

Adding “Stuff” to a Drupal Site

What good is a Web site without information? Drupal makes adding new information much easier to accomplish than it is to describe. Describing this step is severely restricted by the need to be very careful with vocabulary. Because Drupal is a content management system, it is very tempting to refer to everything on the site that is being managed by the system as “content.” This would be technically inaccurate, though the distinction is subtle. Like many systems, Drupal attaches special meanings to some words that we would normally use as general descriptors. *Content* is one of those words; in Drupal, *content* is synonymous with *node*, and nodes are very special things. So please pardon the occasional need to either formally refer to information (or informally call things “stuff”) when a word like *content* would seem to make more sense.

Nodes

Whether a news story, information page, blog post, event on a calendar, or forum topic, a piece of content in Drupal is a node. Nodes are the basic element from which a site is constructed; at a very basic level, nodes are defined as chunks of information consisting of a title and a body that are stored as a single content item for later reference (see figure 2). For example, on a Drupal site, a node for a blog post is a chunk of content that contains the title and body for the post that can be viewed on a page at an address like `?q=node/1234`. The node is the combination of the stored information and the storage and reference structure itself.

Unlike a traditional HTML Web site, where pages are the only level of content creation, breaking information down into smaller nodes gives Drupal much more flexibility. Multiple nodes can be placed on a single page, and

Figure 2
Screen for defining a node in Drupal.

a single node can be displayed on multiple pages. With content and formatting separated, Drupal can break free of a predefined structure. Perhaps this is best described by this explanation: “Drupal treats your content as a giant soup, with each piece of content having properties like a title and author, a ‘published’ flag, a publication date, and so on.”¹ While nodes can be viewed separately, the real power of Drupal is in combining nodes for display. A blog is created in Drupal by collecting and displaying all nodes by a certain author that are of the blog type. In the same way, the default front page of a Drupal site consists of all the nodes with the option Promoted to Front Page option checked.

Customizable display properties are only one of the special capabilities of nodes within Drupal. Nodes are, for instance, also the only pieces of information in Drupal that can be extended by other nodes or pieces of information. For example, a node can have an uploaded file attached to it. Nodes can also reference other nodes; an event node can be extended by a node that handles registrations for the event.

Types of Nodes

The words *node* and *content* are used interchangeably within Drupal because the different types of nodes that can be added to a site are referred to as *content types*. The default installation of Drupal comes with two types of content: stories and pages. Each of these content types is built from a title, a body, and options. Since they both have the same possibilities for a title and a body, the only difference between a story and a page are the default options that define how a node is displayed. By default, stories have the option Promoted to Front Page turned on and so are automatically displayed on the home page for the site. Pages have this option unchecked by default, although they can also be displayed on the front page if desired.

Other types of content are not as similar as the default story and page content types. Sites can add additional content types by installing and enabling additional modules (see chapter 5). You can find modules that create blog posts, polls, biographies, podcasts, and many more types of content. These content types may either extend the basic structure of title and body—polls have a question as a title and then possible answers instead of a single body—or may have different display properties.

About Nodes

I think the genius of what Dries and the community have done is to reduce all of the aspects of social software to their core DNA: content nodes and membership, and then build a platform that could be infinitely extended to allow the assembly of almost any style of online social interaction. —Jeff Whatcott²

Things That are Not Nodes

While nodes are where users add their primary content to Drupal, the CMS has other bits of information that support or define nodes. Comments, taxonomies, menus, and blocks are not nodes in that they cannot be directly

manipulated within the framework; instead, they exist in a symbiotic relationship with the content nodes.

Comments

In Drupal, any node can have comments. Comments cannot exist independent of nodes, but rather are small snippets of information that are linked to a specific node (see figure 3). The Comment module creates the association, stores the comments for each node, and causes them to display with nodes when appropriate. The module also attaches the comments to the node for purposes of searching. As with other node preferences, you can set comments to default as Disabled, Read Only, or Read/Write for each content type you are using by going to the Content Types section of the Administration area. The content settings for each individual node can also be adjusted independently for a higher level of control. Comment permissions can also be set by user role; you can define which roles can access comments, administer comments, post comments, and post comments without approval. With careful manipulation of these options, you can set up a variety of commenting scenarios.

In a Drupal-based library catalog, a library could use comments for staff notes about resources by turning off access to comments for all roles except library staff and then giving the library staff role permission to post comments without approval. Now librarians can leave on library resource nodes notes about weeding or usage that can be seen only by library staff. A library that wanted to let children or YA users talk about upcoming events might want to allow comments but require approval before comments are displayed. This is accomplished by giving the user role the Post Comments permission that holds comments from those users for review by a user

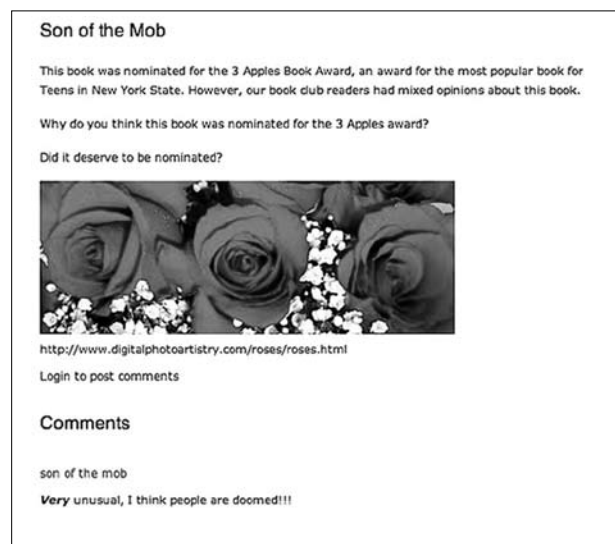


Figure 3
A Drupal page with comments.

with the Administer Comments permission. It is important to remember, however, that since comments are not nodes, they can't be put through the same complex moderation procedures as nodes. Comment moderation is limited to approval, deletion, or minor editing. This is why in Fish4Info, book reviews are implemented as a special type of node that attaches to a book node as opposed to being a comment on a book. Comments are used in other places in the portal, but since they are not nodes, there is very little else that can be done with them.

Taxonomies

Drupal also extends nodes with a taxonomy system powerful enough to satisfy even a librarian. For every content type, administrators can select to attach any combination of controlled vocabularies and/or free tagging. Controlled vocabularies can be set to be single selections or multi-select fields, as well as being either mandatory or optional. This type of taxonomy can also be built as a hierarchy, with additional terms available or hidden after an initial selection (see figure 4). Free tagging fields are enhanced with an auto-fill system that suggests possible existing matches as a user inputs a tag.

Taxonomies are a very powerful addition to nodes, allowing you control over grouping, filtering, and much more. Drupal automatically creates a page for each term in a taxonomy to allow a grouped display. At a very basic level, this can allow a user to display all of the books tagged as being about "friends" simply by clicking on the term *friends* in a single node. Terms are also used as criteria for filters to determine what items are displayed in a view (see chapter 6) and are used by a contributed module as a filter in determining similar items by term. Another contributed module, Tagadelic, will create a tag cloud based on terms in a taxonomy or create a directory display of terms.

Curriculum Area	
Name	Operations
Arts	edit
Careers	edit
ELA	edit
Foreign Language	edit
Health	edit
Library	edit
Math	edit
Science	edit
Social Studies	edit

Figure 4
Drupal's taxonomies provide a high level of organization.

In Fish4Info, taxonomies played a critical role in the development of a Drupal-based catalog. While storing the subject headings in a controlled vocabulary didn't take much inspiration, we did end up using taxonomies in more creative ways as well. By storing author names in a taxonomy, we were able to make use of the automatically created Term page to allow an author's name to be a hyperlink to a list of all works by that person. This simple solution is an excellent example of using the core functionalities of Drupal as a replacement for having to write custom code. Not only was storing author names as a taxonomy an easy way of accomplishing what we wanted to do, it is the accepted best practice for using Drupal.

Menus

As a framework for managing content, Drupal has an advanced menu system that controls not only navigation, but also security and elements of page display. The beauty of Drupal's menus is that many of the most powerful features will just happen, with no customization required (see figure 5). For example, since the menu system is fully integrated with the security controls, users will not see menu items for sections of the site to which they do not have access. If students don't have access to the Faculty Discussion Forum, it will never be displayed, and they won't even know it exists. This means menus are much cleaner and easier to navigate, especially for users with fewer permissions who may be less familiar with a site.

The main menu system is managed as a hierarchy of weighted links to internal pages defined within a menu container (see the sidebar Weight). These links are stored as menu items within a menu container. Menu containers can then be displayed on a site within a block (see

Menu item	Enabled	Expanded	Operations
Compose tips (disabled)	<input type="checkbox"/>	<input type="checkbox"/>	edit
My account	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
Create content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
Page	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
Story	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
Administrator	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
Content management	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
Comments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
Content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
Content types	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
Post settings	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
RSS publishing	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
Taxonomy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit

Figure 5
Drupal screen for managing site navigation.

below). There are two special menu areas that are defined within most themes: primary links and secondary links. As with other menus, a hierarchy of links can be defined to provide additional links after the selection of an initial primary link (see figure 6).

Blocks

The other items discussed in this chapter are clearly not nodes, but blocks can be more confusing. Unlike menus or taxonomies, blocks can mimic nodes and hold information as if they were pages. It must be stressed, however, that blocks are not nodes. Though information can be stored in a block, it cannot be acted upon as with a node. Blocks cannot have comments, cannot be categorized with taxonomy terms, and cannot be referenced by

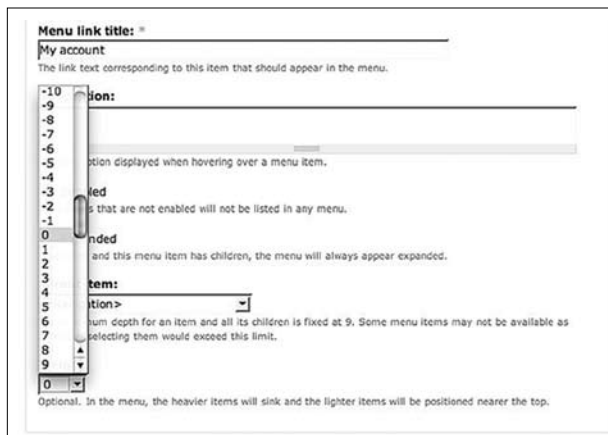


Figure 6
Drupal screen for setting weights for links.

Weight

Menu links and many other items are arranged for display based on a common Drupal weighting system. Items are assigned a weight from -10 to 10. Items default to a neutral 0 weight. Lower numbers will float to the top of a display list, while higher numbers will sink to the bottom. Multiple items with the same weight are arranged in alphabetical order. Since items default to a common 0 weight, this means that lists will start out arranged alphabetically. This concept has been shown to be confusing enough to users that a concerted effort has gone into hiding the weight selectors in Drupal 6 and instead relying on drag-and-drop reordering. Weight will still be encountered in some situations, however.

a menu. Even so, blocks serve a number of very useful purposes within Drupal. Most important, they allow information to be placed around a page in block regions (see the sidebar Jargon Alert). Like menu items, blocks within the same region are weighted to determine order. The default Drupal theme has a user login block at the top of the left sidebar, followed by a block for the navigation menu (see figure 7).

Other blocks will be available for use depending on the modules that have been installed. The Event module provides a block with a monthly calendar, for example, while the Tagadelic module enables blocks to display a tag cloud for each taxonomy. Custom blocks can also be added from the block menu to hold any HTML text or PHP code. These can be used to display brief snippets of information, such as hours a library is open, or even JavaScript to display a widget or badge from a site like Flickr.

Jargon Alert: Blocks, Regions, and Sidebars

Think of blocks as chairs. They are placeholder containers that serve a purpose by being filled by something. Like chairs, though, blocks have to be arranged in an orderly pattern and cannot just be strewn around the library for

(cont.)

Block	Region	Operations
Left sidebar		
+ Navigation*	Left sidebar	configure
Secondary right sidebar		
+ Library Hours*	Secondary right sidebar	configure delete
Before content		
No blocks in this region		
Content		
No blocks in this region		
Featured		
+ Featured Event*	Featured	configure delete
Bottom content		
+ Tag Cloud*	Bottom content	configure delete
+ Powered by Drupal*	Bottom content	configure

Figure 7
Drupal screen for managing blocks.

anyone to trip over. To maintain order on the page, blocks are placed within block regions. Available block regions are defined by each theme installed on a site. Many themes are anchored by sidebars, regions to the right and/or left side of a main content area. Sidebars are no different from other block regions; they are just very common. Other common block regions include header, footer, content top, and content (meaning below the content). Some themes have only a few block regions while others may have more than 20 (see figure 8).

Notes

1. Jeff Eaton, "Content, the Drupal Way," <http://drupal.org/node/176252> (accessed Feb. 15, 2008).
2. Jeff Whatcott, "Dormant Drupal Opportunity?" <http://jeffwhatcott.com/drupal/content/dormant-drupal-opportunity> (accessed Feb. 15, 2008).

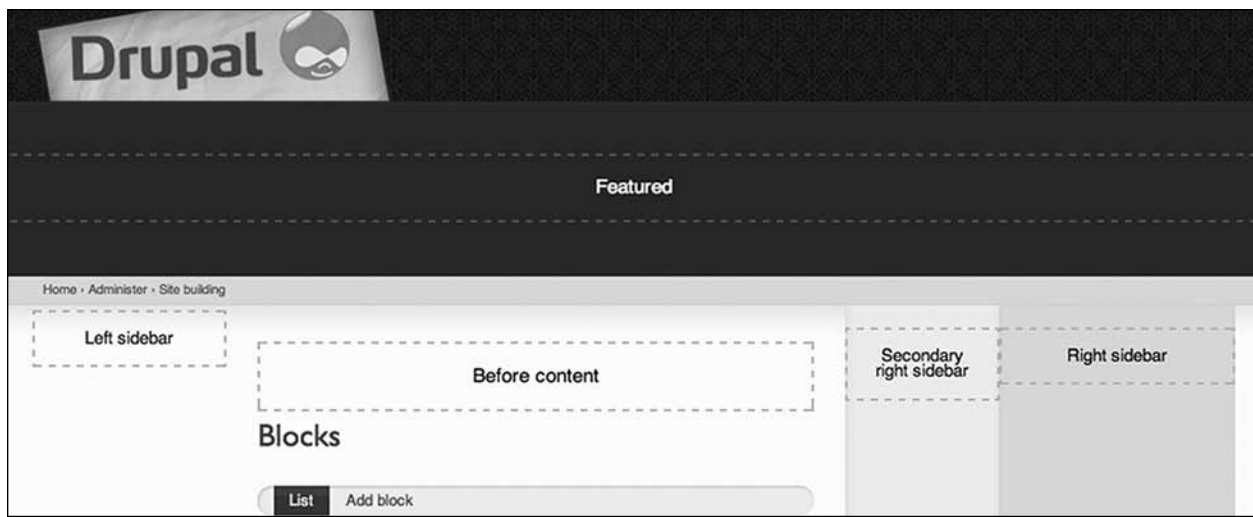


Figure 8
Drupal layout of blocks.