

Choosing and Configuring Modules to Build Dynamic Websites



Using

Drupal

O'REILLY®

*Angela Byron, Addison Berry,
Nathan Haug, Jeff Eaton,
James Walker & Jeff Robbins*

This excerpt is protected by copyright law. It is your responsibility to obtain permissions necessary for any proposed use of this material. Please direct your inquiries to permissions@oreilly.com.

Table of Contents

Foreword	xiii
Preface	xv
1. Drupal Overview	1
What Is Drupal?	1
Who Uses It?	2
What Features Does Drupal Offer?	3
A Brief History of Content Management	4
A Historical Look at Website Creation	4
The Age of Scripts and Databases	6
The Content Revolution	7
How Does Drupal Work?	8
Modules	10
Users	10
Content (Nodes)	11
Ways of Organizing Content	13
Types of Supporting Content	14
Getting Help	15
Conclusion	16
2. Drupal Jumpstart	17
Case Study	18
Implementation Notes	19
Spotlight: Content Management	20
Content	20
Comments	23
Navigation	24
Blocks	26
Hands On: Content Management	29
Creating Content	29
Managing Site Navigation	33

Configuring Blocks	34
Spotlight: Modules	35
Module Administration Page	36
Finding and Installing Modules	38
Removing Modules	40
Hands On: Working with Modules	41
Path Module	41
Administration Menu Module	42
Spotlight: Access Control	46
Configuring User Access	47
User Profiles	49
User Settings	49
Handling Abusive Users	50
Hands On: Creating Roles	51
Hands On: Configuring Permissions	52
Hands On: Contact Form	58
Spotlight: Taxonomy	60
Hands On: Blog	62
Spotlight: Content Moderation Tools	67
Automated Spam Detection Tools	68
Manual Content Moderation Tools	70
Spotlight: Themes	71
Finding a Theme	71
Theme Installation	71
Theme Configuration	71
Blocks and Regions	75
Administration Theme Setting	76
Hands On: Branding the Site	76
Spotlight: Content Editing and Image Handling	78
Content Editing	78
Image Handling	79
Spotlight: Input Formats and Filters	81
Hands On: Setting Up FCKeditor	84
Summary	91
3. Job Posting Board	93
Case Study	94
Implementation Notes	94
Spotlight: CCK	96
Fields and Widgets	97
Widgets	98
Formatters	99
Hands On: CCK	100

Hands On: Job Content Type	101
Hands On: Customizing Field Display	108
Hands On: Job Application Type	111
Spotlight: Views Module	115
Displays	117
Pieces of a View	120
Hands On: The Views Module	123
Jobs View	124
Job Applications View	137
Taking It Further	150
Summary	151
4. Product Reviews	153
Case Study	153
Implementation Notes	155
First Steps: Basic Product Reviews	157
Creating the Product Review Content Type	157
Spotlight: Amazon Module	160
What's Included?	161
Locale	161
Referral Settings	161
Hands On: Adding an Amazon Field	162
Adding the Product Field	163
Finding Product IDs	165
Spotlight: Voting API and Fivestar	165
Hands On: Adding Ratings	167
Adding the Rating Field	167
Turning on Visitor Ratings	168
Hands On: Building a Product List	170
Spotlight: The Search Module	174
The Importance of Cron	175
Searching with Views	177
Hands On: Make the Product List Searchable	177
Spotlight: CSS Injector	179
Hands On: Polishing the Presentation	180
Setting CCK Display Fields Options	180
Configuring CSS Injector	181
Taking It Further	182
Summary	183
5. Wiki	185
Case Study	186
Implementation Notes	187

Wiki Functionality	187
Easy Text Formatting	187
Easy Linking	188
Tracking Changes	188
Human-Readable URLs	189
Listing Changes	189
Hands On: First Steps	189
Creating a Wiki Content Type	189
Removing the Author Information Display	191
Configuring Access Permissions	191
Hands On: Wiki Input Format	191
Configuring the Filters	192
Creating the Wiki Input Format	193
Setting up Format Permissions	195
Adding Content	195
Spotlight: Pathauto	197
Hands On: Pathauto	201
Configuring Settings	201
Spotlight: Drupal's Revision Tracking	203
Hands On: Revisions and Diff	204
Make Revisions the Default	204
Setting Permissions	204
Viewing Revisions and Reverting	205
Using Diff	206
Hands On: New Pages and Recent Edits with Views	207
Recent Posts Listing	208
Recent Edits Listing	208
Taking It Further	211
Summary	212
6. Managing Publishing Workflow	215
Case Study	215
Implementation Notes	217
Hands On: First Steps	218
Spotlight: Actions and Triggers	219
Hands On: Actions and Triggers	220
Configure Actions	220
Assign Triggers	223
Spotlight: Workflow Module	224
Hands On: Creating a Workflow	226
Spotlight: The Workspace Module	230
Hands On: Create Workspaces	231
Spotlight: Views Bulk Operations	233

Hands On: Building an Administration Page	235
Create Workflow Actions	235
Configure the View	236
Taking It Further	236
Summary	238
7. Photo Gallery	241
Case Study	241
Implementation Notes	242
Photo Uploads	242
Thumbnail Generation	243
Photo Galleries	243
Spotlight: ImageField	243
Configuration	243
Hands On: Uploading Photos	246
Photo Content Type	246
Image Galleries	249
Spotlight: ImageCache	251
Presets and Actions	252
Using a Preset	256
Troubleshooting ImageCache	259
Hands On: ImageCache	261
Create ImageCache Presets	261
Configure Photo Field Display	263
Improve Image Quality	264
Hands On: Gallery View	265
Hands On: Latest Photos Block	270
Hands On: Custom Pagers	273
Taking It Further	276
Summary	277
8. Multilingual Sites	279
Case Study	280
Implementation Notes	281
Forum Discussions	281
Knowledge Base	281
Translating User Interface Text	281
Translating User-Generated Content	282
Spotlight: Core Internationalization Features	282
Locale	283
Content Translation	286
Hands On: Installing a Translation	287
Hands On: Configuring Locale Features	292

Language Negotiation Settings	292
Language switcher	293
Hands On: Translatable Content	294
Spotlight: Localization Client	296
Hands On: Translating the Interface	296
Using the Locale Module	296
Using the Localization Client	298
Hands On: Translating Content	301
Translation	301
Spotlight: Internationalization	304
Content Selection	306
Strings	308
Site-Wide Language-Dependent Variables	308
Module Helpers	308
Synchronization	309
Hands On: Internationalization Features	309
Content Selection	309
Site-Wide Variables	311
Content Types	313
Taxonomy	314
Taking It Further	317
Summary	319
9. Event Management	321
Case Study	321
Implementation Notes	322
Event Management	322
Attendance Tracking	323
Hands On: First Steps	324
Creating an Event Content Type	324
Access Control	325
Spotlight: Date Module	325
Date API Module	326
Date Timezone	326
Date Field Types	326
Date Widgets	327
Date Settings	328
Hands On: Adding Dates	333
Set Up the Date Module	333
Add the Date Field	333
Hands On: Upcoming Events View	335
Spotlight: Calendar Module	338
Calendar View Type	338

Hands On: Calendar View	339
Spotlight: Flag module	341
Hands On: Flag Configuration	342
Hands On: Attendee View	344
Taking It Further	345
Summary	347
10. Online Store	349
Case Study	349
Implementation Notes	350
Spotlight: Ubercart Packages	351
Ubercart – core	351
Ubercart – core (optional)	353
Ubercart – extra	357
Spotlight: Ubercart’s Administration Panel	359
Hands On: Setting Up the Store	360
Initial Setup Tasks	361
Spotlight: Products, Product Classes, and Attributes	362
Hands On: Creating Products	365
Initial Setup Tasks	366
Configuring Product Classes	366
Configuring Product Attributes	367
Configuring Product Settings	370
Configuring the Catalog	371
Spotlight: The Ordering Process	375
Hands On: Processing Orders	376
Shopping Cart	377
Taxes	379
Shipping	379
Payment	380
Placing a Test Order	381
Fulfilling an Order	383
Access Control	385
Taking It Further	387
Summary	388
11. Theming Your Site	389
Spotlight: The Theme System	390
.info Files	392
Regions	393
Features	394
CSS	395
JavaScript	395

Template Files	396
The template.php File	398
Hands On: Creating a Custom Theme	401
Make a Copy of the Theme	402
Changing CSS	403
Spotlight: Template Files	405
Hands On: Working with Template Files	406
Modifying a Template File	406
Theming Specific Content Types	410
Overriding a Module's Template File	412
Creating a New Region	413
Spotlight: Advanced Overrides	415
Template Variables	415
Theme Functions	416
Hands On: Using template.php for Overrides	418
Overriding a Template Variable	418
Overriding a Theme Function	419
Taking It Further	421
Summary	422
References	423
A. Installing and Upgrading Drupal	425
B. Choosing the Right Modules	445
C. Modules and Themes Used in This Book	461
Index	465

Event Management

Managing online calendars and event registration can present a huge challenge. Without a dynamic system, the task is nearly impossible. Generating the HTML required to display a calendar and all the various presentation options (day, week, month views, and so on) is unreasonable; and worse, because the events are time-sensitive, remembering to update “next” or “upcoming” event lists can be onerous. Nothing looks worse than having last week’s meeting listed first on your “Upcoming Events” page.

Even with dynamic systems, you tend to be constrained to certain parameters with fixed options. However, by taking advantage of the flexibility of Drupal and building on the powerful base of CCK and Views, you can accommodate nearly any variation on event listings for your site.

To follow along with the hands-on example in this chapter, you should install Drupal using the Events install profile. The completed website will look as pictured in Figure 9-1 and at <http://events.usingdrupal.com>. For more information on using the book’s sample code, see the Preface.

This chapter introduces the following modules:

- Date (<http://drupal.org/project/date>): Provides a CCK field for entering date information, as well as libraries to handle things like time zone conversion.
- Calendar (<http://drupal.org/project/calendar>): A view style for displaying a list of site content in a rich calendar display.
- Flag (<http://drupal.org/project/flag>): A flexible module that enables administrators to add on/off toggle switches to items such as nodes and comments.

Case Study

The Aurora Book Club is a rather social group of local book enthusiasts. They hold semiregular monthly meetings and events for both current and prospective members. Members want to be able to see when and where the next meeting is happening. Additionally, members should be allowed to post their own events to the site. Events

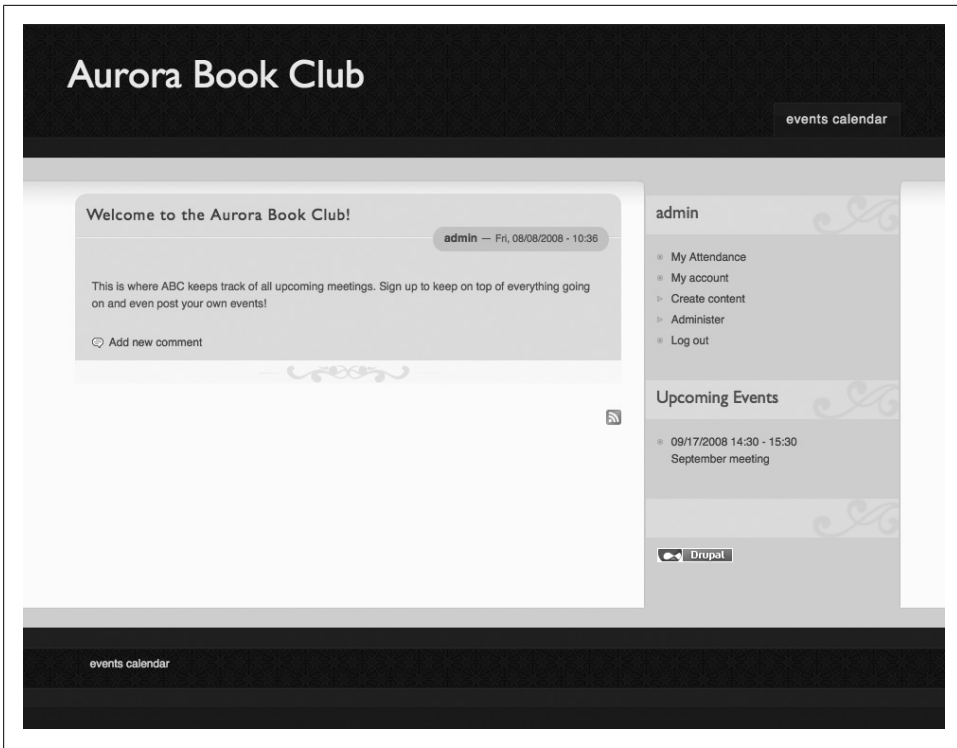


Figure 9-1. The completed Aurora Book Club site

should have start and end times and dates, as well as information about the event and where it will take place. To make it easy to see what is happening soon, there should be a short list of upcoming events in addition to the full calendar. The calendar needs to offer day, month and annual views, and a way for members to subscribe to the club's calendar using Microsoft Outlook or Apple's iCal. Finally, since they would like to know how many cookies to bring and how many chairs to have on hand, the club president has asked that we include a way to track who plans to attend each event.

Implementation Notes

Event Management

The book club has two main options for managing event data in Drupal. The options are indicative of a common trend amongst Drupal's contributed modules. Specifically, there is a longstanding module in the Drupal community simply called Event (<http://drupal.org/project/event>), which natively handles most of the required features. The other option, and the one we will be using in this chapter, is a combination of CCK and Views add-ons: the Date (<http://drupal.org/project/date>) module and the Calendar

(<http://drupal.org/project/calendar>) module, respectively. The main differences are as follows:

- The Event module allows per-content-type “event enabling,” which means that it will add a start and end time to any existing content type. Included with the module is also an entire system for viewing your created events and even a block for listing upcoming events.
- The Date module offers a CCK field type for handling dates. The Calendar module is a Views plug-in that renders a view in a browsable calendar layout.

Both options are capable of meeting all the requirements for the book club’s site. In fact, when compared side by side, they look functionally equivalent: both options allow us to create new events, view a list of upcoming events, and offer rich, full-featured calendar displays. In fact, by working with any content type, Event module still allows us to use CCK to build custom content types for our event data (for example, for holding additional information about events). This significantly blurs the line between the solutions, particularly compared to choice between the Image module and Imagefield, as discussed in Chapter 7. With the Image module, a specific content type (“image”) is defined, whereas with Event you can reuse any custom content types. Furthermore, Event is a single module without any dependencies, whereas Date and Calendar require CCK and Views respectively. So why not use the Event module?

As we’ve discussed in several earlier chapters, the CCK and Views modules represent the future of site building with Drupal. Their added flexibility and granularity present far more flexible options for tailoring the modules to fit our site’s exact needs. By being more narrowly focused, Date and Calendar also implement their specific features more completely. This is particularly true of the Date module, which has several more options for date format support than Event. Also, by going with a CCK and Views native solution, improvements and efficiencies will automatically trickle down as CCK and Views continue to evolve new features.

Essentially, the reason to use Event module would be either for legacy purposes or ease of installation. For more information about event related modules, see the Event category on drupal.org (<http://drupal.org/project/Modules/category/61>).

Attendance Tracking

The other remaining feature, tracking user signups, will be accomplished by using the Flag (<http://drupal.org/project/flag>) module. In previous versions of Drupal, the Signup module (<http://drupal.org/projects/signup>) was typically used to handle the task of managing signups, but it was not yet available for Drupal 6 at the time of writing. Instead, we’ll use the generic building block Flag module instead. Flag (known as “Views Bookmark” in Drupal 5) provides a general-purpose way to have users mark or “flag” a piece of content. For our purposes, this flag will be “attending” versus “not attending.” We

can then make use of the excellent integration between the Flag module and the Views module to build a listing of which members are attending each event.

Hands On: First Steps

First, we'll set up a few basics for our site just using Drupal core. The main thing that we need to start off is a content type to handle our events. Log in to the Aurora Book Club site with the username *admin*, password *oreilly*, if you are using the installation profile.

Creating an Event Content Type

We'll start by creating a new, basic content type just for events. We just need the event name and description along with an easy way to add the event location:

1. Go to Administer→Site Building→Modules (admin/build/modules) and enable the following modules:
 - CCK package
 - Content
 - Text
2. Go to Administer→Content Management→Content types (admin/content/types) and select the “Add content type” tab (admin/content/types/add) to create a new content type called Event, using the settings from Table 9-1.

Table 9-1. Settings for the Event content type

Field	Value
Identification	
Name	Event
Type	event
Description	A book club meeting or social event.
Submission form settings	
Title field label	Name
Body field label	Description
Workflow settings	
Default options	Uncheck “Promoted to the front page”
Comment settings	
Default comment settings	Disabled

3. Click the “Save content type” button.
4. Add a text field for storing the location of the event (that is, where the event takes place. You should have been returned to the Content types administrative page

(admin/content/types) after saving the new type. Click the “manage fields” link for the Event content type (admin/content/node-type/event/fields). Complete the New field form using the values from Table 9-2.

Table 9-2. Settings for adding a location field to the Event content type

Field	Value
Label	Location
Field name	location
Select a field type	Text
Select a widget	Text field

5. Click Save. This will take us to the configuration settings page for the Location field. We will just use the default settings here, so click the “Save field settings” button to finish.

Access Control

Now that we’ve got the content type created and configured properly, we need to grant permissions to our members to allow them to create events:

1. Go to Administer→User Management→Permissions (admin/user/permissions) and set the permissions as shown in Table 9-3. Click the “Save permissions” button when you are done.

Table 9-3. Permissions for the event content type

Permission	anonymous user	authenticated user	editor	site administrator
node module				
create event content		Checked		
delete any event content				Checked
edit any event content			Checked	Checked
edit own event content		Checked		

Spotlight: Date Module

The main building block for the site is our new Event content type. The information that we need it to provide us with is “where” and “when.” We have taken care of the “where” part in our initial setup. The Date module helps us effectively answer the “when” question, in an incredibly flexible manner.

As mentioned previously, our real interest in the Date module is to add a CCK field to our Event content type to indicate date and time. However, looking at the Date module more closely, there are a few extra pieces worth noting.

Date API Module

As its name would imply, the Date API module merely provides a set of library functions for date handling. These functions consist of things like converting between date formats, handling time zones, and even generating date input select boxes. Though covering the full extent of the API is outside the scope of this chapter, it is worth noting that Date API module does not depend on CCK, but is required by both the Date and Calendar modules. In theory, any module in Drupal that perform any sort of date handling or manipulation could take advantage of this module.



For users running PHP4, the Date PHP4 module is required to emulate date manipulation functions introduced in PHP 5.2. This module can optionally make use of an external library for parsing additional dates: the ADOdb date library (http://phplens.com/phpeverywhere/adodb_date_library). Without this library, Date PHP4 primarily handles dates from 1970 to 2038. The library adds support for any date from the year 100 to 3000. For the purpose of the book club, we will be able to just use Date API on its own.

Date Timezone

Drupal core, by default, does all time zone handling based on an offset from Greenwich Mean Time (GMT), represented as the number of hours' difference (plus or minus) from GMT. For example, the offset -0400 is used for Eastern Daylight Time in the United States. This approach has several drawbacks, the most significant of which is in trying to account for Daylight Saving Time, as the offset will change. When the local time changes to Eastern Standard Time, the actual offset from GMT changes from -0400 to -0500.

The Date API comes with a module, Date Timezone, that alters the core behavior to use named time zones rather than hour offsets (for example, America/New York versus -0400). Therefore, Date API can account for Daylight Saving Time accurately in its calculations.

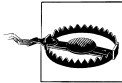
Date Field Types

At its most basic level, Date module defines three CCK field types: Date, Datestamp, and Datetime for adding date fields to content types. The differences among these fields are summarized in Table 9-4.

Table 9-4. CCK fields offered by the Date module

Name	Description	Example	Database Storage
Date	Store a date in the database as an ISO8601 date, used for historical (pre-1000 A.D.) or partial dates (for example, only a year and no day or	2008-08-26T17:02:00	var-char(20)

Name	Description	Example	Database Storage
	month). This field type should be avoided otherwise, as it's extremely expensive to sort and perform conversions on this style of date.		
Dates-tamp	Datestamp field types are stored using the common Unix timestamp format containing the number of seconds since January 1, 1970. As such, these have a limited date range available (1901 A.D.–2038 A.D. on most systems) but are quick to calculate time zone offsets and sort in listings. A legacy format, Datestamps are supported across all database systems.	1219770120	int(11)
Date-time	Datetime field types are stored using the database system's internal "datetime" format for date handling. It has the advantage of being able to use database-specific functions for date handling, including ease of extracting a single part of the date, but with the caveat of inconsistent support across database systems.	2008-08-26 17:12:00	datetime



For much, much more than you ever wanted to know about the pros and cons of various date storage formats, there's an interesting discussion on the Events working group at <http://groups.drupal.org/node/731>.

Because the Aurora Book Club has no intention of moving from MySQL, and all dates will be well within "normal" ranges, we will be using the Datetime field type for our site.

Date Widgets

In addition to the base field types, Date module also defines three CCK widgets for entering Date information, which are pictured in Figure 9-2:

Select list

Presents a series of drop-down lists for each of year, month, day, hour, minute, and second, based on the configured granularity for the date field defined by the data settings.

Text field with custom input format

Provides a simple text field for date entry that will then be converted to the appropriate storage format. The advantage of this widget is that it lets advanced users enter dates much faster. However, for the uninitiated, it can be frustrating if your natural date entry format is not properly recognized.

Text field with jQuery pop-up calendar

Adds an elegant, user-friendly option for Date value entry. This widget uses JavaScript to present a calendar pop-up when a user clicks in the date text field. The user can then click the date on the calendar to select the date that they want.

The book club will be using the "Text field with jQuery pop-up calendar" widget for improved usability.

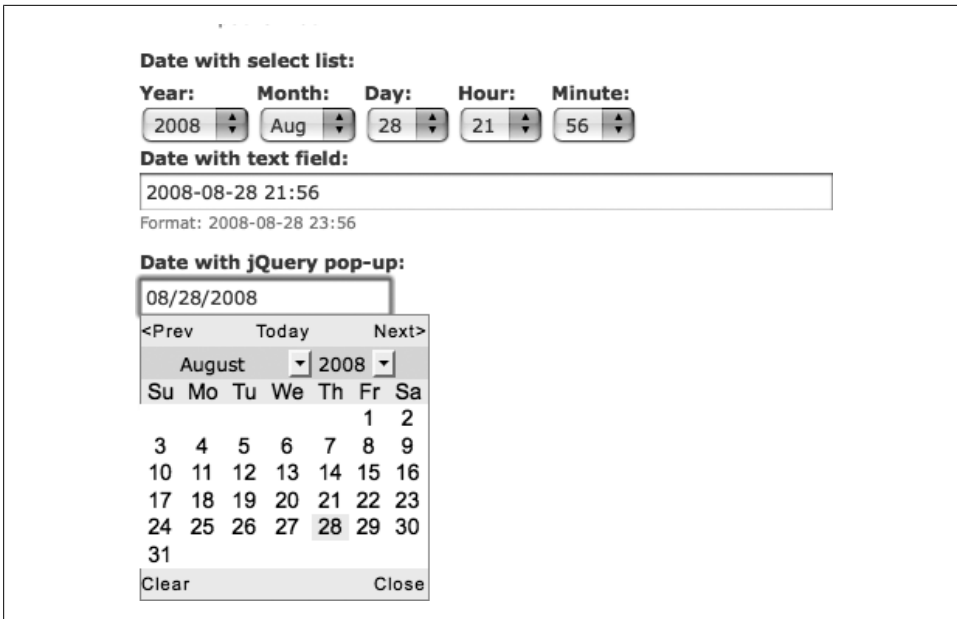


Figure 9-2. Date CCK field widgets

Date Settings

There are quite a few settings available specific to date fields and different from other CCK field types, as shown in Figure 9-3 and Figure 9-4. Some of these are:

Default value

The default value setting gives you a few options to choose from. The Blank and Now settings are pretty straightforward. The Relative setting will let you set the default to a date that is relative to the current time, such as two days from now. To set up a relative default, you must enter a value in Customize Default Value that uses PHP's `strtotime()` syntax, such as `+2 days`. You can find out more about `strtotime()` at <http://www.php.net/manual/en/function strtotime.php>.

Input format

The input format setting dictates the ordering of the fields—year, month, day, hour, minute, and seconds—as they appear on the form for entering the date. Only the values available in the Granularity data setting will be taken into consideration. There is also a “Custom input format” option, which allows an arbitrary date format to be used when using the “Select list” and “Text field with custom input format” widgets for ultimate flexibility. The custom format is set using PHP's `date()` formatting syntax (<http://php.net/date>). The “Text field with jQuery pop-up calendar” does not offer a custom input format, because it must use its format to work with the pop-up calendar.

Event settings

These settings apply only to the *Date with select list* field as it appears in the *Event* content type.

Default value:

Blank
 Now
 Relative

A default value to use for this field. If you select 'Relative', add details below.

— ▷ Customize Default Value

Input format:

2008-08-30 00:17:35

Set the order and format for the date parts in the input form. The format will be adapted to remove values not in the granularity for this field.

***Custom input format:**

The custom format, if provided, will override the input format selected above. See more about custom date formats below.

Years back and forward:

-3:+3

Number of years to go back and forward in the year selection list, default is -3:+3.

Time increment:

1

Increment the minute and second fields by this amount.

— ▷ Customize Date Parts

Help text:

Instructions to present to the user below this field on the editing form.
 Allowed HTML tags: <a> <big> <code> <i> <ins> <pre> <q> <small> <sub> <sup> <p>

Figure 9-3. Date field content type settings

Years back and forward

This setting gives us control over how many years will be listed in the widget for a user to select from. An example of the format used is $-2:+4$, where -2 is the number of years before the current year to show in the list and $+4$ is the number of years after. If the current year is 2009, this would indicate that the years available to select should be between 2007 (-2 years) and 2013 ($+4$). For our event site, there is

probably little use in using a wide range here, particularly in allowing support for events in the distant past.

Time increment

The time increment setting allows us to constrain how many minutes (and seconds, if specified in the granularity settings) will be shown. By default, all 60 minutes are available as options; however, in many instances, dates may be entered only as 15- or 30-minute intervals. Therefore, having all 60 minutes as options makes the drop-down list more cumbersome to use.

Customize date parts

A date consists of the date fields and the label for each field. You can customize where you would like to display the label in relation to its field: above the field, within the field (either as an option in a select list or inserted inside a text field) or none at all.

The “Select list” widget has a few more options. Despite its name, we can actually have text field entry for certain values in the date, mixing drop-downs and text fields. For instance, rather than having a select list of 31 days, we could set Day to be a text field input, in which case Drupal will render the input as select lists for year and month with a small text field for day. This option again allows us full control over the widget and a chance to select the interface easiest to use for our target audience.

Under the global date field settings, we see even more options related to how the field is handled:

To Date

The “To Date” setting allows you to associate an ending date/time, thus making the date field a date span. The setting has three options:

Never

No end date will be associated with this date field.

Optional

The date field can potentially have an end date and an entry widget will be displayed, but may be left blank.

Required

An end date must be entered if either the date field is required or a start date has been entered.

Granularity

The granularity data setting dictates how much information will be retained about the dates supplied. The checkboxes for Year, Month, Day, Hour, Minute, and Second can be selected independently to provide extreme flexibility. For instance, if we wanted only birthday (but not a full birth date), we could select only Month and Day; for example, July 10. For the purpose of event management, the default

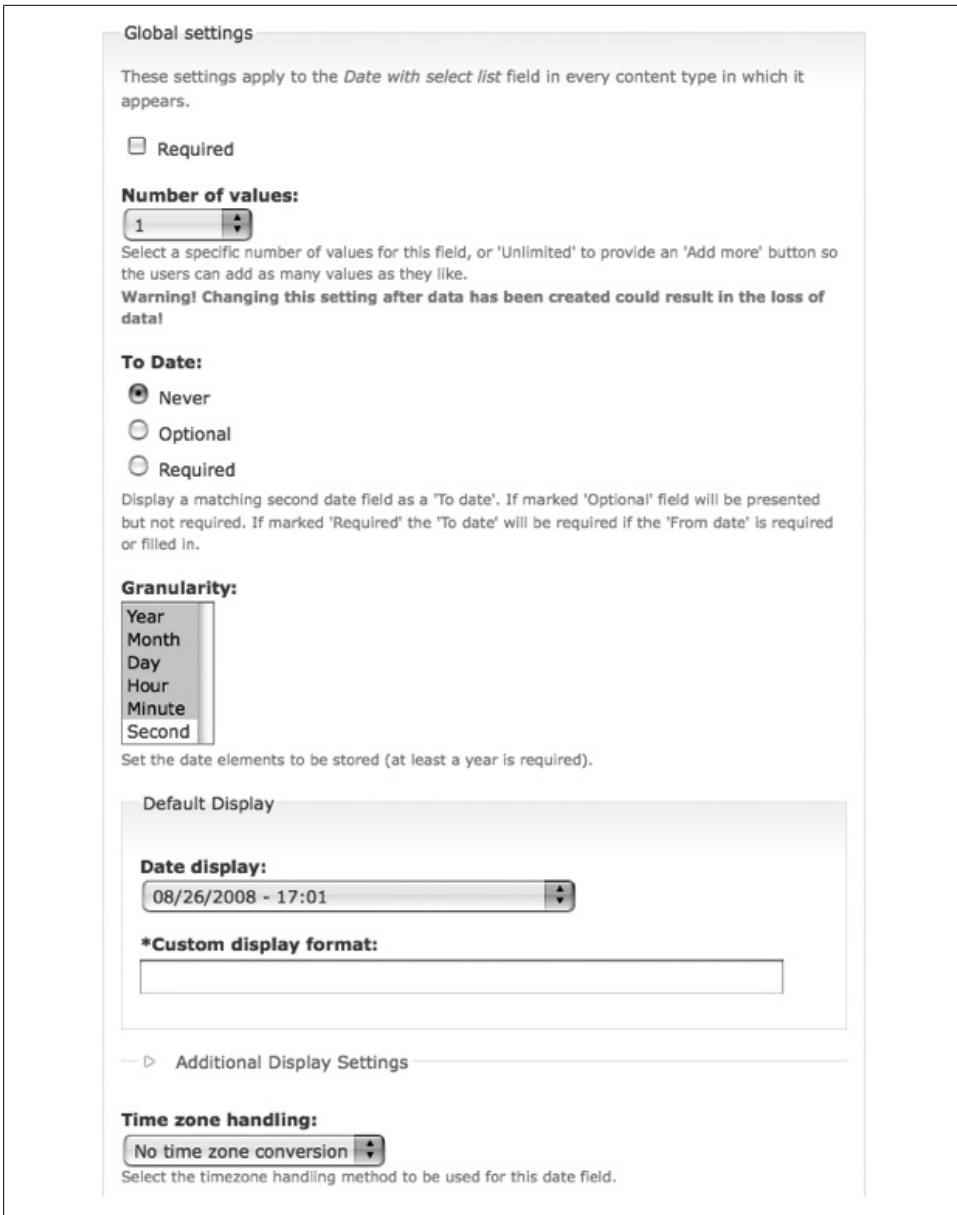


Figure 9-4. Date field settings

selection of Year, Month, Day, Hour, Minute is suitable, which allows us to display the date like July 10, 2008 7:30.

Note that the Granularity setting will impact the date entry widget, in that only the appropriate options will be displayed.

Default display

The default display setting section allows us to configure the format that will be used when displaying the date value. There are four formats that we can configure: default, which is shown on the form in Figure 9-4, as well as short, medium and long, which can be configured by expanding the Additional Display Settings field-set. For each of the display settings, there are two options. The “Date display” drop-down list contains a huge variety of date permutations (month names versus numeric months, 12- versus 24-hour clocks, time zone information as well as ordering of each year, month, day value). Should none of the listed options prove to be suitable, each display option has an additional “Custom display format” text field that allows even further customization, in the same way as the custom input format mentioned above.

Each of the four options—default, short, medium, and long—will be available then as CCK formatters for use in the content display settings as well as for Views and theming.

Time zone handling

The time zone handling settings allow us to configure how time zones should affect the stored date values and whether conversions should be performed. The options are described in Table 9-5.

Table 9-5. Date field time zone options

Option	Description
Site's time zone	The time zone specified for the entire site, specified at Administer→Site configuration→Date and Time (admin/settings/date-time). Useful for making sure each date field shares a consistent time throughout the site, even if users are from different time zones.
Date's time zone	Adds a Time zone drop-down next to the date widget to specify the time zone for the date. Useful for sites where many users from many different time zones will be creating dates.
User's time zone	The time zone, specified in each user's My Account settings if the option is enabled under Administer→Site configuration→Date and Time (admin/settings/date-time). This option is useful if you mainly have events in one time zone, but users from many different places.
UTC	Coordinated Universal Time (UTC), which is informally equivalent to GMT. This is a standard time zone that is the same across all systems.
No time zone conversion	For events with dates only, rather than dates and times, or for sites with both local events and users, performs no time zone conversions on the date.

For the book club, we will not be doing time zone conversions, as all members will be local.

Hands On: Adding Dates

In this section, we will transform our basic Event content type by adding the date component. This will be the linchpin of the book club’s site, as the rest of the site will build from this content.

Set Up the Date Module

1. Go to Administer→Site Building→Modules (admin/build/modules) and enable the following modules (note that if you are using PHP 4, you will need to enable Date PHP4 as well):
 - Date/Time package
 - Date
 - Date API
 - Date Popup
 - Date Timezone
2. Go to Administer→Site Configuration→Date and Time (admin/settings/date-time) and select an option from the “Default time zone” drop-down list that matches your time zone. Click “Save configuration” to save your changes.

Add the Date Field

With all our required modules enabled and set up, we can now customize the *Event* content type we created earlier by adding dates. Note that as we do this, we will only add a single CCK field to handle both the event’s start and end times:

1. Go to Administer→Content management→Content types (admin/content/types) and click the “manage fields” link for the Event content type (admin/content/node-type/event/fields). Complete the New field form with the values from Table 9-6.

Table 9-6. Settings for adding a time field to the Event content type

Field	Value
Label	Time
Field name	time
Select a field type	Datetime
Select a widget	Text Field with jquery pop-up calendar

2. Click Save. This brings us to the date field configuration screen. As the Date module is geared towards event management by default, several of the default settings work well. Enter the values from Table 9-7, and click the “Save field settings” button to complete adding the configuration.

Table 9-7. Date field configuration settings

Field	Value
Event settings	
Default value	Now
Input format	Select a format such as 08/29/2008 — 11:31pm
Years back and forward	-1:+2
Time increment	15
Global settings	
Required	Checked
To Date	Optional
Default Date Display	Select a format such as 08/29/2008–11:31pm
Time zone handling	No time zone conversion

- You should be returned to the “Manage fields” tab (`admin/content/node-type/event/fields/field_time`). Reorder the fields as follows and click Save:
 - Name
 - Time
 - Location
 - Description
 - Menu settings

With the content type fully created and permissions granted, our members can now post events to the site! To do so, go to Create content→Event (`node/add/event`) and complete the form with the settings in Table 9-8. If all has gone well, you should see something like the form in Figure 9-5. Go ahead and create a few more events for the Aurora Book Club.

Table 9-8. Initial example event

Field	Value
Name	September meeting
Time	
From date	09/17/2008—2:30PM
To date	09/17/2008—3:30PM
Location	The Book Nook on Main Street
Description	Our first meeting after the summer break! Andrew and Camryn are bringing cookies.

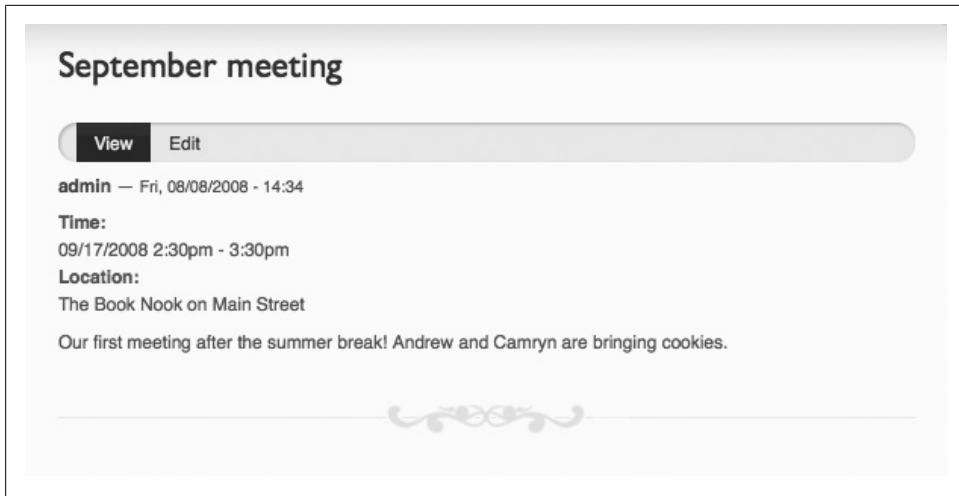


Figure 9-5. Our initial event

Hands On: Upcoming Events View

Now that we've created our event content type and started populating some content, it's clear that we need to add in a way to access all our event data. In the book club's requirements, we had a need for an "upcoming events" listing that would allow members to quickly see the meetings happening in the coming days or weeks. To achieve this, we will use the Views module to create our block. Keep in mind that when building views of event data, we generally want to do our sorting or our limiting on the date field, not the content's created or updated time, as we normally do.

We will create a simple block view of published events where the event's time field is in the future. In terms of the views configuration, having a date value "greater than now" represents dates "in the future." Finally, the view will be sorted in chronological (or ascending) order of the event's date (not the event posting's created date). When completed, this section will look as pictured in Figure 9-6. Clicking the event name link in the block will take you to the full information.

1. Go to Administer→Site building→Modules (admin/build/modules) and enable the following modules:
 - Other package
 - Advanced help
 - Views package
 - Views
 - Views UI
2. Once the modules are enabled, go to Administer→Site building→Views (admin/build/views) and click on the Add tab (admin/build/views/add).

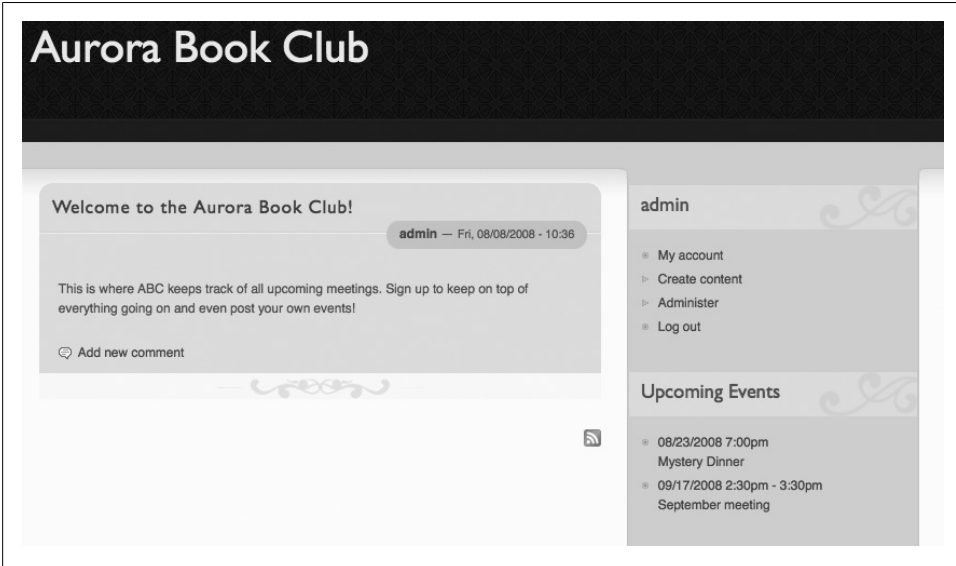


Figure 9-6. The Aurora Book Club site, showing list of upcoming events

3. Fill out the view form using the values from Table 9-9.

Table 9-9. The Upcoming Events view configuration values

View Setting	Value
View name	upcoming_events
View description	A block list of upcoming events
View tag	event
View type	Node

4. Clicking the Next button places us in the main Views interface. From here, we can configure our view. Let's begin with the Basic settings for the Defaults from Table 9-10, clicking Update after setting each one.

Table 9-10. Basic settings for the upcoming events view

Defaults: Basic setting	Value
Title	Upcoming Events
Style	List
	List type: Unordered list
Items to display	Defaults: Items to display: 5

5. Now, we must add some fields to our view. In the Fields section, click the + (plus) icon and add the following fields: "Content: Time (field_time value)" and "Node:

Title.” Configure the settings to match those in Table 9-11 and make sure to click the Update button after each one.

Table 9-11. Upcoming Events view field values

Defaults: Fields	Values
Content: Time (field_time value)	Label: None
Node: Title	Label: (make this blank) Link this field to its node: Checked

6. In the Filters section, we’ll add our required filters by clicking the + (plus) icon:
 - Date: Date
 - Node: Published
 - Node: Type
7. Recall that our view is for upcoming (that is, future) events; therefore, we will filter on dates that are greater than now. Match the filter settings with the values in Table 9-12 and Update them.

Table 9-12. Upcoming Events view filter values

Defaults: Filters	Value
Date: Date	Date field(s): Content: Time (field_time value) Operator: Is greater than or equal to Date default: now
Node: Published	Published: Checked
Node: Type	Operator: Is one of Node type: Event

8. Then, we need to similarly add the Sort Criteria, using the + (plus) icon. Add “Content: Time (field_time value),” ensure that the Sort order is Ascending and click the Update button to finish.
9. To create the block, add a Block display by selecting Block from the drop-down list on the left side of the interface and click “Add display.”
10. Under “Block settings,” under Admin, enter the Block: Block admin description to read Upcoming Events and click Update.
11. Save the view, which should now look like Figure 9-7.
12. Because we created a block view, we should see no change to our site until we enable the block that we’ve created. To do this, go to Administer→Site building→Blocks (admin/build/block). Drag the “Upcoming Events” row to the “Right sidebar” region (or simply change the region value in the drop-down) and click “Save blocks.”

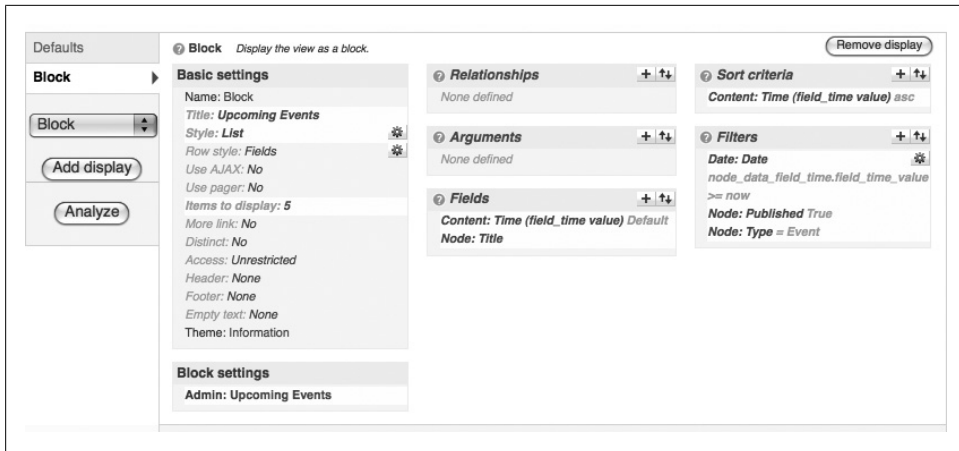


Figure 9-7. Upcoming Events block view

Spotlight: Calendar Module

Although a simple list of upcoming events is very useful (particularly in a sidebar block), the book club has additional requirements for the display of the event data. As is extremely common for event management websites, this site needs an interactive calendar for browsing through past and future events. We will implement this feature using the Calendar module in conjunction with Views.

In addition to creating a nice online calendar, the Calendar module can handle the need for book club members to be able to update their desktop calendars (in Microsoft Outlook or Apple's iCal) with the event information from the book club site. To do this, the desktop applications use a standardized format known as iCalendar (<http://en.wikipedia.org/wiki/iCalendar>), or iCal for short. Calendar comes with the Calendar iCal module, which allows us to easily provide this format for the interested members.

Calendar View Type

The Calendar module provides a new view type that shows the results of a view in a calendar rather than a list or table as with the default view types. This view type is one of the more complicated ones available. It provides full day, week, month, and year views of the event data on our site for us with lots of links between views and paging through days, months, and years. To achieve this rich functionality, Calendar requires certain views arguments to exist, and to be ordered and configured in a certain way.

The Calendar view type then determines which view the user would like to see based on the arguments that exist. For example, if our view URL is *calendar*, the Calendar view will handle the paths described in Table 9-13.

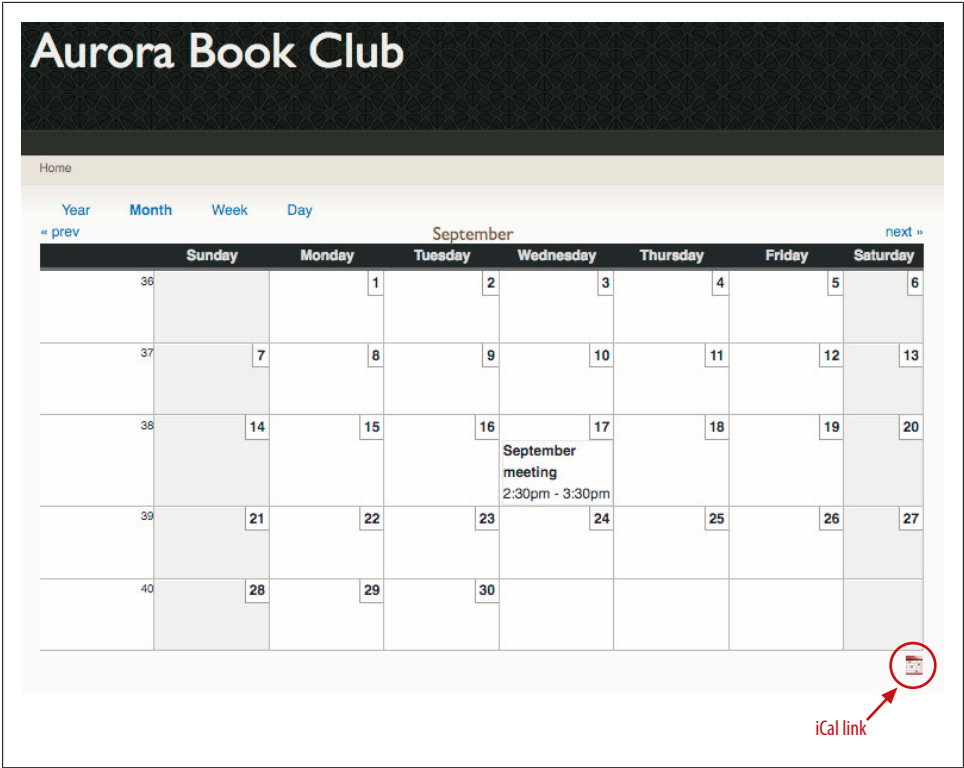


Figure 9-8. Completed event calendar, with iCal link

Table 9-13. Calendar path-based display

Path	Calendar display
calendar	Month view, defaulting to the current month.
calendar/1970	Year view, for the year 1970.
calendar/1970-1	Month view for January, 1970
calendar/1970-1-1	Day view for January 1, 1970

Hands On: Calendar View

In this section, we'll be enabling the Calendar view of book club events. Although this is potentially a daunting task, the Calendar module conveniently comes with a default view that handles most of the difficult bits for us. In this section, we'll alter that default calendar view to fit our requirements.

Figure 9-8 shows the finished Aurora Book Club calendar. Note the small iCal icon in the bottom right. Clicking this link will download the calendar to an appropriate desktop application.

- Go to Administer→Site building→Modules (admin/build/module) and enable the following modules:
 - Date/Time package
 - Calendar
 - Calendar iCal
- Go to Administer→Site building→Views (admin/build/views). You should now see “Default Node view: calendar” listed. Turn it on via its Enable link (admin/build/views/enable/calendar).
- Click its Edit (admin/build/views/edit/calendar) link to open the calendar view for editing.
- We need to make changes to the default Arguments. By default, the calendar view uses the date the node was last changed to place events on the calendar. In the Arguments section, click on the “Date: Date” link and under “Date field(s),” uncheck “Node: Updated date” and check “Content: Time (field_time value)” instead. Then click Update.
- For the Fields section, make the changes noted in Table 9-14 and click Update. This selection will show our event’s date, rather than the node’s last updated date, and will remove the word “Title” from the event name that comes by default.

Table 9-14. Calendar view fields configuration

Defaults: Fields	Value
Node: Updated date	(Remove this field)
Content: Time (field_time_value)	Checked
	Label: None
Node: Title	Label: (make this blank)

- We need to add some filters so that only our published Event content shows in the calendar. Do this by clicking the + (plus) icon next in the Filters section and select the Node: Published and Node: Type filters. Complete the filter configuration according to Table 9-15, clicking Update for each one.

Table 9-15. Calendar view filter configuration

Defaults: Filters	Value
Node: Published	Published: Checked
Node: Type	Node type: Event

- Finally, we need to add a menu item for the calendar page. The default view already provides us with a path of “calendar,” which makes sense for us to keep. To add this to the menu, click on the Calendar Page tab on the left side of the interface and complete the “Page settings” section according to Table 9-16. Click Update after you enter the menu settings.

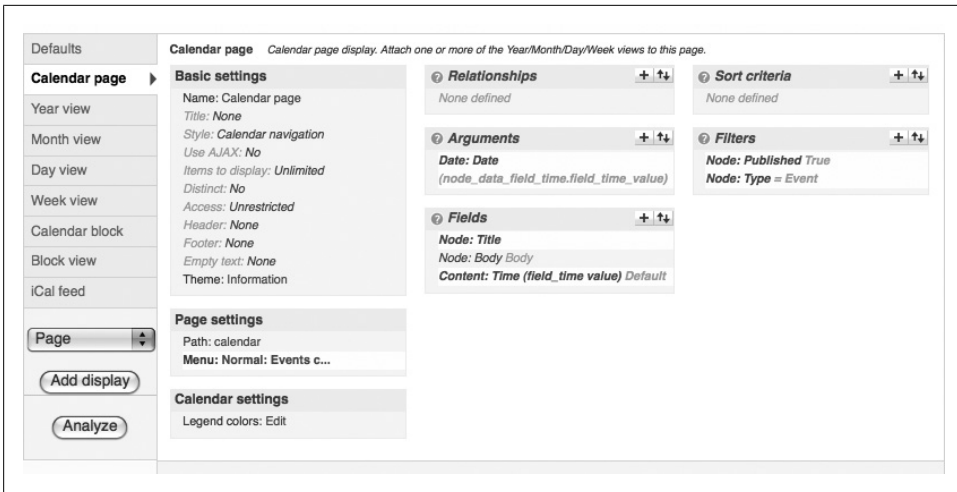


Figure 9-9. Upcoming Events calendar view

Table 9-16. The Calendar view’s Page settings

Page: Page settings	Value
Menu	Type: Normal menu entry Title: Events calendar

8. Save the view, which should look like Figure 9-9.
9. We can now move the Events Calendar menu item into the Primary Links menu. Go to Administer→Site building→Menus→Navigation (admin/build/menu-customize/navigation) and click “edit” in the Events calendar row.
10. Update the “Parent item” to read “<Primary links>” and click Save.

Now we have a working events calendar that can be reached by clicking on the events calendar link in the Primary navigation.

Spotlight: Flag module

The Flag module (formerly known as Views Bookmark) is an incredibly flexible module that allows you to create relationships between users and content on your site. After creating a flag, an item can be marked with the flag a few different ways, including links displayed below content, as shown in Figure 9-10, or checkboxes displayed on the edit form, as shown in Figure 9-11.

Upon installation, Flag module defines a “bookmark” relationship, allowing users to maintain a list of bookmarks (or posts they find interesting) on the site. However, this default behavior only touches on the surface of the flexibility of the Flag module. Some possible uses for Flag include:

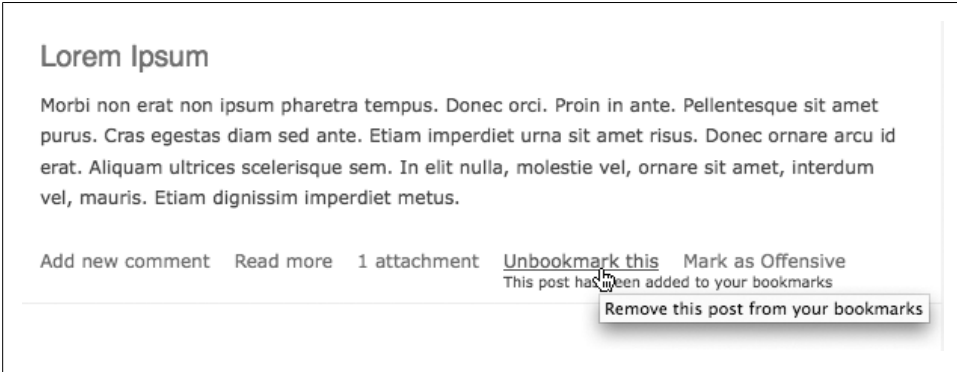


Figure 9-10. Flags as links shown on content

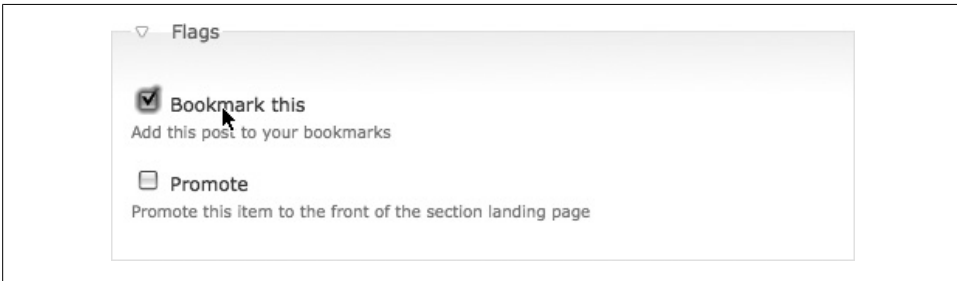


Figure 9-11. Flags as checkboxes shown on the node edit form

- A “favorite” or “bookmark” flag to mark content.
- A “promote” flag (or many different promote flags) that are similar to the default “Promote to Frontpage” checkbox.
- An “offensive” flag for comments and nodes.
- A “friend” flag that allows users to mark other users as friends.
- An “attending” flag for events.

As you can see, there are a variety of uses for flagging content. After creating a flag for some purpose, views can be constructed that create lists of content that has been flagged by users. We’ll use it to let users indicate whether they plan to attend a book club event. Once the flag relationships are created, we can create a view to list the attendees of a particular event.

Hands On: Flag Configuration

In this section, we will configure the Flag module to allow our users to indicate that they are attending our events, as pictured in Figure 9-12.

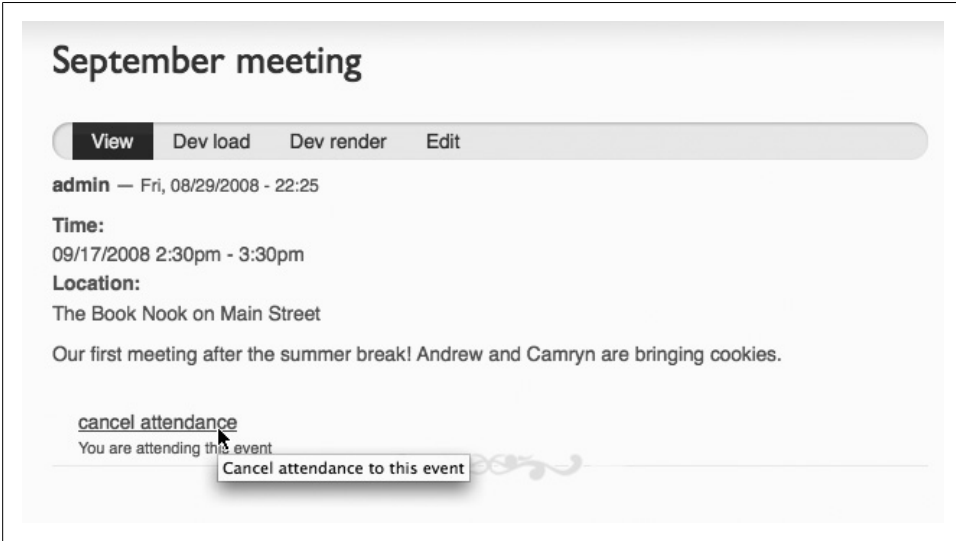


Figure 9-12. Flags as checkboxes shown on the node edit form

1. Go to Administer→Site Building→Modules (admin/build/modules) and enable the “Other package: Flag” module.
2. Now go to Administer→Site Building→Flags (admin/build/flags) to manage the defined flags. By default, the Flag module defines a “Bookmarks” flag when installed. We can either add a new flag or edit the default one to suit our purpose. As we won’t be using “Bookmarks” on this site, we’re going to edit the default. Click the “edit” link next to the bookmark flag (admin/build/flags/edit/bookmarks). Fill out the form according to Table 9-17.

Table 9-17. Attendance flag configuration

Field	Value
Name	attendance
Title	Attendance
Flag link text	attend this event
Flag link description	Attend this event
Flagged message	You are attending this event
Unflag link text	cancel attendance
Unflag link description	Cancel attendance to this event
Unflagged message	You are no longer attending this event
What nodes this flag may be used on	Event: Checked Story: Unchecked
Display options	Display link on node teaser: Unchecked

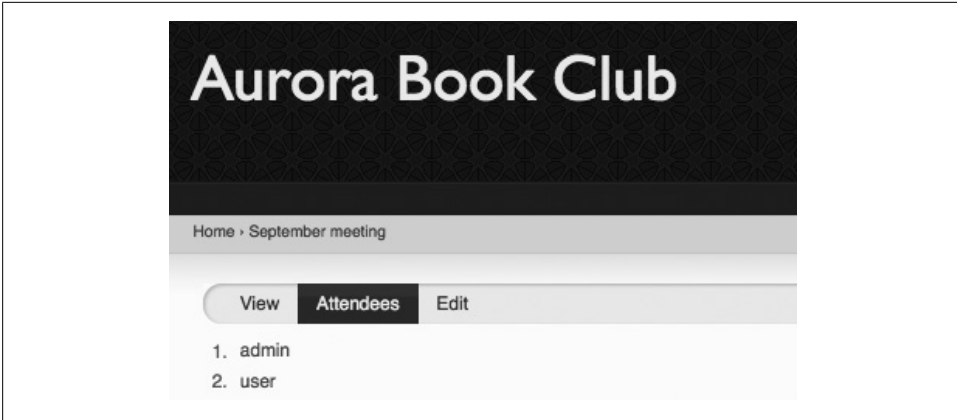


Figure 9-13. Event attendees list view

Field	Value
	Display link on node page: Checked
	Display checkbox on node edit form: Unchecked

3. Clicking Submit will create our attendance flag. We can now go to any events that we previously created and click the “attend this event” link.

Hands On: Attendee View

The Flag module provides a default view that will allow any user to see the content that they have flagged for any flag that we have defined. You should now see a My Attendance link in the Navigation menu, which will show all of the events that you have marked as attending.

However, the book club would also like to see a list of who will be attending each event. For this, we will need to create a new view, pictured in Figure 9-13.

1. Go to Administer→Site Building→Views (admin/build/views) and click on the Add tab.
2. Complete the form according to Table 9-18.

Table 9-18. Attendees view settings

Setting	Value
View name	attendees
View description	Attendees for a given event
View tag	event
View type	User

- The first thing we need is a relationship to give us access to the flag information related to our users in our view. Click the + (plus) sign in the Relationships section and check the “Flags: User’s flagged content” relationship and click Add. Complete the relationship settings based on Table 9-19.

Table 9-19. Settings for the Flags relationship

Defaults: Relationship	Value
Flags: User	Label: attendance
	Include only users who have flagged content: Checked
	Flagged: Attendance

- We want our view to show the users who have flagged a given Event node; therefore, we need to add an argument for the Node ID that was flagged. To do this, we click the + (plus) sign next to Arguments. Check the “Flags: Content ID” argument and click Add. The default values for the argument are fine, so we can click Update.
- Now we can add the fields that we want to display. Click the + (plus) icon next to Fields. Check the “User: Name” field and then Add. Blank out the Label text so that there is nothing there and click Update.
- It would be helpful if the attendees were listed in an ordered list so that we could quickly glance at the total number of people attending. To add this functionality, we need to change the style of the view. Click on Style under “Basic settings” and change it to List. Then set the list type to “Ordered list.” Click Update to save the change.
- To complete the view, we need a Page display. Select Page from the drop-down list on the left side of the interface and click Add Display. Complete the Page settings according to Table 9-20.

Table 9-20. The Attendees view Page settings

Setting	Value
Path	node/%/attendees
Menu	Type: Menu Tab
	Title: Attendees

- Save the view, which should look like Figure 9-14.

Now, when you visit an Event post, you should see an Attendees tab. Clicking on that tab will display a list of all of the users who have said they will attend the event.

Taking It Further

The site we have built covers all of the needs for the club. As they go down the road, they may want to spruce things up a bit. Here are a few modules that could round out the site even more:



Figure 9-14. Event attendees list view settings

Calendar Popup (part of the Calendar module)

The Calendar package also includes this module. This will make it so that when you click on an event in a calendar display, it will appear as a JavaScript pop-up rather than taking you to the node page. This is handy if you want people to be able to browse the details of events without leaving the calendar view page.

Countdown (<http://drupal.org/project/countdown/>)

This module adds a block that shows the time left until an event. This is a nice way to let people quickly know that the next meeting is in four days or four hours.

Flag Actions (part of the Flag module)

The Flag module can be set up to send emails, and to unpublish or delete nodes upon reaching certain flagging thresholds. Although this feature is most commonly used for things like community flagging of spam or offensive content, it can also be used to notify someone by email if, say, more than 10 people will be coming to an event and (for example) a second person needs to be asked to help supply refreshments.

Location (<http://drupal.org/project/location/>) and GMap (<http://drupal.org/project/gmap/>)

Instead of just typing a location in a text field, you can use the Location and GMap modules to let people use a Google map to select the location for each event.

Summary

In this chapter, we have looked at building an event management site for the Aurora Book Club, making use of the Date field for CCK, the Calendar plug-in for Views, and the Flag module for handling attendance. The book club now has a handy calendar for display on the site, as well as being available in iCal format. They also have an easy-to-find list of all the attendees. The site is simple and easy to use, yet fits all of their needs quite nicely.

Here are the modules referenced in the chapter:

- Calendar: <http://drupal.org/project/calendar>
- Countdown: <http://drupal.org/project/countdown>
- Date: <http://drupal.org/project/date>
- Event: <http://drupal.org/project/event>
- Flag: <http://drupal.org/project/flag>
- GMap: <http://drupal.org/project/gmap>
- Location: <http://drupal.org/project/location>
- Signup: <http://drupal.org/project/signup>
- Views: <http://drupal.org/project/views>

Additional resources :

- ADOdb Date Time Library: http://phplens.com/phpeverywhere/adodb_date_library
- Date module handbook: <http://drupal.org/node/92460>
- Event related modules: <http://drupal.org/project/Modules/category/61>
- iCalendar: <http://en.wikipedia.org/wiki/ICalendar>
- PHP date formatting: <http://php.net/date>
- PHP strtotime: <http://www.php.net/manual/en/function.strftime.php>

To purchase this book, go to <http://www.oreilly.com>.