, 2007 02:45AM

[Re: Tagger Hack: Retrieving threads from db](#)

It is best not to hack the code, but that is no news to you. What I normally do when the standard db layer misses functionality for a feature that I am building, is create the extra required db code as part of my module. For example the Spam Hurdles module creates an extra table and has some functions for interfacing with that table.

The thing that you are trying to pull off here is much more complicated than the Spam Hurdles db handling though. I think you have two options here: either hack the core code or fully override the full list page. Overriding would mean that you implement the list functionality in full (front end script + db interfacing). That last one solves the problem of having to hack core code, but it does not meet any requirement for being forward compatible.

Isn't it an option to use the search interface for listing messages by tag(s)? You can override the search functionality more easily than the list functionality and to me, tagging sounds more like a search feature than something to do through the list page.

Functions defined from a mod should be accessible from anywhere in PHP. We did not do anything special to disallow this. It might be that at the time that your db code runs, the module has not yet been loaded. Modules only get loaded as their first hook is called.

Maurice Makaay

Phorum Development Team

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

Ambush Commander / September 13, 2007 03:31AM

[Re: Tagger Hack: Retrieving threads from db](#)

Search overloading would work quite well. What I'm concerned about are the little "extras" that are in `phorum_db_get_thread_list()`. For example, the SQL statement for selecting threads contains a few extra gems: it checks for approval status, it checks the sort column, and it uses an index. Presumably, this could expand in the future, so it's slightly non-forwards compatible, but tolerable.

Phorum Support Forums / 5.1 Phorum Hacks

Tagger Hack: Retrieving threads from db

As for the module loading problem, makes sense. I should make a dummy hook function for those pages, then?

[HTML Purifier](#), standards-compliant HTML filtering

Maurice Makaay / September 13, 2007 09:41AM

[Re: Tagger Hack: Retrieving threads from db](#)

You could indeed add a common_pre hook, with an empty function as its implementation. That would make Phorum load the mod at an early stage.

I will try to think about a way to handle the list madness. At this moment I have no really brilliant ideas on how to handle it in a fun way. Dynamically extending the db layer functionality is the problem here. We do not allow hooks in the db layer and really want to keep it that way. I did do some dynamic query building for the "prune messages" functionality, but I don't think that such type of system would be sufficient for the needs of the tagging mod.

Maurice Makaay

Phorum Development Team

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

Joe Curia / September 13, 2007 01:41PM

[Re: Tagger Hack: Retrieving threads from db](#)

This sounds like a problem similar to creating extra user info. Maurice, would it make sense to create a "Custom Message Fields" section much like the custom profile fields that you have already added?

I know that this is more of a feature request for a future phorum version, but I think it makes sense to have custom message fields and they would really add some power to phorum. Just a few uses would be the tags mentioned above, a ratings system (which could even be used a la slashdot.org), or adding set of available icons which a poster could use to highlight or categorize a post.

So I guess this is me officially asking that "Custom Message Fields" be considered for a future phorum release and darn the expense and the hassle :p

Joe Curia (aka Azumandias)

Modules:	Admin Mass Email	Automatic Time Zones	Enhanced Custom
Profiles	Google Calendar	Post Previews	
	Admin Security Suite	Check Modules for Upgrades	External
Authentication	Group Auto-Email	Private Message Alerts	
	Attachment Download Counter	Custom Attachment Icons	Favorite
Forums	Highlighted Search Terms	Self-Delete Posts Option	
	Attachment Watermarks	Custom Language Database	Forum
Lockdown	Ignore Forums	Threaded Tree View	
	Automatic Message Pruning	Easy Color Scheme Manager	Forum
Subscriptions	Moderated User Group		
Templates:	Generic Integration	Simple Rounded	Tabbed Emerald

Thomas Seifert / September 13, 2007 02:33PM

[Re: Tagger Hack: Retrieving threads from db](#)

they are there, its just called meta without needing extra-definition ;).

Thomas Seifert

Phorum Development Team / Mysnip-Solutions.de

[Custom Phorum and general software development](#)
[worry-free Phorum Hosting](#)

Joe Curia / September 13, 2007 03:10PM

[Re: Tagger Hack: Retrieving threads from db](#)

So it should be possible to save the tags to the meta field without disturbing the other meta data? I'm assuming the meta field holds an array which could hold other arrays? This would then be readable without doing anything else to the db layer?

Joe Curia (aka Azumandias)

Modules: [Admin Mass Email](#) [Automatic Time Zones](#) [Enhanced Custom Profiles](#) [Google Calendar](#) [Post Previews](#)
[Admin Security Suite](#) [Check Modules for Upgrades](#) [External Authentication](#) [Group Auto-Email](#) [Private Message Alerts](#)
[Attachment Download Counter](#) [Custom Attachment Icons](#) [Favorite Forums](#) [Highlighted Search Terms](#) [Self-Delete Posts Option](#)
[Attachment Watermarks](#) [Custom Language Database](#) [Forum Lockdown](#) [Ignore Forums](#) [Threaded Tree View](#)
[Automatic Message Pruning](#) [Easy Color Scheme Manager](#) [Forum Subscriptions](#) [Moderated User Group](#)
Templates: [Generic Integration](#) [Simple Rounded](#) [Tabbed Emerald](#)

Ambush Commander / September 13, 2007 04:43PM

[Re: Tagger Hack: Retrieving threads from db](#)

I considered using the meta field, but decided against it. If I had implemented it that way, I would need to do a table scan in order to find the rows that had what I wanted.

[HTML Purifier](#), standards-compliant HTML filtering

Maurice Makaay / September 13, 2007 04:54PM

[Re: Tagger Hack: Retrieving threads from db](#)

Meta data can do that, but the big problem with meta data in this respect, is that you cannot run SQL queries on the meta data. So it is pretty useless when used for tagging.

Maurice Makaay

Phorum Development Team

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

Joe Curia / September 13, 2007 05:08PM

[Re: Tagger Hack: Retrieving threads from db](#)

So would this put us back to implementing a "Custom Message Fields" concept? It would allow each custom field to be queried, correct? What would it take to implement this idea?

Phorum Support Forums / 5.1 Phorum Hacks

Tagger Hack: Retrieving threads from db

Joe Curia (aka Azumandias)

Modules: [Admin Mass Email](#) [Automatic Time Zones](#) [Enhanced Custom Profiles](#)
[Google Calendar](#) [Post Previews](#)
[Admin Security Suite](#) [Check Modules for Upgrades](#) [External](#)
[Authentication](#) [Group Auto-Email](#) [Private Message Alerts](#)
[Attachment Download Counter](#) [Custom Attachment Icons](#) [Favorite](#)
[Forums](#) [Highlighted Search Terms](#) [Self-Delete Posts Option](#)
[Attachment Watermarks](#) [Custom Language Database](#) [Forum](#)
[Lockdown](#) [Ignore Forums](#) [Threaded Tree View](#)
[Automatic Message Pruning](#) [Easy Color Scheme Manager](#) [Forum](#)
[Subscriptions](#) [Moderated User Group](#)
Templates: [Generic Integration](#) [Simple Rounded](#) [Tabbed Emerald](#)

Thomas Seifert / September 13, 2007 05:10PM

[Re: Tagger Hack: Retrieving threads from db](#)

it would simply take too much load to put custom fields for each message in a separate table!
another query for each message or all messages in one retrieval step.

Thomas Seifert

Phorum Development Team / Mysnip-Solutions.de

[Custom Phorum and general software development](#)
[worry-free Phorum Hosting](#)

Maurice Makaay / September 13, 2007 06:24PM

[Re: Tagger Hack: Retrieving threads from db](#)

Even if you would have such a feature, then it would not really help with the tagging project, because the code in the database layer would still not be using it.

If custom storage is needed for a major feature, then you probably want to implement a custom table for that in the database. I can imagine that for tagging, you want to record more than only the tags. I would want to have track of who added the tags and when they did that, to be able to do some tagging abuse prevention.

Maurice Makaay

Phorum Development Team

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

Joe Curia / September 13, 2007 07:22PM

[Re: Tagger Hack: Retrieving threads from db](#)

I think you underestimated the amount of workin I was asking for from you Maurice. I wasn't thinking of adding custom fields as a mod but rather to the core phorum dev. It is more of an idea for future programming than a practical solution now. I do think it would enhance the usability and customizability (word?) of phorum, but I realize it would take quite a bit of work to make it happen.

Joe Curia (aka Azumandias)

Modules: [Admin Mass Email](#) [Automatic Time Zones](#) [Enhanced Custom](#)

Phorum Support Forums / 5.1 Phorum Hacks

Tagger Hack: Retrieving threads from db

Profiles	Google Calendar	Post Previews	
	Admin Security Suite	Check Modules for Upgrades	External
Authentication	Group Auto-Email	Private Message Alerts	
	Attachment Download Counter	Custom Attachment Icons	Favorite
Forums	Highlighted Search Terms	Self-Delete Posts Option	
	Attachment Watermarks	Custom Language Database	Forum
Lockdown	Ignore Forums	Threaded Tree View	
	Automatic Message Pruning	Easy Color Scheme Manager	Forum
Subscriptions	Moderated User Group		
Templates:	Generic Integration	Simple Rounded	Tabbed Emerald

Maurice Makaay / September 13, 2007 07:49PM

[Re: Tagger Hack: Retrieving threads from db](#)

I do understand that you are talking about the core and the amount of work is not the issue here. About the most valid point for not adding another table for linking individual setting fields to a message, is the performance of the system. We do have this system in place for users, but after implementing and running it, we found out that it performs badly. We really needed user caching then to maintain the Phorum speed.

My view is that for most of the fancy features for which you would want to store extra data in the database for a message, a separate table for storage is the best option. Or of course, extend the existing message table with extra fields. That gives full control over the data and gives you possibilities for adding multiple related fields on which you can query and indexes for speedy queries.

For simple message related storage, the meta data is a perfect and fast tool already and it would be bad to change that into a system which slows down Phorum.

Maurice Makaay

Phorum Development Team

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