

Drupal 5

Getting Started

Drupal 5: Getting Started

Table of Contents

1. Project and Features	1
Features	1
Mission	3
Principles	3
License	3
CMS vs CMF	3
Drupal Project	3
History	3
Druplicon (the logo)	4
Drupal core	4
Security team	4
Documentation and support	5
2. Before you start	7
Technology Stack	7
Requirements	7
Terminology	8
Best Practices	9
3. Installation	11
Download Drupal	11
Create the database	11
Create database with MySQL	11
Create database with PostgreSQL	12
Run the install script	12
Configure your website	13
Create the first account	14
Basic configuration	15
CRON tasks	21
Multi-site	21
4. Basic Content Management	23
Page and story content types	23
Adding content	24
Adding content with images	25
Manually controlling the teaser location	26
Edit / disable primary links	26
Create a new menu	28
Alias content	30
Set up category	31
5. Administration	33
Site Configuration	33
Administration theme	33
Blog APIs	33
Clean URLs	33
Date and time	33
Distributed authentication	33
Error reporting	34
File system	34
File uploads	34
Image toolkits	34
Input formats	34
Localization	35
Performance	35
Search settings	35
Site information	35
Site maintenance	35
Site registry	35

Throttle	36
Site Building	36
Blocks	36
Contact form	37
Menus	37
Modules	37
Themes	38
URL Aliases	38
User Management	38
Access control	38
Access rules	39
Roles	39
Users	39
User settings	39
Content Management	39
Logs	40
Help	40
6. Upgrading your site	41
7. Concepts	42
Backwards compatibility	42
Content, the Drupal Way	42
Filters	43
Translations	47
Introduction to the Content Construction Kit	49

Chapter 1. Project and Features

Don't panic. - Douglas Adams (The Hitchhikers Guide to the Galaxy)

Drupal is an open source content management platform. Equipped with a powerful blend of features, Drupal can support a variety of websites ranging from personal blogs, corporate brochures and large community-driven websites.

While it is not the purpose of this manual to fully educate you on database, web server or operating systems basics specific to your site, this will hopefully introduce you to getting a Drupal based website installed and configured for your needs. For a more technical sounding definition;

Drupal is a web-based content management system. Text and pointers to other kinds of content are stored in a database, dynamically retrieved and composed, and presented to a user in response to a request sent via a web-browser.

Sound scary? It is a complex software product and can at times seem overwhelming, however, anyone willing to invest a little time and effort can learn how to use, setup, configure and maintain a Drupal based web site.

Features

General features

Collaborative Book - Our unique collaborative book feature lets you setup a "book" and then authorize other individuals to contribute content.

Friendly URLs - Drupal uses Apache's mod_rewrite to enable customizable URLs that are both user and search engine friendly.

Modules - The Drupal community has contributed many modules which provide functionality that extend Drupal core. <http://drupal.org/project/Modules>

Online help - Like many Open Source projects, we can't say that our online help is perfect but have built a robust online help system built into the core help text. Available to you on your own site.

Open source - The source code of Drupal is freely available under the terms of the GNU General Public License 2 (GPL). Unlike proprietary blogging or content management systems, Drupal's feature set is fully available to extend or customize as needed.

Personalization - A robust personalization environment is at the core of Drupal. Both the content and the presentation can be individualized based on user-defined preferences.

Role based permission system - Drupal administrators don't have to tediously setup permissions for each user. Instead, they assign permissions to roles and then group like users into a role group.

Searching - All content in Drupal is fully indexed and searchable at all times if you take advantage of the built in search module.

User management

User authentication - Users can register and authenticate locally or using an external authentication source like OpenID or another Drupal website. For use on an intranet, Drupal can integrate with an LDAP server.

Content management

Polls - Drupal comes with a poll module which enables administrators and/or users to create polls and show them on various pages.

Templating - Drupal's theme system separates content from presentation allowing you to control the look and feel of your Drupal site. Templates are created from standard HTML and PHP coding meaning that you don't have to learn a proprietary templating language.

Threaded comments - Drupal provides a powerful threaded comment model for enabling discussion on published content. Comments are hierarchical as in a newsgroup or forum.

Version control - Drupal's version control system tracks the details of content updates including who changed it, what was changed, the date and time of changes made to your content and more. Version control features provide an option to keep a comment log and enables you to roll-back content to an earlier version.

Blogging

Blogger API support - The Blogger API allows your Drupal site to be updated by many different tools. This includes non-web browser based tools that provide a richer editing environment.

Content syndication - Drupal exports your site's content in RDF/RSS format for others to gather. This lets anyone with a News Aggregator browse your Drupal sites feeds.

News aggregator - Drupal has a powerful built-in News Aggregator for reading and blogging news from other sites. The News Aggregator caches articles to your MySQL database and its caching time is user configurable.

Permalinks - All content created in Drupal has a permanent link or "perma link" associated with it so people can link to it freely without fear of broken links.

Platform

Apache or IIS, Unix / Linux / BSD / Solaris / Windows / Mac OS X support - Drupal was designed from the start to be multi-platform. Not only can you use it with either Apache or Microsoft IIS but we also have Drupal running on Linux, BSD, Solaris, Windows, and Mac OS X platforms.

Database independence - While many of our users run Drupal with MySQL, we knew that MySQL wasn't the solution for everyone. Drupal is built on top of a database abstraction layer that enables you to use Drupal with MySQL and PostgreSQL. Other SQL databases can be supported by writing a supporting database backend and creating a matching SQL database scheme.

Multi-language - Drupal is designed to meet the requirements of an international audience and provides a full framework to create a multi-lingual website, blog, content management system or community application. All text can be translated using a graphical user interface, by importing existing translations, or by integrating with other translation tools such as the GNU gettext.

Administration and analysis

Analysis, Tracking and Statistics - Drupal can print browser-based reports with information about referrals, content popularity and how visitors navigate your site.

Logging and Reporting - All important activities and system events are captured in an event log to be reviewed by an administrator at a later time.

Web based administration - Drupal can be administered entirely using a web browser, making it possible to access it from around the world and requires no additional software to be installed on your computer.

Community features

Discussion forums - Full discussion forum features are built into Drupal to create lively, dynamic community sites.

Performance and scalability

Caching - The caching mechanism eliminates database queries increasing performance and reducing the server's load. Not only can the caching be tuned in real time, while your site is under load, but it has been successfully tested under a "slashdotting" and performed extremely well.

Mission

By building on relevant standards and open source technologies, Drupal supports and enhances the potential of the Internet as a medium where diverse and geographically-separated individuals and groups can collectively produce, discuss, and share information and ideas. With a central interest in and focus on communities and collaboration, Drupal's flexibility allows the collaborative production of online information systems and communities.

Principles

Modular and extensible. Drupal aims to provide a slim, powerful core that can be readily extended through custom modules

Quality coding. High quality, elegant, documented code is a priority over roughed-in functionality.

Standards-based. Drupal supports established and emerging standards. Specific target standards include XHTML and CSS.

Low resource demands. To ensure excellent performance, Drupal puts a premium on low-profile coding (for example, minimizing database queries). Drupal should also have minimal, widely-available server-side software requirements. Specifically, Drupal should be fully operational on a server with Apache web server, PHP, and either MySQL or PostgreSQL.

Open source. Drupal is based on the open source philosophy of collaborative free software development and is licensed under the GPL. Drupal is itself open source and builds on and supports other open source projects. Specifically, Drupal is coded in the open source scripting language PHP and supports as primary data sources the open source database formats MySQL and PostgreSQL.

Ease of use. Drupal aims for a high standard of usability for developers, administrators, and users.

Collaboration. Drupal development supports open, collaborative information sharing systems and approaches.

License

Drupal is licensed under the GNU GENERAL PUBLIC LICENSE Version 2. A copy of this license is packaged with the Drupal download. For more information on GNU Licenses see <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.

CMS vs CMF

Drupal is also a solid Content Management Framework. Frameworks allow site owners to make unique customizations to their site by providing a standardized way to interact with the core components. In order to truly extend a site you may have to delve into some code. You can do this by writing PHP snippets for specific pages or blocks, customizing existing contributed modules to your liking or writing complete modules to do exactly what you want. All this while still leveraging the built in core modules and features of the framework less then 2MB download. Drupal core code is documented and available online at <http://api.drupal.org>

Drupal Project

Drupal is more than software -- it is a project and a community.

History

In 2000, permanent Internet connections were at a premium for University of Antwerp students, so Dries Buytaert and Hans Snijder set up a wireless bridge between their student dorms to share Hans' ADSL modem connection among eight students. While this was an extremely luxurious situation at that time, something was missing: There was no means to discuss or share simple things.

This inspired Dries to work on a small news site with a built-in web board, allowing the group of friends to leave each other notes about the status of the network, to announce where they were having dinner, or to share some noteworthy news items.

The software did not have a name until the day after Dries moved out after graduation. The group decided to put the internal website online so they could stay in touch, keep sharing interesting findings, and narrate snippets of

their personal lives. While looking for a suitable domain name, Dries settled for 'drop.org' after he made a typo to see if the name 'dorp.org' was still available. Dorp is the Dutch word for 'village', which was considered a fitting name for the small community.

Once drop.org was established on the Web, its audience changed as the members began talking about new web technologies, such as moderation, syndication, rating, and distributed authentication. Drop.org slowly turned into a personal experimentation environment, driven by the discussions and flow of ideas. The discussions about these web technologies were tried out on drop.org itself as new additions to the software running the site.

It was only later, in January 2001, that Dries decided to release the software behind drop.org as "Drupal." The purpose was to enable others to use and extend the experimentation platform so that more people could explore new paths for development. The name Drupal, pronounced "droo-puhl," derives from the English pronunciation of the Dutch word "druppel," which means "drop."

Druplicon (the logo)

After Drupal had been created, an obvious matter was the choice and creation of a logo. Of course it would have to do something with a drop... or water.

The initial idea was simple: a drop in a circle. It was featured as an "O" in a liquidish "Drop".

When the community grew, the idea came up of a cartoony drop with a face. Steven Wittens created a 3D drop, but the idea didn't get too far mainly because 3D is hard to print, hard to edit, etc.

When the logo-issue had come up again, Kristjan Jansen came up with idea of putting two side-way drops together to form an infinity-sign. When put into a filled circle, it resembled a face. After some more work by Steven Wittens, the Druplicon was created: a stylised drop with the infinity eyes, a round nose and a mischievous smile.

That's the 'story' behind Druplicon... Dries likes the idea that the infinity-eyes symbolise the infinite possibilities that Drupal offers.

Drupal core

Founder and Lead Developer. Drupal was founded by Dries Buytaert, who retains primary control over the software and makes most decisions on proposed changes. In approving or rejecting proposals and patches, he gives special weight to comments made by people he trusts and respects for their past contributions to Drupal.

Core committers. A small team that reviews final proposed changes to new versions. They are the only ones who have write access to the core CVS repository.

Permanent core committers are Dries Buytaert and Steven Wittens.
Current branch maintainers are Neil Drumm for Drupal 5.x

Maintainers. Though not directly making decisions, maintainers have informal responsibility for a designated portion of the core (for example, a particular core module). Individual areas of responsibility are listed in the file MAINTAINERS.txt included with every download of Drupal. Maintainers are appointed by Dries. Core contributors who have made substantive contributions (particularly to a core component not individually maintained) may apply for Maintainer status by writing to Dries. Dries may also individually invite them.

Core contributor. Core contributors are those who contribute code patches or documentation for the Drupal core. The contributions are peer reviewed and then decided on by Dries or another of the core committers. Anyone can submit patches against Drupal core.

Security team

See <http://drupal.org/security-team> for current information on the Security team

See <http://drupal.org/security> for security announcements and to subscribe.

Goals of the security team

Deal with reported security issues.

Constantly review the code for potential security weaknesses.
Provide assistance for contributed modules' maintainers in dealing with security issues.
Provide documentation on how to write secure code.

How to report a security issue

If you discover or learn about a potential error, weakness or threat that could compromise the security of Drupal, please mail your concern to the Drupal security team: security@drupal.org. Please provide as many details as you can about the environment, Drupal version, modules used, their versions and so on.

If you are required to encrypt your report, use the OpenPGP key: `0xA1FDFAC2`.

How we deal with reported security issues

Review the issue and evaluate the potential impact on all supported releases of Drupal.
If it is indeed a valid problem, the security team is mobilized to eliminate it.
New versions are created and tested.
New packages are created and uploaded to Drupal.org.
We will utilize all available communication channels to make it known that a security issue has been found and fixed, and what steps must be taken by Drupal administrators to protect themselves.

Disclosure policy

Our policy is one of full disclosure; we will never withhold information about a security problem and hope that it won't be discovered by others. However, public announcements will only be made when the threat has been addressed and a secure version of Drupal is available. We ask that when reporting a security issue, you observe these same guidelines, and beyond communicating with the security team, do not share your knowledge of security issues with the public at large.

Which versions are supported?

Not all historical versions of Drupal are actively supported. Only the current release and one previous. Versions of Drupal that are not actively supported will not get security releases. It is therefore not recommended to run unsupported versions of Drupal. Please upgrade so that you will benefit from security releases.
The development branch of Drupal core is not intended for production use and while security problems are fixed, security announcements are not issued. If you are using the development branch for testing or evaluation, we assume that you will update your code regularly.
The security team oversees the security of the code found in the core Drupal distribution. The security of contributed modules and code lies with the individual maintainers. See the process below.

Issues with contributed modules

As soon as we learn about a security issue with a contributed module, we forward that to the maintainer of said module with a deadline. As soon as the maintainer fixed the problem, the security team will issue the relevant advisory with instructions on how to upgrade. However, if the maintainer does not fix the problem within the deadline, an advisory will be issued nonetheless, but we will advise to disable the module and we will disable the project as well.

How to get involved?

The most important help you can provide is reviewing proposed patches with a security mindset. You can also help by reporting issues and working with the team on a fix.

Documentation and support

Documentation and support is collaboratively provided by the Drupal community. Members span all Drupal roles and skill levels. Remember that you are dealing with people so etiquette matters. These people are also volunteers, so your words and actions often determine people's willingness to assist you.

Contributions to the online handbooks (<http://drupal.org/handbooks>) can be done by any registered account on drupal.org. Additional rights to edit the handbooks are granted to those members of the community who express interest and request the rights to so.

Chapter 2. Before you start

The beginning is the most important part of the work - Plato

Drupal is a content management system. It is one component of a website architecture and relies on several other components to work properly and perform at it's best. If you are looking for a web host carefully review the requirements section when evaluating their services.

Tip

Drupal 7 will be developed for PHP 5.2. While this does not apply to Drupal 5 or 6 it would reduce later complication to build new sites using PHP 5.2 if possible.

Technology Stack

Drupal is part of a technology stack. It's performance depends on a lot of factors.

Server - A server is a computer/device which provides information or services to computers on a network.

Operating System - The software that the rest of the computer depends on to work. Unix, Linux, BSD and Windows.

Database - A structured collection of records. Drupal uses a database to store most content for your site.

Webserver - The software component responsible for serving web pages. Examples are Apache and Microsoft IIS.

PHP - The PHP Hypertext Preprocessor is a programming language that allows web developers to create dynamic content that interacts with databases.

Drupal - An open-source platform and content management system for building dynamic web sites offering a broad range of features and services including user administration, publishing workflow, discussion capabilities, news aggregation, metadata functionalities using controlled vocabularies and XML publishing for content sharing purposes. Drupal is generally comprised of a mix of core and contributed modules.

Requirements

Drupal is being developed to be web server independent but we have limited or no reports of successful use on web servers not listed here.

A web server that can execute PHP scripts.

- Recommended: Apache <http://httpd.apache.org>

Drupal will work on Apache 1.3 or Apache 2.x hosted on Unices or Windows. The majority of Drupal development is done using Apache so there is more community experience and testing performed. Optional: You can use the Apache extension `mod_rewrite` to allow for clean URLs.

- Optional: Microsoft Internet Information Server (IIS) <http://www.microsoft.com/iis>

Drupal core will work using IIS5 or IIS6 if PHP is configured correctly. You will need to use a third party solution to achieve Clean URLs. In view of Microsoft's support life cycle it is suggested you use IIS6.

To achieve `clean_url`'s you will need to use a third party product.

PHP version 4.3.3 or higher.

Note

We recommend using the latest version of PHP 4.x. or 5.x for security and future compatibility.

PHP memory of approximately 8MB for a Drupal core installation. In reality you will probably need to use a higher setting depending on your site and contributed modules you are using. A good starting point is 16-24 MB.

PHP XML extension (for blogapi, Drupal, and ping modules). This extension is enabled by default in a standard PHP installation; the windows version of PHP has built-in support for this extension. (<http://www.php.net/manual/en/ref.xml.php>)

PHP needs the following configuration directives for Drupal to work (only directives that differ from the default `php.ini-dist` / `php.ini-recommended`):

- PHP XML extension (for blogapi, Drupal, and ping modules). This extension is enabled by default in a standard PHP installation; the windows version of PHP has built-in support for this extension. (<http://www.php.net/manual/en/ref.xml.php>)
- PHP needs the following configuration directives for Drupal to work (only directives that differ from the default `php.ini-dist` / `php.ini-recommended`):

```
session.save_handler: user http://www.php.net/manual/en/ref.session.php#ini.session.save-handler
recommended: session.cache_limiter: none http://www.php.net/manual/en/function.session-cache-limiter.php#22443
```

- Some of these settings are contained in the default `.htaccess` file that ships with Drupal, so you shouldn't need to set them explicitly. Note, however, that setting PHP configuration options from `.htaccess` only works:

```
with Apache (or a compatible web server)
if the .htaccess file is actually read, i.e. AllowOverride is not None
if PHP is installed as an Apache module.
```

- See the PHP manual for how to change configuration settings for other interfaces to PHP. <http://www.php.net/manual/en/configuration.changes.php#configuration.changes>.

A PHP supported database server

- Recommended: MySQL 4.1 or MySQL 5.0. <http://www.mysql.com>

Drupal will work on v3.23.17 and 4.0 but it is strongly suggested you use 4.1 or 5.0 for future compatibility with Drupal 6 which will drop support for older versions of MySQL.

Important

Drupal makes use of some features not available on some inexpensive hosting plans so please check that your host allows database accounts with the following rights: `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `CREATE`, `DROP`, `INDEX`, `ALTER`, `CREATE TEMPORARY TABLES`, `LOCK TABLES`.

Note

If your system/host is running MySQL 4.1 or newer and you receive the error "Client does not support authentication protocol requested by server", address the problem by following the instructions provided by MySQL AB. There is a minor OS issue with some MySQL 5+ installations primarily on Windows but affecting some Unices as well.

- PostgreSQL, version 7.3 or newer. <http://www.postgresql.org>

Some of the contributed modules are not as abstracted from MySQL specific code as everyone would like. If you are familiar with PostgreSQL please file issues with those contributed modules as you find them.

Currently MS SQL and Oracle are not supported but various efforts are underway to supply schemas. Please see discussions in the Enterprise Group if you are interested in working on this.

Terminology

PHP is an HTML-embedded scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly. (<http://us2.php.net/manual/en/faq.general.php>)

Drupal is built around collections of modules. Drupal's *module* system is based on the concept of "hooks". A hook is a PHP function that is named `foo_bar()`, where "foo" is the name of the module (whose filename is thus `foo.module`) and "bar" is the name of the hook. Each hook has a defined set of parameters and a specified result type. To extend Drupal, a module need simply implement a hook. When Drupal wishes to allow intervention from modules, it determines which modules implement a hook and call that hook in all enabled modules that implement it. More information can be found here (<http://api.drupal.org/api/group/hooks/5>). There are two classes of modules in the Drupal environment;

Core modules refer to the modules that are included. With Drupal core you will always have a data migration path for upgrades to new versions.

Contrib or Contributed modules refer to modules contributed to the Drupal community to extend and enhance your Drupal site. Contributed modules can be downloaded here: <http://drupal.org/project/Modules>. Contributed modules are not part of the official release.

Content in Drupal is created in individual 'nodes'. Almost all content in Drupal is stored as a node. When people refer to "a node" all they mean is a piece of content within Drupal, it could be a poll, a story, a book page, etc. Content that is uploaded to your site (images, files) are stored in your files directory and a pointer in the database points to it.

Blocks are boxes of content that may be rendered into certain regions of your web pages. Blocks can be placed in various content areas. They are usually generated automatically by modules, but administrators can create blocks manually.

Users are the accounts registered on your site.

Roles allow you to fine tune the security and administration of Drupal. A role defines a group of users that have certain privileges as specified in Access Control. By default, Drupal comes with two user roles:

Anonymous user: this role is used for users that don't have a user account or that are not authenticated.

Authenticated user: this role is automatically granted to all logged in users.

PHPTemplate Engine is the theme engine that comes with Drupal core. There are alternative theme engines available but the vast majority of experience, documentation and expertise are invested in the core phpTemplate engine.

Modules are plugins for Drupal that extend its core functionality. Here you can select which modules are enabled. Click on the name of the module in the navigation menu for their individual configuration pages. Once a module is enabled, new permissions might be made available

Menus are a collection of links (menu items) used to navigate a website. The list(s) below display the currently available menus along with their menu items. Select an operation from the list to manage each menu or menu item.

Taxonomy or Categories The taxonomy module allows you to classify content into categories and subcategories; it allows multiple lists of categories for classification (controlled vocabularies) and offers the possibility of creating thesauri controlled vocabularies that indicate the relationship of terms), taxonomies (controlled vocabularies where relationships are indicated hierarchically), and free vocabularies where terms, or tags, are defined during content creation.

Best Practices

There is no site that is so small that it is not important to someone. - Steven Peck

If you are going to invest in the time to setup a CMS, then you should make sure you protect your investment with some simple best practices guidelines. These guidelines are only suggestions on things that should be considered when managing a web site or CMS. It is up to you to decide what is appropriate for you to implement for your sites.

- Learn how to setup a test site.

- There are many programs (XAMMP, MAMP, Apache2Triad) that help you setup a local test system easily. You should set one up to play with your site and it also should allow you to test backups and restores of your production site.
- Subscribe to the Drupal security list and when an update comes out, schedule a test update of your site. <http://drupal.org/security>
- Backup your site. Backup both the database and the files on the webserver. *Test your backups!* If you don't test them, you have no idea if you are doing it right.
- Get involved in the community. Doing this will help you follow development, trends and you may just come across that cool idea that solves your problem while helping others.
- The user id 1 account (uid1) is able to do everything on a Drupal site. Drupal access checks will always allow this account to take an action. Another special permission is that the uid1 account is the only account that can execute the update script. Due to these permissions, it is generally not appropriate to use this account for "day to day" purposes
- Set up some appropriate roles (do not forget to update these roles as you add new modules). Some role suggestions are 'site admin', 'user admin', 'site contributor'. What roles you need to create will depend on the type of site you have designed. Note that anyone who can administer users can grant themselves additional permissions.
- Drupal gives you a great deal of power and flexibility when using PHP code in blocks. Unfortunately, a stray typo character or a missing semi-colon breaks PHP. Drupal then attempts to evaluate this broken code on any requested page, the PHP interpreter chokes on it and therefore your whole site is broken. Fortunately, there is a very simple and easy solution. Instead of writing and testing your code inside the *administer > blocks* page, go to create content and create a new story or page node. Use PHP input format, write the code, and the Preview to debug your code. When you are happy and your code is working, copy and paste the code into the block.
- Do not modify Drupal core files.
 - Doing so will complicate, make difficult or near impossible site updates due to Security and bug fixes
 - You will make it difficult for those that come after to maintain the site
 - You could possibly leave your site vulnerable to exploits

Warning

Are there exceptions to this rule? Sure, but this is generally for specific sites or implementations by people who are extremely familiar with the Drupal code base, development practices and security model. Those who properly document their changes and practice proper revision control with their code. If you have to ask, chances are you shouldn't. If you don't ask, chances are you wouldn't listen to start with.

- Plan for the future. You should plan to revisit your site and evaluate it each time a there is a major version release of Drupal. This does not mean you have to upgrade it, but you should evaluate and plan for an upgrade approximately each 12-24 months.
- Plan your site. Drupal provides a good toolset to help you build your site but you still need to plan your site. Good wireframes and proper planning can avoid significant misunderstandings and problems later.

Chapter 3. Installation

Drupal has an automated installation script that automatically populates database tables and sets the correct settings in settings.php file.

While we recognize that there are a number of graphical tools to do some of these operations we cannot account and document the entire range of possibilities. With that in mind, this manual shall cover the basic methods. Some things such as downloading and moving your files to your hosts directories and the specific mouse clicks for setting up your database through graphical interfaces are better covered with your hosts or tools documentation.

We realize that many people skipped the earlier chapters and jumped right to this section, if you get stuck on a word or term, consider going back to review the earlier pages.

Download Drupal

You can obtain the latest Drupal release from <http://drupal.org/>. The files are in .tar.gz format and can be extracted using most compression tools. On a typical Unix command line, use:

```
wget http://drupal.org/files/projects/drupal-x.x.tar.gz
```

```
tar -zxvf drupal-x.x.tar.gz
```

This will create a new directory drupal-x.x/ containing all Drupal files and directories.

Note

On windows systems you can use a number of compression programs. One such is 7-Zip (<http://www.7-zip.org/>). Extract it to a drupal-x.x.tar file, then extract it again.

Move the contents of that directory into a directory within your web server's document root or your public HTML directory:

```
mv drupal-x.x/* drupal-x.x.htaccess /var/www/html
```

Create the database

Drupal requires access to a database in order to be installed. Your database user will need sufficient privileges to run Drupal. Additional information about privileges, and instructions to create a database using the command line are available in INSTALL.mysql.txt (for MySQL) or INSTALL.pgsql.txt (for PostgreSQL).

Take note of the username, password, database name and hostname as you create the database. You will enter these items in the install script.

Create database with MySQL

This step is only necessary if you don't already have a database set-up (e.g. By your host). In the following examples, 'username' is an example MySQL user which has the CREATE and GRANT privileges. Use the appropriate user name for your system.

First, you must create a new database for your Drupal site here, 'databasename' is the name of the new database):

```
mysqladmin -u username -p create databasename
```

MySQL will prompt for the 'username' database password and then create the initial database files. Next you must login and set the access database rights:

mysql -u username -p

Again, you will be asked for the 'username' database password. At the MySQL prompt, enter following command:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES ON databasename.* TO 'username'@'localhost' IDENTIFIED BY 'password';
```

where

'databasename' is the name of your database

'username@localhost' is the username of your MySQL account

'password' is the password required for that username

Unless your database user has all of the privileges listed above, you will not be able to run Drupal.

If successful, MySQL will reply with:

```
Query OK, 0 rows affected
```

To activate the new permissions, enter the following command:

```
FLUSH PRIVILEGES;
```

the database should be created with UTF-8 (Unicode) encoding.

Create database with PostgreSQL

The database must be created with UTF-8 (Unicode) encoding.

1. CREATE DATABASE USER

This step is only necessary if you don't already have a user setup (e.g. by your host) or you want to create new user for use with Drupal only. The following command creates a new user named "username" and asks for a password for that user:

```
createuser --pwprompt --encrypted --no-adduser --no-createdb username
```

If everything works correctly, you'll see a "CREATE USER" notice.

2. CREATE THE DRUPAL DATABASE

This step is only necessary if you don't already have a database setup (e.g. By your host) or you want to create new database for use with Drupal only. The following command creates a new database named "databasename", which is owned by previously created "username":

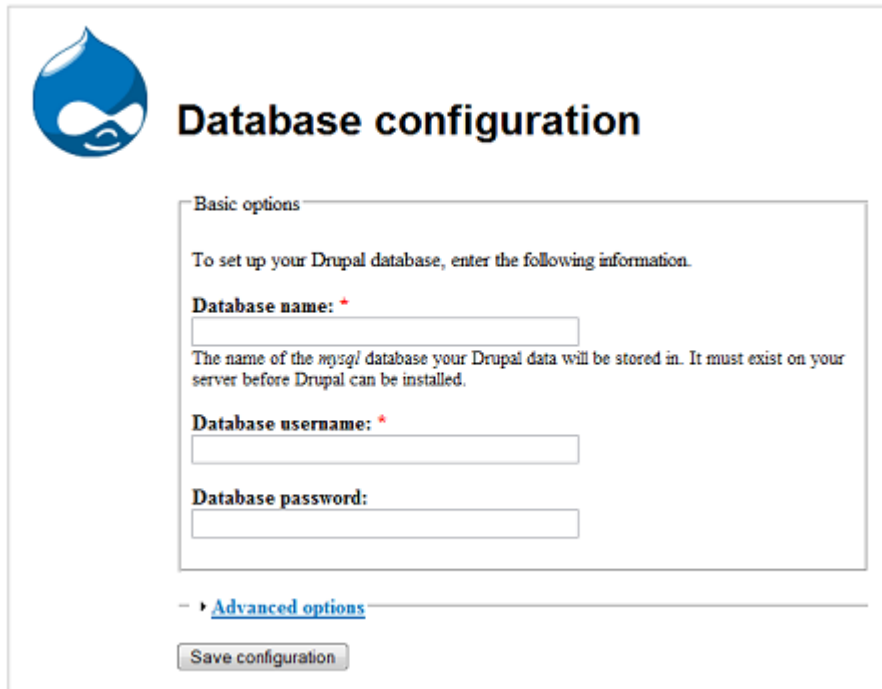
```
createdb --encoding=UNICODE --owner=username databasename
```

If everything works correctly, you'll see a "CREATE DATABASE" notice.

Run the install script

The install script will set the base URL, connect Drupal to the database, and create tables in the database.

To run the install script point your browser to the base URL of your website (i.e. <http://www.example.com>). You will be presented with the "Database Configuration" page.



The screenshot shows the 'Database configuration' screen in Drupal. It features the Drupal logo on the left. The main heading is 'Database configuration'. Below this, there is a section titled 'Basic options' with a sub-heading 'To set up your Drupal database, enter the following information.' This section contains three input fields: 'Database name: *', 'Database username: *', and 'Database password:'. A note below the 'Database name' field states: 'The name of the *mysql* database your Drupal data will be stored in. It must exist on your server before Drupal can be installed.' Below the input fields, there is a link for 'Advanced options' and a 'Save configuration' button.

on success, you will see the Drupal installation complete screen. If there are any error message, review and correct them now.

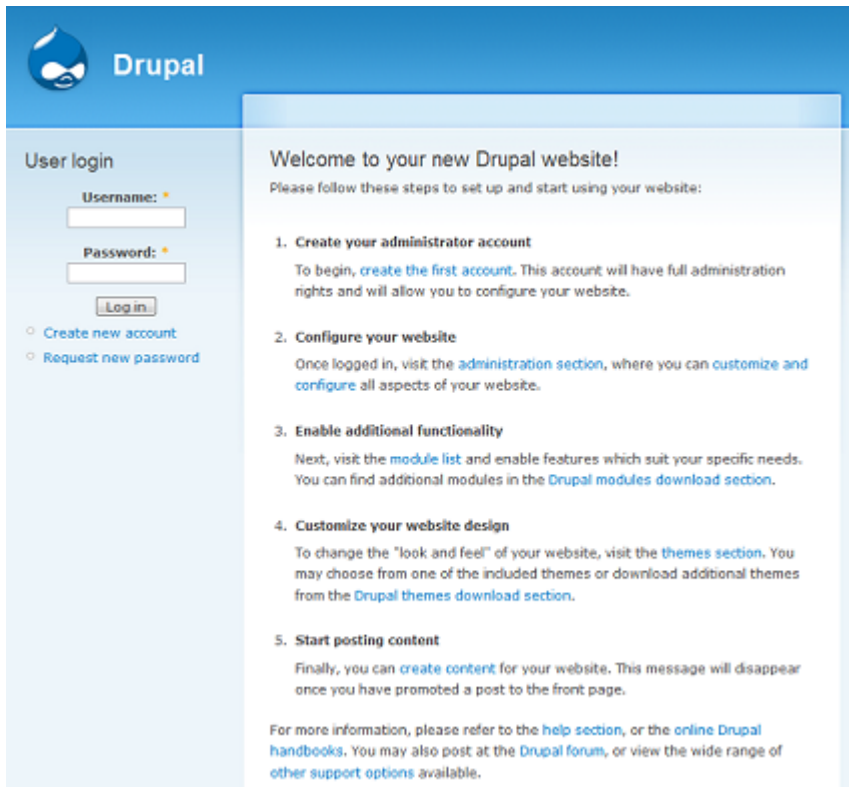


The install script will attempt to write-protect the settings.php after updating it with the information you provide in the installation routine. If you make manual changes to that file later, be sure to protect it again after making your modifications. Failure to remove write permissions to that file is a security risk. The default location for the settings.php file is at sites/default/settings.php. If you are using a multi-site configuration, it may be in another location.

Configure your website

When the install script succeeds, you will be directed to the "Welcome" page. This page outlines, in brief, the basic steps to setup and configure your Drupal site. Take a moment to review them.

These are the basic steps to setup and configure your Drupal site. This message will stay on your front page until you promote something to it (default behavior) or change the front page setting.



At this time you have no configured accounts on your site. The first account you create on step one is the super admin account (you may also see this referred to as UID1). This account will always have access to your entire site so do not share its password. Later, when testing regular account access, you should use a different account.

Create the first account

In "step one" click "create the first account" link. At the user account creation page, enter your details and click 'create new account'.

The image shows the 'User account' creation page. At the top, there are navigation links: 'User account', 'Create new account' (highlighted), 'Log in', and 'Request new password'. Below this, there are two main sections: 'Username' and 'E-mail address'. Each section has a text input field and a small explanatory text below it. The 'Username' section notes that punctuation is not allowed except for periods, hyphens, and underscores. The 'E-mail address' section notes that it is for system emails and will not be public. At the bottom, there is a 'Create new account' button.

This will take you to the user account page for UID1. Don't forget that UID1 is your site's super user. While on the screen you can make note of your randomly generated password, or just enter one you will remember.

The screenshot shows the 'admin' user's account settings page. At the top, there are 'View' and 'Edit' buttons. A green message box contains the following text: 'Welcome to Drupal. You are user #1, which gives you full and immediate access. All future registrants will receive their passwords via e-mail, so please make sure your website e-mail address is set properly under the general settings on the [site information settings page](#). Your password is **db8bpHNU2K**. You may change your password below.'

The 'Account information' section includes the following fields and options:

- Username:** * admin (with a note: 'Your preferred username; punctuation is not allowed except for periods, hyphens, and underscores.')
- E-mail address:** * you@sample.com (with a note: 'A valid e-mail address. All e-mails from the system will be sent to this address. The e-mail address is not made public and will only be used if you wish to receive a new password or wish to receive certain news or notifications by e-mail.')
- Password:** (empty field)
- Confirm password:** (empty field) (with a note: 'To change the current user password, enter the new password in both fields.')
- Status:** Blocked, Active

You may want to set your accounts timezone here as well and click submit.

The screenshot shows the 'admin' user's account settings page after a successful update. At the top, there are 'View' and 'Edit' buttons. A green message box contains the text: 'The changes have been saved.'

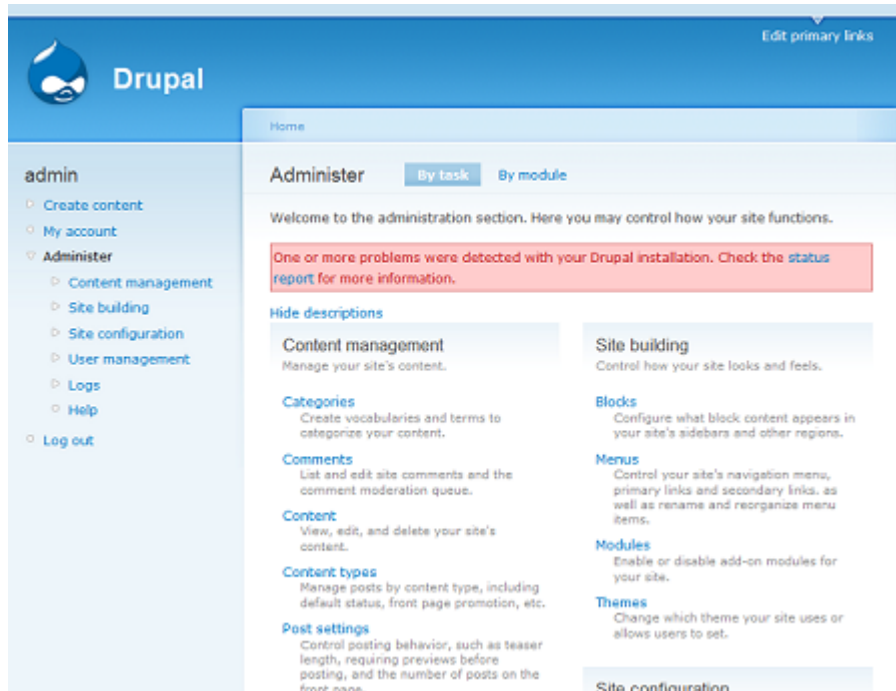
The 'History' section shows:

- Member for 11 min 42 sec

Basic configuration

The next few pages will introduce you to some common administrative tasks in your Drupal site. It is not meant to be a comprehensive configuration guide but an introduction to help familiarize you with where things are located and configured.

Click that Administrator link. This takes you to the administrative section. We will explore more deeply later, for now just an overview on some basic tasks. Note the scary red notice about problems. This is normal at this point and if you follow along we'll show you how to finish configuring your site.



Status Report

The Status Report contains basic information about your site.

Drupal version

Configuration file status. This refers to your sites settings.php file being set to read only

Database schema status. On a new installation this will be 'up to date' but when you install new or updated modules, you should check here. If the database schema is out of date you will need to run /update.php with UID1

File system. The files directory is where all uploaded content in your site is stored. On a new installation this will be an error condition as the files directory will not yet exist

GD Library. If you have the GD library installed on your server

MySQL database (or PostgreSQL) version

PHP version

Unicode library

Web Server

Home > Administer > Logs

Status report

Here you can find a short overview of your Drupal site's parameters as well as any problems detected with your installation. It is useful to copy/paste this information when you need support.

Drupal	5.1
✓ Configuration file	Protected
✗ Cron maintenance tasks	Never run
Cron has not run. It appears cron jobs have not been setup on your system. Please check the help pages for configuring cron jobs. You can run cron manually.	
✓ Database schema	Up to date
✗ File system	Not writable
The directory <code>files</code> does not exist. You may need to set the correct directory at the file system settings page or change the current directory's permissions so that it is writable.	
✓ GD library	bundled (2.0.28 compatible)
✓ MySQL database	4.1.11
✓ PHP	4.3.10
✓ Unicode library	PHP Mbstring Extension
✓ Web server	Apache/2.0.58 (Win32) PHP/4.3.10

At the status report page, you will generally have two errors. One regarding Cron is not configured and the other regarding the file system. Configuring cron will be done later, for now click the run cron manually link to clear the error.

File System

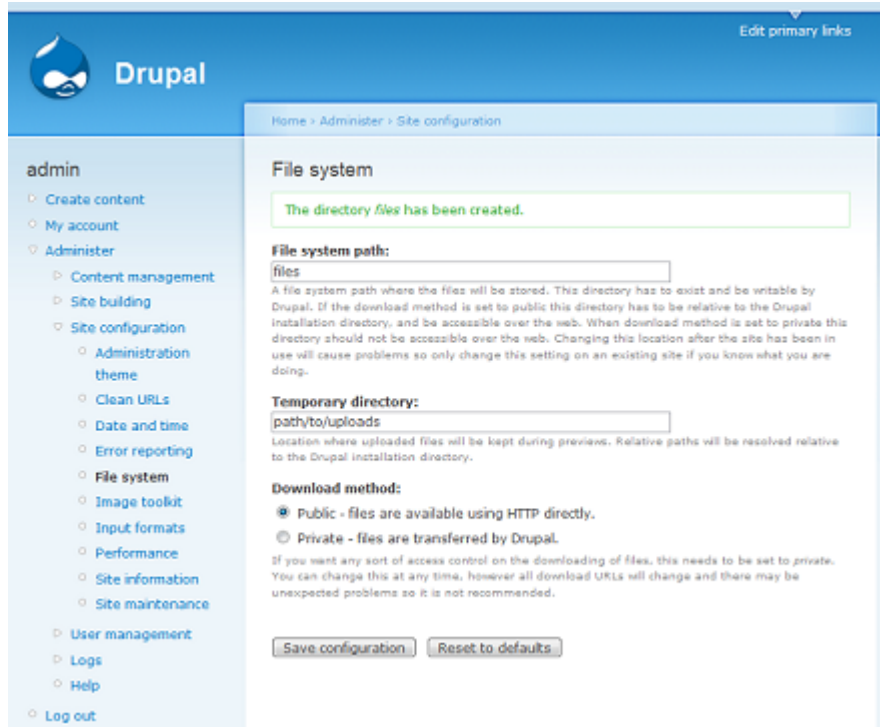
The file system is where all uploaded content is stored. If you use CSS aggregation, then it stores those files there as well. You can do one of two things...

Create it yourself at the root of your site *OR* Click the link in the 'site system settings page' and Drupal will attempt to create the files directory automatically.

Note

In most configurations automatic creation will be fine, in some LAMP configurations this will grant the web server service account rights to the folder and exclude your account direct access. To correct this you will generally need to contact your web host.

If you are still on the Status Report page, you can click the file system settings page link or browse to administer > site configuration > file system manually.



Download method refers to how you control access to your files.

Public - Files can be linked directly and Drupal does not control access.

Private - Drupal controls access to the files. This only works if the /files directory is not in your webroot. The color picker in Garland does not work with if your files are set to Private.

Note

There is a practice that is to put the files directory in your sites directory. This helps centralize the files unique to your site that need to be backed up. For the first site this would be `"/sites/default/files"`. This can be very useful if you plan on using multi-site configurations.

Site Information

Let's change some basic information now. Currently your site name is Drupal and you have the smiling face of Drupicon. While a nice starting point, let's personalize a bit.

In your menu under Administer > Site configuration we will go through some of the available options.

Select Site information;

1. Set the name of your site.
2. Set the email address for your site. (for out going only noreply@sample.com works)
3. Set a slogan for your site
4. The mission statement is optional. The mission statement is available to some themes and only displays on the front page when enabled
5. A footer message if you need one
6. Default name for anonymous users if you allow anonymous user posting
7. Default front page. By default it is set to node. With it set to 'node', it displays, in order, all posts with 'Promote to front page' checked. You can set it to a specific node for a static front page (i.e. node/4)

8. Date and time. Set the default time zone for your site. Review the other options while you are here

Theme Settings

Now that you have your site information a little personalized, let's work a little on the theme. In them menu go to *Administer > Site building* and select Themes.

In a new install, Garland is enabled and the default theme. Click the Configure tab next to List and you will see the Global settings for themes. For now, un-check Logo and check Site Slogan and Save configuration. Take a look at other options available here as well, this is where you can upload your own custom 'favicon' to replace the smiling Druplicon. We like Druplicon but your site should display your identity.

The screenshot shows the 'Themes' configuration page for the 'garland' theme. It features two tabs: 'List' and 'Configure'. Under 'Configure', there are sub-tabs for 'Global settings' and 'garland'. The main content area is titled 'These options control the default display settings for your entire site, across all themes. Unless they have been overridden by a specific theme, these settings will be used.' It is divided into several sections:

- Toggle display:** A list of checkboxes to enable or disable various page elements. 'Logo' is unchecked, while 'Site name', 'Site slogan', 'Mission statement', and 'Shortcut icon' are checked. 'User pictures in posts', 'User pictures in comments', and 'Search box' are unchecked.
- Display post information on:** A section for enabling or disabling the 'submitted by' information. 'Page' is unchecked, and 'Story' is checked.
- Logo image settings:** Includes a checkbox for 'Use the default logo' (checked), a text field for 'Path to custom logo', and an 'Upload logo image' section with a 'Browse' button.
- Shortcut icon settings:** Includes a checkbox for 'Use the default shortcut icon' (checked), a text field for 'Path to custom icon', and an 'Upload icon image' section with a 'Browse' button.

At the bottom of the form, there are two buttons: 'Save configuration' and 'Reset to defaults'.

Next to the Global settings sub tab is a list of enabled themes. On a new install only Garland will be listed. Select Garland now. Assuming you have chosen to use public files earlier, you will see the color selector. You can choose from some existing color sets or choose your own colors using the color picker or HTML color code's. Feel free to experiment.

Themes [List](#) [Configure](#)

Global settings [garland](#)

Color scheme


Color set:

Base color: 

Link color: 

Header top: 

Header bottom: 

Text color: 



Preview



Garland – Fluid Width

Garland

Lorem ipsum dolor

Sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Toggle display

Enable or disable the display of certain page elements.

Logo

Site name

Site slogan

Mission statement

User pictures in posts

User pictures in comments

Search box

Shortcut icon

Logo image settings

If toggled on, the following logo will be displayed.

Use the default logo

Check here if you want the theme to use the logo supplied with it.

Path to custom logo:

The path to the file you would like to use as your logo file instead of the default logo.

Upload logo image:

If you don't have direct file access to the server, use this field to upload your logo.

Shortcut icon settings

Your shortcut icon or Favicon[®] is displayed in the address bar and bookmarks of most browsers.

Use the default shortcut icon.

Check here if you want the theme to use the default shortcut icon.

Path to custom icon:

The path to the image file you would like to use as your custom shortcut icon.

Upload icon image:

If you don't have direct file access to the server, use this field to upload your shortcut icon.

User Management

Under the Administer menu go to user management. Go to User Settings and set user registration as appropriate to your site.

In the User Management menu, select Roles. You will see that there are two default roles.

Anonymous users: These are the rights users not authenticated. (Anonymous users are UID0 in the database)

Authenticated user: this role is automatically granted to all logged in users.

Let's add a role called 'administrator'. Next Edit permissions of the new role and check all the boxes except Use PHP for block visibility and administer filters. Save permissions.

Warning

Any rights that allow for PHP or administering filters should be given with extreme care.

Go to Users and select the Add user tab. Choose a user name and fill in the information. Do not assign a role at this time. Log out and log back in with your new account. See what the menu looks like. When you are done, log back with your first user account, add the administer role to your test account and repeat. See what's different.

Don't get hung up on the user id. number with two exceptions, it's just a sequential number that the account was created on. The only important ones are;

UID0 - anonymous users.

UID1 - first account and super user. In general, don't tie this account to a specific individual.

Note

When testing access to your site or it's features user a different account then UID1. UID1 has access to every part of your site at all times and testing with it can disguise various role/permissions issues.

CRON tasks

Many Drupal modules (such as the search functionality and maintenance to clear log files and cache) have periodic tasks that must be triggered by a scheduled job. To activate these tasks, call the cron.php page by visiting <http://www.example.com/cron.php> --this will pass control to the modules and the modules will decide if and what they must do.

Most UNIX based systems support the crontab utility for scheduling tasks like this. See cron MAN pages for more specific or additional configuration options. The following example crontab line will activate the cron tasks automatically on the hour:

```
0 * * * * wget -O - -q http://www.example.com/cron.php
```

Most ISP's will have instructions on how they allow access to configure cron tasks. Windows operating systems do not use cron, they use the Windows scheduler service.

Multi-site

Multi-site is a very powerful feature in large sites or hosting operations but is often mis-understood. Multi-site is designed as a way to share the same code base for multiple different Drupal based sites. These sites are separate and do not share content or administration. Multi-site just helps with managing the code base, shared themes and modules.

A single Drupal code base installation can host several Drupal-powered sites, each with its own individual configuration and database.

Additional site configurations are created in subdirectories within the 'sites' directory. Each subdirectory must have a 'settings.php' file which specifies the configuration settings. The easiest way to create additional sites is to

copy the 'default' directory and modify the 'settings.php' file as appropriate. The new directory name is constructed from the site's URL. The configuration for `www.example.com` could be in `'sites/example.com/settings.php'` (note that `'www.'` should be omitted if users can access your site at `http://example.com/`).

Sites do not have to have a different domain. You can also use sub domains and subdirectories for Drupal sites. For example, `example.com`, `sub.example.com`, and `sub.example.com/site3` can all be defined as independent Drupal sites. The setup for a configuration such as this would look like the following:

```
sites/default/settings.php
sites/example.com/settings.php
sites/example.com.site3/settings.php
sites/sub.example.com/settings.php
sites/sub.example.com.site3/settings.php
sites/www.sub.example.com.site3/settings.php
sites/www.sub.example.com/settings.php
```

If you are installing on a non-standard port, the port number is treated as the deepest sub domain. For example: `http://www.example.com:8080/` could be loaded from `sites/8080.www.example.com/`. The port number will be removed according to the pattern above if no port-specific configuration is found, just like a real sub domain.

Each site configuration can have its own site-specific modules and themes in addition to those installed in the standard 'modules' and 'themes' directories. To use site-specific modules or themes, simply create a 'modules' or 'themes' directory within the site configuration directory. For example, if `sub.example.com` has a custom theme and a custom module that should not be accessible to other sites, the setup would look like this:

```
sites/sub.example.com/settings.php
sites/sub.example.com/themes/custom_theme
sites/sub.example.com/modules/custom_module
```

Chapter 4. Basic Content Management

Now that your installation is running, here are some basic instructions about managing content.

Page and story content types

Drupal comes with two preconfigured content types enabled by default, "Page" and "Story". In previous versions of Drupal these were actually separate modules, currently they are merely starting content types with different names and default configurations. They can be renamed or deleted. By default:

A page doesn't post author information, timestamps or comments by default.

A story does post author information, timestamps or comments by default and is set to automatically promote all posts to the front page.

As they are processed as two different node types, they can have different editing permissions and even be themed differently. These defaults serve as preconfigured examples and can be changed. To see these settings look in the Content Management section of the Administer page.

The description for Content types is "Manage posts by content type, including default status, front page promotion, etc." Click on *Content types*

Name	Type	Description	Operations
Page	page	If you want to add a static page, like a contact page or an about page, use a page.	edit delete
Story	story	Stories are articles in their simplest form: they have a title, a teaser and a body, but can be extended by other modules. The teaser is part of the body too. Stories may be used as a personal blog or for news articles.	edit delete

On the Content types page you will find a listing of the available content types for your site. This list depends on what modules you have enabled and if you have created any additional content types. For now click on the *edit* link in the *Operations* column for *Page* content type.

There are three areas on this page. Identification, Submission form and Workflow.

For now scroll down to the bottom. In a new Drupal install you will have Default options, default comment settings and attachments. These settings control the default behavior of the content type for users without special permission.

Default options:

- Published
- Promoted to front page
- Sticky at top of lists
- Create new revision

Users with the administer nodes permission will be able to override these options.

Default comment setting:

- Disabled
- Read only
- Read/Write

Users with the administer comments permission will be able to override this setting.

Default options:

Published - This determines if site visitors can see this content.

Promoted to front page - If your front page is set to /node (default Drupal behavior) then this will post the content to the front page.

Sticky at top of lists - Posts will stay at top of lists. Normal behavior is most recent date first.

Create new revision - This will save a copy of each edit to a given piece of content.

Default comment settings control whether you allow comments on a given content type. Requires that the comment module be enabled.

Various contributed modules can add other options here.

Adding content

In this example you will learn how to add content to a Drupal site.

For the very basics of adding content.

1. Select *Create content*
2. Select *Story*
3. Submit story form

Submit Story

Title: *

Body:

[Input format](#)

Log message:

An explanation of the additions or updates being made to help other authors understand your motivations.

4. In the Title text box enter something, Test story title
5. In the Body area enter some text.
6. Test story content

The screenshot shows a content editor interface. At the top, there's a title field with the text 'Test story title' and two buttons: 'View' and 'Edit'. Below the title is a 'Title:' label and a text input field containing 'Test story title'. Underneath is a 'Body:' label and a large text area containing Lorem Ipsum text. Below the body are several expandable sections: 'Input format', 'Log message:' (with an empty text area), 'Menu settings', 'Comment settings', 'Authoring information', and 'Publishing options'. At the bottom of the form are three buttons: 'Preview', 'Submit', and 'Delete'.

7. Click the *Submit* button
8. You should now see your post.
9. Test story post

Adding content with images

With Drupal you can add images to your content. How it is displayed still depends on the theme you are using and the CSS included handles the IMG tag.

1. Enable upload module
2. Set your role as allowed to upload files
3. Check your space and attachment settings *administer > site configuration > file uploads*
4. Enable your content type for uploads
5. Create content
6. Attach file - uncheck List file, copy the file path under the box once it's uploaded.

Now you can use `` in the content area.

Note

Assumes your filter allows the use of ``

Manually controlling the teaser location

Teaser settings (configured in Post settings) are global for all content created after the value is set. If you wish to change the teaser length of existing content, you will need to edit each node and re-save it for the change to be reflected.

In order to manually set where break point is in your post, you must use the `<!--break-->` tag before the sites trimmed post character count is reached. Everything above this tag will appear in the teaser.

Note

Make sure there are no spaces in the tag or it will be ignored. You can place the tag in the middle of a paragraph. The full text will not notice the tag so the paragraph will look normal, but the paragraph will be cut short in the teaser

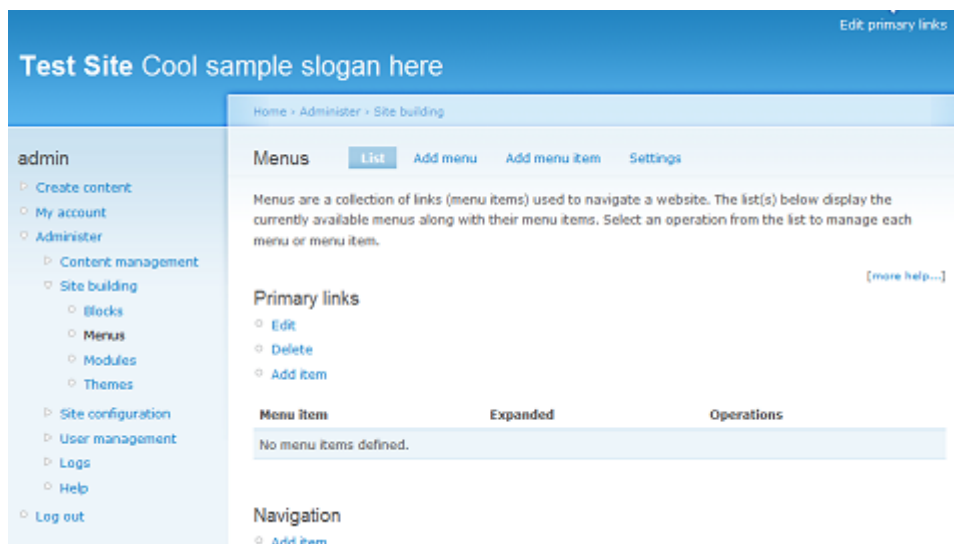
Edit / disable primary links

In this example you will learn how and where to modify the 'Edit primary links' on a new Drupal install

On new sites you have this link in the upper right hand corner that says *Edit primary links*

To add an item to the Primary links menu

1. Click on Edit primary links. This takes you directly to the menu page. You can see the navigation path on the left *Administrator > Site building > Menus*



2. Under the Primary links menu, click on Add item link. This will take you to the Add menu item window.
3. Enter the Title, Description (you see this on hover) and Path. For this example we'll use node/3 which is one of the test posts you created earlier if you were following the examples.

Menus List Add menu Add menu item Settings

Title: *
Test menu item
The name of the menu item.

Description:
Test menu item
The description displayed when hovering over a menu item.

Path: *
node/3
The path this menu item links to. This can be an internal Drupal path such as `node/add` or an external URL such as `http://drupal.org`. Enter `<front>` to link to the front page.

Expanded
If selected and this menu item has children, the menu will always appear expanded.

Parent item:
Primary links

Weight:
0
Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer the top.

Submit

4. Select Submit. You should see something like this.

Test menu item

Test Site Cool sample slogan here

Home > Administer > Site building

Menus List Add menu Add menu item Settings

Menus are a collection of links (menu items) used to navigate a website. The list(s) below display the currently available menus along with their menu items. Select an operation from the list to manage each menu or menu item. [\[more help...\]](#)

Primary links

- Edit
- Delete
- Add item

Menu item	Expanded	Operations
Test menu item		edit disable delete

5. If you click on `Test menu item` you will go to that contents page.

To disable *'Edit primary links'*

1. Choose the Settings tab, you should this page.

Menus [List](#) [Add menu](#) [Add menu item](#) **Settings**

Primary and secondary links settings

Primary and secondary links provide a navigational menu system which usually (depending on your theme) appears at the top-right of the browser window. The links displayed can be generated either from a custom list created via the [menu administration](#) page or from a built-in list of menu items such as the navigation menu links.

Menu containing primary links:

Menu containing secondary links:

If you select the same menu as primary links then secondary links will display the appropriate second level of your navigation hierarchy.

Content authoring form settings

The menu module allows on-the-fly creation of menu links in the content authoring forms. The following option limits the menus in which a new link may be added. E.g., this can be used to force new menu items to be created in the primary links menu or to hide admin menu items.

Restrict parent items to:

Choose the menu to be made available in the content authoring form. Only this menu item and its children will be shown.

2. Menu settings tab

Menus [List](#) [Add menu](#) **Add menu item** [Settings](#)

Title: *

The name of the menu item.

Description:

The description displayed when hovering over a menu item.

Path: *

The path this menu item links to. This can be an internal Drupal path such as `node/add` or an external URL such as `http://drupal.org`. Enter `<front>` to link to the front page.

Expanded
If selected and this menu item has children, the menu will always appear expanded.

Parent item:

Weight:

Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer the top.

3. In *Menu containing primary links* select *No primary links* and submit. That will disable it.

Create a new menu

In this example we will create a new menu 'Example menu', add some links and place it on the left side of our site.

1. Navigate to *administer > site building > menus*

2. Select the *Add menu* tab item
3. For the title enter *Example menu* and select to return to the menu list view

Menus List **Add menu** Add menu item Settings

Enter the name for your new menu. Remember to enable the newly created block in the [blocks administration page](#).

[more help...]

Title: *
Example menu
The name of the menu.

Submit

Tip

See the *more help...* link? Don't forget that much of Drupal has contextual help built in.

4. Scroll to the Example menu you just created and select *Add item*

Example menu

- Edit
- Delete
- Add item

Menu item	Expanded	Operations
No menu items defined.		

5. Enter a title and description. For the path enter *node/3*. The parent item should be our Example menu and then select submit. Add a second item to your menu.

Menus List Add menu **Add menu item** Settings

Title: *
Test node 3
The name of the menu item.

Description:
Test menu item
The description displayed when hovering over a menu item.

Path: *
node/3
The path this menu item links to. This can be an internal Drupal path such as `node/add` or an external URL such as `http://drupal.org`. Enter `<front>` to link to the front page.

Expanded
If selected and this menu item has children, the menu will always appear expanded.

Parent item:
Example menu

Weight:
0
Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer the top.

Submit

6. You should now have an Example menu with at least two items

Example menu

- Edit
- Delete
- Add item

Menu item	Expanded	Operations		
Test node 3		edit	disable	delete
Test node 4		edit	disable	delete

7. So now you have a new menu. Menu's generate blocks. So we need to go to the blocks menu to make it visible.
8. Navigate to *Administer > Site building > Blocks*
9. Blocks are set per theme, not globally so make sure the theme you are using is selected.
10. In the region drop down next to the Example menu, select left sidebar and then select Save blocks

[\[more help...\]](#)

Block	Region	Weight	Operations
Left sidebar			
User login	left sidebar	0	configure
Navigation	left sidebar	0	configure
Disabled			
Example menu	<none>	0	configure
Primary links	<none>	0	configure
Recent comments	right sidebar	0	configure
Syndicate	content	0	configure
Who's new	header	0	configure
Who's online	footer	0	configure

11. Your menu should now be in your left sidebar.

Experiment with moving it around and see what happens. Click the *configure* option next to the block to explore various visibility options and don't forget the *more help...* link in the blocks section.

Alias content

In this example you will learn how to activate a module, check for permissions and alias the URL path.

By default, a typical Drupal URL will be `www.example.com/?q=node/3` or `www.example.com/node/3`. While this is useful, Drupal core comes with a module that can make it nicer.

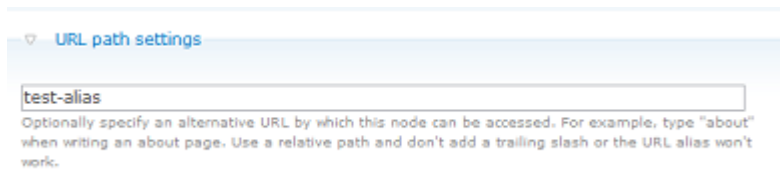
1. Navigate to *Administer > Site building > Modules*
2. In the core options module list locate Path module.
Note the description "Allows users to rename URLs."
3. Select the box next to it and select save configuration on the bottom.
4. Navigate to *Administer > User management > Access control*

Note

Many modules add features which often will need to be assigned to your sites roles. By default new features are not enabled for any role. UID1 has access to all your sites features.

5. Scroll down to the path module entry. You will see two new permissions
administer URL aliases - this allows an administrator to edit/update aliases
create URL aliases - this allows users to alias nodes
6. Choose the role you wish to grant this ability to and then select *save permissions*
7. You can either create new content or edit a previously created node.
8. Scroll down on the edit page and find the collapsed fieldset *URL path settings*.
9. Enter an alias for your content and submit your page.

For this example we'll use test-alias



10. You should now be able to browse to your content at www.example.com/test-alias
11. Browse to Administer > Site building > URL aliases.

This is where you can edit and add multiple aliases to content. Select the *more help...* link for some examples of how to alias other Drupal URLs.

Set up category

In this example you will learn how to set up a basic category and assign it to a specific content type.

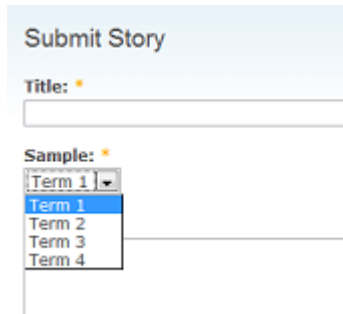
The taxonomy module is one of the most popular features because users often want to create categories to organize content by type. A simple example would be organizing a list of music reviews by musical genre. Taxonomy what? In the menu UI it was renamed to Categories.

1. Navigate to *Administer > Content management > Categories*
2. Select the *Add vocabulary* tab
3. In Vocabulary name enter a name (Sample) and a description (This is a sample category)
4. Scroll down and in Types select Story
5. Under Hierarchy select disabled
6. For this example we will also select *Required*.
7. Select *Submit* and you will be taken back to the Categories page.
8. Select *add term*
9. In the List terms page fill out the values and select submit

Term name: Term 1

Description: Term 1 description

10. Create 3 to 4 different terms
11. Navigate to *Create content > Story*
12. Between the Title and the Body of the story you will have the categories. The red asterix indicates mandatory.



The image shows a screenshot of a web form titled "Submit Story". It features a "Title:" label with a red asterisk indicating it is mandatory, followed by an empty text input field. Below this is a "Sample:" label, also with a red asterisk, followed by a dropdown menu. The dropdown menu is open, showing four options: "Term 1", "Term 2", "Term 3", and "Term 4". The "Term 1" option is currently selected and highlighted in blue.

Now that you have covered the basics, play around with categories to see how they suit your site.

Chapter 5. Administration

A new installation of Drupal defaults to a very basic configuration with only a few active modules and minimal user access rights. This is a reference overview of the various administrative options in your Drupal installation.

The administration section is by default task based and organized into five major areas with an additional link to help in the menu.

Content management: Manage your sites content.

User management: Manage your site's user, groups and access to site features.

Logs: View system logs and other status information.

Site building: Control how your site looks and feels.

Site configuration: Adjust basic site configuration.

Help: Online help with a glossary , module configuration instructions and links to module settings pages. The core Help module must be on for you to see this information. Help information is only displayed for modules that are activated.

Options available in the next several pages depend on modules activated.

Site Configuration

The site configuration area will be primarily of use when first setting up your site and rarely visited afterwards. Some contributed modules will add one or few time configuration options in this section.

Administration theme

Choose which theme the administration pages should display in. If you choose "System default" the administration pages will use the same theme as the rest of the site.

This can be useful if your sites theme is not flexible enough (to narrow, etc) to accommodate the administration areas of your site or you don't wish to theme those areas for a custom theme.

Blog APIs

The blog API module enables a post to be posted to a site via external GUI applications. Here is where you enable your content types to be accessible via these external tools.

Clean URLs

This option makes Drupal emit "clean" URLs (i.e. Without ?q= in the URL.) In order to enable this, you will need to run the clean URL test.

Date and time

Select the default site time zone. Set whether your site allows for user configurable time zones and the date time format.

Distributed authentication

Using this your site can "call home" to another Drupal server. By calling home to drupal.org and sending a list of your installed modules and themes, you help rank projects on drupal.org and so assist all Drupal administrators to find the best components for meeting their needs.

If you want to register with a different server, you can change the Drupal XML-RPC server setting -- but the server has to be able to handle Drupal XML. Some XML-RPC servers may present directories of all registered sites.

To get all your site information listed, go to the site information settings page and set the site name, the e-mail address, the slogan, and the mission statement

Authentication service. If enabled, your Drupal site will accept logins with the user names of other Drupal sites, and likewise provide authentication for users logging into other Drupal sites, based on their user accounts here.

Error reporting

Here is where you configure your sites error handling. At it's basic you can create and set a node for a custom 403 or 404 errors rather than have your sites users encounter a Page not found.

Here also is where you set Error reporting behavior and how long log entries are kept for. Error reporting is where Drupal, PHP and SQL errors are logged. On a production server it is recommended that errors are only written to the error log. On a test server it can be helpful to write logs to the screen until deployed.

File system

A file system path where the files will be stored. This directory has to exist and be writable by Drupal. If the download method is set to public this directory has to be relative to the Drupal installation directory, and be accessible over the web. When download method is set to private this directory should not be accessible over the web. Changing this location after the site has been in use will cause problems so only change this setting on an existing site if you know what you are doing.

By default Drupal will automatically attempt to create a /files directory at the root of your site if it doesn't exist. In most configurations automatic creation will be fine, in some LAMP configurations this will grant the web server service account rights to the folder and exclude your account direct access. To correct this you will generally need to contact your web host.

There exists a configuration practice that is to put the files directory in your sites directory. This helps centralize the files unique to your site that need to be backed up. For the first site this would be "/sites/default/files". This can be very useful if you plan on using multi-site configurations or wish to simplify your sites backups.

File uploads

Basic file handling for your site.

Maximum resolution for uploaded images: The maximum allowed image size (e.g. 640x480). Set to 0 for no restriction, and whether they are listed by default. What extensions are allowed. (Default: jpg jpeg gif png txt html doc xls pdf ppt pps odt ods odp).

You also need to set a maximum file size per upload and total file size per user. Defaults are 1 MB each.

Note

The settings in your php.ini file limit the actual maximum file size per upload.

Image toolkits

Drupal can use different image toolkits. The commonest one available is the GD2 toolkit. The default settings are for JPEG quality of 75%.

Drupal provides for adding additional image processing libraries. These come in the form of inc files that you copy to the /includes files. One for Image Magic can be found with the contributed image module.

Input formats

Input formats define a way of processing user-supplied text in Drupal. Every input format has its own settings of which filters to apply. Possible filters include stripping out malicious HTML and making URLs clickable.

Users can choose between the available input formats when submitting content.

Below you can configure which input formats are available to which roles, as well as choose a default input format (used for imported content, for example).

Note

(1) the default format is always available to all roles, and (2) all filter formats can always be used by roles with the "administer filters" permission even if they are not explicitly listed in the Roles column of this table

Localization

Drupal provides support for the translation of its interface text into different languages. This page provides an overview of the installed languages. You can add a language on the add language page, or directly by importing a translation. If multiple languages are enabled, registered users will be able to set their preferred language. The site default will be used for anonymous visitors and for users without their own settings.

Drupal interface translations may be added or extended by several courses: by importing an existing translation, by translating everything from scratch, or by a combination of these approaches.

Performance

Enabling the cache will offer a significant performance boost. Drupal can store and send compressed cached pages requested by anonymous users. By caching a web page, Drupal does not have to construct the page each time someone wants to view it.

Search settings

The search engine works by maintaining an index of the words in your site's content. You can adjust the settings below to tweak the indexing behaviour. Note that the search requires cron to be set up correctly.

Site is indexed per cron run and can be throttled to reduce impact on the site.

Site information

Site name, default email address slogan footer name for anonymous users and the default front page is set here.

Site maintenance

When set to "Online", all visitors will be able to browse your site normally. When set to "Off-line", only users with the "administer site configuration" permission will be able to access your site to perform maintenance; all other visitors will see the site off-line message configured below. Authorized users can log in during "Off-line" mode directly via the user login page.

To take it out of maintenance mode, log in at `sample.com/?q=user` with the first user that was created for your site (UID1)

Site registry

Register with a Drupal server. If enabled, your Drupal site will register itself with the specified Drupal XML-RPC server. For this to work properly, you must set your site's name, e-mail address, slogan and mission statement. When the Drupal XML-RPC server field is set to `http://drupal.org/xmlrpc.php`, your web site will register itself with drupal.org. Requires the cron feature to be enabled.

Send system information. If enabled, your site will send information on its installed components (modules, themes, and theme engines). This information can help in compiling statistics on usage of Drupal projects.

Send statistics. If enabled, your site will send summary statistics on the number of registered users and the total number of posts. No private information will be sent. These data help to improve the ranking statistics of Drupal projects.

Allow other Drupal sites to register. If enabled, your Drupal site will allow other sites to register with your site and send information to this site. This functionality can be used to maintain a list of related sites.

Throttle

If your site gets linked to by a popular website, or otherwise comes under a "Denial of Service" (DoS) attack, your webserver might become overwhelmed. This module provides a congestion control throttling mechanism for automatically detecting a surge in incoming traffic. This mechanism is utilized by other Drupal modules to automatically optimize their performance by temporarily disabling CPU-intensive functionality.

Auto-throttle on anonymous users The congestion control throttle can be automatically enabled when the number of anonymous users currently visiting your site exceeds the specified threshold. For example, to start the throttle when your site has 250 anonymous users online at once, enter '250' in this field. Leave this value blank or set to "0" if you do not wish to auto-throttle on anonymous users. You can inspect the current number of anonymous users using the "Who's online" block.

Auto-throttle on authenticated users The congestion control throttle can be automatically enabled when the number of authenticated users currently visiting your site exceeds the specified threshold. For example, to start the throttle when your site has 50 registered users online at once, enter '50' in this field. Leave this value blank or set to "0" if you do not wish to auto-throttle on authenticated users. You can inspect the current number of authenticated users using the "Who's online" block.

Auto-throttle probability limiter. The auto-throttle probability limiter is an efficiency mechanism to statistically reduce the overhead of the auto-throttle. The limiter is expressed as a percentage of page views, so for example if set to the default of 10% we only perform the extra database queries to update the throttle status 1 out of every 10 page views. The busier your site, the lower you should set the limiter value.

Site Building

The Site Building section will primarily be used for structural configuration of your site. Blocks location, visibility, which theme you are using or have made available, modules installed and whether you are using URL Aliasing.

Blocks

Blocks are boxes of content that may be rendered into certain regions of your web pages, for example, into sidebars. Blocks are usually generated automatically by modules (e.g., Recent Forum Topics), but administrators can also define custom blocks.

The region each block appears in depends on both which theme you are using (some themes allow greater control over block placement than others), and on the settings in the block administration section.

The block administration screen lets you specify the vertical placement of the blocks within a region. You do this by assigning a weight to each block. Lighter blocks (those having a smaller weight) "float up" towards the top of the region; heavier ones "sink".

A block's visibility depends on:

Its region placement. Blocks with no region assigned to them are never shown.

Its throttle check box when throttle module is enabled. Throttled blocks are hidden during high server loads.

Its page visibility settings. Blocks can be configured to be visible/hidden on certain pages.

Its custom visibility settings. Blocks can be configured to be visible only when specific conditions are true.

Its user visibility settings. Administrators can choose to let users decide whether to show/hide certain blocks.

Its user-role visibility settings. Administrators can choose to let blocks be visible only for certain user roles.

Its function. Some dynamic blocks, such as those generated by modules, will be displayed only on certain pages.

Module blocks. Some modules generate blocks that become available when the modules are enabled. These blocks can be administered via the blocks administration page.

Administrator defined blocks. Administrators can also define custom blocks. These blocks consist of a title, a description, and a body which can be as long as you wish. Block content can be in any of the input formats supported for other content.

Contact form

The contact module enables the use of both personal and site-wide contact forms, thereby facilitating easy communication within the community. While personal contact forms allow users to contact each other by e-mail, site-wide forms allow community members to contact the site administration from a central location. Users can specify a subject and message in the contact form, and also request that a copy of the e-mail be sent to their own address.

Users can activate/deactivate their personal contact forms in their account settings. Upon activation, a contact tab will appear in their user profiles. Privileged users such as site administrators are able to contact users even if they have chosen not to enable this feature.

Note that the contact tab will not appear when a user views his or her own profile; only when viewing another user's profile, if that user's contact form is enabled.

If the menu module is enabled, a menu item linking to the site-wide contact page is added to the navigation block. It is disabled by default, but can be enabled via the menu management page. Links to the contact page may also be added to the primary and secondary links using the same page.

Menus

Menus are a collection of links (menu items) used to navigate a website. The menu module provides an interface to control and customize the powerful menu system that comes with Drupal. Menus are primarily displayed as a hierarchical list of links using Drupal's highly flexible blocks feature. Each menu automatically creates a block of the same name. By default, new menu items are placed inside a built-in menu labelled Navigation, but administrators can also create custom menus.

Drupal themes generally provide out-of-the-box support for two menus commonly labelled primary links and secondary links. These are sets of links which are usually displayed in the header or footer of each page (depending on the currently active theme). Any menu can be designated as the primary or secondary links menu via the menu settings page.

Menu administration tabs:

- On the administer menu page, administrators can "edit" to change the title, description, parent or weight of a menu item. Under the "operations" column, click on "enable/disable" to toggle a menu item on or off. Only menu items which are enabled are displayed in the corresponding menu block. Note that the default menu items generated by the menu module cannot be deleted, only disabled.
- Use the "add menu" tab to submit a title for a new custom menu. Once submitted, the menu will appear in a list toward the bottom of the administer menu page underneath the main navigation menu. Under the menu name there will be links to edit or delete the menu, and a link to add new items to the menu.
- Use the "add menu item" tab to create new links in either the navigation or a custom menu (such as a primary/secondary links menu). Select the parent item to place the new link within an existing menu structure. For top level menu items, choose the name of the menu in which the link is to be added.

Modules

Modules are plugins for Drupal that extend its core functionality. Here you can select which modules are enabled. Click on the name of the module in the navigation menu for their individual configuration pages. Once a module is enabled, new permissions might be made available. Modules can automatically be temporarily disabled to reduce

server load when your site becomes extremely busy by enabling the throttle.module and checking throttle. The auto-throttle functionality must be enabled on the throttle configuration page after having enabled the throttle module.

It is important that update.php is run every time a module is updated to a newer version.

Themes

Select which themes are available to your users and specify the default theme. To configure site-wide display settings, click the "configure" task. Alternately, to override these settings in a specific theme, click the "configure" link for the corresponding theme. Note that different themes may have different regions available for rendering content like blocks. If you want consistency in what your users see, you may wish to enable only one theme.

URL Aliases

The path module allows you to specify aliases for Drupal URLs. Such aliases improve readability of URLs for your users and may help Internet search engines to index your content more effectively. More than one alias may be created for a given page.

Some examples of URL aliases are:

```
user/login => login
```

```
image/tid/16 => store
```

```
taxonomy/term/7+19+20+21 => store/products/whirlygigs
```

```
node/3 => contact
```

The path module enables an extra field for aliases in all node input and editing forms (when users have the appropriate permissions). It also provides an interface to view and edit all URL aliases. The two permissions related to URL aliasing are "administer URL aliases" and "create URL aliases".

This module also comes with user-defined mass URL aliasing capabilities, which is useful if you wish to uniformly use URLs different from the default. For example, you may want to have your URLs presented in a different language. Access to the Drupal source code on the web server is required to set up these kinds of aliases.

User Management

Access control - Determine access to features by selecting permissions for roles.

Access rules - List and create rules to disallow usernames, e-mail addresses, and IP addresses.

Profiles - Create customizable fields for your users.

Roles - List, edit, or add user roles.

Users - List, add, and edit users.

User settings - Configure default behavior of users, including registration requirements, e-mails, and user pictures.

Access control

Permissions let you control what users can do on your site. Each user role (defined on the user roles page) has its own set of permissions. For example, you could give users classified as "Administrators" permission to "administer nodes" but deny this power to ordinary, "authenticated" users. You can use permissions to reveal new features to privileged users (those with subscriptions, for example). Permissions also allow trusted users to share the administrative burden of running a busy site.

Access rules

Set up username and e-mail address access rules for new and existing accounts (currently logged in accounts will not be logged out). If a username or e-mail address for an account matches any deny rule, but not an allow rule, then the account will not be allowed to be created or to log in. A host rule is effective for every page view, not just registrations.

Roles

Roles allow you to fine tune the security and administration of Drupal. A role defines a group of users that have certain privileges as defined in user permissions. Examples of roles include: anonymous user, authenticated user, moderator, administrator and so on. In this area you will define the role names of the various roles. To delete a role choose "edit".

Anonymous user: this role is used for users that don't have a user account or that are not authenticated.

Authenticated user: this role is automatically granted to all logged in users.

Users

Drupal allows users to register, login, log out, maintain user profiles, etc. Users of the site may not use their own names to post content until they have signed up for a user account. You can filter users by a combination of role, permission and status here.

User settings

You can set public registrations to one of three states;

Only site administrators can create new user accounts.

Visitors can create accounts and no administrator approval is required.

Visitors can create accounts but administrator approval is required.

Optionally you can *Require e-mail verification when a visitor creates an account* and enter some user registration guidelines

Note

If this box is checked, new users will be required to validate their e-mail address prior to logging into to the site, and will be assigned a system-generated password. With it unchecked, users will be logged in immediately upon registering, and may select their own passwords during registration.

You can customize the generic email message sent by your site. There are several variables available to allow you to do this.

Enable or disable user picture support, set the path, dimensions and maximum file size along with entering picture guidelines for your site.

Content Management

Books - Manage site's books and orphaned book pages.

Categories - Create vocabularies and terms to categorize your content.

Comments - List and edit site comments and the comment moderation queue.

Content - View, edit, and delete your site's content.

Content types - Manage posts by content type, including default status, front page promotion, etc.

Forums - Control forums and their hierarchy and change forum settings.

News Aggregator - Configure which content your site aggregates from other sites, how often it polls them, and how they're categorized.

Post settings - Control posting behavior, such as teaser length, requiring previews before posting, and the number of posts on the front page.

RSS publishing - Configure the number of items per feed and whether feeds should be titles/teasers/full-text.

Search content - Search content by keyword.

Logs

The watchdog module monitors your web site, capturing system events in a log to be reviewed by an authorized individual at a later time. The watchdog log is simply a list of recorded events containing usage data, performance data, errors, warnings and operational information. It is vital to check the watchdog report on a regular basis as it is often the only way to tell what is going on.

Recent log entries - View events that have recently been logged.

Recent hits - View pages that have recently been visited.

Top 'access denied' errors - View 'access denied' errors (403s).

Top 'page not found' errors - View 'page not found' errors (404s).

Top referrers - View top referrers.

Top search phrases - View most popular search phrases.

Top pages - View pages that have been hit frequently.

Top visitors - View visitors that hit many pages.

Access log settings - Control details about what and how your site logs.

Status report - Get a status report about your site's operation and any detected problems.

Note

Ensuring that you have a scheduled task to run cron.php on your site is important. It updates and clears out older log entries. Failure to do this can cause you site to run more slowly over time.

Help

Drupal sites contain online help files and a basic vocabulary which are often overlooked. Make sure that the help module is enabled and then check it out. It covers all Drupal core modules.

Chapter 6. Upgrading your site

Upgrading your Drupal site whether for a point release or a full release follows the same basic steps.

1. Backup your database and Drupal directory - especially your "sites" directory which contains your configuration file and added modules and themes, any contributed modules in your "modules" directory, and your "files" directory which contains uploaded files.

Note: for a single site setup the configuration file is the "settings.php" file located at sites/default/settings.php. For multisite configuration the configuration file is located in a structure like the following:

```
sites/default/settings.php
sites/example.com/settings.php
sites/sub.example.com/settings.php
sites/sub.example.com.path/settings.php
```

More information on multisite configuration is located in the INSTALL.txt file.

2. Log on as the user with user ID 1. User ID 1 is the first account created and the main administrator account. User ID 1 needs to be logged in so that you can access update.php (step 9) which can only be run by user ID 1. Do not close your browser until step 10 is complete.
3. Place the site in "Off-line" mode, to mask any errors from site visitors.
4. Disable contributed modules and switch to a core theme (Blue Marine or Garland).
5. Remove all of the old files and directories from the Drupal installation directory.
6. Unpack the new Drupal files and directories into the Drupal installation directory.
7. Copy the backed up "files" and "sites" directories to the Drupal installation directory. If the original .htaccess or robots.txt files have been modified, copy the backed up versions of these files to the installation directory as well.
8. Verify the new configuration file to make sure it has the latest and correct information.
9. Re-install contributed modules.

Note

Note: make sure the version of a module matches your version of Drupal. Modules from previous versions may not be compatible with the current version. Check <http://drupal.org/project/Modules> for the version of a module to match your version of Drupal.

10. Run update.php by visiting <http://www.example.com/update.php> (replace www.example.com with your Drupal installation's domain name and path). This step will update the database to the new Drupal installation.

Note

if you are unable to access update.php do the following:

Open update.php with a text editor. - There is a line near top of update.php that says \$access_check = TRUE;. Change it to \$access_check = FALSE;. - As soon as the script is done, you must change the update.php script back to its original form to \$access_check = TRUE;.

11. Finally, return site to "Online" mode so your visitors may resume browsing.

Chapter 7. Concepts

Drupal 5 was released on January 15, 2007, the six year anniversary of the 1.0 release. It is a mature product and has generated its own practices and concepts. Some of these are shared by most content management systems available while some are Drupal specific. The following are a collection of articles contributed by various members of our community.

Backwards compatibility

Once a given major version of Drupal core is released, its API and schema is frozen. Point releases will be periodically generated for bug fixes and security issues.

For all major releases, Drupal core provides an update path for your sites data. Major versions do not provide backwards compatibility for the Drupal API. This is a long term project philosophy to allow the code base to remain agile and as fast as possible. You should plan your site with an eye towards considering major version updates 12-24 months.

Content, the Drupal Way

by Jeff Eaton

Organizing content in Drupal can be very liberating -- or very frustrating, depending on what methods you're used to using. Almost all information in Drupal is stored as a 'node', the basic unit of content. By default, there is no hierarchy or structure imposed on these nodes: they do not reside in specific 'sections' of your site, and different kinds of nodes (images, blog posts, news articles, etc.) are not automatically grouped or sorted into different groups. Rather than hard-coding specific hierarchies or styles of display, 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. Specific kinds of pages, specific sections of your site, and so on are created by pulling up any content with certain properties and listing it. Some examples can help clarify this concept.

- The front page, at <http://www.example.com/node>

The default front page of a Drupal site is a blog-style list of the latest stories posted. This simply pulls up any content nodes with both the 'published' and 'promoted to front page' flags set to TRUE. If any pieces of content also have the 'sticky' flag set to true, they'll be moved to the top of the list, which is otherwise sorted by the date the content was posted. Different content types can be configured to have these flags on or off by default, and different user roles (editors, authors, etc.) may be given permission to override these defaults on a per-post basis. In a stock installation of Drupal, 'story' content has the 'Published to front page' flag set to true by default, while 'page' content (often used for 'about us' information, and so on) does not.

- The taxonomy module

This module allows administrators to use sections, categories, and groups of keywords to label different types of content. For example, you might use the Taxonomy module to set up a list of topics that news stories can be filed under, and a field for authors to enter free-tagging keywords on their blog post. When the Taxonomy module is used in this way, the descriptive terms associated with a piece of content become just another property to use when organizing. The Taxonomy module provides a dedicated listing page for each one of these descriptive terms, displaying all the content associated with the term. Other third-party plugin modules also use these descriptive terms when sorting and filtering content in other locations.

- The Blog module

Drupal's Blog module implements group blogging features with just two additions: a custom 'blog' content type, and a set of new listing pages that show only blog entries, and blog entries by specific authors. With it turned on, <http://www.example.com/blog> becomes a sitewide blog, and <http://www.example.com/blog/1> becomes the blog, filtered to show only posts by user #1.

- The Views module

The Views module is a popular third-party plugin module for Drupal that gives site administrators the ability to define new listing pages and sidebar blocks that display content matching almost any list of criteria. For example, a list the to 10 pieces of content that are published, but NOT 'page' content -- can be sorted by the number of comments they've received and used as a 'most popular content' section.

The important concept to remember is that all content 'nodes' in Drupal are treated equally. Different sections of the site, and different kinds of organizational structures, are built by filtering down content based on properties like number of comments, creation date, author, associated keywords, and so on. You can have menu links to stand alone pages aliases with path module stand alone or nested. By planning your structure -- and sometimes adding new contributed modules that layer additional properties onto your content nodes -- you can build complex and powerful sites that grow with your organizational needs.

Filters

by Robert Douglas

[http://www.lullabot.com/articles/drupal_input_formats_and_filters]

Processing textual content for output in a browser is one of Drupal's most critical tasks. Without such processing we would all have to become masters at typing in HTML text! In this article I will explain what filters and input formats are, why they are important, how they are used, and why they impact the security of your site.

Filters and Input Formats


The pillars of Drupal's text handling are filters and input formats. A *filter* is a set of rules that can be applied to transform text in some way. Some filters strip certain HTML tags or security hazards from text. Other filters look for special patterns and expand the text in a meaningful way. Other fun-oriented filters, such as the Pirate Filter (<http://drupal.org/project/pirate>), rewrite the text altogether (in this case, to make it "talk like a pirate"). Filters know how to do one thing, and do it well; text in, filtered text out.

Some filters have extra configuration options. The HTML filter, for example, strips all but an allowed set of HTML tags from text. The set of allowed tags can be determined by the administrator.

An *input format* is an ordered collection of filters. Any text that is being displayed to the browser should be run through the filters in an input format first. The input format then applies all of the filters, in the right order, so that one filter feeds its output to the next, forming a chain. This chaining of filters can be the source of great flexibility as well as great confusion. The flexibility comes from the fact that filters can be made to work together, the confusion comes from the case where filters inadvertently work against each other, one filter undoing the work of the previous filter. I'll show examples of both.

Input vs Output

Drupal captures input in its raw form, saving whatever gets submitted straight to the database without alteration. Then, before displaying any such content in the browser, Drupal processes the text by choosing an input format to apply. Why doesn't Drupal apply the filters in an input format before saving input into the database? The answer is simple; flexibility. If you were to change the text that a user has input before saving it in the database, you could never get back to the original state. You could never change your mind about the configuration of the filters. By filtering on output, not on input, Drupal gives the site administrator the option of changing how content is displayed at any time. As an example, imagine that you notice the users on your site using character patterns to represent smiley faces. I know, that stuff is so 1998 :P But just for fun, let's say they're doing it ;-) You look around and find the Smiley Filter (<http://drupal.org/project/smileys>) on Drupal.org, and install it. Now all of the keystroke

patterns that your users had been using can be displayed as images  This ability to change is only available if the input is saved verbatim and filtering is done on output.

Meet Drupal's core filters

Here is a rundown of the filters that Drupal ships with:

- **HTML Filter:** The HTML filter is primarily responsible for removing HTML tags from text. It can be configured to allow any number of tags (white list) and it will remove the rest. It removes them either by stripping them, or by escaping them into entities like this: `<div>`. If tags are escaped, they show up in the output as visible tags: `<div>Some text</div>`. The set of tags that are allowed by default include: `<a>` `` `` `<cite>` `<code>` `` `` `` `<dl>` `<dt>` `<dd>`

The final task of the HTML filter is to add a spam link deterrent to anchor tags. The deterrent, proposed by Google, gives search engines a tip about which links to follow when crawling the web. If this option is enabled, `rel="nofollow"` will be added as an attribute of all anchor tags.

- **Line Break Converter:** This filter converts line breaks into `
` or `<p>` tags depending on whether a single or double line break is found. This preserves the paragraph formatting in the text that is input.
- **URL Filter:** Any web or email addresses that are found in the text will be converted to clickable links, thus saving the user the hassle of having to type ``
- **PHP Evaluator:** The PHP Evaluator is the most radical of all Drupal's core filters. It looks for text enclosed in `<?php ... ?>` and evaluates it as PHP code. This effectively allows you to program and extend Drupal just by submitting content to the site! In 99% of cases, this is a bad idea, and the initial attraction of harnessing such power should be weighed by a healthy sense of fear. If you really need to write PHP code to accomplish what you're trying to do, writing a module is usually a better idea (and not that hard in most cases). Furthermore, in the wrong hands, the PHP Evaluator is an enormous security risk. A malicious attacker, with the PHP Evaluator at their disposal, could wipe out your database and take control of your web server.

Drupal's core input formats

Drupal also comes with three input formats pre-defined.

- **Filtered HTML:** This is the workhorse input format that is used most of the time for displaying posts such as blogs, pages, forum topics and so forth. It combines the URL Filter, the HTML Filter and the Line Break Converter in a way that allows users a small set of HTML tags for formatting while taking care of paragraphs and URLs behind the scenes. This is also the default input format for new Drupal installations. More on default input formats later.
- **PHP Code:** This input format consists of only one filter, the PHP Evaluator filter. This input format is to be used when the goal is embedding PHP code in a post.
- **Full HTML:** The Full HTML input format applies only the Line Break Converter filter. No HTML tags are stripped and no weblinks are converted to anchor tags.

Order matters

When an input format consists of more than one filter, the ordering of the filters has a huge impact on what the final output is. The Filtered HTML input format has three filters, the URL Filter, the HTML Filter, and the Line Break Converter. Here is the order in which they are executed in a new Drupal installation:

Name	Weight
URL filter	0
HTML filter	1
Line break converter	2

`<h1>The quick brown fox jumps over the lazy dog.</h1> King Phillip came over from Germany swimming.

Every good boy deserves fudge.`

`http://drupal.org`

The first filter is the URL filter. It will find the URL which we have in this text and make it into a proper anchor tag:

Before

`http://drupal.org`

After

```
<a href="http://drupal.org" title="http://drupal.org">http://drupal.org</a>
```

The second filter is the HTML Filter. The text contains two HTML tags that are not on the whitelist, namely `<h1>` and `
`. Thus, they will be stripped.

Before

```
<h1>The quick brown fox jumps over the lazy dog.</h1> King Phillip came over from <em>Germany</em> swimming.<br><br>Every good boy deserves fudge.
```

After

The quick brown fox jumps over the lazy dog. King Phillip came over from `Germany` swimming. Every good boy deserves fudge.

Finally, the Line Break Converter gets its chance. It looks for line break characters (`\n`, `\n\n`, etc.) and replaces them either with `
` or encloses blocks of text in `<p>...</p>` tags. The function responsible (http://api.drupal.org/api/5/function/_filter_autop) for this is quite cunning, and was inspired by code from WordPress (our debt of gratitude).

So what went wrong here? Well, nothing, technically. But the output is unlikely to be what the user expected. First of all, the `<h1>` tag was stripped, which is a good thing because it is what the Drupal administrator wanted. Second, the place where the user wanted to make a line break using `

` is totally different than what the user might expect. After all, the final rendered HTML contains a `
` tag, so why were the `
` tags from the user stripped out? And why weren't they replaced by something more intelligent by the Line Break Converter? The answer can be seen by looking at the text as it gets passed from the HTML Filter to the Line Break Converter. Because `
` isn't on the whitelist of allowed tags, the HTML Filter takes them out, leaving the Line Break Converter no clues to follow concerning the user's wish for a line break between "swimming." and "Every good boy".

Changing the order

Let's look at the example above and see what would happen if we change the order of the filters. This is done by clicking to Administer -> Site configuration -> Input formats -> (Filtered HTML) configure -> Rearrange. Here I've changed the order so that the HTML Filter comes after the Line Break Converter.

Name	Weight
URL filter	0
Line break converter	2
HTML filter	3

As in the first example, the first filter is the URL Filter, so the output from that will be the same. The second filter, though, is now the Line Break Converter. Here is what happens to our text coming from the URL Filter and going into the Line Break Converter:

Before (as received from the URL Filter)

```
<h1>The quick brown fox jumps over the lazy dog.</h1> King Phillip came over from <em>Germany</em> swimming.<br><br>Every good boy deserves fudge.
```

```
<a href="http://drupal.org" title="http://drupal.org">http://drupal.org</a>
```

After

```
<h1>The quick brown fox jumps over the lazy dog.</h1> <p>King Phillip came over from <em>Germany</em> swimming.<br><br>Every good boy deserves fudge.</p>
```

<p>http://drupal.org</p>

Interesting to note is that no paragraph tag was placed on "The quick brown fox". This is because <h1> elements are block level elements (which get their own line breaks in rendered HTML), so the Line Break Converter ignores them. Also interesting is that we have many instances of <p> and
 tags, even though we're about to go into the HTML Filter which is configured to strip those tags out. Here is the final output with the new ordering of the filters (line breaks added for readability):

The quick brown fox jumps over the lazy dog. King Phillip came over from Germany swimming. Every good boy deserves fudge. http://drupal.org

What a mess =>

What's the real solution in this case? Well, the original order of filters worked better, so consider leaving it *URL Filter -> HTML Filter -> Line Break Converter*. One way to fix the problem would be to configure the HTML Filter to allow
 and <p> tags. This gives HTML savvy users control over paragraph formatting. The other solution would be to submit a patch against the HTML filter (see filter.module) to have it replace
 tags with \n line break characters so that the Line Break Converter will pick up on them. Guess which solution is easier :P

Input Formats and User Roles

Not all Drupal users are created equal. Some are anonymous users, some are authenticated users, and some have other user roles that allow them to have greater privileges than normal authenticated users. Furthermore, one Drupal user on every site is the super-user (user #1) who can do anything. The privilege of using an input format can be assigned to users on a per-role basis. This is an important mechanism that exists for allowing some trusted users to have access to some filters while denying this access to less trusted users.

Look at the screen Administer -> Site configuration -> Input formats. It lists all of the input formats that have been established. The default Drupal installation comes with three, as noted above. On any Drupal site, one input format has to be designated as the default input format. This is indicated by the radio button in the Default column. To guarantee the presence of at least one input format, the default format cannot be deleted. The others can be, however, and you might consider deleting any input formats (such as the PHP code format) that you don't plan on using.

Home > Administer > Site configuration

Input formats [List](#) [Add input format](#)

Input formats define a way of processing user-supplied text in Drupal. Every input format has its own settings of which filters to apply. Possible filters include stripping out malicious HTML and making links clickable.

Users can choose which input format to use when submitting content.

Below you can configure which roles, as well as choose a default input format (for supported content, for example).

Note that (1) the default format is always available to all roles, and (2) all filter formats can always be used by roles with the "administer filters" permission even if they are not explicitly listed in the Roles column of this table.

Default	Name	Roles	Operations
<input checked="" type="radio"/>	Filtered HTML	All roles may use default format	configure
<input type="radio"/>	PHP code	No roles may use this format	configure delete
<input type="radio"/>	Full HTML	No roles may use this format	configure delete

[Set default format](#)

On the configuration screen for an input format, you will see a listing of all the roles for users on your site. For any input format besides the default you can specify which user roles are privileged to use that input format. The default input format is automatically available to all users in all roles on your site and this cannot be changed.

Filters and Security

Filters and security go hand-in-hand. Without filters, there would be no security for your site as malicious attackers would have free reign in using scripts to deface your site, subject your users to phishing scams, and steal important data such as passwords.

The heart of the security offered by filters comes from the HTML Filter and the calls it makes to `filter_xss` (http://api.drupal.org/api/5/function/filter_xss) and `check_plain` (http://api.drupal.org/api/5/function/check_plain). These are the functions that Drupal uses to prevent attacks based on user input. For this reason, all of your user submitted output should be run through the HTML Filter. It is tempting to ignore this advice, especially if you are having troubles getting the configuration settings just right for your purposes. Don't ignore this advice. You may end up sorry.

Also worth reiterating is the fact that the PHP Evaluator filter poses an extreme risk if it can be used by anyone but highly trusted, PHP-competent site administrators. Most sites will be better off deleting the PHP code input format and not extending use of the PHP Evaluator filter to anyone.

Finally, it should be obvious that the Full HTML input format, which does not use the HTML Filter, is insecure and should be offered only to those users who can be trusted not to ruin your site. Most sites will be better off deleting this input format.

Many More Filters Available

The fun with filters is that modules can offer their own filters. The number of filters available is large, and I can't possibly cover them all, but you can get a feel for the possibilities by looking at the Filters and Editors category (<http://drupal.org/project/Modules/category/63>) of Drupal modules on Drupal.org. Here are some interesting modules that offer filters:

- Amazon Filter - http://drupal.org/project/amazon_filter : Provides a text filter to insert amazon book title/links, cover images, and themable formatted information using a simple `[amazon {title|cover|info}]` tag.
- BBCode - <http://drupal.org/project/bbcode> : Allows users to specify markup using BBCode.
- Code Filter - <http://drupal.org/project/codefilter> : Renders syntax-highlighted PHP code. This module is used on Drupal.org.
- DruTex - <http://drupal.org/project/drutex> : A LaTeX renderer that can, among other things, render mathematical formulas and generate PDFs of nodes.
- HTML Corrector - <http://drupal.org/project/htmlcorrector> : Corrects corrupt HTML in nodes and comments. This is useful for cases where users forget to close tags, or for where the teaser view breaks the HTML.
- Inline Filter - <http://drupal.org/project/inline> : Uses a `[inline:filename.jpg]` syntax to allow for inline images or file links.

Conclusion

The filtering of output is an essential part of web publishing and one of Drupal's great strengths. Understanding the difference between input formats and filters, and how to configure each, is an essential step in becoming a great Drupal site administrator. Drupal modules can implement filters to make your site powerful and fun.

Translations

by Konstantin Käfer

Drupal 5.x provides translation of the entire user interface out of the box. To distribute translations, Portable Object (.po) files are used. These files can be generated using GNU gettext (<http://www.gnu.org/software/gettext/>), an open source message translation framework. Currently, there are about 2400 strings in Drupal.

Obtaining translations

On Drupal's translation page (<http://drupal.org/project/Translations>), you can currently find more than 40 translations of Drupal. Pick your language, check what Drupal version you're using and click on the "Download" link in that row. Note that not all translators are always keeping up with the most recent Drupal release. It may take them some weeks or months to finish a translation for a new version of Drupal.

After downloading, extract the file. You will end up with at least these files:

<langcode>.po – This is the actual translation and contains all translated strings (e.g. de.po, fr.po, ru.po).
installer.po – This file contains only the files required

Note that there is no guarantee that a fully translated package is available for your language. Most "major" languages are fairly complete, but there are also exceptions. Similarly, a less common language may be at a 100% of completion. That is because translators are not a company with a dedicated localization team but a group of volunteers who are doing translation work on an entirely voluntary basis. It also depends upon you to help translation Drupal in your language.

Installing Drupal in a foreign language

After copying Drupal to your servers, don't yet visit the URL in your browser. Copy install.po in your install profile's folder (if you run the default installation, that's profiles/default) and rename it to <langcode>.po.

When you point your browser to your Drupal installation, you should get a language selection screen. Select your language and click on next. As long as the installer has been translated to that language, it should now be in the selected language.

Unfortunately, it is not possible to automatically import the full translation during the installation process. Installing Drupal in another language currently has the only benefit of translating the preconfigured content types to the install language.

Adding a new language to Drupal

Even if you installed Drupal in a foreign language, it will still be in English by default. To enable another language, go to the module administration screen (at Administer » Site building » Modules) and enable the "Locale" module. Then, go to Administer » Site configuration » Localization and click on "Import" if you want to import a translation. Choose the <langcode>.po file from the zip file you downloaded, select the appropriate language and click on "Import". Note that this can take a while.

If you want to start a translation from scratch (not recommended!), click on "Add Language" and provide details about that language. On the "Manage strings" tab, you can now start searching for untranslated strings and translate them.

To enable another language, go to the "Manage languages" tab, check the "Enabled" checkbox of your language, maybe set it to "Default" and save.

Help translating Drupal

As previously mentioned, translations are created by the community for the community. Therefore, your help is greatly appreciated. If you want to help with an existing language, read their readme files (in the translation zip file) or look at further information on the project page on drupal.org.

To get started translating, see the Translator's guide (<http://drupal.org/translators>). Every group has a slightly different translation process, but basically they work with a lot of small .po files. You can obtain the single small files (usually one for each module) through CVS. Go to <http://cvs.drupal.org/viewvc.py/drupal/contributions/translations/> and pick your language. Make sure that you have selected the right Drupal version in the "Sticky Tag" dropdown menu. With a program like poEdit (<http://www.poedit.net/>), you can create a .po file from a template and translate it, or open an existing file and complete or modify the translations.

Translating contributed modules

To create translation templates (the files actual translations are based on) for custom modules, use the potx module. Install it and go to Administer » Site configuration » Translations and click on the “Extract strings” tab. You can now select the modules you want to generate templates for and export them.

By convention, translation templates and actual translations for a contributed module reside in a subdirectory named “po” in the module’s folder. Before translating a contributed module, check there if someone already went through the effort to translate it. In those “po” directories, the files are named according to the <langcode>.po scheme.

The processes for importing a translation for a contributed module is similar to importing a translation. Instead of creating a new language on the “Import” tab, select an existing language

Introduction to the Content Construction Kit

by Karen Stevenson

In Drupal 5.x you have the ability to create as many types of content as you like. The contributed module CCK builds on that ability by allowing you to add custom fields to that content. If you use the Views module, CCK also makes all your custom fields available in any View.

Installation

Install CCK the same way other modules are installed, download and unzip the tarball from <http://drupal.org/project/cck> and add it to the 'modules' folder on your site, then go to admin/build/modules to activate it.

CCK is actually a group of modules. The primary module is called 'Content' and it is required in order to use any other CCK modules. The tarball includes a few other modules, which comprise the 'Core CCK' modules. They include:

- Text
- Number
- Nodereference
- Userreference
- Optionwidgets
- Fieldgroup

There are many other CCK modules available. You can see them by going to <http://drupal.org/project/Modules/category/88>. Once you are familiar with CCK, you can download and install any of these other modules you want to use. The best way to get started with CCK is to begin with just the CCK core modules. The core modules are well supported and it's easier to start with just a few alternatives, so start by enabling only the core CCK modules.

Add a New Field

Go to admin/build/content where you will see a list of content types that are available. There are a couple included as a part of a standard Drupal installation, and you can use one of them or add a new content type of your own. For our example, let's add some fields to the 'story' content type.

Choose the option to 'edit' the 'story' type. Once you get to the screen where you can edit the content type, you will see additional tabs that say 'Manage fields', 'Display fields', and 'Add new field'. If you choose 'Manage fields' or 'Display fields' you will see nothing, since you have not yet created any fields, so start by selecting the option 'Add new field'.

On the 'Add new field' screen you will be asked for a name for the field. You can type anything here, including spaces, but whatever you type here will be used to create the actual field name that is used in the database. If you type 'My Text', the actual field name will become field_my_text. You'll use that field name later if you want to do any custom themeing.

Note

The machine name of the field cannot be changed later!

Below the field name you will see a list of the fields that are available. The core fields are 'Text', 'Integer', 'Decimal', 'Nodereference', and 'Userreference'. Those are the types of content you can create with those fields. The 'Text' field will create content that is stored in the database as text. The 'Integer' and 'Decimal' fields create content that is stored in the database as numbers. Nodereference and Userreference store either the node or user ids in the database to allow you to link to those node or user records.

Under each type of field are several options for the way the data will be input for this field (CCK calls these 'widgets'). They control the way the input form will look for this field. The available options are different for each field, but include things like a text field, which is a blank box where a user can type text, auto-complete fields where a user can type text and the system will try to complete the text for them from values available in the database, select lists, checkboxes, and radios.

The screen will look like:

Create new field

Name: *

The machine-readable name of the field.
Allowed characters : unaccentuated a-z, numbers and _ . All other characters will be discarded.
You'll be able to choose a human-readable label for the field on next page

Field type: *

Node Reference

- Select List
- Autocomplete Text Field

Integer

- Text Field
- Select list
- Check boxes/radio buttons
- Single on/off checkbox

Decimal

- Text Field
- Select list
- Check boxes/radio buttons
- Single on/off checkbox

Text

- Select list
- Check boxes/radio buttons
- Single on/off checkbox
- Text Field

User Reference

- Select List
- Autocomplete Text Field

Start with something easy, a 'Text' field with a text field widget, by selecting 'Text' > 'Text Field'.

Next, you will be taken to a screen where you can define more things about the way this field will work. It will look like:

New Field Form

Widget settings

These settings apply only to the *My Text* field as it appears in the *Story* content type.

Widget: *

- Select list
- Check boxes/radio buttons
- Single on/off checkbox
- Text Field

Label: *

Rows: *

— Default value

Display in group:

 ▼

Select a group, in which the field will be displayed on the editing form.

Help text:

Instructions to present to the user below this field on the editing form.

Data settings

These settings apply to the *My Text* field in every content type in which it appears.

Required

Multiple values

Text processing:

Plain text

Filtered text (user selects input format)

Maximum length:

The maximum length of the field in characters. Leave blank for an unlimited size.

Allowed values list:

The possible values this field can contain. Enter one value per line, in the format key|label. The key is the value that will be stored in the database and it must match the field storage type, text. The label is optional and the key will be used as the label if no label is specified.

— **Php code**

You'll notice that the list of possible widgets shows up again on this screen. That's because you have an option to change your field to use a different widget once the field has been created. Skip past this section for now.

Some of the things you can define include the following:

- Label

Create a label here. This is a chance to create a label that has different components than the machine name of the field. It can contain spaces or non-ASCII characters, and it can be changed later. The machine name cannot be changed.

- Rows

This is an option that you only see for the 'Text' field. If you leave it set to '1', the default, you will get a regular textfield in your form, which will be stored as a varchar field in the database. If you choose a number greater than '1', you will get a large textarea in your form, set to the number of rows you select here, stored as a text field in the database.

- Default value

You'll see a collapsed area where you can input a default value for this field. You can use this or leave it blank.

- Group

If you have enabled the Fieldgroup module, you'll see an option to select a group to place this field into. If you have not created any groups, this won't have any effect.

- Help text

Anything you type into the 'Help text' box will appear to end users as a description for this form element.

- Required

If you select the option to make this field required, users will have to fill this field out and will not be able to save new content without doing so.

- Multiple values

The multiple option is one of the most powerful and confusing parts of CCK. If selected, users will be able to select or create more than one of these items. One of the most common uses of this is to create a text or number field that uses a select or checkbox widget where you want to allow users to select more than one item.

- Text processing

This is another option that only applies to the Text field. Here you identify if the user should be allowed to select a filter for text processing. The option to use filter process will only make sense if the field is being created as a textarea field by selecting more than one row.

- Maximum length

Select the maximum length for the field or leave it blank for unlimited length. If this is a textfield, this value will be used to shorten the available field to the selected size.

- Allowed values list

This rarely will make any sense or be useful unless you choose to use a select or checkbox widget. If you use such a widget, this is where you can define the list of options that users can select from.

- Save the field

Select or fill out these values, then save the field. You'll be returned to the 'Manage Fields' screen.

Manage Fields

Once you have at least one field created, this screen becomes your main place to see and organize your fields. You'll see both fields you have created with CCK and fields added by other modules that CCK can't control. For instance, you'll see the content 'Title' and 'Body' on this screen. Each field has a weight that controls the order it appears when content is created or edited. If you have enabled the Fieldgroup module, you'll see your fields organized by groups. On this screen you can change the weight and group for each field. If you have no groups to select from, use the 'Add new group' to create them.

The weight controls the order that this field will appear in. Fields with heavier weights will appear after fields with lighter weights. Keep in mind that there are non-CCK fields in the content, too, and they have their own weights. For instance, the 'Title' field has a default weight of -5 that can't be controlled by CCK. If your content type has a body, the body has a weight of 0. If your content type uses taxonomy, the taxonomy has a weight of -3. So you need to select your field weights to fit in among other things in the content. If you have a lot of fields, it may be hard to fit them all in between other elements on the form. That's where fieldgroups come in handy, since you can put fields into groups, then organize them by weight within their group.

Display Fields

The 'Display Fields' screen is where you can control when field values and labels will appear in your content. Each field is listed on the screen, and you can select whether the label should be hidden or appear above or next to the field value.

For each field, you can choose how the field will be displayed in the teaser and the full node view by selecting the 'formatter' to use in that view. The exact formatters available differ depending on the type of field you are using. For the 'Text' field, you have a choice of 'Default', 'Plain text', 'Trimmed', or 'Hidden'. 'Default' will display the normal view of the text, as HTML if html is used, and filtered with the selected filter. 'Plain text' will display

the raw text from the field with no HTML. 'Trimmed' will trim long textareas down to the length selected for post teasers, and 'Hidden' will not display the field at all. So you could choose to hide this field in the teaser, but display it in the full view.

Create Content

Once you have created field for your content type, given them weights and put them in groups, and set up the display options, you're ready to create new content. Go to `node/add/story` (or select 'Create content' in your navigation, then choose the content type you added your fields too), and you will see an edit form that contains your new fields. Test that the fields behave as you intended by filling them out and saving the form.

That's it, you've now created custom fields in your content type! Go back and try the same thing with other core CCK fields and widgets, then try adding other CCK field modules to the mix.

CCK and Views

If you have installed Views, CCK will automatically make all your custom fields available in Views. Once you understand how Views works, create a new view and you'll see that all your custom fields appear as options in several places. You can add them to the 'Fields' section of your view, as Views 'Filters', as 'Sorts', and as 'Arguments'. You'll be able to choose from the same 'formatters' as you saw on the 'Display fields' screen, and have an option to display multiple value fields on individual rows in the results, or grouped together in a single row.