

Maurice Makaay / September 07, 2006 12:03AM

[Making life easier for module writers](#)

Phorum 5.1.16 has just been released. In this new version, a couple of simple enhancements were implemented to facilitate module writing. Most of the changes were made to allow for writing more self-contained modules that do not need manual intervention at installation/upgrade time for moving script and template files to the right locations within the Phorum directory.

In this post I will explain the new features that can be used.

Templates that are stored within the module directory

It's now possible to easily access the Phorum template system from within your modules. Before, using your own templates could only be done by manually copying the required template files to the appropriate Phorum templates directories. This is no longer needed. Storing and using templates can now be fully contained in a module.

To use this new feature, you have to create a templates directory in your module's directory, which uses the same template structure as Phorum's main templates directory. If you write a module named "foo", which needs a template called "bar.tpl", then you can store this in the file:

```
./mods/foo/templates/default/bar.tpl
```

If you have a main Phorum template named "yourtemplate", for which you need a modified "bar.tpl", you can put the modified template in:

```
./mods/foo/templates/yourtemplate/bar.tpl
```

If a module template is requested, Phorum will automatically check if there is a template file available for the active template. If there is none, then the template file in templates/default will be used. This is a huge improvement over copying the template files to Phorum's templates directory, because now you do not have to copy the file to each template directory that is in use.

After you have created the templates that are used by your module, you can access them by prefixing the template name by "<module name>:". So for the example template from above, you could use the following methods to access the template:

Loading the template from (a module's) PHP code:

```
$template = phorum_get_template("foo::bar");  
print $template;
```

Including the template from another template file:

```
{INCLUDE foo::bar}
```

Addon scripts that can be contained in a module

Addon scripts are scripts that implement functionality that cannot be implemented through hooking into the standard module hook system. Mostly, these are scripts that implement totally new functionality instead of tweaking existing functionality.

A special script was added to the Phorum release, to make it possible to write scripts that act as addon scripts, but which are implemented through a module. The big advantage of doing this is that installing and maintaining becomes much easier for admins that are using your addon. Installing and upgrading will no longer include moving files to the right directories in Phorum. Instead, installing becomes as easy as installing and activating a standard module.

For information on how to use the new `addon.php` script for your addon, please read the `addon.php` script itself. In that file I documented the use of this script.

Better module support for the posting editor

The hook `"posting_custom_action"` was added to the editor code. This hook is at a convenient place for doing all kinds of processing in the message editing process. If you need to handle custom fields that were added to the editor, then this is the hook to use for processing them. Also, if you want to change meta data for a message, then this is the designated hook for doing that.

Which template the editor code shows, is now determined by the variable `$PHORUM["posting_template"]`. This allows for using your own templates for separate posting screens.

I wrote a new poll module from scratch, as a showcase for the new features in the posting editor code and the use of module templates. This module implements the poll editor as a fully separate screen in the posting process. Check out the code of the [Topic Poll Module](#) for an example of how to use the new possibilities.

Future changes

Phorum module writing is still in development and module authors will probably hit some (possibly brick) walls now and then. Those authors are invited to tell us what those walls are, so we can see if adjustments are needed to make Phorum more module friendly. Some things that I have already been thinking about are:

- better documentation (will be done for Phorum 5.2);
- a system for scheduling hook priorities (e.g. to always run the bbcode hook as the last hook). A beta system for this is already implemented in the Phorum 5.2 development tree;
- a library containing functions for recurring module tasks;
- more information in the module's `info.txt` (like module version and compatibility).

More ideas for module writing are welcome at all time.

Maurice Makaay

Phorum Development Team

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

Oliver Riesen-Mallmann / September 07, 2006 11:18AM

[Re: Making life easier for module writers](#)

Hi Maurice,

Quote

To use this new feature, you have to create a templates directory in your module's directory, which uses the same template structure as Phorum's main templates directory.

What about if I want to support various version of Phorum (for example 5.1 and 5.2) with my module templates? How to handle this?

Regards
Oliver

Using Phorum since 7/2000: forum.langzeittest.de (actual version 5.2.23)

Modules "Made in Germany" for version 5.2: [Author as Sender](#), [CarCost](#), [Close Topic](#), [Conceal Message Timestamp](#),
[Format Email](#), [Index Structure](#), [Mailing List](#), [Pervasive Forum](#), [Spritmonitor](#), [Terms of Service](#) and [German Language Files Package](#).

Maurice Makaay / September 07, 2006 10:37PM

[Re: Making life easier for module writers](#)

See [this thread](#) for an answer on that question.

Maurice Makaay

Phorum Development Team

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

cactux / September 17, 2006 08:28AM

[Re: Making life easier for module writers](#)

Quote

more information in the module's info.txt (like module version and compatibility).

I would suggest :

module_url : web page or thread address of the module

version_url : an url giving back the latest version of the module, so an automatic check could be done to see if the currently running version is the latest or not. Something similar to what's done currently for Phorum version.

Cactus : [www.cactuspro.com]

Maurice Makaay / September 17, 2006 10:42AM

[Re: Making life easier for module writers](#)

Sounds like good ideas to me. Thanks.

Maurice Makaay

Phorum Development Team

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