Talking The Talk (A phpBB Primer)

By icarus
2003-06-13

Printed from DevShed.com
URL: http://www.devshed.com/Server_Side/PHP/PHPBB/

Code Pool

One of the nicest things about PHP is the huge support the language enjoys in the developer community.

This developer support is one of the prime reasons for the language's increasing popularity, and for its rapid incorporation of cutting-edge new technologies. Since PHP development is handled by a team of programmers spread across the globe, and they in turn are supported by a worldwide community of users and developers, it's no surprise that support for new technologies appears far more rapidly in PHP than in many of its competitors.

PHP's base in the developer community has also given rise to a huge number of very successful open-source projects - among them, the Smarty template engine (http://smarty.php.net/), the phpMyAdmin MySQL database administrator (http://www.phpmyadmin.net/), the ADODB database abstraction layer (http://php.weblogs.com/ADODB) and the PEAR code repository (http://pear.php.net/). These open-source applications play an important role in helping regular developers (like you and me) speed up their development cycle, by offering a rich, robust and feature-rich code base that can be used as the foundation for new projects.

Why my sudden fascination with the PHP code pool? Because I've been playing with the language a lot lately, and one of my recent projects involved implementing a PHP-based portal which included a discussion forum. At first, I thought that I would have to design and build this forum myself from scratch - until a search on Google turned up a very neat little PHP application named phpBB, which offered all the features I needed. Installation was a snap, the package was stable, full-featured, and easy to customize, and - best of all - it worked like a charm, allowing me to add Slashdot-type discussion boards easily and transparently to my PHP-based site.

If this is something you've always wanted to do, but never known where to begin, this is your lucky day - over the next few pages, I'm going to guide you through the process of installing, using and customizing phpBB on your PHP-based Web portal. Keep reading!

The Big Picture

First, a little background. The project that prompted me to look at phpBB involved setting up a Web site for one of our customers, a company involved in the Web hosting business. This Web site was supposed to offer detailed descriptions of their UNIX and Windows server hosting plans, corporate information, customized solutions and the like. One of the more important components of the Web site was a discussion board for the company's technical support staff to interact with customers, answer questions and build an interactive community of users.

Needless to say, our customer was very excited about the discussion forum, looking at it as a great way to post announcements to their customers, to create a searchable repository of commonly-asked questions and their answers, and to build a sense of community between their customers. I was therefore tasked with putting together a full-featured bulletin board, one which was stable, full-featured and simple to use. My initial thought was that I'd be spending the next few weeks coding it, until I came up with the bright idea of first Googling the term "php bulletin board"...which gave me a list of options, all of them meeting (and in some cases, exceeding) the customer's requirements.

One of the more interesting options on the list was phpBB, which, in the words of its authors, is "...a high powered, fully scalable, and highly customisable open-source bulletin board package [...] based on the powerful PHP server language and your choice of MySQL, MS-SQL, PostgreSQL or Access/ODBC database servers." It provides a simple, user-friendly discussion board for portal members, and includes support for features like message posting and replying, message threading, subject/body search, themes, private messages and many others.

Building Blocks

In order to get started with phpBB, you'll need a working PHP installation (I'm currently running PHP 4.2.3), a MySQL database server (I'm running MySQL 4.0.10) and a copy of the latest release (currently version 2.0.4) of the package; you can download a copy from http://www.phpbb.com/downloads.php

Setting up the package is pretty simple. The first step is to uncompress the source archive into a directory under your Web server (mine is called "medusa" here) root.

```
$ cd /usr/local/apache/htdocs/

$ tar -xzvf /tmp/phpBB-2.0.4.tar.gz
```

This should create a directory (named "phpBB2/" for phpBB 2.0.4) which contains the following files:

```
$ ls -l /usr/local/apache/htdocs/phpBB2
drwxrwxr-x 2 www www 4096 Jan 15 19:04 admin
-rw-rw-r-- 1 www www 5489 Jan 15 19:04 common.php
-rw-rw-r-- 1 www www 0 Jan 15 19:04 config.php
drwxrwxr-x 2 www www 4096 Jan 15 19:04 contrib
drwxrwxr-x 2 www www 4096 Jan 15 19:04 db
drwxrwxr-x 2 www www 4096 Jan 15 19:04 docs
-rw-rw-r-- 1 www www 810 Jan 15 19:04 extension.inc
-rw-rw-r-- 1 www www 3596 Jan 15 19:04 faq.php
-rw-rw-r-- 1 www www 46523 Jan 15 19:04 groupcp.php
drwxrwxr-x 3 www www 4096 Jan 15 06:17 images
drwxrwxr-x 2 www www 4096 Jan 15 19:04 includes
-rw-rw-r-- 1 www www 14332 Jan 15 19:04 index.php
drwxrwxr-x 3 www www 4096 Jan 15 19:04 install
drwxrwxr-x 3 www www 4096 Jan 15 19:04 language
-rw-rw-r-- 1 www www 7097 Jan 15 19:04 login.php
-rw-rw-r-- 1 www www 12059 Jan 15 19:04 memberlist.php
-rw-rw-r-- 1 www www 36338 Jan 15 19:04 modcp.php
-rw-rw-r-- 1 www www 34935 Jan 15 19:04 posting.php
-rw-rw-r-- 1 www www 73271 Jan 15 19:04 privmsg.php
```

-rw-rw-r-- 1 www www 3726 Jan 15 19:04 profile.php
-rw-rw-r-- 1 www www 40443 Jan 15 19:04 search.php
drwxrwxr-x 3 www www 4096 Jan 15 19:04 templates
-rw-rw-r-- 1 www www 23070 Jan 15 19:04 viewforum.php
-rw-rw-r-- 1 www www 7232 Jan 15 19:04 viewonline.php
-rw-rw-r-- 1 www www 45110 Jan 15 19:04 viewtopic.php

The next step is to create a database for your phpBB data. Start up the MySQL command-line client, log into the database server, and create a database named "phpbb".

$ mysql

mysql> CREATE DATABASE phpbb;
Query OK, 1 row affected (0.03 sec)

You can check that the database has been created,

mysql> SHOW DATABASES;
+------------+
| Database |
+------------+
| mysql |
| phpbb |
| test |
+------------+
3 rows in set (0.08 sec)

and then log out.

All that's left now is to create the base tables for the application, configure it to talk to the database, and set an administrator password. phpBB makes all this easy to accomplish via a single-click PHP script, which performs all these tasks for you automatically.

Make sure your Web server is running, and point your browser to the "install/install.php" script of your phpBB installation. For example, if the application was installed to the root of the Web server, you would surf to "http://medusa/phpBB2/install/install.php", where you'd see something like this:

This is the initial application configuration screen, which requires you to enter basic information about your system - credentials for the MySQL database, a username and password for the forum administrator, the Web server host name and application location, a prefix for the phpBB database tables and an email address for administrator notifications. Note that this application configuration module allows you to perform both a new install and an upgrade of an existing installation; you should obviously pick the former in this case.

Once you've entered all the required data, submit the form, and the installation script will go to work creating the tables for you. On successful completion, you should see a screen with the following message.

You can verify that the tables have indeed been created by going back to the MySQL command line and checking for yourself.

```
mysql> USE phpbb;
Database changed

mysql> SHOW TABLES;
+-----------------------+
| Tables_in_phpbb |
+-----------------------+
| phpbb_auth_access |
```

| *phpbb_banlist* |
| *phpbb_categories* |
| *phpbb_config* |
| *phpbb_disallow* |
| *phpbb_forum_prune* |
| *phpbb_forums* |
| *phpbb_groups* |
| *phpbb_posts* |
| *phpbb_posts_text* |
| *phpbb_privmsgs* |
| *phpbb_privmsgs_text* |
| *phpbb_ranks* |
| *phpbb_search_results* |
| *phpbb_search_wordlist* |
| *phpbb_search_wordmatch* |
| *phpbb_sessions* |
| *phpbb_smilies* |
| *phpbb_themes* |
| *phpbb_themes_name* |
| *phpbb_topics* |
| *phpbb_topics_watch* |
| *phpbb_user_group* |
| *phpbb_users* |
| *phpbb_vote_desc* |
| *phpbb_vote_results* |
| *phpbb_vote_voters* |
| *phpbb_words* |
+----------------------+
28 rows in set (0.01 sec)

All there? Good. Before you get into actually configuring your forum for use, there is one more thing you need to do - delete the "install/" and "contrib/" directories from the main application directory.

$ rm -rf /usr/local/apache/htdocs/phpBB2/install

$ rm -rf /usr/local/apache/htdocs/phpBB2/contrib

Leaving these directories behind opens up a serious security hole in your forum...so serious, in fact, that phpBB will refuse to function if they're present, stubbornly displaying the following error until you delete them:

All done? Flip the page, and let's set some global parameters for the application.

Starting Up

Once the application has been properly installed, phpBB will automatically log you in as the administrator and drop you into the administration module so that you can begin configuring the system for use. In case it doesn't, point your browser to the "admin/" directory under the installation (continuing the example above, the complete URL would be "http://medusa/phpBB2/admin/") and you should see something like this:

Each section of the administration module deals with a specific aspect of the system - here's a brief list:

Forum Administration: This section allows you to create and manage discussion forums, organize them into categories, and decide the access levels for each.

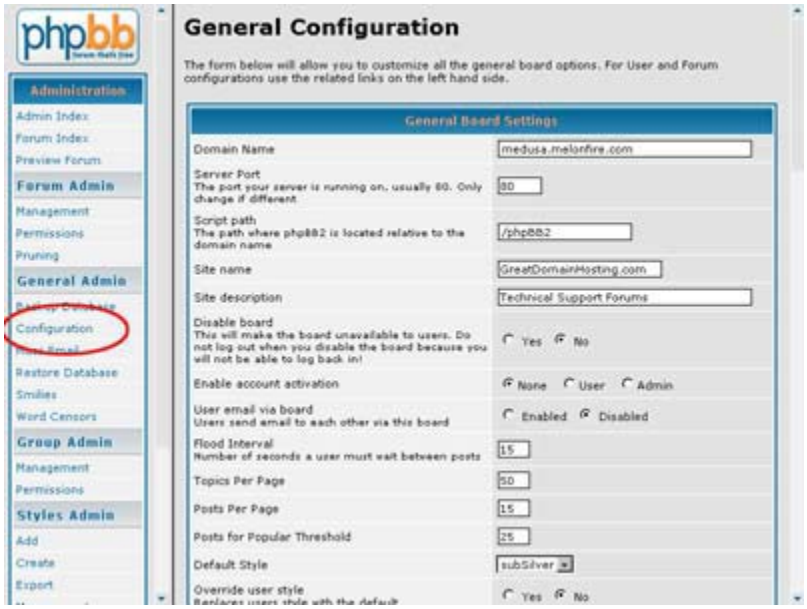
General Administration: This section allows you to configure the application for service, controlling variables like file paths, posts to display per page, date and time formats, default user and forum settings, and mail server configuration. It also allows you to back up and restore the system database, and configure smileys, avatars and word control lists.

Group Administration: This section allows you to organize users into groups, and assign permissions to these groups.

Style Administration: This section allows you to define layouts and style sheets for the system, allowing you to add new themes, alter the currently-selected theme, and export themes.

User Administration: This section allows you to manage the users registered in the system, assigning them appropriate permissions and setting access control rules.

Your first stop should be the "Configuration" component of the "General Administration" section.



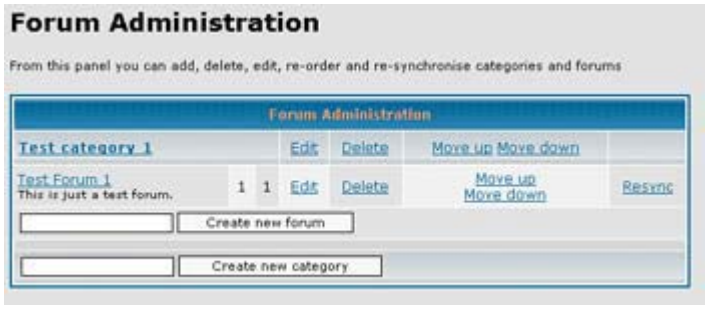
As you will see, most of the items here are fairly self-explanatory. You should go through them and alter those you see fit...although most of the time, the defaults will do just fine. Pay special attention to the domain name, script path and SMTP server settings, errors in this may cause the application to malfunction.

Once you're done, save your settings, and flip the page to configure the forums themselves.

An Open Forum

phpBB allows you to create an unlimited number of discussion forums, and organize them into an unlimited number of categories. Obviously, such power should be used with restraint, and an effort should be made to organize your forums into distinct categories that are easily accessible to the users.

Continuing the example begun a few pages ago, my task was to create two categories of discussion forums, one for Windows hosts and the other for UNIX hosts. Each of these categories would contain forums dealing with specialized aspects of the hosting plan, such as email management, FTP, scripting languages, databases and so on. The place to go next, therefore, is the "Forum Administration" section, and the "Management" component within it.



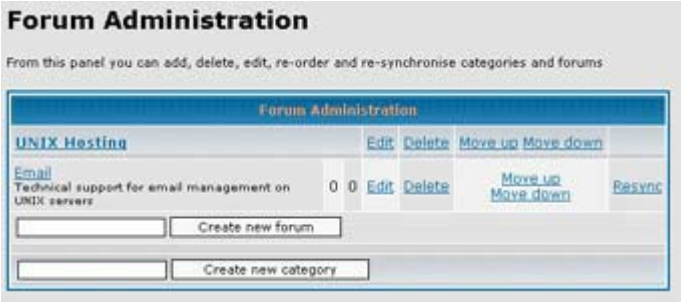
By default, as you can see above, phpBB sets up a single category containing a single discussion forum. This is useful for testing purposes; however, when you're setting it up in a live environment, you

will probably want to delete it using the controls provided.

You should then use the "Create New Category" command to add new categories to the system,

and then use the "Create New Forum" command to add forums within these categories. phpBB will allow you to define some variables for the forum: its name, description (make this as clear as you can, since it will be displayed below the forum name in the main index), its category, and whether old posts should automatically be removed. Here's an example of what this might look like:

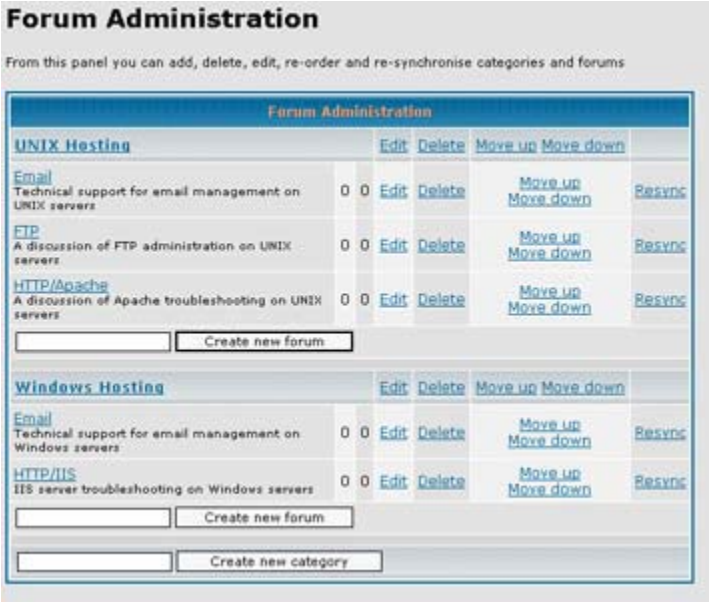
Once you save the configuration, the new forum should appear (under the appropriate category) in the main "Forum Administration" page.



Once all your forums have been created, the next step is to decide who can post to each forum. phpBB supports a number of different access levels for each forum, accessible via the "Permissions" components of the "Forum Administration" section: "public" (unregistered users may post), "registered" (only registered users may post), "private" (only selected users may post) and "moderators" (only moderators may post). Forum access levels should be carefully selected keeping in mind the goals of the forum and the type of users expected.

In the example above, the hosting company was not interested in creating public forums, as even casual visitors would be able to post to these. Rather, they wanted the forums restricted to registered users, so that each post would be tagged with an audit trail. Based on this requirement, I altered access levels for all the forums created thus far to "registered".

When you're done, you can go back to the "Forum Administration" main page to see a list of all the forums created thus far - mine looked something like this:



This is also a good time to back up the database, so that the system can be reinstalled from scratch if needed. Go to the "Backup Database" component within the "General Administration" section, and start the backup.

The output (which may take a minute or two) of this operation will be a single SQL file containing the SQL commands needed to recreate the forums from scratch. You should keep a copy of this in a safe place, so that you don't need to go through the process of forum creation above again.

A Moderate Approach

Next up, user creation. Continuing the example above, since the forums on the previous page were primarily intended for technical support, our customer requested us to create accounts for their five-member technical support department. Since these users would moderate the various forums, they would need appropriate moderator privileges, so as to allow them to maintain and administer their forums.

In phpBB, this is a two-step process: registering the users and then assigning them moderator privileges. The first part can be easily accomplished by haveing users register themselves through phpBB's user registration module, accessible via the main page of the application (in our example

setup above, "http://medusa/phpBB2/").

Selecting the "Register" command brings up a user registration form, which each user has to complete to gain access to the forum (since I made the forums accessible to registered users only on the previous page).

Important elements of this form are the user name and password (obviously!), and a valid email address - phpBB uses this email address to send the user email notification of forum replies, and to resend new passwords if needed. The user may optionally provide other personal information, and set various profile options such as the timezone for messages and whether email notification and private messaging is enabled for the account (these options may also be altered later via the "Profile" link at the top of every application page).

Once the user's account has been activated, he or she may log into the system and begin using it to post messages in the various forums. Before we get to this, though, there is one additional step to perform with reference to the example above: assigning moderator privileges to the five tech support users. Therefore, once I received notification that the five tech support members had successfully registered, I logged back into the phpBB administration module and clicked my way into the "Permissions" component of the "User Admin" section.

This section allows you to select users and assign them moderator privileges on a per-forum basis. Moderators will then see an additional "Moderator Control Panel" for each forum when they log back into the board; this control panel allows them to delete offensive posts, move posts to other forums, split discussion threads and perform other, forum-level administrative tasks.

This "Permissions" component of the "User Admin" section is also the place to assign specific users administrative rights to the system. phpBB has two basic security levels, "user" and "administrator". As you may remember, when phpBB is first installed, it asks you to define an administrative user; this user is automatically assigned administrator-level access to the system. New users, as they register, are give user-level rights; however, you can alter this and give specific users administration-level rights at any time.

The User Experience

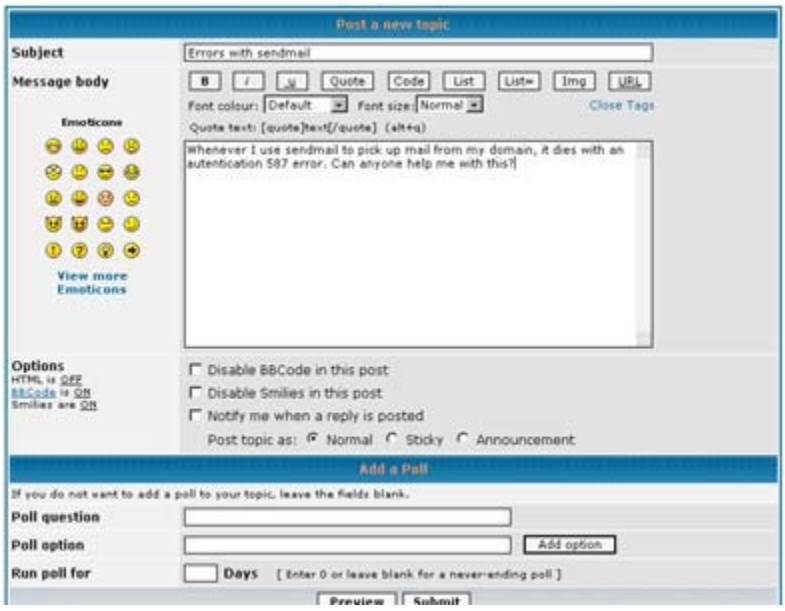
With the system configured, all that's left is to turn it live and let users begin registering. These registered users have a number of different functions available to them. They can read messages in the various forums, post new messages or reply to existing threads, search for messages meeting specific criteria, send private messages to other users, and make changes to their profile.

The starting point for all this activity is the application's main page; this page contains the forum index, which contains a list of forums, broken down into various categories (these are the same categories and forums you configured a while back).

The forum list is followed by a short user summary, which contains statistics on the number of users registered and the number of users online.

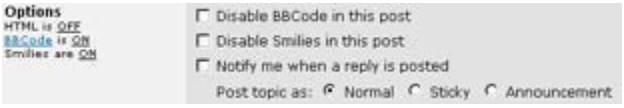
Selecting a forum will produce a list of all threads in the forum, together with information on the number of threads, number of posts, number of views and author of latest post.

Users can post a message by selecting the "New Topic" command, which will pop up a form like this:

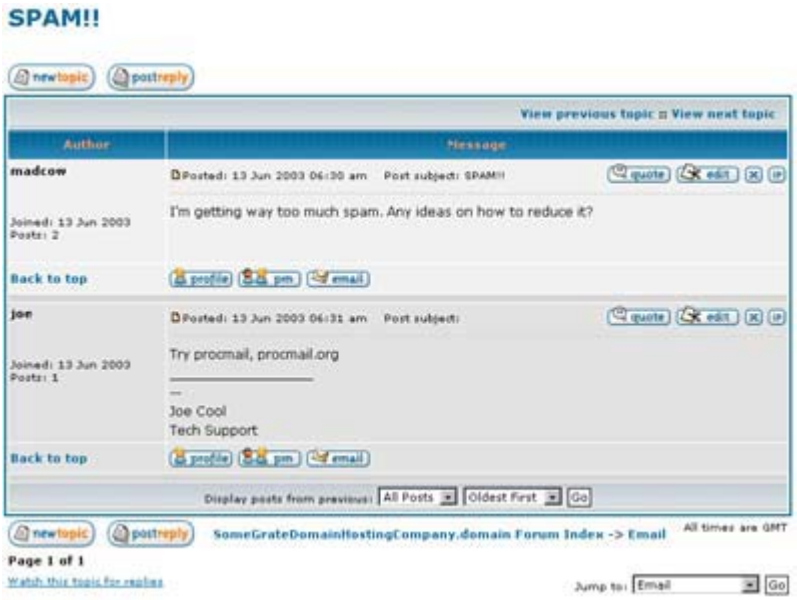


Depending on how the board has been configured, users may be able to use HTML code and smileys in their posts, or attach files or images to their messages.

It is interesting to note that users with administrative rights have the option to post announcements (posts which appear at the top of every forum) and "sticky" messages (posts which are always present on the first page of each forum). In the scenario above, administrators would use this option to post announcements of server upgrades or downtime notices.



Selecting a particular thread will drill one level deeper into the forum, producing a list of the starting post and all replies to it.



Adding a message to a thread is easily accomplished via the "Post Reply" button at the end of every page, which pops up a message entry form similar to the one above. Messages posted using this form are added to the forum, and appear at the end of the message list. Each message is accompanied with the name of the user who posted it, together with a rank and the user's number of posts and date of registration. It's possible to find out more about the user by clicking the "Profile" link next to the corresponding post,

or send the user a private message (that is, a message which does not appear in the forum but is only visible to the user) with the "Private Message" link.

Private messages appear in the recipient's inbox; additionally, if the recipient is logged in at the time, and the system is configured to allow it, a private message may also appear as a small pop-up window in the recipient's browser.

It's also possible for users to "watch" particular threads,

so that they are automatically notified of replies to those threads. This feature was particularly useful for the hosting company in the scenario above, as it allowed both users and technical support staff to keep track of hot-button issues and respond to them rapidly.

Finally, all users have a fairly powerful search engine at their disposal, which can be used to search for posts by keyword or author. Searches may be restricted to specific forums or categories, and may be sorted by time, author, title, forum or topic. Wildcards and Boolean expressions may be used to create complex search queries, and the results may be displayed as either topics or posts. Here's what a typical search and results page might look like:

Doing More

The operations listed in the previous pages of this tutorial cover the most common things you would do with phpBB - creating forums, assigning moderators, posting and replying to messages, and searching forum archives. However, phpBB also offers a number of other features, which power users can exploit to customize every single aspect of the application.

* Groups: phpBB allows administrators to organize individual users into named groups, and assign permissions to these groups as a whole, rather than to users individually. This makes it easy to create discrete collections of users, and to assign special permissions and moderator status to them. A single user may thus have numerous different privileges, depending on the groups he or she belongs to.

In the scenario above, for example, as the company expands. it might make sense to create two groups of support personnel, one for UNIX servers and the other for Windows servers, assign these groups moderator status for the appropriate forum categories, and then simply add and delete users from these groups as needed, rather than assign and maintain permissions individually.

* Private forums: phpBB makes it possible to create so-called private forums, access to which is restricted to a select group of users. This capability comes in handy to build gated user communities whose members are known to each other and require a private area to communicate. In the scenario above, for example, the various technical support engineers might have a private forum of their own, in which internal issues or problems might be discussed and tracked; this forum would not be available to the company's customers at large, serving only as a meeting place for staff and thus offering a greater degree of privacy.

* Ranks: phpBB assigns ranks to users based on the number of posts they have placed in the various forums. These ranks can be customized to your special needs, so that more active forum participants are easily identified and given special status. For example, in the scenario above, forum participants who are particularly helpful might be flagged with a special rank so that other, newer users can identify them and approach them for assistance.

* Styles: phpBB comes with a powerful template engine which makes it possible to completely separate the HTML user interface from the business logic. This makes it possible to easily alter the layout of the application, and to change fonts, colours and images for specific branding requirements. The phpBB site contains extensive documentation on how to build these templates, together with numerous samples, and the application itself includes a module to import, export and manage these styles. For the project discussed above, this feature came in handy when it was time to integrate phpBB into the main company Web site - I was able to easily alter the main elements of the phpBB interface and the headers, footers and colours to more closely match the rest of the Web site's user interface.

* Polls: Users have the ability to post polls in a forum; these polls can contain multiple options, and other users can respond to them while participating in the forum. phpBB autmatically tracks poll responses and displays graphical summaries of the responses as they come in.

* Private messages: phpBB comes with a private messaging system that allows board users to quickly and transparently communicate with each other. This messaging system is similar to email, in that messages appear in the recipient's private "inbox" and are not displayed to other forum members. If the board is configured suitable, private messages may also pop up immediately on the recipient's display. This private messaging mechanism provides a fast way for forum users to communicate with each other privately.

* Smileys and BBCode: Smileys are a popular way of expressing emotions, and phpBB comes with an entire gallery of custom-created smileys that can be used in posts. Additionally, phpBB also comes with its own custom formatting language, BBCode, which allows you to mark up your posts with colours, underlines, italics and other effects.

* Word blocks and user bans: phpBB comes with powerful tools to help moderators and administrators cut down on offensive posts. The phpBB administration module comes with the ability to automatically reject posts containing specific words, and users can also be banned (restricted for posting) if the moderators find their posts offensive, annoying or generally off-topic.

* Avatars: phpBB allows users to personalize their profiles with the use of custom "avatars", graphical icons which appear under each user's name in the forums. Users can choose from pre-defined avatars or even create their own; these avatars may be serious, playful, happy or sad, depending on the personality the user wishes to project. On boards with a large number of users, avatars can help

distinguish users from each other, and can also substantially add to the personality and spirit of the board.

* Mass email: Forum users are required to provide their email address during the registration process; this creates a rapidly-growing database of contact information that can be used for annoucements or targeted marketing activities. In order to assist in this process, phpBB includes a component to send email to all the users registered in the system, or even to members belonging to a specific group.

As you might imagine, these and many other features have conspired to make phpBB one of the most full-featured Web-based discussion boards available today. It's robust and simple to install - I was able to get it up and running in twenty minutes flat - and its Web-based administration module makes it easy to administer and maintain. It's also free (always a Good Thing, especially for organizations with limited budgets) and it comes with fairly good documentation and help files. If you're looking to add a discussion board to your Web site, you could do a lot worse!

Link Out

And that brings us to the end of this tutorial. In this article, I introduced you to phpBB, an open-source discussion forum, and explained (with a small case study) how you can use it to add interactivity to a Web site. This article discussed the process of installing the application, setting up its database and tables, and configuring it for service. I showed you how to create discussion forums and organize them into categories, add users, assign moderators and administrators, and post and search for messages in the various forums.

These are the most common operations an administrator or user would need to perform in the context of a discussion board; however, phpBB also offers a number of additional features for more specialized requirements, and the latter part of the tutorial examined these in brief as well.

While phpBB is one of the better options available, there's certainly no shortage of alternatives when it comes to Web-based discussion forums. In case you're looking for alternatives, you should consider visiting the following links:

Drupal, at http://www.drupal.org/

Sporum, at http://www.sporum.org/

BazookaBoard, at http://www.bazookaboard.com/

NeoBoard, at http://www.neoboard.net

PHPTalk, at http://www.phptalk.org/

Zorum, at http://zorum.phpoutsourcing.com

The official Web site for phpBB is http://www.phpbb.com/ Now that you've seen a preview, drop by and get yourself a copy. And once you've got an active forum community running on your site, don't forget to drop me a line and tell me all about it. Until then...stay healthy!

Note: Examples are illustrative only, and are not meant for a production environment. Melonfire provides no warranties or support for the source code described in this article. YMMV!

This article copyright Melonfire 2000-2002. All rights reserved.