# Building Online Communities with Drupal, phpBB, and WordPress

Robert T. Douglass, Mike Little,
and Jared W. Smith

Apress®

**Building Online Communities with Drupal, phpBB, and WordPress**

**Copyright © 2006 by Robert T. Douglass, Mike Little, and Jared W. Smith**

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN (pbk): 1-59059-562-9

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Matt Wade
Technical Reviewers: Steve Potts, James Walker
Editorial Board: Steve Anglin, Dan Appleman, Ewan Buckingham, Gary Cornell, Tony Davis, Jason Gilmore, Jonathan Hassell, Chris Mills, Dominic Shakeshaft, Jim Sumser
Project Manager: Sofia Marchant
Copy Edit Manager: Nicole LeClerc
Copy Editor: Marilyn Smith
Assistant Production Director: Kari Brooks-Copony
Production Editor: Lori Bring
Compositor: Linda Weidemann
Proofreader: Linda Seifert
Indexer: Rebecca Plunkett
Artist: Kinetic Publishing Services, LLC
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at http://www.apress.com in the Source Code section.

CHAPTER 4

# Adding Contributed Modules

One of Drupal's great strengths is the ease with which new functionality can be added in the form of contributed modules. The clear and well-defined hook system (`http://drupaldocs.org/api/head/group/hooks`) allows modules to interact with all of the Drupal subsystems, including the user, menu, taxonomy, filtering, and node-handling systems. As a result, more than 350 modules have been contributed to the Concurrent Versions System (CVS) repository (`http://cvs.drupal.org/viewcvs/contributions/`) at `http://drupal.org/project/Modules`.

The contributions repository is not only large, but it is also very diverse in its offerings. Some of the modules are for specific Drupal versions; some are well maintained, but some are not. Some are easy to install; some require patches to the core Drupal code. Many offer similar or duplicated functionality. This diversity is a great strength of Drupal, but it also means that you need to know which modules are best for any given situation and task.

This chapter will cover the installation, configuration, and use of a number of the best and most popular Drupal modules. The selection covered ranges from making input easier with WYSIWYG editors, to protecting your site from comment spam, to letting site users organize themselves into groups that share interests. These modules, in conjunction with the core modules delivered with Drupal, will provide you with a broad set of tools that you can apply to a diverse array of web sites.

## Getting Drupal Modules

You can find a list of the available modules for the current release at `Drupal.org`:

```
http://drupal.org/project/Modules
```

Each module has a download link to an automatically generated archive file, which is updated with the latest changes from `Drupal.org`'s CVS repository on a regular basis.

First, I'll give you an overview of the modules covered in this chapter, and then I'll explain the general installation process.

### Introducing Some Useful Modules

Some of the most useful modules are those that help with creating content on your site. These include the TinyMCE module (WYSIWYG editor) and the Image and Image Assist modules (for image galleries and in-line images in posts).

The Flexinode module allows you to define your own data types (node types), complete with fields of varying types. The data collected with your defined node types can be shown in table form with sortable columns to facilitate searching.

The Event and Location modules can be used alone or in tandem to attach time and place information to any node type. Combined with the Flexinode module, the Event and Location modules allow you to create custom events calendars that not only track events, but also offer location information based on zip codes and geocoding, as well as deep linking to mapping services.

Organic groups are a way to encourage members of your site to organize into smaller working groups to share resources and have a space for online collaboration. The Organic Groups module introduces node-level permissions so that groups can have private content not visible to those who aren't members of the group. Group photo albums and RSS feeds for each group are among the other features available.

Protecting your site from abuse by spammers is the focus of the Spam module. Based on powerful Bayesian filters that continuously learn from the spam content that is submitted to your site, this module may be the only thing that keeps spammers from turning your site into an online billboard for drugs and sex products.

The Devel and Database Administration modules offer convenient tools for maintaining your database and developing Drupal code. They give you a window into the inner workings of Drupal, exposing not only the contents of your Drupal database, but also the actual SQL queries that are used to build each page. Turn to these modules to help solve database-related problems.

## Installing Contributed Modules

The process for installing a contributed module typically consists of the following steps:

1. Make a backup of your site's database. This is especially important if the module requires that changes be made to your database. *Never neglect this step.*

2. Copy the entire module folder into the `modules` directory of your Drupal installation. Usually, the whole folder can be copied, but you may want to exclude some files that don't need to be on the server. The `.mysql` and `.pgsql` files, for example, can be omitted.

3. Update the database schema, if necessary. For a MySQL database, use the following GNU/Linux command:

   ```
   mysql -u user -p drupal < module.mysql
   ```

   Replace *`user`* with the MySQL username, *`drupal`* with the database being used, and *`module`*`.mysql` with the `.mysql` file distributed with the module. If your database is using table prefixes, you will need to update the database definition file directly to reflect this.

4. Apply any patches, if necessary.

5. Enable the module by navigating to *administer* ➤ *modules* (`admin/modules`) and checking its Enable box.

6. Configure the module, including setting new permissions defined by the module. As explained in Chapter 2, you administer permissions from *administer* ➤ *access control* (`admin/access`).

A module will typically consist of the following files contained in a folder of the same name as the module:

- `README.txt`: Always start here. The `README` file will explain what the module is for, what it can do, and any extra information you might need in deciding whether or not to install the module.
- `INSTALL.txt`: This file will provide directions for installing the module.
- `CHANGELOG.txt`: This file contains a running account of how the module has been altered or updated. This information is useful if you are upgrading a module and want to see how the newest version differs from the version you are currently using.
- `.module`: This file contains the PHP code that makes the module work. It typically consists of the module-specific implementations of the relevant Drupal hooks, as well as any other code needed for the module.

The following are some of the optional files sometimes included with a module:

- `.mysql` and `.pgsql`: These files contain the SQL to build the database schemas that will be used by the module. Not all modules require altering the database schemas, and not all modules support both MySQL and PostgreSQL. This should be explained in the `INSTALL` file for the individual modules.
- `update.php`: If present, this file can be used to change from one version of a module to a newer version. Make sure to consult the `INSTALL` file directions before applying the updates.
- `.patch` files: These files are present if any other Drupal files, core or otherwise, need to be patched in order for the module to work.
- `MAINTAINERS.txt`: This file lists the people who have write access to the CVS repository and who are willing to fix bugs and perhaps add features to the module.

The following sections include specific instructions for installing each of the modules described.

# TinyMCE Module

All of the textual content submitted to a Drupal site is intended first and foremost for display as HTML in a web browser. Because of this, any formatting of the text—such as boldface, color, alignment, and so on—must be realized in HTML. In addition, the inclusion of images and hyperlinks within the text also must be achieved with HTML. This poses a problem to the end users who may have limited or no knowledge of HTML tags. Requiring these users to learn how to use tags is an invitation for badly formatted content and frustrated users. Fortunately, a number of solutions exist for this problem.

Drupal has at least three full WYSIWYG editors: TinyMCE (`http://drupal.org/project/tinymce`), Htmlarea (`http://drupal.org/project/htmlarea`), and FCKeditor (`http://drupal.org/node/16118`). The TinyMCE editor achieves a very nice balance between powerful features and ease of use, and it integrates well with the Image Assist contributed module and core Upload module for comfortable inclusion of images in posts. TinyMCE has the following features:

- Role-based profiles, which let you decide which user roles can use the editor, and how it looks for each of them
- More than 50 editor buttons that can be turned on or off for each profile
- A switch to easily turn the editor on or off when composing content

## Installing the TinyMCE Module

The Drupal TinyMCE module has two parts: the Drupal module, which you can obtain from `Drupal.org`, and the TinyMCE project itself, which is hosted at SourceForge (`http://sourceforge.net/projects/tinymce/`). The `INSTALL.txt` file that is packaged with the Drupal module provides a link to the particular SourceForge download that is to be used. Both the Drupal module and the TinyMCE project are updated often, so you are encouraged to follow the installation instructions from both projects carefully to ensure compatibility.

Here are the steps for installing TinyMCE:

1. Download the TinyMCE module from `http://drupal.org/project/tinymce`.
2. Unpack the download and copy the `tinymce` folder into your Drupal `modules/` directory.
3. After making a backup of your site and database, update the database using the SQL in the `tinymce.mysql` or `tinymce.pgsql` file.
4. Follow the directions in the `tinymce/INSTALL.txt` file for obtaining the appropriate copy of the TinyMCE project.
5. The code from the TinyMCE project is also contained in a directory called `tinymce`. Place this directory inside the `modules/tinymce` directory so that the resulting structure is `modules/tinymce/tinymce/`, as shown in Figure 4-1.

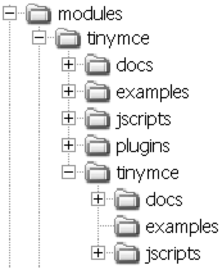


**Figure 4-1.** *The TinyMCE directory structure*

6. Navigate to *administer* ➤ *modules* (`admin/modules`) and enable the module.

After completing these steps, each text area on your entire site will have an additional link in the bottom-left corner. This link, *enable rich-text*, activates the default TinyMCE editor for that particular text area. The default configuration of the editor, as shown in Figure 4-2, is very minimal, with only nine basic buttons enabled.

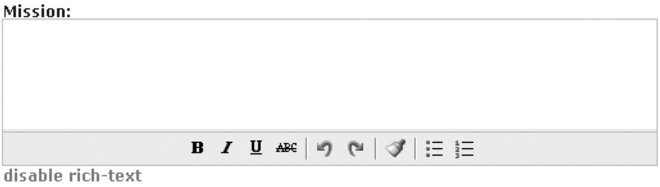


**Figure 4-2.** *The default TinyMCE editor in action*

## Configuring the TinyMCE Module

To configure TinyMCE, you need to set the appropriate Drupal permissions and input formats. Then you can configure whether TinyMCE appears as the editor by default and set up role-based profiles.

### TinyMCE Permissions

Two permissions for the TinyMCE module are defined: Access TinyMCE and Administer TinyMCE. Any user role that should be able to use the editor needs the Access TinyMCE permission.

### TinyMCE Input Formats

The TinyMCE module inserts HTML into posts. As the default input format filters many of the elements that TinyMCE uses to format text, you will need to create a new input format specifically for use with TinyMCE. You should set this input format to be the default for those user roles that use TinyMCE. As described in Chapter 2, to configure input formats, select *administer* ➤ *input formats* (`admin/filters`).

In order to be compatible with the full range of TinyMCE elements, the input format should allow the following set of elements:

```
<a> <blockquote> <br> <em> <hr> <img> <li> <ol> <p> <span> <strike>
<strong> <sub> <sup> <table> <tbody> <td> <tr> <u> <ul>
```

### TinyMCE Default State

With the appropriate input format and permissions in place, you can begin configuring the TinyMCE module by navigating to *administer* ➤ *settings* ➤ *TinyMCE* (`admin/settings/tinymce`). This page will list all of the role-based configuration profiles that you have created, as well as the Default Tinymce State setting.

In all of the text areas where the TinyMCE editor can appear, there is a link that toggles between the editor and a plain text area. The Default Tinymce State setting determines whether the text area or the editor is loaded first by default. Users are able to override this setting in the TinyMCE Settings area of their user account page. What you choose as a default largely depends on who your users are and whether they will find the rich-text editor more of a convenience or a hassle.

## Role-Based Profile Configuration

One of the TinyMCE's most attractive features is the vast control you have over how the editor looks, which buttons it displays, and what functionality is enabled. In order to make these decisions, you need to create one or more configuration profiles that can be assigned to your existing user roles.

From the TinyMCE settings page (`admin/settings/tinymce`), click *Create new profile* (`admin/settings/tinymce/add`). This leads to a new configuration screen with a wealth of options.

After naming your new profile, select the roles to which it will apply. The roles available will be those that have the Access TinyMCE permission. Each role can belong to only one profile, so the roles that you select here for this profile will not be available if you create another profile.

### Editor Display and Formatting Options

You can set the following options for this profile's editor display and formatting options:

**Make TinyMCE visible on**: You can decide whether to show the editor on all text areas or only on those pages that you specify. For indicating specific pages, you can use any valid Drupal paths and use the asterisk (*) as a wildcard. You can also use `<front>` to indicate the front page (as you defined it on the `admin/settings` page).

**Buttons**: The Buttons group of options lets you decide exactly which buttons will be active in the editor for this particular profile. More than 50 buttons are available, ranging from basic formatting (boldface, italic, and so on) to scripts to check spelling and clean the HTML output.

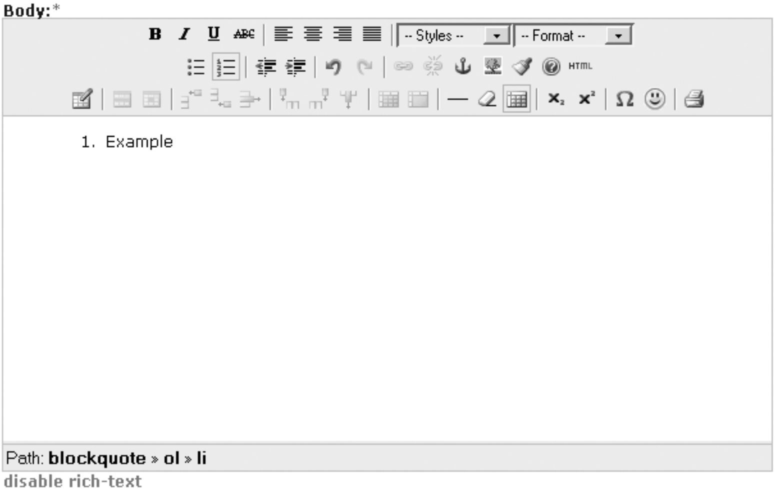
---

■ **Tip** Try out the buttons before you activate them by visiting the TinyMCE web site: `http://tinymce.moxiecode.com/example_full.php?example=true`.

---

**Editor appearance**: The Editor appearance group of options lets you customize how the editor should appear.

- The Toolbar Location option lets you set whether the editor's toolbar appears above the text area (top) or below it (bottom).
- The Toolbar Alignment option sets the editor's toolbar to appear on the left or right, or in the center.
- The Path Location field refers to a context-sensitive display of the Document Object Model (DOM) hierarchy within the rich-text editor. Wherever the cursor is placed, the path will show you which HTML element it is dealing with, as well as that element's parents, all the way up the hierarchy within the editor. The example in Figure 4-3 shows the cursor in an `<li>` with an `<ol>` as its parent, which, in turn, is in a `<blockquote>` element, resulting in the path `blockquote » ol » li`. The main purpose of the path information is to help people with an understanding of HTML figure out why something in the editor works the way it does, which can be useful if achieving a particular format is proving to be a tricky task.



1. Example

Path: **blockquote** » **ol** » **li**

**Figure 4-3.** *Path information in the TinyMCE editor*

- The Enable Resizing Button option allows you to grab the bottom-right side of a text area and drag it to any size you need.

**Block formats**: The list of block formats allows you to remove options from the block element selector, which appears next to the styles selector in Drupal's Advanced theme. By default, the block format selector contains nine options, which correspond to the following HTML elements: `<p>,` `<address>`, `<pre>`, `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`. These elements are all known as block-level elements because the default behavior for browsers is to place them on their own line. The rich-text editor allows any highlighted text to be promoted to block level using these controls. In reality, it is rather uncommon for all of them to have special meanings on any given web site. If this is the case on your web site, you can remove the irrelevant options from the list in the Block Formats field and avoid tempting your web users to add meaningless markup. Note that it is impossible to add elements other than these nine.

### Cleanup and Output Options

TinyMCE can help guarantee the quality of the HTML that users enter into your web site. It does so by offering three different types of processing or cleanup of input.

**Verify HTML**: Setting Verify HTML to True will help eliminate HTML errors from user input. This includes detecting and closing unclosed elements, and stripping the `<html>`, `<head>`, and `<body>` tags. Occasionally, the editor may be used to make modifications to an entire HTML document, including the `<html>` and `<head>` tags. For this case, it is essential that you set Verify HTML to False, as the `<head>` tags will be stripped otherwise.

**Preformatted**: The Preformatted option, when set to True, will replace white space characters with their HTML entities, thus functioning like the `<pre>` tag. This might be a good idea if your site specializes in ASCII art, but in most other cases, it is probably a bad idea, as many web users will try to format their posts with the spacebar, a practice that should be discouraged.

**Convert <font> tags to styles**: This is a cleanup task that is highly recommended, as the `<font>` tag has been deprecated (see `http://www.w3.org/TR/REC-html40/present/graphics.html#edef-FONT`). When this option is enabled, `<font>` tags will be converted to `<span>` tags with style attributes to replace the function of the `<font>` tags.

### CSS Settings

One of the most significant advances in modern web development is the widespread adoption of Cascading Style Sheets (CSS) as the preferred means for adding style elements to HTML. Gone are `<font>` tags, where size and color are embedded directly into the HTML. Most in-browser WYSIWYG editors, including TinyMCE, offer font, size, and color selectors that allow web users to apply their favorite shades of purple or pink to their postings, effectively marring the design and color scheme of your carefully designed web site. This is the reason why these elements are, by default, turned off in the Drupal TinyMCE module.

■ **Note** The styles selector is visible only in the Advanced theme.

Fortunately, there is a better way to offer these same features in a controlled and limited way that gives the web users the tools they want and eliminates the danger of them overstepping their bounds, stylistically. With TinyMCE, you can identify a style sheet that defines text attributes the user is encouraged to apply. TinyMCE will look at the style sheet and offer each defined style in a drop-down box, so it is easy for your users to apply style information.

The Editor CSS field gives you three options for where TinyMCE should look to find this style sheet:

**Use theme CSS**: This grabs the `style.css` file that is included in your Drupal theme. Figure 4-4 shows the list of style names available when using this option with the default Drupal Blumarine theme. While this option guarantees that the user won't be able to do anything that doesn't fit with the look and feel of your site, it is probably not the best choice. First, the style names are rather cryptic and theme-developer-oriented. Terms like *box*, *block*, and *node* don't really describe how the resulting style will look. Furthermore, some styles, like *error*, don't need to be accessible to web users.

**Define CSS**: This is the best option, and it also requires the most extra effort on your part. When you select this option, TinyMCE takes the value from the next field, CSS Path, and looks there for the style sheet to use.

**CSS path**: Here, you can enter any web-accessible URL. The variables `%h` and `%t` are available and are dynamically replaced with your host name and the path to the current theme, respectively. This allows a site that supports different themes for different users to have a fitting TinyMCE style sheet for each theme.

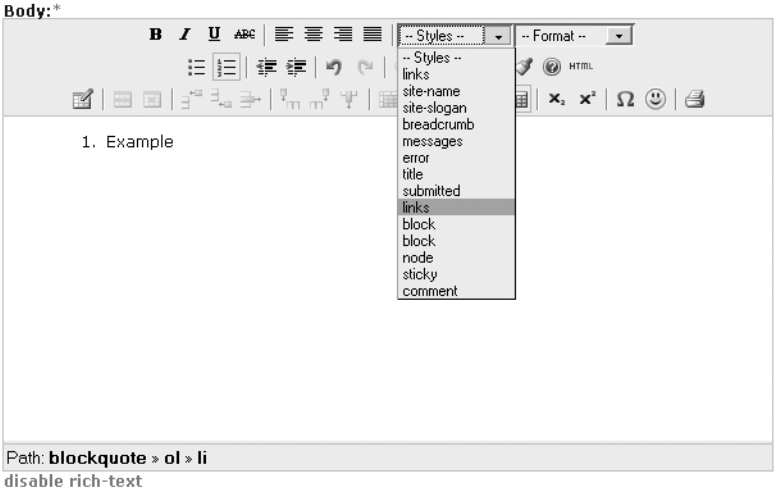


**Figure 4-4.** *Styles available with the Bluemarine theme*

---

■ **Caution** TinyMCE, which is visible only on editing screens, imports the style sheet you define in the CSS Path field itself. You will need to add the import for this style sheet to your theme as well. Otherwise, the style sheet will be loaded only when editing, and the styles visible in TinyMCE will not take effect when the content is being viewed normally.

---

Exercise 4-1 demonstrates how to add a custom style sheet for use with TinyMCE.

## Exercise 4-1. Add a Custom Style Sheet for TinyMCE

Here are the steps for adding a custom style sheet containing the styles that should be available to your site's users:

1. Create a file named `tinymce_styles.css` with the following class rules. TinyMCE recognizes class rules and will ignore other selectors like element rules and ID rules.

```
.color-red {color: red;}
.color-blue {color: blue;}
.color-green {color: green;}

.font-sans { font-family: Arial, sans-serif; }
.font-serif { font-family: Roman, serif; }
.font-mono { font-family: Courier, monospace;}
.font-cursive { font-family: cursive; }
.font-fantasy { font-family: Palette, fantasy; }
.font-blackletter { font-family: serif; }
```

2. Place the `tinymce_styles.css` file in your theme directory; for example, *`drupal`*`/themes/bluemarine/`.

3. Navigate to *administer* ➤ *settings* ➤ *TinyMCE* (`admin/settings/tinymce`) page, and choose an existing profile or click *Create new profile* (`admin/settings/tinymce/add`). Set Editor CSS to Define CSS and CSS Path to `%h%ttinymce_styles.css`. Note that the variables `%h` and `%t` include trailing slashes already. This will evaluate to something like the following:

   ```
   http://www.yourdomain.com/drupal/themes/bluemarine/tinymce_styles.css
   ```

4. Add the following bolded lines to the `<head>` section of your theme's `page.tpl.php` file, or the appropriate counterpart if you are not using PHPTemplate as a theme engine:

   ```
   <?php print $styles ?>
   <style type="text/css" media="all">
    @import "<?php print $directory ?>/tinymce_style.css ";</style>
   <script type="text/javascript"> </script>
   ```

Now when you use the TinyMCE editor, you should have a styles selector that includes the class definitions you added, as shown in Figure 4-5.

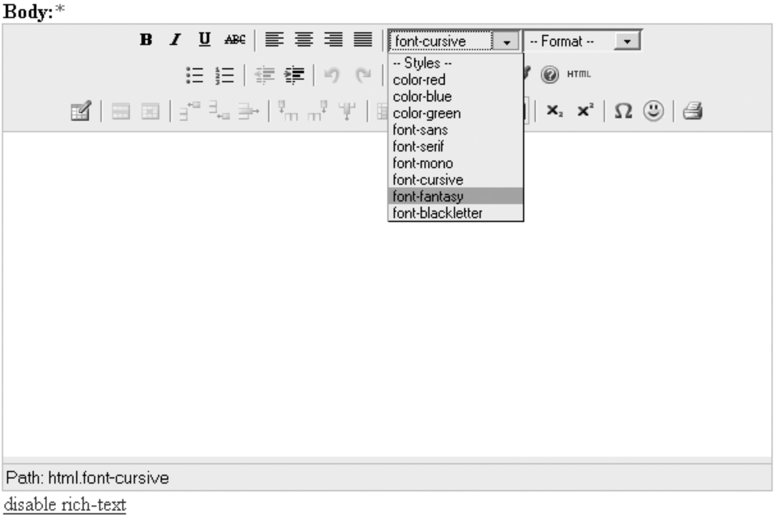


**Figure 4-5.** *TinyMCE with user-defined CSS classes*

If you apply a custom style to some text, click *disable rich-text*, and inspect the HTML, you will see that TinyMCE has surrounded portions of your text in `<span>` elements, like this:

```
<span class="font-serif">You can apply styles to your text.</span>
```

If you want to give the classes in your style sheet different names—perhaps to make them more descriptive than *box*, *block*, or *node*—or if you want to exclude some of the classes in the style sheet you are using, the CSS Classes field is the tool to use. It allows you to enter a list of *name=value;* pairs, which determine the name of the style shown to your users (*name*) and the class name that it corresponds to (*value*). For example, suppose that you want to offer the box and block styles with the names *introduction* and *summary*, and you don't want the node style to be available at all. You could achieve this by entering the following as the value for the CSS Classes field:

```
introduction=box;summary=block;
```

The names *introduction* and *summary* would appear in the list of available styles, and node would be inaccessible to the user.

# Image Module

Images are one of the most popular forms of web content. Whole online communities can be built around the idea of sharing pictures, and with the popularity of digital cameras, almost everyone has photos ready to be uploaded. For the tasks of uploading photographs to the web site, generating thumbnails and standard sizes, and organizing them into galleries, the Image module is the tool to use.

## Installing the Image Module

Installing the Image module is quite straightforward. Just download the module from `http://drupal.org/project/image`, place the entire `image` folder in the `modules` directory, and activate the module from the `admin/modules` page. If you wish to use the ImageMagick library (`http://www.imagemagick.org/script/index.php`) for converting and resizing images, you will also need to move the `/modules/image/ image.imagemagick.inc` file into the `/includes/` directory of your Drupal installation. If you wish to use the Image module's gallery-building functions, you need to make sure that the Taxonomy module is enabled.

## Configuring the Image Module

To configure the Image module, you need to set the appropriate Drupal permissions and conversion library. Then you can configure file paths, file sizes, and galleries.

### Image Module Permissions

The Image module defines two permissions: Administer Images and Create Images. Users with the Administer Images permission will be able to edit the gallery structure and define the various image sizes. Users need the Create Images permission in order to add images to the site.

### Image Module Conversion Library

The most important issue in configuring the Image module is choosing and setting up the conversion library that you will use. The conversion library will be given the responsibility of taking the original images that are uploaded to the site and resizing them to standard sizes and thumbnails. Drupal automatically has support for the GD2 library (`http://www.php.net/image`), which has been bundled with PHP since version 4.3. The Image module adds support for the ImageMagick library as well. To choose which library should be used, navigate to *administer* ➤ *settings* (admin/settings), and in the Image Handling group, select an image processing toolkit.

If you choose a library other than GD2, you will need to save the changes and then fill in the additional field to specify a path to the binary library file that you have chosen. If you want to use ImageMagick, for example, you will first need to select ImageMagick Toolkit, save the changes, and then enter a value for the Location of the "Convert" Binary field. This is the absolute path to the convert (convert.exe on Windows) file.

■ **Note** The Image module needs either the GD2 or ImageMagick library to convert images.

### Image Module File Paths

The Image module requires a directory inside the files directory for the storage of your images. When you select *administer* ➤ *settings* ➤ *image* (admin/settings/image) for the first time, Drupal will create two directories for you: files/images and files/images/temp. If you do not want to use the defaults set by Drupal, you have the opportunity to designate a different path and name in the Default Image Path field. Note that this path is relative to the files directory.

### Image Module Derivative Sizes

When a new file is uploaded via the Image module, at least two (smaller) derivative sizes are created. The Image Sizes group on the Image module settings page (admin/settings/image) includes a table where these sizes can be determined. Each different derivative size has a name and dimensions. Two names, *thumbnail* and *preview*, are required by the Image module. You have the opportunity to define up to three additional names and the sizes for all five formats. The various forms of any given image file can then be accessed using the path node/*{nid}*&size=*name*, where *nid* is the node ID and *name* is one of the format names. If you want your web users to be able to access the uploaded photos in their original sizes as well as in the derivative sizes, you need to check the Allow Users to View Original Image option.

### Image Module Galleries

The Image module has basic taxonomy-based image galleries. You configure galleries by selecting *administer* ➤ *image galleries* (admin/image). The form for adding galleries (admin/image/add) looks similar to the form for adding category terms, and with good reason. The Image module creates a vocabulary of categories to manage your galleries (see Chapter 2 for more on categories). You can build galleries hierarchically by assigning a parent in the Parent field and order them with the Weight field, just as with categories.

■ **Note** The gallery functions are dependent on the Taxonomy module. Make sure the Taxonomy module is enabled before trying to build galleries.

After you have configured the galleries, you can take a peek at what is going on behind the scenes by visiting the list of vocabularies: *administer*➤ *categories* (`admin/taxonomy`). You will see the gallery structure you created is really a vocabulary managed by the Taxonomy module.

### Uploading and Viewing Images

When you are ready to upload images to your site, select *content*➤ *image* (`node/add/image`). The process for uploading images is straightforward, and the module will automatically make thumbnails and various derivative sizes on the server.

Since images are nodes, you make the same decisions about whether they should be promoted to the front page as with any other node type. In addition, you have the option of viewing your images in the context of the galleries you specified. If you have the Menu module activated, you can go to `admin/menu` and activate the navigation link for the galleries. Otherwise, you can add it to your primary or secondary links with the path `image`. (The Drupal path to your image galleries is `image`.)

■**Tip** The Image module defines two blocks, named Latest Image and Random Image, which you can turn on from the block administration page (`admin/blocks`).

## Image Assist Module

With the aid of the Image module, you can upload images to a Drupal web site and arrange them in galleries. In order for those images to be truly useful, however, you also need a means of incorporating them into your site's content. This is the job of the Image Assist (Img_assist) module. This module places an extra icon on content-creation forms. Clicking this icon launches a pop-up window for selecting and inserting images into the text of the node you are editing. This is the easiest way to incorporate in-line images into forums, blogs, pages, and so forth.

### Installing the Image Assist Module

These are the steps for installing the Image Assist module:

1. Download the latest version of Img_assist from `http://drupal.org/project/img_assist`.
2. Place the entire `img_assist` folder in the `/modules/` directory.
3. Make a backup of your database.
4. Load the database definition from the file appropriate to your database, either `img_assist.mysql` or `imag_assist.pgsql`.
5. Navigate to *administer*➤ *modules* (`admin/modules`) and enable the module.

## Configuring the Image Assist Module

To configure the Image Assist module, you need to set permissions and input formats. Then you can configure access, image output, and preview settings.

### Image Assist HTML Use and Permissions

Image Assist can achieve the task of inserting images into posts in two ways:

- Insert special non-HTML tags called *macros* into the post. These macros are then replaced with HTML at page-load time by the content filter system.
- Insert finished HTML into the post, without any further filtering.

The macro tag approach has the advantage that you can change the template for inserted images on a live site, and the change will take place for all previously inserted images. This is helpful if you might be updating the look of your site sometime in the future. It also has the advantage that you can force your users into using Image Assist and only Image Assist, as opposed to typing the HTML for images directly into their posts. You could do this by using the input filters to disallow `<img>` tags in the posts.

On the other hand, using macro tags requires that you take the extra step of adding the Image Assist filter to your input format. Another disadvantage is that you cannot edit the HTML that is generated from within your post, so you cannot customize the appearance of your images on a per-node basis. Also, at page-load time, you might see a slight performance reduction compared with the pure HTML method, since Image Assist must run the regular expressions to find and replace the macro tags.

Here is a list of questions to answer when making the decision of whether to go with Image Assist macro tags or pure HTML:

- Do you need to restrict users from using the `<img>` tag directly?
- Do you need the HTML template for in-line images to remain consistent across your site over time, even if you decide to change it?
- Do you sometimes wish to change or customize the HTML for your in-line images for a particular post?

If you answered yes to either of the first two questions, the Image Assist macro tags are right for you. If you answered yes to third question, you probably want to use pure HTML.

The macro tags are used by default. There is no setting that makes pure HTML the default and only choice. The good news is that users with the Choose Format Type permission have the choice of whether to use the macro tags or pure HTML.

Regardless of which permission you set, however, you must tend to the input formats.

### Image Assist Input Formats

Navigate to *administer* ➤ *input formats* (`admin/filters`) and look at your various formats. As an example, click *configure* for the Filtered HTML format. The two pertinent filters are the HTML filter and the Inline Images filter. If you want the macro tags to work, you need to include the Inline Images filter. If you want to be able to insert pure HTML using Image Assist, you need to configure the HTML filter to accept `<img>`, `<div>`, and `<a>` tags. Otherwise, your images will be filtered out. Finally, you need to check the order of the filters. Click *rearrange* and confirm that the Inline Images filter comes after (has a bigger number than) the HTML filter.

■ **Note** If you alter the Image Assist template to include tags other than `<img>`, `<div>`, and `<a>`, you may need to update your HTML filter to include the additional tags.

### Image Assist Access Settings

Image Assist adds an icon below the text areas on your site. When you click this icon, you see a pop-up window that you can use to add images to the text area. The first configuration choice you make after navigating to *administer* ➤ *settings* ➤ *img_assist* (`admin/settings/img_assist`) is whether you want to allow these icons to appear with all text areas, even where it really makes no sense to add images, or if you want to make a list of paths to specify where the icons should appear.

The Display Img_assist On field gives you two choices: on all text areas (the default) or on specific pages. If you select on specific pages and save the settings, a Pages text box will appear. In the Pages text box, you can make a line-separated list of paths on which the Image Assist icon should appear below text areas.

### Image Output Settings

The Image HTML Template field on the Image Assist settings page (`admin/settings/img_assist`) lets you edit the template for in-line images using eight placeholder variables to represent the dynamic parts.

- `%node-link`: URL to the image node.
- `%img-link`: URL to the (original) image file.
- `%src`, `%width`, `%height`, and `%alt`: Values to the common parameters in an `<img>` tag.
- `%caption`: Text to appear with the image. This can be configured to use the body portion of the image node.
- `%image-class`: A CSS class name that defaults to `image`. This can be overridden by editing the macro tag, so different image styles can be exposed from your style sheets.

Here is an example of an image HTML template:

```
<div class="%image-class">
  <a href="%node-link">
    <img src="%src" width="%width" height="%height" alt="%alt" /></a>
  <div class="caption">%caption</div>
</div>
```

When using the pop-up window to insert images, the Image Assist module performs a nice favor for you in that it loads the body text of the image nodes in case you want to use that as the image caption. The Preload Image Captions field, set to enabled by default, lets you turn this feature off.

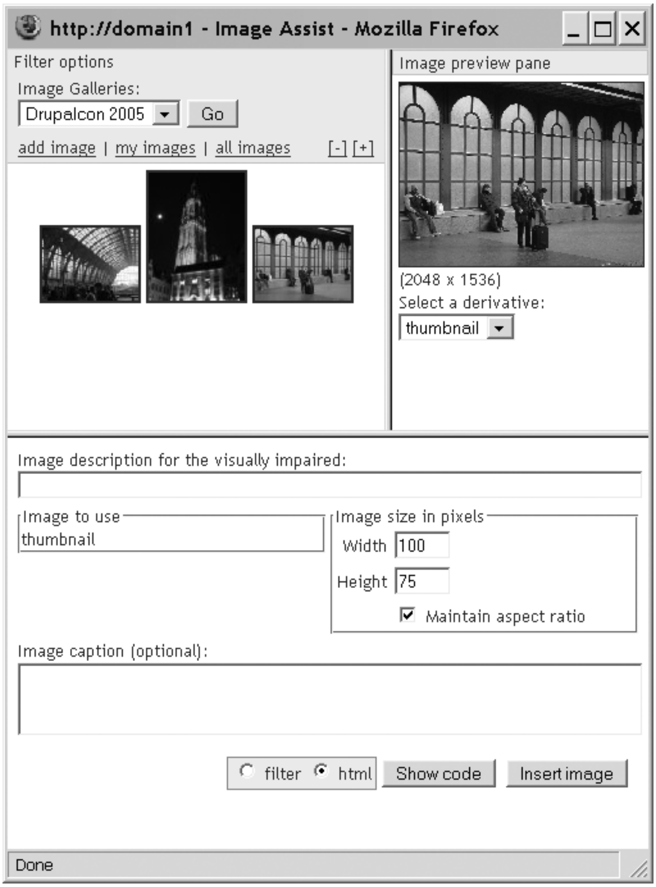
### Image Assist Preview Settings

The Default Derivative Selection field on the Image Assist settings page (`admin/settings/img_assist`) lets you decide which derivative size should be used as a default from the list of possible derivative sizes. It takes the list of derivatives from the Image module's settings.

The final two settings, Default Width of Image Thumbnail Previews and Max Number of Images to Preview, allow you to customize the size and number of thumbnail previews that appear in the pop-up window.

## Using Image Assist

Now that you've configured the Image Assist module, you can use it to insert in-line images into posts. When you create a new story, blog, page, or similar node type that has one or more text areas for content, you will notice the Image Assist icon directly below the text areas. Click this icon, and the Image Assist pop-up window will appear (so make sure your browser allows pop-ups for your site), as shown in Figure 4-6.



http://domain1 - Image Assist - Mozilla Firefox

Filter options

Image Galleries: Drupalcon 2005

Image preview pane

(2048 x 1536)

Select a derivative: thumbnail

Image description for the visually impaired:

Image to use: thumbnail

Image size in pixels

Width 100

Height 75

☑ Maintain aspect ratio

Image caption (optional):

○ filter ● html

Done

**Figure 4-6.** *The Image Assist icon and pop-up window*

Use this pop-up window as follows:

- If you set a value for Image Preview filter on the Image Assist settings page, you can use the Filter options to select the images you want to see by category. Each selection box represents a vocabulary. Click Go to update the selection.
- If the thumbnails are too small to see clearly or too large to fit nicely on the page, use the plus and minus links to change their size.
- You can upload an image by clicking the *add image* link. It works just like adding an image with the Image module.
- Click a thumbnail to see the preview version in this frame.
- You can enter some text in the Image Description for the Visually Impaired field. This will be the value of the `alt` parameter in your `<img>` tag.
- The Image to Use setting lets you choose between the available derivative sizes of the image for inclusion in your post.

---

**Caution** Before deciding to include the original image, make sure to refer to the Image Size in Pixels dimensions, to see if the image is appropriately sized to fit on the page without ruining your theme's layout.

---

- If the original is too big and the thumbnail is too small, you can adjust the size of the displayed image. Leave Maintain Aspect Ratio checked, and the height and width parameters will react to any changes by updating the other.

---

**Caution** The Width and Height fields set the `height` and `width` parameters of the `<img>` tag and do not actually resize the image. This is not the best practice, as it can result in more bandwidth use, slower page loading, and poor-quality images. Use these fields only when really necessary.

---

- You can edit the image caption, as represented by the `%caption` variable in the image HTML template. If you have enabled the Preload Image Captions setting on the Image Assist settings page, the body of the image node will appear here. You can edit it to fit your caption needs without changing the original node.
- The Filter and Html radio buttons are available to those users who have the Choose Format Type permission. Choosing Filter (the default) will result in a macro tag. Selecting Html will result in pure HTML.
- You can always preview the code by clicking the Show code button. The code that will be inserted into your post will appear in a second pop-up window for your review.

Finally, when you're ready, click the Insert image button, and the macro tag or HTML will be inserted into the text area where the cursor appeared before you opened the Image Assist window.

Here is an example of the macro tag that is generated:

```
[img_assist|fid=11|thumb=1|alt=Antwerp train station|caption=Antwerp train station]
```

You can remove images from posts simply by deleting the macro tag or HTML that Image Assist generated.

# Flexinode Module

The very popular Flexinode module is designed to give you control over the information that your web site collects. It enables you to define your own data types, called *flexinodes*, with their own fields. These then function just like other content types (blogs, images, polls, and so on) in that web users can create them, edit them, delete them, and so on. The Flexinode content types can be promoted, moderated, and categorized, and they can participate in any other function Drupal provides for node types.

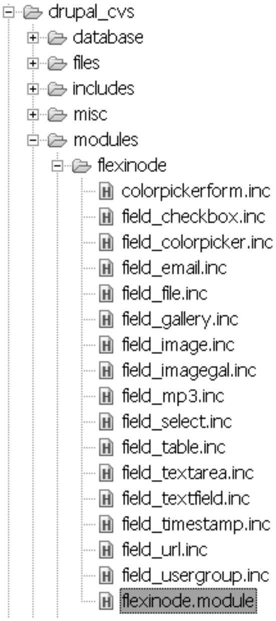
The examples of data types you can create are endless. How about an exercise log tracking what you've done at the gym, or a system to catalog all of the songs a composer has written? Virtually any information that you can imagine collecting through a web interface can be modeled with flexinodes.

## Installing the Flexinode Module

Follow these steps to install the Flexinode module:

1. Download the latest version of Flexinode from `http://drupal.org/project/flexinode`.
2. Place the entire `flexinode` folder in the `/modules/` directory.
3. Load the database definition from the file appropriate to your database, either `flexinode.mysql` or `flexinode.pgsql`.
4. Move all of the files in `flexinode/contrib` into the `flexinode` directory.
5. Delete the empty `flexinode/contrib` directory.
6. After making a backup of your database, import the database definitions (`flexinode.mysql`) using the tool of your choice.
7. Navigate to *administer* ➤ *modules* (`admin/modules`) and enable the module.

Flexinode consists of the Drupal module (`flexinode.module`) and a number of included files representing the various field types that a flexinode can have. Seven of these field types were written by the module's original author (Jonathan Chaffer), and a number of contributed field types were written by various other people. When you put the entire `flexinode` folder into the `modules` directory, you moved all seven of the core field types there. In the archive file that you downloaded from `Drupal.org` is a folder named `flexinode/contrib`, which contains the various contributed field types. After perusing the README text, you can decide which of the contributed field types you would like to have available. These should also go into the modules/flexinode folder on the web server. In the end, an installation of Flexinode with all available field types will look like Figure 4-7.



```
drupal_cvs
    database
    files
    includes
    misc
    modules
        flexinode
            colorpickerform.inc
            field_checkbox.inc
            field_colorpicker.inc
            field_email.inc
            field_file.inc
            field_gallery.inc
            field_image.inc
            field_imagegal.inc
            field_mp3.inc
            field_select.inc
            field_table.inc
            field_textarea.inc
            field_textfield.inc
            field_timestamp.inc
            field_url.inc
            field_usergroup.inc
            flexinode.module
```

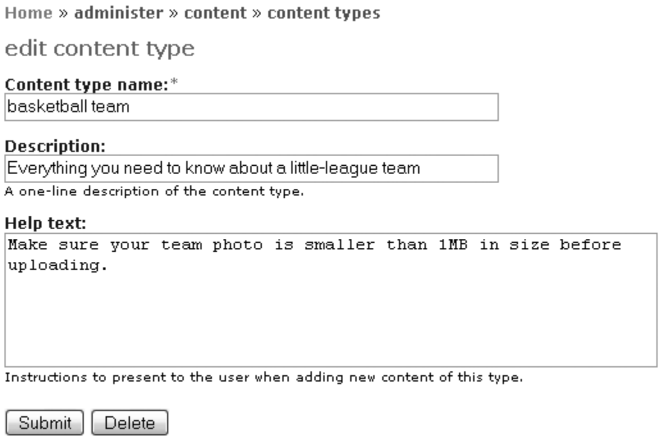
**Figure 4-7.** *Flexinode directory structure*

## Adding Custom Node Types

The first step in creating a custom flexinode type is naming and describing it. Then you can add your content type fields. As an example, let's say you want to create a type called Basketball Team and populate it with fields to track such things as a list of players, the team colors, and the team's motto.

### Content Type Creation

To create a flexinode type, choose *administer* ➤ *content* ➤ *content types* ➤ *add content type* (admin/node/types/add_type). The content type name and description that you choose for your new type are analogous to the names and descriptions you can see for existing node types when you click create content (node/add). The help text will appear at the top of the form when you or other web users create new instances of the custom node. Use this field to give any special instructions that might be helpful to your users when creating new nodes of this type. Figure 4-8 shows an example of defining a new flexinode type named Basketball Team.



edit content type

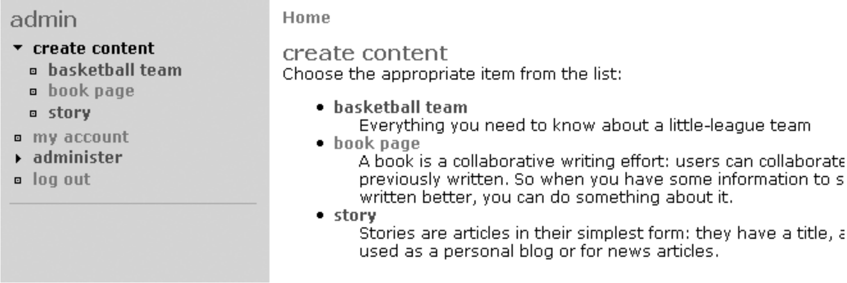
Content type name: *
basketball team

Description:
Everything you need to know about a little-league team
A one-line description of the content type.

Help text:
Make sure your team photo is smaller than 1MB in size before uploading.
Instructions to present to the user when adding new content of this type.

**Figures 4-8.** *Defining a new flexinode type named Basketball Team*

The name and description you give your new content type appear with its choice on the content-creation form. Figure 4-9 shows the Basketball Team type added to the list of content choices.

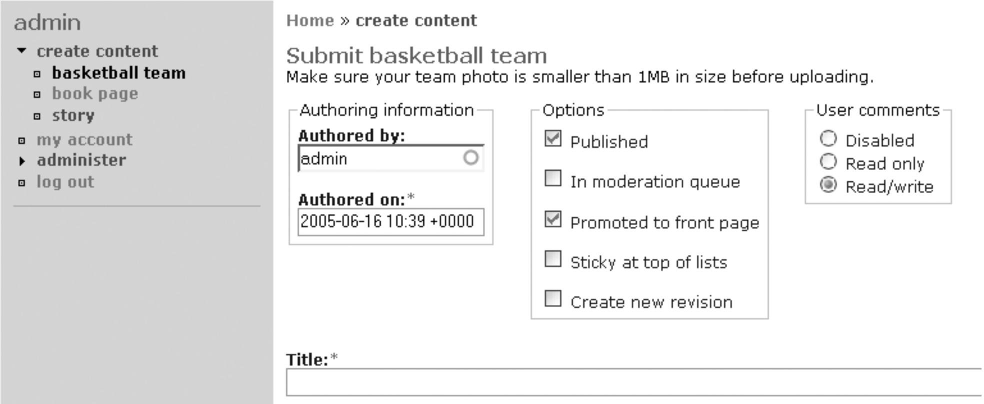


create content
Choose the appropriate item from the list:

- **basketball team**
  Everything you need to know about a little-league team
- **book page**
  A book is a collaborative writing effort: users can collaborate
  previously written. So when you have some information to s
  written better, you can do something about it.
- **story**
  Stories are articles in their simplest form: they have a title, a
  used as a personal blog or for news articles.

**Figure 4-9.** *After creating the Basketball Team flexinode type, it appears in the list of content type choices. Notice its description shows up in this list.*

The help text you added for your new content type is displayed on the content-creation form for that type, as shown in Figure 4-10.



Submit basketball team

Make sure your team photo is smaller than 1MB in size before uploading.

Authoring information

Authored by: admin

Authored on:* 2005-06-16 10:39 +0000

Options

☑ Published
□ In moderation queue
☑ Promoted to front page
□ Sticky at top of lists
□ Create new revision

User comments

○ Disabled
○ Read only
● Read/write

Title:*

**Figure 4-10.** *When you select to use the Basketball Team flexinode type, its help text appears near the the top of the form.*

## Content Type Fields

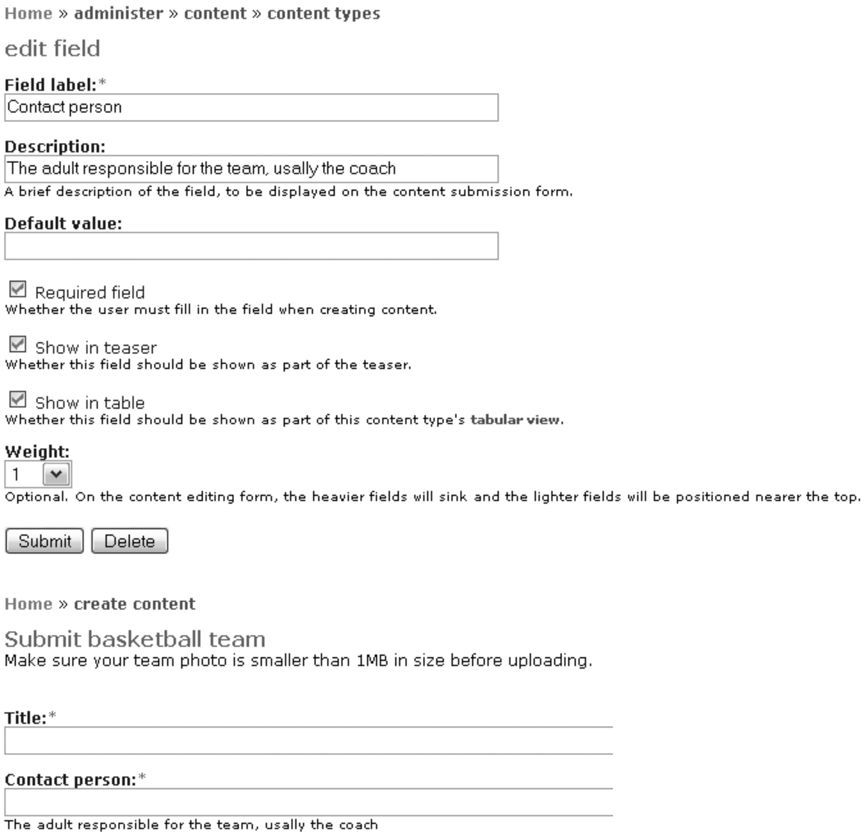
Next, you will want to add some fields to your new content type. In its empty state, it is nothing but a title. Your task is to determine what information you want to collect and how to best represent it.

To add a field, navigate to *administer* ➤ *content* ➤ *content types* ➤ *list* (`admin/node/types`). From this page, you can track all of your flexinode content types. The list of links on the right (*add checkbox*, *add file*, *add image*, and so on) correspond to the field type files that you included in the `modules/flexinode` folder during installation. Clicking one of these links will add a field of this type to the data model of your flexinode.

For the Basketball Team example, let's start with *add textfield* to add a field for the name of the team's contact person (the default Title field can be used for the team's name). The screen for adding a text field has seven fields:

- Field Label is the only required field on this form. The value entered here becomes the label on the text field you are adding to the Basketball Team type.
- The text you enter in the Description field will appear directly underneath the field and serves as a help text in case it isn't clear from the field label what this field is for.
- Entering a value in the Default Value field will prefill the field.
- If the field is to be required, so that it will be impossible to submit the form without filling in a value for it, check the Required Field check box.
- The teaser of a node is the summary version that is shown in lists. If this field is to appear as part of the teaser, check the Show in Teaser check box.
- Each flexinode type has a tabular view where all nodes of that type can be browsed, complete with sortable columns. If this field is to be one of the fields in that table, check the Show in Table check box.
- The Weight value orders the field among all other fields on the form.

Figure 4-11 shows an example of the edit field form for the text field and what it produces on the Baseball Team creation form.



Home » administer » content » content types

edit field

Field label:*
Contact person

Description:
The adult responsible for the team, usally the coach
A brief description of the field, to be displayed on the content submission form.

Default value:

☑ Required field
Whether the user must fill in the field when creating content.

☑ Show in teaser
Whether this field should be shown as part of the teaser.

☑ Show in table
Whether this field should be shown as part of this content type's **tabular view**.

Weight:
1
Optional. On the content editing form, the heavier fields will sink and the lighter fields will be positioned nearer the top.

Home » create content

Submit basketball team
Make sure your team photo is smaller than 1MB in size before uploading.

Title:*

Contact person:*
The adult responsible for the team, usally the coach

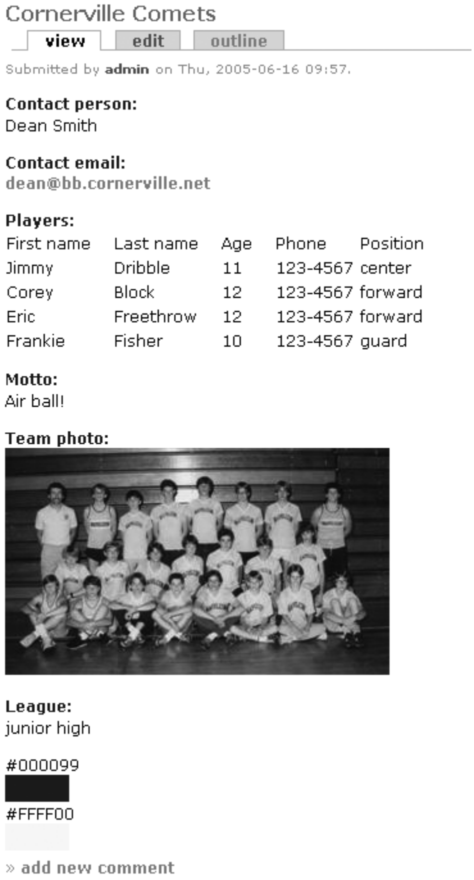
**Figure 4-11.** *Adding a text field to the form*

These seven fields appear on all field type forms. A text field is the simplest field type and has only the standard seven configuration options. Others, such as drop-down menu and table, have extra configuration options in addition to the seven standard ones.

Adding a drop-down menu to the form, for determining which league a team plays in, is similar to adding the text field. Start by clicking *add dropdown menu* from the `admin/node/types` screen. In addition to the seven standard fields, the form for adding a drop-down menu has a group called Options. The values you enter here will become the various options in the content-creation form later. Clicking the More button provides additional fields in case you need more options.

In the same way, you will be able to add the other fields needed to complete the Basketball Team form. The example in Figure 4-12 uses various core and contributed field types, including the following:

- An e-mail address field for Contact email
- A table field for Players
- A text field for Motto
- An image field for Team photo
- Color pickers for the Team color fields



Cornerville Comets

Submitted by **admin** on Thu, 2005-06-16 09:57.

**Contact person:**
Dean Smith

**Contact email:**
dean@bb.cornerville.net

**Players:**

| First name | Last name | Age | Phone | Position |
|---|---|---|---|---|
| Jimmy | Dribble | 11 | 123-4567 | center |
| Corey | Block | 12 | 123-4567 | forward |
| Eric | Freethrow | 12 | 123-4567 | forward |
| Frankie | Fisher | 10 | 123-4567 | guard |

**Motto:**
Air ball!

**Team photo:**

**League:**
junior high

#000099
■
#FFFF00
□

**Figure 4-12.** *A finished basketball team node*

## Table View for the Content Type

Once you have finished designing your flexinode content type and have created a number of actual nodes—basketball teams, in this example—you can view the results in the tabular view that flexinode offers. There is no link offered to this tabular view, so you must create one yourself using the primary or secondary links, or perhaps with the Menu module.

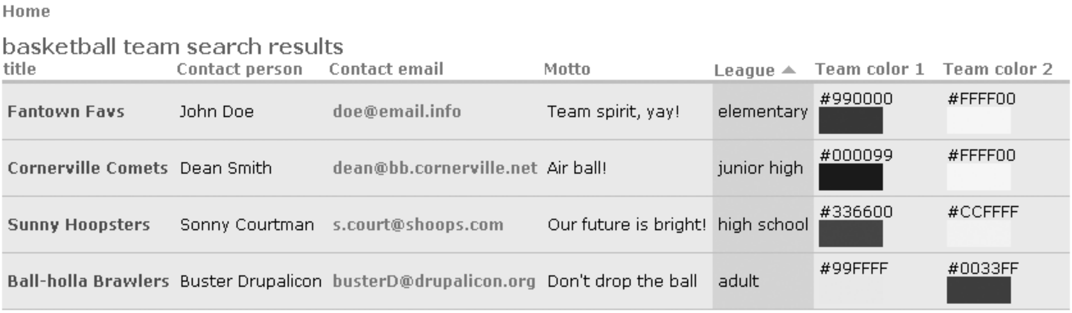
The path to the table view takes the form `flexinode/table/type_id`, where *`type_id`* is the numerical ID assigned to the content type you wish to view. To find out exactly what this ID is, you will need to do a little sleuthing. Go to the content type editing screen for the type that you are interested in: *administer* ➤ *content* ➤ *content types* ➤ *edit content type*. The URL in your browser's address bar should look something like this:

```
http://localhost/drupal/?q=admin/node/types/edit_type/1
```

From this URL, you see that the type ID number is 1, thus the URL to the table view for this content type is as follows:

```
http://localhost/drupal/?q=flexinode/table/1
```

Figure 4-13 shows the table view for the sample Basketball Team flexinode type.



Home

basketball team search results

| title | Contact person | Contact email | Motto | League ▲ | Team color 1 | Team color 2 |
|---|---|---|---|---|---|---|
| Fantown Favs | John Doe | doe@email.info | Team spirit, yay! | elementary | #990000 | #FFFF00 |
| Cornerville Comets | Dean Smith | dean@bb.cornerville.net | Air ball! | junior high | #000099 | #FFFF00 |
| Sunny Hoopsters | Sonny Courtman | s.court@shoops.com | Our future is bright! | high school | #336600 | #CCFFFF |
| Ball-holla Brawlers | Buster Drupalicon | busterD@drupalicon.org | Don't drop the ball | adult | #99FFFF | #0033FF |

**Figure 4-13.** *The table view for basketball teams*

# Event Module

Sites interested in publicizing information about upcoming events (concerts, meetings, deadlines, and so on) will find the Drupal Event module very useful. An *event*, in Drupal's view, is any node that has a start, and possibly an end, time. This flexibility opens the door for any node type to become an event. Events can be viewed in calendars, and the calendars can be filtered to show events based on node type and taxonomy categories.

As any node type can be "event-enabled," you can use the Flexinode module to create various event types. A site for a music school could create Concert, Rehearsal, Lecture, and Holiday (when the school is closed) event types. The Concert and Rehearsal event types could then use an ensemble taxonomy vocabulary to show when and where ensembles are rehearsing or playing concerts. It would then be possible to show calendars and schedules for queries such as "concerts and rehearsals for the symphony orchestra in May and June."

The Event module works particularly well for tracking the "when" aspect of events. Later in the chapter, I will show you how you can use the Location module to track the "where" part of events.

## Installing the Event Module

Follow these steps to install the Event module:

1. Download the latest version of the Event module from `http://drupal.org/project/event`.

2. Perform the recommended backup of your database, and then import the database definitions (`event.mysql` or `event.pgsql`) using the tool of your choice.

3. Place the Event module with all its files under `modules/event`.

4. Navigate to *administer* ➤ *modules* (`admin/modules`) and enable the module.

Once the Event module has been installed, you need to configure it and also decide which node types are to become event-enabled. After you've configured the Event module and enabled at least one node type to be event-aware, you will be ready to create some events.

## Configuring the Event Module

Select *administer* ➤ *settings* ➤ *event* (`admin/settings/event`) to see the configuration settings for the Event module. Depending on where your site's audience is located, you need to decide how to save and display time zones. You also have options for the display of events.

### Time Zone Input

The first field, Event Time Zone Input, is a means of locating the event, answering the question, "In which time zone does this event take place?" You have three options:

**Use the sitewide time zone**: This setting uses the site-wide setting that you set from the site settings page (`admin/settings`). If the web site and your events are in the same place, such as for an office intranet or a web site about sports in Boston, this option is appropriate.

**Use the time zone of the user editing or creating the event**: This uses the user's time zone (as configured on the user's profile page). This assumes that users will create only events that are local to them and that they have correctly configured their personal profile to reflect their time zone. This seems like too many assumptions to be practical in most cases.

**Allow users to set event time zones**: This setting lets users set the time zone for each event. If your site is going to track events from a large geographical region, this option is probably the best choice.

### Time Zone Display

The Event Time Zone Display field answers the question, "The time for this event should be displayed adjusted to which time zone?" Imagine that your site is configured to Eastern Standard Time (EST). Jim in Chicago uses your site to announce his upcoming concert. which starts at 18:00 (6:00 p.m.). Since he and his concert are in Central Standard Time (CST), he saves the event with CST. George in California, which is on Pacific Standard Time (PST), visits the site and sees the announcement for Jim's concert. What time should George see: 18:00 CST (the event's time zone), 16:00 PST (George's time zone), or 19:00 EST (the web site's time zone)? These are your three choices for the Event Time Zone Display field:

**Use the event's time zone**: If it is an event that you want to attend (in the place that it is happening), this is the most logical setting. Since you will attend the event in person, you will naturally be in the same time zone as the event.

**Use the user's time zone**: Let's say the concert is also being broadcast by radio or webcast. Now you will definitely want to see the time of the concert adjusted to the time zone you are in, thus making this setting the better choice.

**Use the sitewide time zone**: This is probably not the best setting for most cases.

The Time Notation Preference field determines whether times will be displayed in 12- or 24-hour notation.

### Event Block

The Event module defines a block that shows upcoming events. You can enable this block from the block administration page (`admin/blocks`). On the Event module configuration page, the Upcoming Event Block Limit setting allows you to control how many events will be displayed in the block.

■ **Note** The Event module defines two blocks: Calendar to Browse Events and List of Upcoming Events. You can activate and configure these blocks by navigating to *administer* ➤ *blocks* (`admin/blocks`). See Chapter 2 for details about administering blocks.

### Event Overview Options

The events calendar has five different views: day, week, month, table, and list. Use the Default Overview field to select which of these views should be the default calendar view. If you choose Table, you can also use the Table View Default Period setting to specify how many days are to be shown.

When looking at the overview of events on a calendar view, it is possible to filter the display based on node type and taxonomy category. The queries for these filters are built by constructing special URLs, as described next. The Taxonomy Filter Controls field sets the display of this filter:

**Never show taxonomy filter control**: This allows you to hide the taxonomy filter control.

**Only show taxonomy filter control when taxonomy filter view is requested**: This shows the filter control only when a taxonomy query is built into the URL query.

**Show taxonomy filter control on calendar views**: This shows the filter control by default.

The same three options can also be applied to the Content Type Filter Controls field, which lets you filter the calendar view by node type.

## Event-Enabling Node Types

For each type that you wish to be an event type, the content-creation form will have two extra fields for the start date and end date. As I mentioned earlier, a convenient way to make event types is by using the Flexinode module to create node types specifically for the purpose. These flexinode types should answer the *what* part of the question “What type of event is this?” Building on the music school example, you could define a Concert flexinode type that included flexinode fields to describe the repertoire and ticket-buying information. The Event module would take care of the *when* part: “When does this concert take place?”

To make a particular node type event-enabled, go to *administer* ➤ *settings* ➤ *configure* (for the particular content type) and set the Show in Event Calendar field. The default setting for this field is Never, which means that a node is not event-enabled, will not have the start and end fields, and will not show up in the calendar. Change this to either of the following settings:

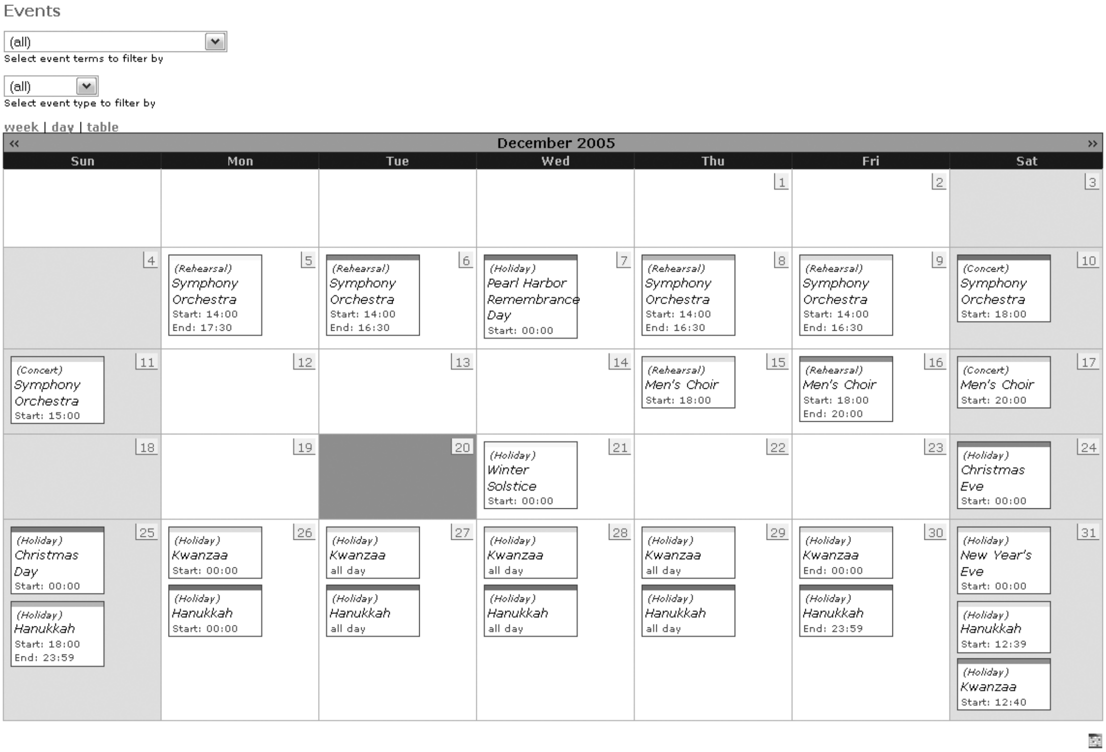
**All views**: This means that the content type is event-enabled, will have the start and end fields, and can appear in calendars with other event types.

**Only in views for this type**: This also means that the content type is event-enabled and will have start and end fields, but it will show up only in calendars specific to this type. In other words, it will have a completely separate calendar all to itself.

## Viewing Events

As noted earlier, the content-creation forms for your event types (`node/add/`*`type`*) will now have fields for entering starting and ending dates and times. If the ending time is earlier than the starting time, it is ignored, which is convenient for events where no ending time is specified. If an event spans more than one day, it will show up on all of the intermediary days in the calendar.

When you’re viewing an event-enabled node, the start and end times will be shown, and a link to the calendar view will appear at the bottom. Clicking the *calendar* link will show the calendar with the day of the event highlighted. (The Drupal path to the current events calendar is `event`.) Figure 4-14 shows an example of a calendar view.



Events

(all)

Select event terms to filter by

(all)

Select event type to filter by

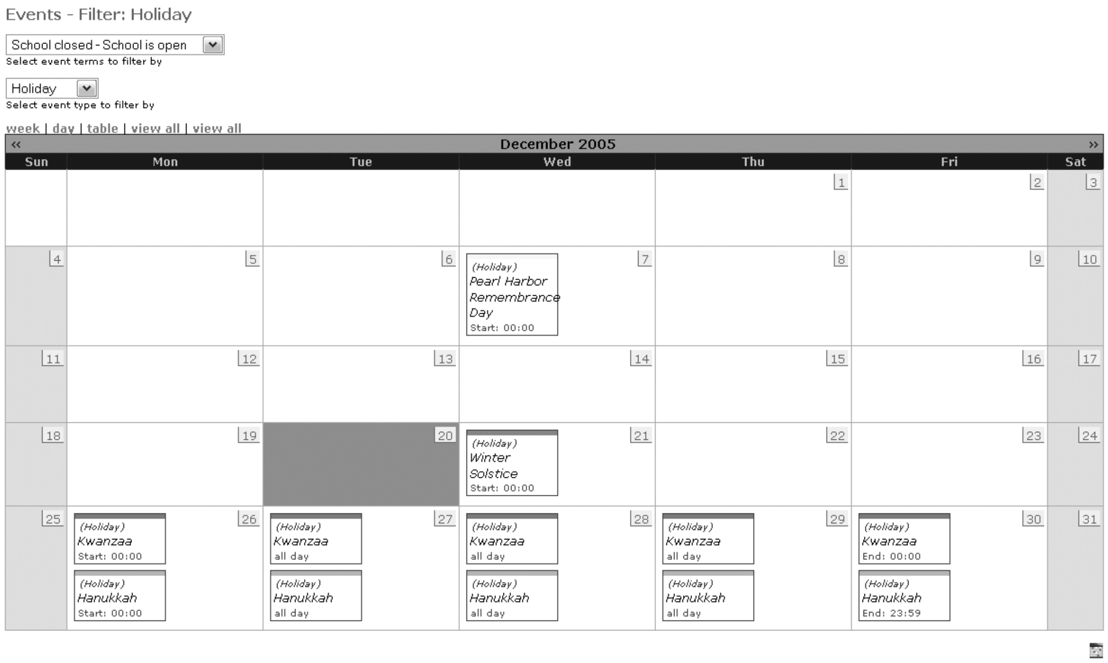
week | day | table

« December 2005 »

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
| | | | | 1 | 2 | 3 |
| 4 | 5<br>(Rehearsal)<br>Symphony Orchestra<br>Start: 14:00<br>End: 17:30 | 6<br>(Rehearsal)<br>Symphony Orchestra<br>Start: 14:00<br>End: 16:30 | 7<br>(Holiday)<br>Pearl Harbor Remembrance Day<br>Start: 00:00 | 8<br>(Rehearsal)<br>Symphony Orchestra<br>Start: 14:00<br>End: 16:30 | 9<br>(Rehearsal)<br>Symphony Orchestra<br>Start: 14:00<br>End: 16:30 | 10<br>(Concert)<br>Symphony Orchestra<br>Start: 18:00 |
| 11<br>(Concert)<br>Symphony Orchestra<br>Start: 15:00 | 12 | 13 | 14 | 15<br>(Rehearsal)<br>Men's Choir<br>Start: 18:00 | 16<br>(Rehearsal)<br>Men's Choir<br>Start: 18:00<br>End: 20:00 | 17<br>(Concert)<br>Men's Choir<br>Start: 20:00 |
| 18 | 19 | 20 | 21<br>(Holiday)<br>Winter Solstice<br>Start: 00:00 | 22 | 23 | 24<br>(Holiday)<br>Christmas Eve<br>Start: 00:00 |
| 25<br>(Holiday)<br>Christmas Day<br>Start: 00:00<br>(Holiday)<br>Hanukkah<br>Start: 18:00<br>End: 23:59 | 26<br>(Holiday)<br>Kwanzaa<br>Start: 00:00<br>(Holiday)<br>Hanukkah<br>Start: 00:00 | 27<br>(Holiday)<br>Kwanzaa<br>all day<br>(Holiday)<br>Hanukkah<br>all day | 28<br>(Holiday)<br>Kwanzaa<br>all day<br>(Holiday)<br>Hanukkah<br>all day | 29<br>(Holiday)<br>Kwanzaa<br>all day<br>(Holiday)<br>Hanukkah<br>all day | 30<br>(Holiday)<br>Kwanzaa<br>End: 00:00<br>(Holiday)<br>Hanukkah<br>End: 23:59 | 31<br>(Holiday)<br>New Year's Eve<br>Start: 00:00<br>(Holiday)<br>Hanukkah<br>Start: 12:39<br>(Holiday)<br>Kwanzaa<br>Start: 12:40 |

**Figure 4-14.** *The month of December showing rehearsals, concerts, and holidays*

The calendar view comes with selection boxes that you can use to filter the events being displayed. This is particularly useful when you're looking at a crowded calendar. The first selection box allows you to filter the calendar by taxonomy category, and the second filters by node type. Figure 4-15 shows an example of filtering by a taxonomy category.



Events - Filter: Holiday

School closed - School is open

Select event terms to filter by

Holiday

Select event type to filter by

week | day | table | view all | view all

« December 2005 »

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
| | | | | 1 | 2 | 3 |
| 4 | 5 | 6 | 7<br>(Holiday)<br>Pearl Harbor Remembrance Day<br>Start: 00:00 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 18 | 19 | 20 | 21<br>(Holiday)<br>Winter Solstice<br>Start: 00:00 | 22 | 23 | 24 |
| 25 | 26<br>(Holiday)<br>Kwanzaa<br>Start: 00:00<br>(Holiday)<br>Hanukkah<br>Start: 00:00 | 27<br>(Holiday)<br>Kwanzaa<br>all day<br>(Holiday)<br>Hanukkah<br>all day | 28<br>(Holiday)<br>Kwanzaa<br>all day<br>(Holiday)<br>Hanukkah<br>all day | 29<br>(Holiday)<br>Kwanzaa<br>all day<br>(Holiday)<br>Hanukkah<br>all day | 30<br>(Holiday)<br>Kwanzaa<br>End: 00:00<br>(Holiday)<br>Hanukkah<br>End: 23:59 | 31 |

**Figure 4-15.** *The month of December showing only holidays when school is open (a taxonomy category)*

All of the parameters governing time range, category, and type of event can be built into the URL. This makes URL manipulation an effective tool for querying a Drupal site for its event information. Here is how event URLs are built:

```
?q=event/$year/$month/$day/$view_type/$content_type/$taxonomy_terms/$duration
```

The event URL variables are as follows:

- `$year`: A four-digit integer, such as 2005.
- `$month`: An integer from 1 to 12 representing the month.
- `$day`: A two-digit integer representing the day of the month to start from. Leading zeros are required, as in 03.
- `$view_type`: The type of calendar layout. Values include `month`, `week`, `day`, `table`, `ical` for an iCal export, and `feed` for an RSS feed.
- `$content_type`: A list of node types, separated by a plus sign (`+`). Flexinode types require only the integer value of the type. For example, to view story nodes and flexinode type 1, you would use `story+1`. The value `all` will show all types.

- `$taxonomy_terms`: A list of term IDs, separated by a plus sign (`+`). For example, to view entries assigned to the taxonomy terms with IDs 4 and 9, you would use `4+9`. The value `all` will show all terms.
- `$duration`: The number of days to display. Currently, only the table view observes this setting.

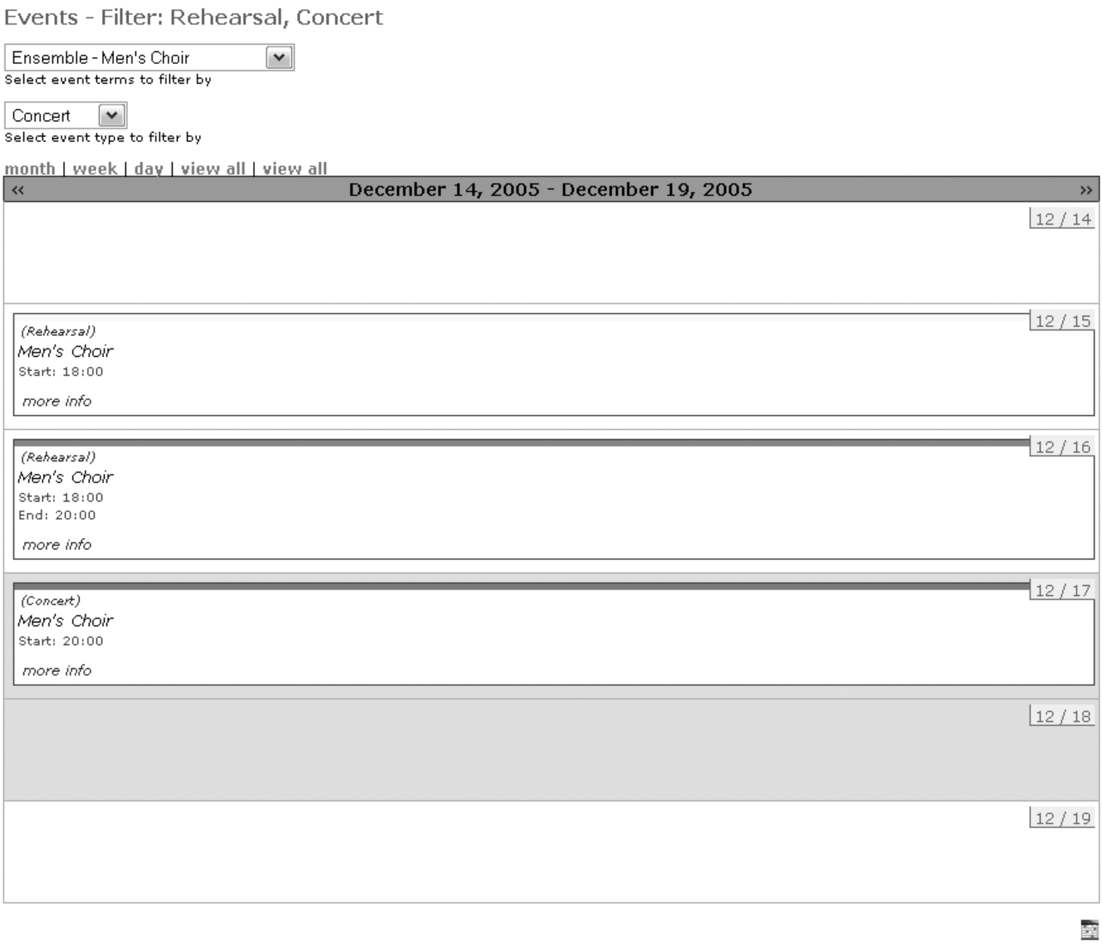
For example, suppose you wanted to see all of the Men's Choir concerts and rehearsals for the five-day period December 14 to 19. Here's what the URL would look like:

```
http://domain/?q=event/2005/12/14/table/1+2/10/5
```

This URL is built as follows:

- `$year`: `2005`
- `$month`: `12`, for December
- `$day`: `14`, the starting day
- `$view_type`: `table`
- `$content_type`: `1+2`, which means Concert and Rehearsal, since these are flexinode 1 and flexinode 2 types
- `$taxonomy_terms`: `10`, the taxonomy term for Men's Choir
- `$duration`: `5`, for five days

The result is a subset of the events shown in a tabular view, as illustrated in Figure 4-16.



Events - Filter: Rehearsal, Concert

Ensemble - Men's Choir

Select event terms to filter by

Concert

Select event type to filter by

month | week | day | view all | view all
«
December 14, 2005 - December 19, 2005
»
12 / 14
12 / 15

(Rehearsal)
Men's Choir
Start: 18:00
more info

(Rehearsal)
Men's Choir
Start: 18:00
End: 20:00
more info

(Concert)
Men's Choir
Start: 20:00
more info

**Figure 4-16.** *Men's Choir concerts and rehearsals from December 14 to 19*

## Exporting Event Information

You can export your Drupal site's event information for external use by any program that supports the webcal protocol, such as iCal (`http://www.apple.com/macosx/features/ical/`) or as an RSS feed. The icon at the bottom right of the screen underneath the calendar will export the current display for iCal.

You can generate both RSS and iCal format using the URL manipulation techniques described in the preceding section, by substituting `ical` or `feed` for the `$view_type` variable. For example, for an RSS feed, the URL for displaying the Men's Choir rehearsals and concerts could be rewritten like this:

```
q=event/2005/12/1/feed/1+2/10
```

# Location Module

Adding location information to content can be very useful. Whether it is a shipping address, a meeting place, or the place where a photograph was taken, location adds an interesting dimension and opens up new possibilities for how communities form and how web sites are used. The Location module provides a set of location services, including appending location information to content and users, doing proximity searches, and linking to external map providers.

Much like the Event module, the Location module lets you choose what content should have location information, and the concept of a location-enabled node type is analogous to the event-enabled node type, but with different information. Furthermore, the Location module complements the Event module in a very meaningful way in that it can be used to answer the question "Where does this event take place?" To draw on the examples in the Event module section of this chapter, the Concert and Rehearsal flexinode types could be location-enabled, making the events calendar even more useful.

## Installing the Location Module

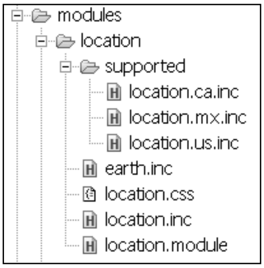
The Location module is made up of the following parts:

- A public API definition describing how to work with location information in Drupal. This can be found in the file `location_API.txt`.
- `location.module`, an implementation of the Location module
- `location.inc`, a set of common tools
- `earth.inc`, a set of tools for working with latitude and longitude values
- A set of 243 files found in `location/supported`, each representing a different country, providing location services that range from a list of states or provinces to proximity searches and geocoding of postal addresses
- `location/database/zipcodes.mysql`, the definition for an optional zip codes table, which is required for the advanced features like geocoding
- A growing number of `location/database/zipcodes.`*`country-code`*`.mysql` data dumps to provide zip code support for specific countries (the United States and Germany are currently available)

Follow these steps to install the Location module:

1. Download the latest version of the Location module from `http://drupal.org/project/location`.
2. Make a backup of your database.
3. Load the database definition from the file appropriate to your database, either `location.mysql` or `location.pgsql`.

4. Transfer the necessary files to the web server. As the `location` folder contains several very large data dumps and a couple hundred country-specific files, it is worth the effort to select only the files that you actually need. Figure 4-17 shows an example of a minimal directory structure for the Location module, which will be able to support locations in Mexico, the United States, and Canada.



```
modules
  location
    supported
      location.ca.inc
      location.mx.inc
      location.us.inc
    earth.inc
    location.css
    location.inc
    location.module
```

**Figure 4-17.** *Location module directory structure*

5. If you wish to work with the finer features of the Location module, create a table called `zipcodes` using the file `supported/zipcodes.mysql`. Then, depending on which data dumps are available, import zip code data into this table. The `supported/zipcodes.us.mysql` file, for example, provides the data dump for the United States. If you intend to run multiple location-enabled sites on the same database server, you might consider putting the `zipcodes` table in a separate database and sharing it among all the sites. Since it is used only for looking up zip code information and is never written to, this is a safe and efficient approach. See Chapter 6 for details on running multiple sites and database table prefixing.

6. Navigate to *administer* ➤ *modules* (`admin/modules`) and enable the module.

As with the Event module, your next steps are to configure the module and make some nodes location-enabled.

## Configuring the Location Module

Select *administer* ➤ *settings* ➤ *location* (`admin/settings/location`) to see the configuration settings for the Location module.

### Country Selection

As a convenience, the Location module lets you set a default country for your web site with the Default Country Selection field. This should be the country that generates the largest portion of your traffic. Whichever country you set here will be selected by default in all location forms.

If the locations your site tracks are all within one country, you can simplify matters even further by checking Hide Country Selection, which saves your users the step of selecting the country. The default country will be used instead.

### Distance Units

Distances are an issue only when you're using the Location module to do proximity searches, which is possible only if you are working with one of the available zip code databases. In this case, you can set the Distance Unit field to select the units in which distances will be displayed for location-based searches.

■**Note** The Drupal path to the location module's proximity search screen is `search/location`.

### Location Display

The Location module handles the task of displaying location information (an address, basically) for those nodes that have it available. You might want to display this information some other way. The Toggle Location Display field allows you to turn off the default display of location information. The idea is that you would then write a themable function that would display the information in the form that you want. See Chapter 5 for instructions for writing custom themable functions.

### User Location Information

You can collect location information for nodes and users. Knowing where your users are located can be very helpful, depending on the type of site you are building. For example, if your site is being used for a music school to track orchestra rehearsals and concerts, it might be very useful for your users to be able to add their address to their profile information for the sake of organizing carpools to rehearsals. The User Locations field turns this functionality on or off.

### Country Location Features

The last section on the Location module settings page, entitled "Enable all *available* features for locations from the following countries," is a list of countries that have specific features available. It corresponds directly with the files found in the `modules/location/supported` folder.

The vast majority of the countries listed offer the feature of knowing their states or provinces and nothing else. Thus, the major practical implication of enabling ten or so countries from central Africa is that you will need to scroll through all these countries and their provinces when entering location information for specific nodes later. At the time of this writing, only three countries have significant features to offer: the United States, Canada, and Germany. The United States has the most full-featured set, including proximity searches and deep linking into map services such as Google or Yahoo! Maps for addresses. Germany and Canada have only partially implemented sets of features.

■**Tip** If you would like to implement further location features for a country that does not have them, the `location_API.txt` file provides an excellent description of the programming API that needs to be used. The `supported/location.us.inc` file is a complete sample implementation of the API.

### Location-Enabling Node Types

With the basic configuration options settled, the next step is to decide which node types should collect location information and exactly which information is desired or required. Node type configuration takes place from the `admin/node/configure/types` page, which lists all of the node types. Click *configure* for any of the listed node types, and you will see a section of the form titled Locative Information. The Enable for Locations field does just what it says. The rest of the fields allow you to customize the location form by making fields invisible, optional, or required, based on your specific needs.

## Organic Groups Module

One of the hallmarks of vibrant online communities is the presence of a common interest or theme that unites those who are involved. Whether this common interest is relatively broad or narrow, there are likely to be smaller but related topics that are of interest only to a subset of the community. For instance, on a music school site, the brass players might want to organize their own rehearsals and brass ensemble concerts. The ability for these people to band together and form a group is the main offering of the Organic Groups (OG) module.

Groups have their own home page, can select their own theme (design template), can have content that is visible only to members of the group, and can have image galleries. Most important, they can be created and joined by users as the need arises, thus growing and thriving based on the current interests of the community. This is what makes them organic.

Until this point, none of the content on your Drupal site has had any sort of node-level access control. This means that if user A and user B have the same roles, they both are able to see the same content. The Organic Groups module is one of the modules that implement node-level permissions to decide who can see content. If user A is in a group and user B is not, and if that group's content is private, user B will not be able to see the same content that user A sees, despite the fact that they have the same roles.

Organic Groups isn't the only module in the contributions repository that implements node-level access controls. Other modules that provide access controls are Node Privacy by Role (`http://drupal.org/project/node_privacy_byrole`), Taxonomy Access Control (`http://drupal.org/project/taxonomy_access`), and Nodeperm Role (`http://drupal.org/project/nodeperm_role`). Unfortunately, these modules tend to interfere with each other. This is a basic weakness in the way Drupal's permissions are handled and is a topic of hot debate in the developer community. Be warned: do not try to mix multiple node-access modules!

---

■ **Caution** Do not try to use the Organic Groups module with other node-level access modules, as they are not designed to work together.

---

### Installing the Organic Groups Module

Please take the time to back up your database before installing the Organic Groups module. While this is good advice for every module that has its own database definitions, it is especially important in this case, as the installation of the Organic Groups module will make fundamental changes in the database's data, and that will alter the way the database behaves. Specifically, there is a table, `node_access`, which controls whether a given user can view a given node. In the absence of a node-level access module such as Organic Groups, this table contains one row that grants access to all nodes to every user. Organic Groups will delete this row and instead manage node permissions on a per-node basis. While the Organic Groups module itself provides a mechanism for uninstalling itself, it is best to secure a backup before installing the module.

Download the latest version of the Organic Groups module from `http://drupal.org/project/og`. After making your backup, you can load the database definition from the `og.mysql` or `og.pgsql` file. Copy the `og` folder into your `modules` folder, and activate the Organic Groups, the Organic Groups Albums (og_album), and Taxonomy modules from the `admin/modules` page.

The Organic Groups module depends on the Image module and the Organic Groups Albums module to provide each group with its own photo gallery. Installing the Image module was covered earlier in this chapter.

---

■**Note** In order for your groups to have their own image galleries, you will need the Image, Taxonomy, and Organic Groups modules. The Organic Groups module comes packaged with the Organic Groups download. The Image module is available from `http://drupal.org/project/image`. The Taxonomy module is a core module and is discussed in Chapter 3.

---

## Activating Group Blocks

The Organic Groups module makes almost no sense without the Group Details block activated, so the logical place to start is on the block administration page (`admin/blocks`). Enable this block and decide where it should be located (right, left, weight, and so on). It is not necessary to specify paths where this block should or shouldn't appear, because the module takes care of this for you.

You can also decide whether you would like to activate the Group Subscribers and Group Albums blocks.

## Configuring the Organic Groups Module

Select *administer* ➤ *settings* ➤ *og* (`admin/settings/og`) to see the configuration settings for the Organic Groups module.

### Access Control Status

First, check the current status of the Organic Groups access permissions. In the Module Status section at the top of the page is a status message that should read "Organic groups access control is currently enabled" and a button that will should be labeled Disable when you first view this screen. Confirm that access control is enabled. If the status message tells you that the Organic Groups access control is disabled, use the button (which is labeled Enable) to enable access control before proceeding with further configuration.

You can use the Disable button to disable Organic Groups if you want to stop using it for whatever reason. The Organic Groups module can be enabled or disabled as many times as you like without negative side effects or loss of previous group-related information.

**Caution** Deactivating the Organic Groups module from the module administration page (`admin/modules`) without first disabling the module from the Organic Groups settings page (`admin/settings/og`) will lead to problems.

### Submission Guidelines

The Explanation or Submission Guidelines provide you with a chance to offer guidance to anyone who is in the process of creating a new group. Depending on how you configure the Organic Groups module, there will be different options available in the form for creating new groups, so you are best advised to come back to this once you have made other fundamental decisions and experimented with the module a bit. Only then will you be able to judge which parts of the process might not be clear to your user base and thus what instructions they should be given.

### Visibility of Posts

Someone must decide whether a given node should be visible only within a group or whether it also should be visible to others not in the group. There are cases for both usages. If groups are being used on your site to do private secretive work and the Organic Groups module is being used to provide this privacy, then the nodes that are created for that group should be visible only to the members of that group and no one else. On the other hand, if groups are being used to organize content that should otherwise be visible to nongroup members, nodes created for that group should remain accessible to the wider public. The Visibility of Posts field lets you decide who gets to make these decisions. It offers three choices:

- Visible only within the targeted groups
- Visible within the targeted groups and on other pages
- Visibility to be determined by the author/editor using a check box on the posting form

The first two options let you, the site administrator, make the visibility decision. The third option defers the decision to the user who creates content. The content-creation form will include a check box titled Public, and the node creator is left to make the choice.

### Maximum Posts and User Pictures

The Maximum Posts on Group Home Page field determines how many content teasers will be shown on the group's home page before pagination begins. Checking the Show Member Pictures field will result in avatars being shown for content posts. However, this works only if user pictures have been configured (by selecting *administer* ➤ *settings* ➤ *users*).

### Audience Required

Also found on the content-creation forms is a check box for each group to which a user belongs. The Audience Required setting controls whether this check box is a required field. Making the check box Required will disallow users from submitting content before assigning the content to at least one group. Set this to Required to make sure that the site's content exists only within the context of groups. Otherwise, choose Optional to give users the choice of whether their content should be posted to a group.

### Omitted Content Types

The Omitted Content Types field allows you to exclude certain node types from the groups functionality altogether. This means that group-based visibility will not apply to nodes of this type. Excluded node types will simply have site-wide visibility. This is an effective way to help establish sections of the site that are for the community at large, while reserving other areas for groups that form around specific areas of interest.

## Configuring Organic Groups Albums

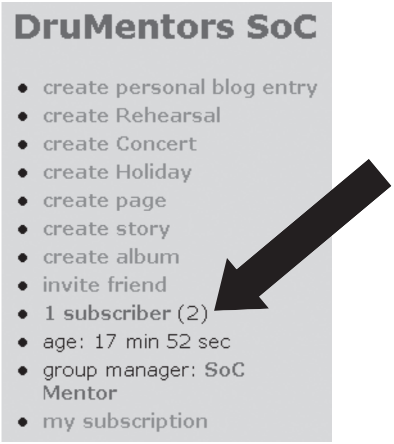
In order to let your users' groups make photo albums, you need to have a free-tagging taxonomy vocabulary that can be used to manage the structure of the albums. An appropriate vocabulary will be created for you when you navigate to *administer* ➤ *settings* ➤ *og_album* (`admin/settings/og_album`). Visiting the page for the first time will prompt Drupal to create and configure a vocabulary called Group Albums. If you wish to use a different free-tagging vocabulary, you can update the Select Vocabulary field.

## Creating Groups

Groups are another content type, just like blogs, pages, and stories. To create a group, you use the *group* link on the *Create content* page (`node/add`). In this case, the group node that you create serves as a container for all of the other content and activities of your group. You have the same general services that Drupal provides for other types of nodes, such as the ability to be categorized with the taxonomy system. Note that you must have the Create Groups permission to create a group.

When you create a new group, you will be faced with several decisions, among them the name and description of the group. You also get to decide how new group members will be admitted by setting the Selective field, which has the following options:

> **Open**: This means that subscription requests are accepted immediately. Open is the easiest to maintain and the most organic.
>
> **Moderated**: This means that subscription requests must be approved. If the group is moderated, the person who created the group is the manager. The manager will have the chance to approve or deny the request for subscription via the *subscribers* link in the Group Details block. This link shows not only how many current subscribers there are, but also how many requests for subscription have been made (in parentheses, as shown in Figure 4-18).
>
> **Invite only**: With this setting, subscriptions must be created by an administrator. If the group is made to be invite only, the moderator should add users by clicking the *add subscribers* link that is visible as a tab on the page listing all of a group's subscribers. Users are added by entering a comma-separated list of usernames.
>
> **Closed**: Subscriptions are managed fully by the administrator. Users cannot choose to join or invite others.



DruMentors SoC

- create personal blog entry
- create Rehearsal
- create Concert
- create Holiday
- create page
- create story
- create album
- invite friend
- 1 subscriber (2)
- age: 17 min 52 sec
- group manager: SoC Mentor
- my subscription

**Figure 4-18.** *The Group details block and the link to manage subscriptions*

■**Note** The moderator of a group is the person who created the group. It is not currently possible to have more than one moderator. To change the moderator of a group, a user with the Administer Nodes permission must edit the group node and change the Authored By field to the username of the new moderator. Since the Administer Nodes permission is almost synonymous with being a site administrator, you should not grant this permission solely for the sake of changing group moderators. Until this facet of the Organic Groups module is better developed, it is advisable to either disallow the changing of moderators or to handle all requests to do so yourself. Be careful to whom you grant Administer Nodes permission!

## Managing Groups

The Group Details block is the control panel for a group. It is visible on a group's homepage. You can add content, manage subscribers, create albums, manage e-mail subscriptions, and perform other group-related tasks from the Group Details block.

For example, you can arrange to receive e-mail notification whenever something gets posted to a group of which you are a member. In the Group Details block (see Figure 4-18), click the *my subscription* link to see and set the status of your e-mail subscription.

# Spam Module

Recent years have seen an explosion in web site spam. *Spam* is any content posted to a web site that is unwanted or has an ulterior motive other than being part of the online community. The most common ulterior motive is getting links to third-party web sites published, in pursuit of the higher search engine rankings that come with the elevated page rank that their web sites enjoy when links from external sites point to them.

Some people are willing to do almost anything to get a link to their web site posted on your web site. They will sign up for accounts, and then post forums and make blog entries with blatant advertisements for their products. They will comment on other people's content, sometimes masking their intent in a thin veneer of compliments before getting to the business of self promotion. They will use scripts to find the holes in your site and flood you with postings.

Spammers show a lot of resourcefulness and absolutely no mercy. If given the chance, they will turn your web site into a wasteland of Viagra, poker, and payday loan advertisements. It is important to the health of your online community that you have a strong defense against this malicious activity. Just as people need to have means to deal with spam e-mail in their inbox, you need to have a way to deal with spam posts on your web site. You will find this defense in the Spam module.

## Detecting Spam

The Spam module can be configured and trained to detect content of any kind that is considered spam, including comments and node types. The administrator has configuration options that allow the Spam module to automatically unpublish that content and/or notify the administrator.

Up to four different mechanisms can be used to identify content as spam: the Bayesian filter, custom filters, URL counting, and the Distributed Server Boycott List.

### Bayesian Filter

The Bayesian filter learns to detect spam by being shown content that has been identified as spam by the site administrator. The best way to describe this method is to quote Jeremy Andrews, the author of the Spam module.

> *The Bayesian filter does statistical analysis on spam content, learning from spam and non-spam that it sees to determine the likelihood that new content is or is not spam. The filter starts out knowing nothing, and has to be trained every time it makes a mistake. This is done by marking spam content on your site as spam when you see it. Each word of the spam content will be remembered and assigned a probability. The more often a word shows up in spam content, the higher the probability that future content with the same word is also spam. As most comment spam contains links back to the spammer's websites (ie. to sell Prozac), the Bayesian filter provides a special option to quickly learn and block content that contains links to known spammer websites.*

For more information about Bayesian filtering, see `http://en.wikipedia.org/wiki/Bayesian_filtering`.

### Custom Filters

As the site administrator, you can define custom filters that increase the probability of certain words and patterns to indicate spam. The filters will cause content to be marked as follows:

- Blacklisted content will be definitely marked as spam.
- Whitelisted content will definitely be marked as not spam.
- Graylisted content is marked as either usually spam or usually not spam, increasing or decreasing the likelihood of it being matched as such by the Bayesian filter.

### URL Counting

Since the most common goal of spam content is to publish links back to the spammer's web site, a logical and effective method for identifying spam revolves around counting URLs in posts. The Spam module can be configured to mark content or comments as spam if they have more than a certain number of URLs in them, or alternatively, if the same URL appears more than a certain number of times.

### Distributed Server Boycott List

The Distributed Server Boycott List is a set of lists of IP addresses of servers that are known to be open relays, open proxies, or to have other vulnerabilities that allow anyone to deliver e-mail to anywhere, through that server. The Spam module's fourth anti-spam mechanism uses these lists to check the IP addresses of users posting content and see if they are known e-mail spammers. Presumably, e-mail spammers are also content/comment spammers.

## Installing the Spam Module

Follow these steps to install the Spam module:

1. Download the latest version of the Spam module from `http://drupal.org/project/spam`.

---

■ **Note** At the time of writing, this module was being rewritten. You are encouraged to follow developments on the Spam 2.0 module at `http://www.kerneltrap.org/jeremy/drupal/spam/`.

---

2. After backing up your data, import the file `spam.mysql` (or `spam.pgsql` if appropriate) into your database.

3. Although the Spam module comes delivered with many subfolders and extra files, the only one you need to get started is `spam.module`. Create a folder called `spam` in your `modules` directory and move `spam.module` into it.

4. Since the Spam module learns from experience and builds lists of words, URLs, and IP addresses associated with spam, the experience of other web site administrators can be very useful in avoiding spam. To this end, there are a number of database dumps of the spam filters from active sites that have dealt with large quantities of spam. These files are found in the `contributed/spam_tokens` folder that comes with the Spam module download. Optionally, you can load these files into your database.

5. Navigate to *administer* ➤ *modules* (`admin/modules`) and enable the module.

## Configuring the Spam Module

To configure the Spam module, you need to set the appropriate Drupal permissions. You also need to visit the module's settings page and set up your filters and other spam prevention options.

## Spam Module Permissions

The permissions structure of the Spam module is designed to let you divide your users into roughly three groups: those who can decide what content is or is not spam, those who are trustworthy and never create spam, and everyone else who cannot be trusted.

The Access Spam Rating and Administer Spam Rating permissions can be given to user roles who will help train the filter and identify spam. When looking at content, they will be able to mark it either as spam or not spam, thus assigning a new score (1 or 99) and biasing the spam filter's future handling of similar content. As this is an essential activity, if you are to have an effective filter, you will want to make sure that you have trustworthy people in your community who have these permissions, so that spam content will be quickly identified and marked as such. However, keep in mind that anyone with these permissions can unpublish anything on your site simply by marking it as spam, so trust is essential.

The Bypass Spam Filter permission can be granted to any user role who will never submit spam. This would certainly include user roles that receive the Access Spam Rating and Administer Spam Rating permissions. When users with the Bypass Spam Filter permission create content, it will not be passed through the Bayesian filter. The content can be marked as spam later, however. The only advantages to using this permission are a small performance gain, since less processing is done upon submitting content, and that no content will be falsely marked as spam for these users (a very small danger to begin with). It has the negative side effect that content from these users won't automatically train the filter. Thus, assigning this permission to too many users isn't a good idea.

## Filters for Content Types

Select *administer* ➤ *settings* ➤ *spam* (`admin/settings/spam`) to see the configuration settings for the Spam module. The group of settings titled Filter gives you the chance to determine which content types will be eligible for spam filtering. As a rule of thumb, you should allow the Spam module to filter any content that can be created anonymously or by users whose trustworthiness cannot be guaranteed (that is, you don't know them personally). The options are as follows:

**Filter comments**: This should be checked in most cases. There are known scripts for attacking Drupal sites where the URLs for nodes are systematically probed and comments are posted. The scripts craftily hit low-numbered nodes (low-numbered nodes are usually older content on most sites) and stop before becoming very conspicuous, in a bid to create comment spam but not be detected. Since these scripts target comments, it is advisable to enable this option.

**Filter open relays**: This corresponds to the Distributed Server Boycott List method of spam filtering, described earlier. It uses the IP address of the user posting content and compares it to published lists of known spammers. Any content published from an IP address found on these lists will have a greatly increased chance of being marked as spam.

**Filter spammer URLs**: This instructs the Spam module to pay greater attention to the URLs in content and afford them special treatment. The URLs in posts marked as spam will be branded as positive identifiers for spam content, and any new content containing the same URLs will be marked as spam.

### URL Limits

The Limits group on the Spam module's settings page deals with the number of URLs that can appear in either comments or normal content. For each, you can set the highest number of allowed URLs and the number of repeated URLs. Be careful when setting these that you don't preclude normal legitimate use of your web site. For example, if your site is supposed to have reviews of a certain type of web site, it is conceivable that someone would want to include several hyperlinks to reference the same page of the site being reviewed. On the other hand, setting these limits to something reasonable (between 3 to 5 in most cases) will catch and block the attempts of some spammers.

### Actions for Identified Spam

What should happen when the spam filter identifies something as spam? That question is answered by the settings in the Actions group on the Spam module's settings page.

You can have the content or comment automatically unpublished by checking the Automatically Unpublish Spam field. This might be the best option if spam or potentially embarrassing content could hurt your business or reputation if allowed to persist on the site for any amount of time. Use this option if you are too busy to constantly police your site (better to be safe than sorry). On the other hand, it is possible (though rare) for the Spam module to identify false positives, and users whose content gets unpublished may get very irritated with your site.

The Notify Admin When Spam Detected field causes the administrator of the site to be notified whenever spam is identified. This should be checked as long as you leave possible spam published (Automatically Unpublish Spam is unchecked), or if you are worried about the chance of false positives. That way, you can closely monitor the Spam module's activities and see that it is doing its job correctly.

### Advanced Spam Module Configuration

For everyone who wants to get the Spam module installed and running, the preceding configuration is all that needs to be done. If you're interested in playing around under the hood of the module and tweaking the Bayesian filter parameters, check the Advanced Configuration check box on the `admin/settings/spam` page and click Submit to see advanced options.

The following are two advanced options that you might find interesting (and that don't introduce any risks):

**Display spam rating**: This setting, under the Tools heading, will let you see the spam rating for any content or comments. This is a rating from 1 to 99, and the higher the rating, the more likely it is that it is spam.

**Collect statistics**: If you instruct the module to collect statistics, you will have access to interesting information, such as how many spam comments or content submissions have been posted and when the last occurrence was.

## Managing URL Filters

Whenever content is identified as spam, the Bayesian filter takes the opportunity to extract all of the URLs contained within and pay special attention to them. The logic is that the spammers' golden eggs are the URLs, and therefore their most telling fingerprint.

Navigate to *administer* ➤ *spam* ➤ *URL filters* (`admin/spam/urls`), and you will see an overview of the URLs that the Spam module recognizes as spam URLs. This page also shows how many times each URL has been matched as spam, how many times it has been identified with content that is not spam, the spam probability that it carries (from 1 to 99), and the date of the last match.

This page can be useful for removing false positives from the filter in case a spammer does something nasty, like including a link to a legitimate site in her spam, thus tricking the spam filter into thinking that URL, too, is spam.

You can also use this page to take preventative measures and block URLs that you know are spammer sites from other sources. Your e-mail inbox could be a good source of these URLs. Every one of those free software and cheap drugs sites that are advertised in e-mail spam deserves to be fed to your site's spam filter. To do this, simply add the domain to the Add New URL Filter section and click Add New URL Filter. If the same witless spammer decides to expand and target your web site as well as your inbox, he will be out of luck.

## Creating Custom Filters

Next to the URL Filters tab is a tab named Custom Filters (`admin/spam/custom`). Custom filters are a tool for you to help teach the filter by feeding it words, phrases, or regular expressions and telling it what to do with them when they are matched.

You create a new custom filter by entering a word, phrase, or Perl-compatible regular expression into the Custom Filter field. If it is a regular expression, you also need to check the Regular Expression check box, and your expression will be checked for validity when you submit the form. Then you need to determine what is to be done with the content that is found to match this filter by making a selection in the Match Effect field. The options are as follows:

- Always spam (blacklist)
- Usually spam (graylist)
- Usually not spam (graylist)
- Never spam (whitelist)

These various classifications affect the probability rating that the content is spam. Remember that the whole point of the Bayesian filter is to establish this rating, so your custom filter is simply taking part in all of the other calculations that will occur when any post or comment is being considered. Here are the technical details, taken from the help text for custom filters:

> *If your filter defines "always spam," this increases the chances the new content will be marked spam by 200%. If your filter defines "usually spam," this increases the chances the new content will be marked spam by 50%. If your filter defines "usually not spam," this decreases the chances the new content will be marked spam by 50%. And if your filter defines "never spam," this decreases the chances the new content will be marked spam by 200%.*

Since the same content can match several filters, they can play complementary or opposing roles. If a comment matches one Always Spam filter and five Usually Not Spam filters, the weight will be in favor of usually not spam, and the chances that the comment will be marked as spam will be reduced by 50%.

Whitelisting filters should be used sparingly and only in the context of training the filter not to repeat false positive identifications that it has made. The reason for this should be clear: if you open up the door for content to automatically be considered never spam, you can't be too surprised when spammers start walking through it.

If you are being bombarded with spam from a single source and you can block this source with one filter, and you are positive that the filter will not have any false positives, you might elect to check the Automatically Delete Spam check box. When checked, content matching this filter will be deleted from your site without any warning or notification. This is intended as an extreme defense when your site is suffering from the efforts of a concerted attack. Due to the extreme nature of this option, it is recommended only in last-resort cases. In most normal cases, you will want to have the option of browsing spam content to double-check that no false positives are being made.

## Using Other Filters

In the `contributed/custom_filters` folder in the Spam module archive, you will find some SQL data dumps of custom filters that others have contributed based on experience from their sites. As noted earlier, you can choose to import these data dumps into your database and benefit from them as well. Here are some examples from the file `teledyn.mysql`:

```
/\b(mortgage|credit|kredit|casino|poker|debt)\S*\.\w{2,3}\b/i
/\b(texas-holdem|onlinegames|blackjack|bingo)\S*\.\w{2,3}\b/i
```

These expressions look for domains that contain any of the list of words *mortgage*, *credit*, *casino*, *bingo*, and so on. The domain `bingo123-use.la` would be matched, for example.

If you decide to use third-party filters, make sure to monitor the content that is being marked as spam to be on the lookout for false positives.

# Database Administration Module

The Database Administration (dba) module allows you to view, edit, repair, and create backups for your Drupal database, all from within the administrator's web interface. It is a practical and convenient way to run SQL queries and view the results. It can also be used to import the database definitions for new modules, possibly saving you a trip to another application or command-line tool.

## Installing the Database Administration Module

To install the dba module, download the latest version from `http://drupal.org/project/dba`. Then create a `modules/dba` folder and copy the `dba.module` file into it. If you are using PostgreSQL as your database, you need to take the extra steps found in the `README.pgsql` file. Finally, activate the module from the `admin/modules` page.

## Configuring the Database Administration Module

The most important configuration decisions you need to make concerning the dba module have to do with user permissions. You can also set options for automatic backups and checking the database integrity.

### Database Administration Module Permissions

To quote from the module's documentation:

> *If a user is granted (or manages to acquire) 'dba administer database' permissions, they are able to directly alter the database. At minimum, they are able to modify data, and possibly to drop tables. Depending on how you have defined your database permissions, the user may also be able to modify other databases unrelated to your Drupal installation. Use at your own risk!*

Clearly, the Dba Administer Database permission is not to be granted lightly.

The second permission that this module defines is Dba View Database, which allows users possessing it the chance to see the contents of the entire database. This includes user e-mail, the encrypted hash of user passwords, any private profile information, and so forth. Judicious use of this module and sparing grants of these two permissions are advised.

### Automatic Database Backups

To configure the dba module, navigate to *administer* ➤ *settings* ➤ *dba* (`admin/settings/dba`). One of the useful features of this module is the ability to make backup SQL dumps of your database, and it can mail these backups to your e-mail address. Dumps can be made from individual tables, in which case, the backup filename will bear the name of the table, or of several tables or even the entire database. The following are settings that pertain to database backups:

**Default backup filename**: In the case of multiple tables, the name of the backup file is initially taken from this field, although this can be changed at the time you initiate the backup.

**Automatically backup database every *(period of time)***: The automatic backups are disabled by default, but if you set this field to any of the given time periods, backups will be made as often as you have specified.

**Automatic backup path**: This is the path to the directory on your server where the backups will be saved. Backups are not automatically deleted, so this is a cleanup task that will be left to you.

**Compress automatic backups**: If compression libraries are available to PHP on your server, you can check this option, and backups will arrived in a compressed format, greatly reducing the size of the backup files.

**Mail backup to administrator**: Checking this option will result in the automatic database backups being mailed to the administrator's e-mail account (as defined on the site settings page, `admin/settings`).

### Database Integrity Checks

The dba module, when used with the MySQL database, allows you to check the data integrity of your database. This is useful for routine maintenance as well as diagnosing problems. The Default Check Type field lets you tell MySQL which type of check to make. Here are the various options on the `admin/settings/dba` page, as described by the MySQL documentation (`http://dev.mysql.com/doc/mysql/en/check-table.html`):

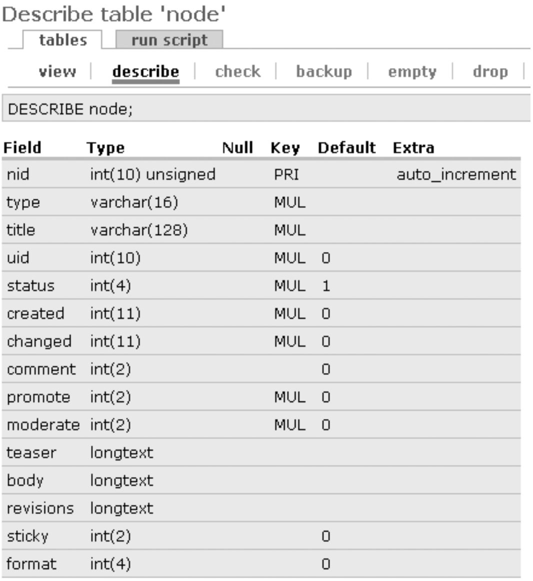
- QUICK means don't scan the rows to check for incorrect links.
- FAST means to check only tables that haven't been closed properly.
- CHANGED means to check only tables that have been changed since the last check or haven't been closed properly.
- MEDIUM means to scan rows to verify that deleted links are okay. It also calculates a key checksum for the rows and verifies this with a calculated checksum for the keys.
- EXTENDED means to do a full key lookup for all keys for each row. This ensures that the table is 100% consistent, but takes a long time!

In the unlikely event that the data integrity of your database becomes compromised, there is a chance that it can be automatically repaired. The Repair Option field lets you determine when the repair link for a table is visible. The best setting is Automatic, as this not only hides superfluous links, but also serves to alert you (by the presence of the link) when the data integrity has been broken.

## Using the Database Administration Module

To see an overview of the tables in your database, select *administer* ➤ *database*. For each table, you are shown the number of rows and have the following options:

- The View option shows the contents of the table.
- The Describe option shows the data structure of the table, as shown in Figure 4-19.



Describe table 'node'

```
DESCRIBE node;
```

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| nid | int(10) unsigned | | PRI | | auto_increment |
| type | varchar(16) | | MUL | | |
| title | varchar(128) | | MUL | | |
| uid | int(10) | | MUL | 0 | |
| status | int(4) | | MUL | 1 | |
| created | int(11) | | MUL | 0 | |
| changed | int(11) | | MUL | 0 | |
| comment | int(2) | | | 0 | |
| promote | int(2) | | MUL | 0 | |
| moderate | int(2) | | MUL | 0 | |
| teaser | longtext | | | | |
| body | longtext | | | | |
| revisions | longtext | | | | |
| sticky | int(2) | | | 0 | |
| format | int(4) | | | 0 | |

**Figure 4-19.** *The node table described*

- The Check option checks the table (and its indexes) for errors and integrity.
- The Repair option attempts to fix errors found with check. See the MySQL documentation (`http://dev.mysql.com/doc/mysql/en/repair-table.html`) for details.
- The Backup option saves one or more tables in the form of a SQL dump to a file that you specify.
- The Empty option deletes all data from the table.
- The Drop option removes the table from the database.

### Running Queries and Scripts

The dba module facilitates querying the database through SQL that you enter directly into the web browser or by using a file stored on your local machine. Using SQL that you type directly into the browser is a convenient way to run custom queries if you are familiar and comfortable with using SQL. To do this, select *administer* ➤ *database* and click the Query tab (`admin/database/query`).

You can also run scripts from files, which is an excellent tool for adding database definitions for modules that you are installing from files. To do this, select *administer* ➤ *database* and click the Script tab (`admin/database/script`).

## Developer Tools (Devel) Module

The Devel module is an indispensable tool for programmers and system administrators who want to know how Drupal accesses the database. It collects all of the SQL queries that are made during the loading of a page and shows them as output at the bottom of the screen, along with the time they took and whether the same query was called more than once. This is an excellent way to start to understand what goes on behind the scenes in Drupal, as well as a means for spotting bottlenecks in performance. The Devel module also defines a couple of helper functions to be used for debugging while programming, integrates with Xdebug (`http://www.xdebug.org/`) to provide profiler information, and includes a number of scripts for generating dummy database data for the purposes of testing.

### Installing the Devel Module

Download the latest version of the Devel module from `http://drupal.org/project/devel`. Create a `modules/devel` folder and move the `devel.module` file into it. Enable the module from the `admin/modules` page. The archive download comes with a `devel/generated` folder, which contains various scripts for automatically generating content in the database for the purpose of testing. Move these files to the server only if the site is a development site and you intend to use the scripts. Otherwise, you are best advised to omit them.

If you are using Xdebug and wish to have access to profiler information through the Devel module, you should add information to either your `php.ini` file or the `.htaccess` file. For `php.ini`, add the following:

```
xdebug.auto_profile=1
xdebug.auto_profile_mode=3
xdebug.output_dir='/php'
xdebug.default_enable=1
```

For .htaccess, add this:

```
php_value xdebug.auto_profile=1
php_value xdebug.auto_profile_mode=3
php_value xdebug.output_dir='/php'
php_value xdebug.default_enable=1
```

## Configuring the Devel Module

To configure the Devel module, set its one permission and adjust its settings for the page timer and query log, if desired.

### Devel Module Permissions

The Devel module defines one user permission, Access Devel Information, which should be enabled for any roles that are to be involved in development or optimization of the site. Only users with this permission will see the page timer and query log when they are turned on, as described next.

### Page Timer and Query Log

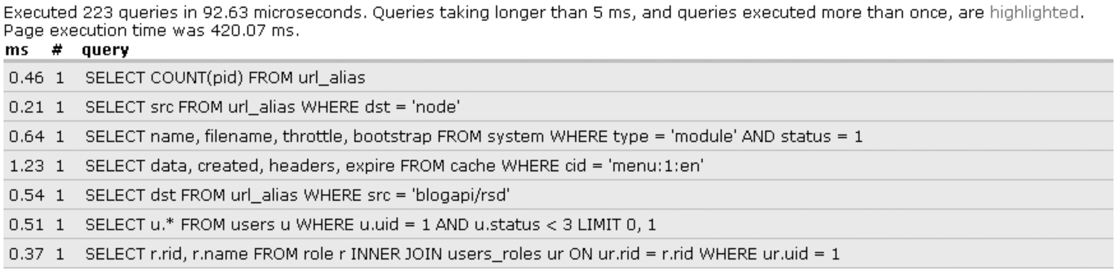
To configure the Devel module, navigate to *administer* ➤ *settings* ➤ *devel* (admin/settings/devel). The settings page has options for enabling the page timer and the query log. The information they generate will be visible to users with the Access Devel Information permission.

The Query Execution Threshold field is the threshold in milliseconds for queries to be considered slow. Queries that exceed this threshold or execute more than once will be highlighted in the query log that appears at the bottom of each screen (when this log is turned on). You can adjust this threshold to give you the most meaningful data.

In many cases, Drupal will perform a set of queries and actions, only to redirect you to a screen or path other than the one you requested. A classic example of this is the 404 Not Found screen. For example, you request the page for node/4711, but there is no node with that ID, so Drupal tells you it can't find it. The problem this presents for debugging is that the SQL queries for the first part of the operation—the part where Drupal looks up 4711 and doesn't find anything—are not included in the query log because of the redirection that occurs thereafter. The Display Redirection Page field takes care of this problem by alerting you when a redirect is about to occur and waiting for your input before the redirect is carried out. This gives you the chance to inspect the queries that occur prior to the redirection as well as after.

## Viewing Timer and Query Log Information

Using the page timer and query log is a simple matter of browsing your site with them enabled and enjoying the bounty of information that they produce. Figure 4-20 shows an excerpt from a timer and query log.



Executed 223 queries in 92.63 microseconds. Queries taking longer than 5 ms, and queries executed more than once, are highlighted.
Page execution time was 420.07 ms.

| ms | # | query |
|---|---|---|
| 0.46 | 1 | SELECT COUNT(pid) FROM url_alias |
| 0.21 | 1 | SELECT src FROM url_alias WHERE dst = 'node' |
| 0.64 | 1 | SELECT name, filename, throttle, bootstrap FROM system WHERE type = 'module' AND status = 1 |
| 1.23 | 1 | SELECT data, created, headers, expire FROM cache WHERE cid = 'menu:1:en' |
| 0.54 | 1 | SELECT dst FROM url_alias WHERE src = 'blogapi/rsd' |
| 0.51 | 1 | SELECT u.* FROM users u WHERE u.uid = 1 AND u.status < 3 LIMIT 0, 1 |
| 0.37 | 1 | SELECT r.rid, r.name FROM role r INNER JOIN users_roles ur ON ur.rid = r.rid WHERE ur.uid = 1 |

**Figure 4-20.** *The page timer and query log*

## Using Developer Functions

If you are writing code for Drupal, the following four functions can help you while you work:

- `dprint($str)`: Works like the PHP function `print( string arg )` function (`http://php.net/print`) but inside `<pre>` tags.
- `dprint_r($arr)`: Works like the PHP function `bool print_r ( mixed expression [, bool return] )` (`http://php.net/print_r`) but inside `<pre>` tags.
- `ddebug_backtrace()`: Works like the PHP function array `debug_backtrace ( void )` (`http://php.net/debug_backtrace`) but inside `<pre>` tags.
- `devel_variable()`: Outputs the global Drupal configuration variables array `$config`.

You can add these functions to the source code files or to text areas that are evaluated as PHP with the PHP filter as tools for debugging.

## Emptying the Cache

Something that every new Drupal developer learns is that the cache table in the database can be emptied without ill effect to the integrity of the database or its data. In fact, when you're programming certain elements, such as the `hook_menu` function for a module, it is necessary to empty the cache from time to time. When developing Drupal, especially when making changes to the `hook_menu` function of a module or when dealing with filtered text, you usually need to clear the cache before your changes will become visible. Menus, variables, and other key elements are cached to improve the performance of the page-building process.

The Devel module offers a convenient link for developers so that they don't need to keep running database queries manually to clear the cache. With the Devel module enabled, an *Empty cache* link appears in the main navigation menu (`devel/cache/clear`).

■ **Note** There is almost no reason to clear the cache if you are not developing Drupal modules. But if you accidentally clear the cache, don't worry—you haven't hurt anything.

# Summary

The repository of contributed Drupal modules is vast and diverse in scope. Installing and configuring Drupal modules tends to follow a common pattern that includes moving files into the `modules` folder on the web server, possibly updating the database schema, and enabling the module from the `admin/modules` screen. Modules usually have a settings page, which can be accessed via *administer* ➤ *settings*.

In this chapter, you learned how to install, configure, and use some useful modules: TinyMCE, Image, Image Assist, Flexinode, Event, Location, Organic Groups, Spam, Database Administration, and Devel.

In this and the previous chapters, you've learned how Drupal sites work and what you can do with them. In the next chapter, you will dive into the world of theming your Drupal site so that you can control how every element looks and make your site a visual masterpiece.

Based on an abstract theme layer, Drupal's theme system allows you to access and control the HTML that is created without needing to modify the source code for modules or core files. This is not only good software architecture, but it also opens the door for sophisticated usage such as running multiple sites (each with their own theme) off a common codebase. Running multiple Drupal sites is covered in Chapter 6.