Library Technology Reports Respond to Your Library's Digital Dilemmas

Eight times per year, Library Technology Reports (LTR) provides library professionals with insightful elucidation, covering the technology and technological issues the library world grapples with on a daily basis in the information age.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>“Changing the Way We Work”</td>
<td>Michelle Boule, Social Sciences Librarian, University of Houston, TX</td>
</tr>
<tr>
<td>February/March</td>
<td>“The Preservation of Digital Materials”</td>
<td>Priscilla Caplan, Assistant Director for Digital Library Services, Florida Center for Library Automation, Gainesville, FL</td>
</tr>
<tr>
<td>May/June</td>
<td>“Drupal in Libraries”</td>
<td>Christopher Harris, Coordinator, School Library System, Genesee Valley BOCES, Le Roy, NY and Andy Austin, Library Technologies Specialist, Genesee Valley BOCES, Le Roy, NY</td>
</tr>
<tr>
<td>July</td>
<td>“On the Move with the Mobile Web: Libraries and Mobile Technologies”</td>
<td>Ellyssa Kroski, Reference Librarian, Columbia University, New York, NY</td>
</tr>
<tr>
<td>August/September</td>
<td>“WorldCat Local at the University of Washington”</td>
<td>Jennifer Ward, Steve Shadle, and Pam Mofjeld, University of Washington Libraries</td>
</tr>
<tr>
<td>October</td>
<td>“Libraries on the MUVE (Multi-User Virtual Environments)”</td>
<td>Thomas A. Peters, Principal, TAP Information Services, Blue Springs, MO</td>
</tr>
</tbody>
</table>

Library Technology Reports 2008, Vol. 44

Drupa in Libraries
by Andy Austin and Christopher Harris
Drupal in Libraries

Andy Austin and Christopher Harris
About the Authors

Andy Austin is a Library Technologies Specialist with the Genesee Valley BOCES, where he spends much of his time developing Drupal sites. In addition to his BA in English from SUNY Geneseo and MLS from the University of Buffalo, he has New York State teacher certification as a school library media specialist. When not spending time with his wife, Vicky, and two young sons, Eric and Nate, he can be found maintaining the Drupal Marc project or infrequently posting on his personal Drupal site, http://posttext.com.

Christopher Harris, author of the Infomancy blog, is the coordinator of the school library system for Genesee Valley BOCES, an educational services agency that supports the libraries of 22 small, rural districts in western New York. In addition to blogging on Infomancy, Christopher is a technology blogger for School Library Journal on Digital Reshift as well as a regular technology columnist. He was a participant in the first American Library Association Emerging Leaders program in 2007 and was named a Library Journal Mover and Shaker for 2008. Christopher is an avid gamer as well as a dedicated reader. He lives with his wife, an elementary librarian, in Le Roy, New York.
# Table of Contents

## Chapter 1—Welcome to a New Paradigm 5
- Content Management Systems 5
- Benefits of a CMS 5
- Why Drupal? 6
- Who Is Drupal? 7
- Notes 7

## Chapter 2—Getting Started With Drupal 8
- System Requirements 8
- Installation 8

## Chapter 3—Adding “Stuff” to a Drupal Site 10
- Nodes 10
- Types of Nodes 11
- Things That are Not Nodes 11
  - Comments 11
  - Taxonomies 12
  - Menus 12
  - Blocks 13
- Notes 14

## Chapter 4—User Management 15
- Using Roles 15
- Assigning Permissions 16

## Chapter 5—Customizing the Drupal Look 18
- The Drupal.org Web Site 18
- Themes 19

## Chapter 6—Modules 21
- Building Modules 21
- Don’t Hack Core 22

## Chapter 7—The Trifecta 23
- CCK 23
- Views 24
- Panels 24
- Putting It All Together 25

## Chapter 8—Other Modules of Note 27
- Core Modules 27
  - Blog 27
  - Book 27
  - Path 27
  - Poll 28
  - Profile 28
  - Upload 28
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributed Modules</td>
<td>28</td>
</tr>
<tr>
<td>Advanced Forum</td>
<td>28</td>
</tr>
<tr>
<td>Biography</td>
<td>28</td>
</tr>
<tr>
<td>ImageField/ImageCache</td>
<td>29</td>
</tr>
<tr>
<td>Nodequeue</td>
<td>29</td>
</tr>
<tr>
<td>Pathauto</td>
<td>29</td>
</tr>
<tr>
<td>Signup</td>
<td>29</td>
</tr>
<tr>
<td>Workflow</td>
<td>29</td>
</tr>
<tr>
<td>CCK Fields</td>
<td>29</td>
</tr>
<tr>
<td>Link</td>
<td>29</td>
</tr>
<tr>
<td>NodeReference</td>
<td>30</td>
</tr>
<tr>
<td>FileField</td>
<td>30</td>
</tr>
<tr>
<td>Much More</td>
<td>30</td>
</tr>
<tr>
<td>WYSIWYG</td>
<td>30</td>
</tr>
<tr>
<td>Notes</td>
<td>30</td>
</tr>
<tr>
<td>Chapter 9—Case Studies</td>
<td>31</td>
</tr>
<tr>
<td>Fish4Info: School Library System of Genesee Valley BOCES</td>
<td>31</td>
</tr>
<tr>
<td>E-Branch in a Box: Idaho Commission for Libraries</td>
<td>33</td>
</tr>
<tr>
<td>Notes</td>
<td>36</td>
</tr>
<tr>
<td>Chapter 10—Resources</td>
<td>37</td>
</tr>
<tr>
<td>Drupal in Libraries</td>
<td>37</td>
</tr>
<tr>
<td>Drupal Resources</td>
<td>37</td>
</tr>
</tbody>
</table>
Welcome to a New Paradigm

Content Management Systems

Libraries are about content: acquiring it, storing it, indexing it, retrieving it, and presenting it. Content management systems (CMS) help libraries accomplish these tasks on the Web by providing a back-end structure for a Web site so that the authors can focus on content. Unlike a traditional Web site, where HTML defined both the content and the formatting in a single document, a CMS uses databases and newer Web languages to store content and define formatting separately. This means that instead of using this:

```html
<H1>This is the Title</H1>
```

a CMS might use this:

```html
<div style="title">$site_title</div>
```

So why should you care? The real power of a CMS is that when you separate content from formatting, you can more easily change either without having to recode your entire Web site.

Instead of storing the text in the code for the page, a CMS instead uses a reference to a variable stored in a database. Similarly, the formatting for the title in this example is defined in a stylesheet instead of within the page code. So why is this so important? This separation of content and formatting is the base technology that makes possible everything discussed herein. When content is approached in this separated fashion, it can be presented for editing in a graphical WYSIWYG (What You See Is What You Get) environment. If the content is edited and stored outside of the Web site itself, you can assign different users different levels of permission for editing content. Furthermore, since adding content doesn’t involve editing the code of the Web site, it becomes much easier to make more frequent updates. In the following chapters, you will learn how Drupal, an open-source content management system, makes use of the separation of content and formatting to let you create a powerful, dynamic library Web site.

Benefits of a CMS

Content management systems can be defined by three common attributes beyond the separation of content and formatting:

- They provide a framework for creating, managing, and publishing Web-based content.
- They provide a secure environment with managed user roles.
- They provide extensions for enhanced capabilities.

Though small library Web sites may have only a few pages, larger sites can run to hundreds or thousands of pages. CMS sites are based on dynamically generated chunks of content that can be more easily displayed for online management (see the sidebar HTML vs. PHP). This is especially important given the ease with which new content can be created in a WYSIWYG environment by users who may not have been able to participate in a code- and file-based system. With a CMS, work can be distributed across multiple authors without sacrificing site security: a department head can publish news updates directly to her or his page, while another staff member can write con-
tent that has to be approved before it is displayed. Finally, most of the available CMS applications can be extended to accomplish new tasks or provide new data types through small plug-in modules that meet a specific need.

Open Source CMS
http://opensourcecms.com

Why Drupal?

To call Drupal a CMS might undersell it a little bit. The Drupal community often refers to Drupal as a content management framework. This is intended to convey the idea that Drupal is not a fixed system but a framework on which you can build your own systems. When we started using Drupal, we were looking for a CMS, so we were looking at products like WordPress and Mamba. Other groups approach Drupal as an alternative to rapid Web development tools like Ruby on Rails or CakePHP, neither of which would be considered a CMS. Drupal offers a nice learning curve as you move from using it as a CMS to using it to develop your own Web applications. You can use it out of the box as a very nice, dynamic Web site. As you learn more about configuring it from the Web interfaces, though, it becomes very flexible. Then when you are ready to start writing your own code, it provides you a powerful set of internal APIs (Application Protocol Interfaces) to streamline your coding.

With the continued threat of legislation that would ban social Web sites from schools and libraries, we wanted to find a way to continue to provide access to social Web applications in an educational setting. There is no dearth of open-source projects looking to duplicate popular Web services like del.icio.us, Flickr, and the like. But we were also looking for a Web application to help us manage our projects and track issues and problems. The fear was that we would end up with a huge number of applications to download, install, and maintain on our servers.

We had already been using Drupal for a number of tasks, including our public Web site, extranet, and a shared electronic resources purchasing system. As we continued to work with the system, it became obvious that we could use Drupal for almost anything we needed: shared department calendars, project management, bookmarks, knowledge base, and more. With a staff of three librarians, our system cannot become experts on many different Web applications. Drupal provides a stable and flexible platform upon which we can base our Web development.

The major project that we have been working on in Drupal is a next-generation library portal called Fish4Info. This site features a library catalog built in Drupal along with book reviews, pathfinders, an events calendar, and much more.
Who Is Drupal?

As with most successful open-source projects, Drupal's greatest asset is its community. Drupal was created by Dries Buytaert as a Web board on a shared dorm Internet connection, developed into the Web site drop.org, and finally distributed as an open-source Web platform. Through the development of Drupal 5 and much of Drupal 6, Dries continued his studies full-time as a PhD student in computer science, spending his days working on Java and his nights guiding the development of Drupal. Since graduation, Dries has found venture-capital funding and has formed Acquia to provide commercial support to the Drupal community, much as Red Hat provides support for Linux users.

Unlike the larger-than-life personalities that dominate some open-source projects, Dries has acted more as a guide than a dictator. He spends much of his time offering guidance, reviewing patches, and helping to settle debates. Perhaps this is because of the open nature of the Drupal community that Dries has fostered, but the success of Drupal is mostly due to the number of talented developers who contribute their work to help move the project forward. Most of the active developers in the Drupal project work as Web site developers and have a real stake in the growth of the project. By developing and sharing code, they make each other's work that much easier. But much of the development is done with a "scratch your own itch" mentality: projects are most often done when they are going to be used by the developers coding them. In addition, many of the new and cool features that get put into Drupal end up being invisible to the lay user and can be appreciated only by users who are themselves building Web sites.

To counter the developer-centric current within the Drupal community, Jeff Robbins of Lullabot started the meme “Drupal Will Save the World,” which seeks to put an idealistic spin on Drupal development—the thought being that Drupal can provide a powerful activist platform for the disadvantaged. Drupal proved its worth as a community tool in 2004, when staffers in the Howard Dean campaign used Drupal to create DeanSpace, later named CivicSpace, which is widely used in the political realm for campaign sites. But to really save the world, Drupal needs to be more accessible to disadvantaged communities who do not have the technical background to build these Web sites. Along these same lines, the University of Minnesota Libraries stepped in to do formal usability testing on Drupal 6, seeking to greatly improve the usability of its interfaces.

The result of the pairing of a strong development community with a focus on usability is a powerful framework that comes with a usable front-end CMS.

Notes

Drupal is an open-source project released under the GNU General Public License Version 3, or GPLv3. A detailed overview of the licensing can be found in the Drupal installation archive or on the GNU General Public License page. This license requires that any projects developed using Drupal also be released under the same GPL. It is important that anyone considering the use of Drupal read and understand the requirements of the GPLv3 and how it interacts with any institutional policies or expectations for development projects.

GNU General Public License page
www.gnu.org/copyleft/gpl.html

**System Requirements**

Drupal is designed to run under what is commonly referred to as a LAMP environment. This refers to the Linux operating system, the Apache Web server, the MySQL database, and the PHP language. This environment is built from open-source software, which means the only startup cost for Drupal is the hardware for a Web server. Most commercial Web hosting companies provide these services as a default package, though you may need to confirm that all of Drupal’s specific requirements for settings in Apache and PHP are met. The core Drupal software will run under Microsoft’s Server software using the IIS Web server, but user-contributed add-ons and advanced functionality may not be fully supported. At this time, Drupal cannot use Microsoft’s SQL database, so Windows users would still need to install MySQL and PHP.

The school library system of Genesee Valley BOCES is running Fish4Info for over 30 libraries, along with quite a few additional Drupal sites, on a single server. The server is, as servers run, about average. It has dual processors and 4GB of memory and uses a RAID setup for faster hard drive access. We expect this $8,000 server to scale up to meet our Drupal needs for the next few years as we add more libraries to Fish4Info. For smaller sites, a basic Web server will run about $3,000 for a decent amount of memory and RAIDed hard drives. Libraries without the technical support or infrastructure to run their own server can easily purchase Web hosting services for their Drupal site.

**Installation**

Drupal installation is actually very easy once the required support systems of Apache, MySQL, and PHP are set up. To start using Drupal, download the latest stable version from the Drupal.org Web site and copy the files from the downloaded archive to your Web documents directory. When you download Drupal, it will come in a compressed .tar.gz file. You need to extract this file to add it to your site. If you are using Linux or Mac OS X, this is no problem. In Windows, you may find that your regular zip-centric compression utility does not know how to handle .tar.gz files. You can use the free software Extract Now at to access the files.
After downloading and extracting the installation files, the first thing you will need to do is create a database within MySQL to hold the information from Drupal. Once you have a database created, you can run the Drupal installation program by pointing your browser to the root of the directory into which you installed Drupal. The installation dialog will ask for the name of your MySQL database as well as the database username and password (see figure 1). Assuming there are no problems, Drupal will then proceed to install itself with a default set of configurations. If you do encounter problems, you can find additional help in the handbooks on Drupal.org.

When everything is installed, you will be taken to your new Drupal Web site. The next thing you must do is create your first user. Click on the link Create the First Account under Step 1. This first account is a super-user account that has complete access to the entire Web site. Protect this account password! After logging in with your new user, you can click on the Administer link in the left sidebar. This will take you to the administration section, which will show a message about some problems with your installation. Click on the Status Report link to see any issues that Drupal may have found. After this, you are ready to begin working with your site.

**Testing Drupal: No Server Required**

If you want to take a look at Drupal and work through some of the information provided here, there is a way to install a test site that does not require a Web server. It must be stressed, however, that the method discussed here is for testing purposes only and is not intended for use as a live server. To try Drupal on a Windows computer, you can download and install XAMPP, a single Windows program that provides an easy-to-use package of Apache, MySQL, and PHP. XAMPP is a free download. There is also a Mac OS X version available for XAMPP, but we had better luck with MAMP, a Mac-only free download. Once you set up one of these packages, follow the regular installation instructions.

**XAMPP download**

**MAMP download**

![Figure 1](image)

Drupal installation dialog.