The Drupal Cookbook (for beginners)

German version

Purpose

To help Drupal "newbies" who seemed overwhelmed on Drupal.org at first.

Background

The tips in this book are based upon a test site on PC, running Windows *(Windows-specific guidelines)*. Pretty much all of this is directly applicable to building a site directly on a web server. Although it was based on Drupal 5, most of the process is the same for Drupal 6, although a few of the navigation buttons may have been renamed or moved.

Myths

You don't have to know how to program (especially in PHP) to use Drupal. It's an advantage to have some basic knowledge of PHP, HTML, and CSS, but it is not required.

Some good resources for you:

- W3 Schools - for virtually everything Internet-related.
- The Official PHP Site - the full PHP implementation.
- PHP Builder - some tutorials and code ready-to-use.

This tutorial uses HTML and tweaked CSS, but not one line of PHP code. Everything is precompiled under the hood of: **Drupal!**

Some may think the learning curve for Drupal is going to be steep and it will take you months, or even years, to get a web site up and running? Wrong. Hogwash! Your first, largely static, web site with 36 pages can be up in less than a week after you installed your first copy of Drupal. Then it's just a matter of sorting out hosting arrangements and domain names with a helpful hosting provider.

You **can** do it! Yes, **YOU**.

Using Drupal.org Site

A few things you need to know before you post anything on Drupal.org site:

- Search to see if the subject you care about has already been covered. That is to avoid having duplicate postings & issues.
- Put only one question subject for each posting, this can shorten the time and make it easier for you get an answer.
- If you don't understand the responses to your question, don't hesitate to ask again and add "I'm a newbie, can you say this in easier to understand terms?"
- When loking for help, read the tips for posting in the Drupal Forums. Try to

completely describe your situation/case. Explain what you have done so far and exactly what you want to do.

- For example, if a page is not showing up correctly, it could be:
  1. Browser problem, so make sure you tell the reader what browser you're using. *(Name, Version ...Plugins ..)*
  2. Always mention which release of Drupal you're using (4.7.x, 5.x, 6.x etc)
  3. Any contributed modules you're using(add version) that may be at play.
  4. Also wouldn't hurt to know which version of PHP and MySQL you're using*(or your Hosting Provider)*.
- Here some basic resources:
  1. Terminology (Terms).
  2. Drupal Jargon.
  3. Common english expressions & acronyms
     - **Node**: This is, in simple terms, a unit of content. It may for example be a "page," "story," "book page," or "blog entry." These different *types* of *content* are also referred to as "Content types".
     - **Theme**: This is the way your site is displayed (or rendered) to the end-user. The graphic look, layout and colors of Drupal sites are defined by the themes.
     - **Block**: This is a "container" for pieces of content on your displayed page. You may also have "Recent posts," "Sponsor links," or "Who's online."
     - **Menu**:A menu displays a list of links. Drupal menus are often located in a column on the left. For example, your **navigation menu**.

Typing Convention

Throughout this site, as well as the Drupal site, you will see things like `Administer >> Access control >> User management >> Roles`. This means click on "Administer" in the navigation menu, then "Access control," then "User management," and then "Roles."

Occasionally you will see refer to "production" or "live" sites. These terms are pretty much interchangeable. The latter term is more modern and accepted in reference to web sites and means: "the site that your end-users interact with". The former term is largely synonymous but is a more "traditional" data processing term.

Some Preliminary Advice

These are some recommendations before you start with Drupal:

- Never try anything for the first time on a live site. Use a test site that uses the same modules and same data (different database).
- Don't try to make the "perfect site" on your first shot. Muddle through for a while until you understand a better way. Stressing over the perfect solution will **slow you down**.
- Start by learning and using the basic functionality of Drupal. Find what it *can* do first before working towards exactly what you *want* it to do. Once you are confident with "core" features, you may start to use more complex contributed modules like: Views, Category, CCK, and Organic Groups modules.
  These modules and some others require a good bit of understanding to master and it might discourage you if you try to dive in too fast. However, over time you will come to realize that these are some of the most powerful and flexible modules out there. *[Nancy's note: also stay away from access control (security) modules*

until later. They can really destroy your site if you don't know what you're doing.]

- Customize one of the default themes before creating your own, the Theme Developer's Guide in the handbook is a big help.
- When you run into a problem with a module, make sure to read the "readme," then do several searches with different terms. Only rarely have my problems not already been answered in the forums or in an issues queue.
- Use google search in this most effectively way: `subject site:drupal.org` You can often quickly find your issue - and solutions to it - already posted somewhere.
- To make a site really successful, make it work for the users, not against them. Once you have the basic site set up, get your friends, family, or anyone who will talk to you to look at your site and give honest feedback, or do user testing for a more formal approach (I run a company intranet and I do user testing every couple of months).
- Participate in the forums when you can. It's surprising how explaining something to someone else helps you understand it yourself.
- Go easy on what I call "gadgets" such as useless blocks, images, and graphics that clutter the page. I prefer simplicity and I only place something on the page if it is needed. This of course depends on your application.

Good Luck!

Drupal Is Supposed to be Easy?

Drupal is very powerful and flexible. That means it must have a significant degree of complexity. Do you think the folks at Myspace don't have their terminology or managed to roll out that site in a day? I don't know who told you Drupal was easy, but many people make it harder than it has to be by thinking they need to understand everything at once.

Terminology is necessary in order to properly convey what one is trying to say or ask. If you talk about "that box-like thingy on the right side of my screen" you could be referring to many things. Contrast that with "the Author Information block in the right sidebar" - now you are precise and everyone knows exactly what you mean. You've told them what it is, where it is, and even how it got there and part of how you've styled it.

Start by trying to understand the basic parts of Drupal, don't try to understand everything at once. For example, it is imperative that you know what a node is (look in my book). Then understand what content types are. Learn the basic parts of the rendered page (header, footer, left and right sidebars, and the center, or content areas). Check out the administration pages so you have some idea where things are, even if you don't understand them all today.

It's all fine and good to have "MySpace" as your target, but you are one person with a new tool. The people that put that together are many and using tools that they already were familiar with. (BTW, I find MySpace to be rather illogical.)

Just start by getting something up and visible. Then celebrate that you've done that. Now you're ready to move on to more wonderful things, but do it one step at a time. Don't add tons of modules right away; get comfortable with what you have. Add modules one at a time and get familiar with them - one at a time.

As for making Drupal easier and more logical, you're welcome to submit feature requests or explain why something is not done in the most logical manner. But don't demand it, or threaten to abandon Drupal if you don't get it your way. And certainly don't resort to name calling or derogatory comments.

A. Getting Started

There are a number of ways to set up a test environment on your own local computer. Numerous applications and tutorials for a variety of operating systems are located in the Setting up a development environment section of the handbook.

Reasons to run a local development server:

- Developing locally allows you to work when not online.
- Getting your local server running, even if it is with a simple installation like WAMP, will help you start thinking in terms of server processes and databases. As you get deeper into Drupal that knowledge will pay off.
- Everything you put on the web is searched, archived and hangs around for a long time. Do you want your inevitable learning mistakes displayed for the world to see on Google?

This tutorial uses the example of building a site on a PC with Windows using the DeveloperSide.net package.

This package has already integrated the following things:

- Apache 2.2 HTTP Server
- MySQL 5.0 Database
- PHP 5.2 and Perl 5.8 Scripting Languages
- GUI WAMP-stack Controller
- Dynamic DNS Client
- Tomcat Servlet/JSP Container
- mod_aspdotnet ASP.NET Host Interface
- OpenSSL Cryptography Toolkit
- mod_security Web Application Firewall
- phpMyAdmin MySQL Administration
- Joomla
- Drupal
- WordPress
- MediaWiki
- phpBB

Do note that any package, like the DeveloperSide one, that includes Drupal for you may not always have the latest secure version of Drupal. It is, therefore, recommended that you check the version immediately and upgrade Drupal if needed when using these packages.

I followed their instructions, which built me a working system! For more instructions see the Web.Developer page in the development environment section. I don't remember if it was automatic or not, but you will find it useful to have the "Web-Developer Controller" icon on your desk top.

The only "fly in the ointment" was that when I went to the Drupal web site to start pulling my modules and themes, there was a big announcement on the front page saying that a new security release for Drupal 5 had been released and was highly recommended.

I downloaded the latest stable release. I then unzipped it (using WinZip).

Of course, that created a directory called "Drupal-5.1," (the latest version at the time this was written) but the other software I had installed was looking for a directory called "Drupal." Well, by getting Apache and its services shut down, I managed to rename the two directories so that 5.1 was now called "Drupal." It worked! I now had a running 5.1 system!

If you have not set up your site using a package that includes Drupal, it is still very easy to install in a few minutes. You can find complete instructions in the handbook Getting Started section under the version number you are installing. Here are the directions for Drupal 5.

B. Basic Configuration

Whether you run one site or several, there are some basic things you should do right now. Here's what I do right off the bat; the advantage to doing it in the "root" database is that when I make copies for my other sites this has already been done. I'd give you a link to something on the Drupal site, but I never found anything like this.

1. Go to `Administer>>User management>>Roles` and create an "administrator" role.
2. Go to `Administer>>User management>>Users` and create a user entry for yourself. This allows you to test the site by changing your role to meet your needs.
3. Go to `Administer>>User management>>Access control` and allow the "administrator" role to do everything.
4. While you're there, go ahead and set what the "authenticated users" (logged in) and "anonymous user" (not logged in) can do, such as using your contact form. This is not engraved in stone; you can change it any time you want.
5. Go to the `Administer>>Site configuration>>Site information` page and, near the bottom, set the "Default front page" to "node." As long as you're on this page, set basic defaults for the other fields. I don't know about everyone, but I don't like, when I visit a site, being called "Anonymous" so I change the designation to "Visitor."
6. If there are any modules (core or contributed) that you use on all sites, go ahead and enable them now ((`Administer>>Site building>>Modules`). For example, you will probably use "Page" on all sites, and maybe "Story." I am finding more and more uses for "Book."

   I do recommend turning on (enabling) the "Path" core module so you can use "normal" names for your pages.

   If you want to use the contact form to email anyone from the site, be sure to enable the "Contact" module.

7. The same goes for themes.

There are a few things I recommend that you do in all your databases, so this is a good time to do it:

- Turn on "CLEAN URLS" to make your site more user friendly. Go to `Administer>>Site` configuration>>Clean URLs. At the bottom of the verbiage there is a link to run the "Clean URLs Test." If it passes, then the "Enable" radio button will un-dim. Click on that. (If the URLs stop working for some reason here are instructions to unset clean URLs.)

- In order for me to create any kind of content, I go to `Administer>>Site configuration>>Input formats` and set "Full HTML" as the default until I get the site ready to go live. Then I still allow administrators (like my other ID) to use that format. Do this now and you will avoid a very common problem with building your site.
- I don't like having "Promoted to front page" as a default for content, so I go to `Administer>>Content management>>Content types` and turn that off - in each format.
- While you're there, decide on your default comment mode. Go to `Administer>>Content management>>Comments>>Settings` and set the comments to be entered on a "separate page" and make sure that "Preview comment" is set to "Required."
- Now, let's turn on the Contact form so your users can send you a message. Go to `Administer>>Site building>>Menus` and locate the "Contact" item. Click on the "enable" link. Remember that later on you will want to go to `Administer>>Site building>>Contact form` and finish setting that up.

C. Creating Multiple Sites On a Local Computer

Need another test site? Here's how to do it the "easy" way. [**Hint**: if creating multiple sites is desirable make a list of the desired sites before reading through these instructions completely. Some steps can be done in bulk to save time.]

Why create extra sites? In addition to my having several sites running already, I had some ideas in the back of my head, not the least of which being a site where I could document everything I do (like this book). I also had some idea for other sites that I might put up in the future. So before you totally pooh-pooh the idea, give it a few minute's thought. And you can always change your mind later; it just might be a bit messier then.

At the very least, I would create a "working" site other than my "root" site. This makes it easier to start all over again if you get totally out of control later on.

This may look like a long process, but it's deceiving because I spell it out in detail. It is expanded and updated from the post "Running multiple sites on a local PC (localhost) from a single codebase, using Windows." For more details on the "official" stance on the sites directories, read Setup of /sites directory for multi-site.

If you want to get deeper into creating multiple sites either locally or on your server, you can search the forums at drupal.org with the term "multi-site"; there is also a group devoted to this subject at http://groups.drupal.org/multisite. Within the Handbooks, there is a good section: Multi-site installation and set-up.

- Open phpMyAdmin (using "other" in Web-Developer Controller)

  start bulk loop 1

- On the left, select the Drupal51 database. This is the one that was created by the package installation.
- Click on "Operations"
- Scroll down to "Copy Database to:"

- Enter the new database name.
- Verify that the radio buttons are clicked for:
  - Structure and data
  - CREATE DATABASE before copying
  - Switch to copied database
- Click on the Go button just below this area.
- When the copy is complete, click on "SQL".
- In the "Run SQL query/queries on database" box, enter:

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

  admin is what I call my administrative role (this is the "super-user" name);

  superpw is the password for this user (I tend to use the same one for all my databases to make it easy).

  Hint: I created and saved this in c:/www/drupal/grant.txt so I could copy and paste.
- Change *databasename* to the new database's name.

  end bulk loop 1

- Close phpMyAdmin.
- Go to the www/Drupal/sites directory.

  start bulk loop 2

- Copy one of the site folders (e.g. `default`) and name it for your new site.
- Open the directory, then open the settings.php file in Notepad.
- Change the `$db_url` and `$base_url` lines. The `$db_url` line should have the name of the database you just created. The `$base_url` will be how you want to access the site.
- Close the file.
- If you are going to have site-specific modules or themes, go ahead and create directories in this site to hold them (named "modules" and "themes").

  end bulk loop 2

- Navigate to the /www/Apache22/conf folder.
- Locate and open the "httpd.conf" file (**Hint**: I always make a copy of things I'm about to change, just in case I mess up.)

  I did notice that my version of Apache sets `index.html` ahead of `index.php`, so don't have an `index.html` in your directory.

- Find the line that says "# Virtual hosts." Remove the "#" from the next line. This allows you to make all your other changes in a separate, and less dangerous file.
- Save it.
- Navigate to the /www/Apache22/conf/extra folder.
- Open the httpd-vhosts.conf file.
- At the bottom of the existing list, comment out ("#") the examples.

  start bulk loop 3

- At the bottom of the existing list, enter:

```
<VirtualHost *:80>
    DocumentRoot /www/drupal/
    ServerName databasename
</VirtualHost>
```

- Change *databasename* to the new database's name.

  end bulk loop 3

- Close the file.
- Navigate to the /windows/system32/drivers/etc folder.
- Open the Hosts file with Notepad. ***Hint***: these two steps can be done by using "other" in Web-Developer Controller.

  start bulk loop 4

- Add a line that says:

```
127.0.0.1   databasename
```

- Change *databasename* to the new database's name.

  end bulk loop 4

- Close the file.
- Go back to Web-Developer Controller.
- Click on Apache2 (top left).
- Click the Stop Service button.
- Wait for it to change to "stopped."
- Click the Start Service button
- Wait for it to say "running."

Go for it. You can now start the browser and enter `http://databasename`.

For more details on directories for multiple sites, see Setup of /sites directory for multi-site.

D. Error Pages

Occasionally, a user may do something that confuses Drupal, such as typing a wrong page name or trying to access content they shouldn't. These will generate 404 and 403 errors, respectively.

A recent SEO newsletter, they mentioned the value of letting Drupal handle these errors:

> Your unique 404 error page should look like a regular page of your site. It should include your site's header, footer and navigation bar so that the site visitor can easily click on another area of your site. The content of this unique 404 error page should contain text explaining that the page selected is no longer available along with contact information so the site visitor has the option of emailing or calling your company.

This was one of those "Duh" moments for me. How obvious it is that you should make it easy for the user to get "back into" your site.

The same more or less goes for the "access denied" (403) error message. Let them know they did a no-no and try to explain why.

Just go to "Create content" and select "Page." I title them "Access Denied" and "Page Not Found" but you can call them whatever makes sense to you and your users. When you submit them, note the node ids. Then go to `Administer >> Site Configuration >> Error Handling` and enter "node/*nnn*" in the appropriate boxes.

Page Not Found

Here's the HTML for my 404 page:

```
<p>Sorry! The page you were looking for no longer exists. We redesigned our site and many of the pages have
changed.</p>
<p>&nbsp;</p>
<p>If you are unable to find something on our new site or have a question about our site or services feel free to
<a href="/contact">contact us</a>.</p>
<p>&nbsp;</p>
<p>--Webmistress</p>
```

Access Denied

Here's the HTML for my 403 page:

```
<p>We're sorry, but you must have permission to view the page you requested.</p>
<p>&nbsp;</p>
<p>If you are already a registered member of this site, please try <a href="user">logging in</a>.</p>
<p>&nbsp;</p>
<p>If you are not a member, you need to <a href="/join_us">join us</a>.</p>
<p>&nbsp;</p>
<p>If you have any questions about our site or group, please feel free to <a href="/contact">contact us</a>.</p>
<p>&nbsp;</p>
<p>--Webmistress</p>
```

Don't worry that you haven't created the "join_us" page yet. This is an advantage to having URL Alias support (the Path module) enabled. Just add to your to-do list to create this page when you get to the "Creating Content" step a few chapters later in this book.

E. Accessing Your Test Site(s)

Okay, great, now we have all that software installed. But how do we use it?

First, open the "Web-Developer Controller" that you put on the desk top. Look at the top left. You want to see, and should, if Apache and MySQL are running. If they are, you're just about ready. If not, select Apache, then click on the "Start Service" button. Within a few seconds it should change to "Running." Now select "MySQL" and start it.

Fabulous, we have the software ready for us. Now let's get us ready.

Fire up your favorite browser. In the Address field type in `http://localhost/drupal/` to access your "root" database. Since you've done everything right, you're now into your Drupal database.

Congratulations! Now the work begins.

F. Adding Modules and Themes

Let me first say that a newbie shouldn't worry a lot about adding modules and themes at first. Work on the basics of your site first, then worry about add-ons.

Themes are largely a matter of taste. For example, I have no idea why anyone would use a "fixed width" theme, but lots of people do. One nice thing about themes are they are pretty much independent of your content (later on you can look at the many submissions that are dependent on content).

Contributed **modules** are ways to add or extend functionality of your site. The only module I, personally, consider necessary is the Nodewords (a.k.a Meta Tags) module; in my opinion, it should be promoted to "core" status. This one allows you to add the "content," "keywords," and "robots" meta tags to your pages. This is useful if you're interested in your search engine rankings. You will also find that many contributed modules also require the Views module; I go ahead and make that a standard one for my sites.

Now, if you experiment with different themes and modules, as I know you will, despite my suggestions, you should also look at the Update Status (core in D6) and Site Documentation modules to make sure you are current and to document and clean up the mess your experimentation will make. Here are some suggestions on choosing the release: Strong stomach?

Installation

Installing a module or theme is pretty much the same until you get to enabling them. Now keep in mind that I use a Windows based PC (development) and Linux servers (on my live sites).

1. Go to the Drupal site and click on the "Downloads" tab. Then select either "Modules" or "Themes" depending on what you're after.
2. Locate the module or theme you want.
3. Make sure there is a version for the version of Drupal that you are using. D5 modules and themes will not work on D6.
4. I always click on "Find out more" and read the stuff again. This gives you the chance to see if there is support for your release of Drupal. You can also look at pending bugs and feature requests - it might change your mind.
5. Download the proper release. (I put them in a Drupal folder in "My Downloads.")
6. Unzip the downloaded file (I use WinZip). It may tell you that there is only one file in the zipped file; click "yes" or "OK."
7. Extract the code to to your `/sites/sitename/modules` or `themes` folder. If you are not running multiple sites, this would be `/sites/all/modules` or `themes`.
8. That's it! Now you need to enable it.

Modules

Modules

You enable a module at `Administer>>Site building>>Modules`. The non-core modules are listed farther down. With 5.x, they now show you some of the inter-module dependencies. I turn them on and "Save configuration" in order of the dependencies. For example, "Views UI" requires "Views", so I turn on "Views" first, save the configuration, then turn on "Views UI." and save again.

Most modules introduce some kind of menu items. Those will generally appear automatically when the modules is enabled. A few menu items will not show up until the permissions are set (the next step). And even fewer require you to take action to add the menu items, but the module will have instructions on how to do that.

Ah, now the real work begins. Go to `Administer>>User management>>Access control` to select who can use the features of the new module.

If the module introduced new content types, go to `Administer>>Content management>>Content types` and configure them. Don't forget this may also affect your "Input formats" (`Administer>>Site configuration>>Input formats`) and "Categories" (or taxonomy, `Administer>>Content management>>Categories`); you'll have to check those too.

Okay, now you can start using the new module.

This Site

My documentation site is a relatively "vanilla" implementation of Drupal.

Core Modules Enabled	Contributed Modules in Use
• Blog • Book • Comment • Contact • Help • Menu • Path	• Codefilter - Provides tags for automatically escaping and formatting large pieces of code. • Meta Tags (Nodewords) - Allows users to add meta tags, eg keywords or description. • Site Documentation - Documents and cleans up your configuration.

To get some idea of what modules are available, check these links: module handbook and contributed modules handbook.

Themes

You enable a theme at `Administer>>Site building>>Themes`

If it has never been enabled on this site, you will have to check the "enable" box and then click the "Save configuration" button at the bottom.

To set up how the theme works, click on the "Configure" **link** (not the tab at the top).

Fill in the fields.

Save the configuration.

Don't leave the page yet.

Logo and Favorite Icon

Now use that "Configure" tab

I prefer to do this part under the "Global Settings" but it can be done theme-by-theme if you prefer.

The "Default Logo" is that little picture (usually) in the upper left corner of the page. For example, on the "Bluemarine" theme, it's the Drupal logo.

If you want to change it, here's how:

1. First, find out what size it is because you'll want your own logo to be about the same. (If you are comfortable with HTML and CSS you can also edit the theme's code to accommodate your logo rather than resizing the image. How to edit theme code is not covered in this beginner's guide.)

Theme Name	Width	Height
Bluemarine	48	55
Chameleon	49	57
Garland	64	73
Minelli	64	73
Pushbutton	144	63
Fancy	80	80

2. Under "Logo image settings" either type in the path to your logo, or upload it from your PC.
3. **Note**: Neither one of those options turns off the "Use default logo" check box. You must select the correct check box yourself.
4. The "Shortcut icon" (a.k.a. the favorite icon, or "favicon.ico") is the same way. If you want to change this, you must specifically say, "Hey, Drupal, I'm changing this."

Now you click the "Save configuration" button. If you did this in "Global settings" it affects all themes (assuming they behave properly); if you did it for a single theme, then only that theme is changed.

For a list of all available themes, check Themes.

HINT: Going to make a few (or a lot) of changes to a standard theme? Think about copying it over to your `/sites/sitename/themes/` folder and renaming it. Then you can do anything you want and still be able to undo it easily by recopying. If the changes ae a bit bigger, think about contributing it back to the community (with your name, of course).

G. Creating Content

"Wow, I've done all this and I still don't have any content on my site!" Well, let's fix that.

First, let me explain that the page your visitor sees first upon entering your site is usually called the "home" page. Drupal calls this the "front" page, much like a newspaper. This page is special to Drupal. I know you're in a hurry, but read about both "pages" and "stories" before you decide which to use to create your front (home) page.

Content Types

Page

Drupal says, "If you want to add a static page, like a contact page or an about page, use a page." If you're used to building web sites with HTML, this is what you've done in the past. In general a "page" is going to stand on its own and will probably have a menu entry. You may also later add it into a book. When I created my first two sites (based on former static HTML sites, I made the front page a "page;" I have since changed to a "story."

Story

Drupal says, "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."

Okay, you've seen the Drupal site and noticed that there are "pieces" all over the place. Look at the front page; you'll see several announcements with space between them - those are "stories."

I have now switched my sites to use stories on the front page. The "welcome" message is one story. I have an announcement story node that any of my admins can edit. If you want to have weather or cartoons, a story is a good idea. Another use is if you are on a net ring - put the ring links into a story.

Book Page

Drupal says, "A book is a collaborative writing effort: users can collaborate writing the pages of the book, positioning the pages in the right order, and reviewing or modifying pages previously written. So when you have some information to share or when you read a page of the book and you didn't like it, or if you think a certain page could have been written better, you can do something about it."

Another way to use a book is to collect related information together. A book has its own navigation, so it can also be used to de-clutter your menu.

Blog Entry

You probably already know what a blog is, but just in case: A blog is a diary, collection of thoughts, or other time-ordered content. The Blog Entry content type is added by the blog module. The blog module allows you to have a multi-user blog, meaning that every user can have their own personal blog. This adds titles and breadcrumbs to indicate the blog authors' names. Generally if you only have a single blog for a site then it is best to use the Story node for blog entries.

So have you decided what kind of content you want? No, okay, just start with a page; it's easy. As you go create your content, be thinking about the menu as well.

NOTE: If you want to set your front page (Administer >> Site configuration >> Site information) as "node," then you need to have at least one piece of content marked as "promoted to front page." If you don't do this, you will keep getting that "Welcome to your new Drupal site" message.

Another Handbook section you may find useful is Creating new content.

Adding Images to your text

There is a choice of facilities for adding images to text items, with pros and cons of each.

1. The Image module, and associated features. Makes each image into a Drupal node, which ads a lot of capability.

· Image_Attach, which adds a separate image field to the target node, pointing at the image node. Provides simple image upload, but little other control.
· Image_Assist which embeds the image in a text field. Provides visual image selector, upload, and control over size and left/right float. Adds necessary HTML to text field.
· drupalimage plug-in to TinyMCE editor which makes Image Assist work with a TinyMCE field, which displays the result as a WYSIWYG image (though not styled fully according to your theme).

Also untested further features, including:
· Bulk upload facility
· Interface with the Drupal Gallery support module.
· Similar interface to the Acidfree support module.

2. The CCK ImageField. Very similar to Image Attach, but just uploads image into a filestore file, and again contains little extra control over e.g. sizes or styles. It is almost invariably used with Imagecache to give fine control for resizing.

3. IMCE(demo at http://ufku.com/drupal/imce/demo). Provides facilities to upload and search for images on the server. Functionally similar to the Image_Assist/drupalimage combination, but the integration with TinyMCE is neater for inserting images, and there is more control over the attributes of the image once inserted. BUT the image filing and select window does not look so nice – to the point of affecting usability, and there are bad things on the Blogs about the associated gallery function.

4. Or go to FCK Editor. From the demo seems as good an editor as TinyMCE, and has its own image upload and filing mechanism. But:
· No automatic creation of thumbnails etc. (cf Drupal Image)
· Images are simple separate filestore files – may be a benefit depending on what one wants?

I have oscillated quite a bit, but (at the moment) am going with the Image module:
· Install the Image module as normal.
· Do the stuff in http://mybesinformatik.com/tinymce-and-drupal5 to add the drupalimage plugin to TinyMCE.
· Tune the settings in the TinyMCE Profile to show the features required.
· Create a Taxonomy to allow tagging of images for easier retrieval.

Front pages

After following the next few steps, you will be able to easily add / change your front page anytime in the future.

1. Creating your front page

After logging in as Administrator, select

Create content > Page

from the left menubar, and create your own content that you would like to publish as a front page. If you are done, hit „Submit” to see the results. Notice that the current URL (the path to your newly created page) looks like this

http://www.example.com/?q=node/# (normal)
http://www.example.com/node/# (using clean URLs)

where # means a number. We will need that number, so write that down somewhere or just try to remember that.

2. Front pages

Now you can either choose to

- **2.1.** Set a single page as your front page or
- **2.2.** Promote your page to the front page, this way adding it to the top of the front page posts. (Only the first 5-10 lines will be published there (as a preview), you will have to click on it to see the whole page).
- **2.3.** Use the Front module to fully customize your front pages (theming, different user roles...)

2.1. Setting your page as Front page

After creating your custom page, select

Administer > Site Configuration > Site Information

At the bottom of this page where it says „Default front page” you will have something like this

http://www.example.com/?q= (normal)
http://www.example.com/ (using clean URLs)

and an input field next to it. That is where you have to enter

node/#

where # is the node number that you previously wrote down. By pressing „Save configuration” your front page automatically becomes the previously created page. You can reset that any time you want, by entering only

node

into the input field. (That is the default value).

2.2. Promote your page as Front page

If you want to promote your page to the front page, you should go back to step 1. but now don't press "Submit", or edit your newly created page (navigate to this address)

http://www.example.com/?q=node/#/edit (normal)
http://www.example.com/node/#/edit (using clean URLs)

where # is the node number.
Either way, at the bottom you will find a drop down menu called „Publishing options", there you will have to check the „Promoted to front page" checkbox and that's it. Press „Submit".

2.3. Front pages with Front Module

If you really need more control over your front page, you can use the Front module to

1. Set a different theme to your front page
2. Set different front pages depending on user roles (admin, anonymous, authenticated)
3. Insert custom php snippets into the front page

You can find the Front module at: http://drupal.org/project/front
After installing the Front module, select

Administer > Site configuration > Advanced front page settings

There the drop down menus are pretty self-explaining. If you click on any of the „Front page for _user role_" you will find

- **Body**: you can input text, html and even php snippets that you want to display
- **Select Type**: you can set other user roles to have the same front page as this, have default or custom themes be applied to it, etc.
- **Redirect to**: you can set a location where users should be redirected to

If you are satisfied with your settings, press „Save configuration" and you are done.

G1. Creating a Page

Click on "Create content" in the menu, then select what kind.

The title and body are pretty self-explanatory. Below these are a number of collapsible fields. "Input format" controls what you can put into the page; I'm assuming you are the "super-user" (user/1), so give yourself the priviledge of "Full HTML."

If you have Nodewords installed, the next section is "Meta Tags." It's pretty well documented.

Use the log message to provide information that might be useful to other authors who may edit your document later, or provide your rationale for making edits to your own or other people's content. The log message is not visible to users without the appropriate content editing rights.

As logs are recorded on a per-revision basis, each time you revise your content and make a log, the log message is stored with that particular revision of the article only. If you are have editing rights over a page you can see a list of revisions (if you selected the option to make a revision when you edited the document) and their associated log messages by clicking on the revisions tab.

Hopefully, you already set your default "Comment settings" but the front page rarely deserves comments, so you can disable them.

If you turned on the "Path" core module, you'll have URL path settings next. You can enter a "normal" name here rather than being required to use "node/2" when you refer to it later on. **Hint:** if you are converting a site that was static pages, you might want to go ahead and add ".htm" or ".html" to the name so that the search engines will continue to find the page.

"Menu settings" is the next section. "Title" is what will show in the menu normally. "Description" will show when the user hovers the cursor over the menu item. "Parent item" allows you to create multi-level menus that collapse and expand. "Weight" allows you to set a relative position within the menu; unfortunately, some Drupal core items are hard-coded at 0.

Chances are you won't need "Authoring information" unless you want to attribute the page to someone else. The other use for this section is to control the page or story order when they are based on the time and date it was created.

Finally, "Publishing options," which you set defaults for earlier, right? You want the page "Published" so it will show up. If this is your "Welcome" page, you'll want it "Promoted to front page."

"Submit" it.

Congratulations, you now have some content on your site!

G2. Creating a Story

Most of what goes for a page is also true for a story. You can largely consider the two types to be interchangeable, and it is goodness to have at least two content types because conflicts can arise in the way content types are used (for example, taxonomy "collisions").

Stories have a "teaser" or opening statement intended to grab the reader's attention. The length of the teaser is set in one of two ways:

- In `Administer>>Content management>>Post settings`. The default there is 600 characters. You can change that.
- By specifically identifying a break point with `< !--break-->` [without the space] in your content (before the default limit).

<u>Note</u>*: You may see some places that tell you to use <break> to set a teaser point. This*

> *was originally changed in 5.0 and created a considerable controversy, so it was backed out.*

A story probably shouldn't have a menu entry. If you use the general "convention" that a "page" is for static or generated content that stands alone, and "story" is for collections of related content (e.g. RSS feeds, newsletter articles, press releases, etc.), then a story is usually going to be displayed with other stories, so which one would be the menu item? Generally the menu for a set of stories will be a description of how they are selected for display.

You may want to promote the story to the front page. For your "Welcome" message, you probably also want to make it "Sticky at top of lists." Unfortunately, there is no core "weight" feature, so you have to play with the dates and times in the "Authoring" section to control the order. (Or you can use the Weight module.)

What's a Teaser?

This is from a post by zoon_unit on January 10, 2007.

A "teaser" is essentially a snippet of text designed to tell the user the content of a post without reading the entire post. Since most writers have embraced the common journalistic style of explaining the nature of an article in the first paragraph, teasers work well for most articles.

Here's what happens:

1. A node contains an entire article.
2. Drupal's "teaser" function, "node_teaser," strips the first x number of characters from the article and makes it available as content. The exact length is determined by the value set in Drupal's `Administer » Content management >> Post settings`.
3. So, you list a bunch of articles on a page. You want the articles to display only a snippet of text from the full article, so that you can fit a bunch of articles on a page without requiring the user to page down through tons of text. If the user likes the "teaser" content of the article, they will click on the article's title and see the full content of the article on its own page. In a sense, teasers function like summaries of an article, except that the software decides where to cut off the text. If you want to determine where a teaser article ends, you can insert the comment tag to instruct Drupal exactly where to fashion the break between full text and teaser text.

G3. Creating a Book Page

Generally, I only create **one** book page for each book. This is the first one, usually the introduction. All other pages will be added as "Child pages" (to be accurate, child pages will still have the "book page" content type set for them).

In addition to the things I said about pages, a book page has a "Parent." For the first page, this will be "<top-level>." If you use "Add child page" the "Parent." selection should be already filled in with the book's name. If you create another book page, you would need to make sure you properly select which book or page the new one is to be made a child of.

The top page of the book probably deserves a menu entry. The rest do not, unless they are really special. Remember, a book has its own navigation.

You are currently reading a "book."

G4. Creating a Blog entry

Blog entries are a little different. Assuming you enabled the "Blog" module, you should see a "My blog" entry in your navigation menu. When you click on that, there will be a "Post new blog entry" link on that page.

If the blog is a diary, you probably want to use the date for the title.

If it's a collection of thought, give it a meaningful title.

Type in the content of the entry.

Your blog will always show the most recent entries on the beginning page when visitors view it.

H. Custom Blocks

Following is a simple example of custom block. For more information on blocks in general, see the Blocks page of the handbook.

Adding A 'Contact Information' Block

A business or support group should always let people know how to contact them. One easy thing is to include your mailing address on your pages. This is about the easiest kind of block to start with.

1. Go to *Administer>>Site building>>Blocks*. It should already be sitting on your default theme, but if not, select the right one.
2. Click on the "Add block" tab.
3. Fill in the "Description" and "Body." Here's a sample body:

   Example Organization<br>
   123 Main St.<br>
   Mytown, State Zip<br>
   USA<br>
   (123) 456-7890

4. Save the block.
5. Now you can "Configure" the block to add the block's title and define it's "Visibility".
6. Follow the Configure link next to the block and enter "Contact Information" as the Block's title.
7. Decide if you want to allow users to turn the block on or off, and, if so, which roles should have that ability. You can leave this with no changes to allow everyone to see it. Then choose which pages it will be shown on; Leave this empty to show the block on all pages.
8. Save the block.
9. Now you're back on the block list. Find the block you just created in the list and

choose a "Region". You can use the "Weight" parameter to set its position with in the selected area; again, I like the address at the bottom, so I use a heavier weight.
10. Click on the "Save blocks" button.

Adding a Last Updated Statement

It is fairly common practice, especially on a group site to let the visitors know when the site was last updated. This example requires you, the webmaster or administrator (sometimes called the "super user") to maintain the block. There are ways to automate this, but for right now, we'll do this manually.

- Follow the same process as for the 'Contact Information' block, giving this one a slightly heavier weight to sink it to the bottom.

  ```
  <em>Site Late Updated on Feb. 12, 2007</em>
  ```
- For this block, lets set it to show on every page except the home (front) page. So under "Page specific visibility settings," I clicked on the "Show on every page except the listed pages." radio button, and entered `<front>` in the pages box.

Creating a Newest Posts Block

Creating a newest posts block is easy using Views. With slight modification you can use this to list comments or list nodes by score if you have a rating module installed. The information below is for version 5.X of Drupal. Things may be different in Drupal 6.X

Step 1 - Installing Views

Download install the Views Module. Follow the instruction that come with it.

Step 2 - Block Title

Determine what to call this block.

Step 3 - Number of Posts

Determine how many posts you want in the block. You might also want to decide what information you want to be displayed. For this tutorial we are going to display titles, but you could display the teaser as well. The amount of information that is displayed could affect your choice as to the number of posts to display.

Step 4 - Starting Views

Login to the admin page of your Drupal site and go to Site Building > Views.

Step 5 - Block Basic Information

Once you are on the Administer Views page, click on add. Fill in the basic information - Name, Description, and Access Privileges. The name must be alphanumeric and you must use underscores for spaces.

Step 6 - Basic Block Setting

Creating the block is very easy. Move down the page until you see word "Block" as a hyperlink. Clicking on it will expand the block section.

Click on the check-box "Provide Block". This means that Views will provide a block for you. You will find it in the top of the block section.

Now we want to tell it how to display the output. You have several options but for this the best is "list View". Since we are just going to list the titles. If you wanted more information, you could use the "Teaser List" to provide the name and description of the post. So, where it says "List Type" you will select "List View"

Finally, you need to tell how many nodes you want in the list. Just fill in "Nodes per Block" with the number that you want. If you are using a theme like "Denver", you might want to use 5 for the top regions. If you are placing in a side bat you could use 10 or more. Depends on how fast your content changes.

Step 7 - Selecting Fields

After that is finished, you want to tell views what field you want to display. The next section below "Block" is the "Fields" section. Click on "Fields" to open it. Then select the field that you want. Then click on the "Add Field" button. For this example, we want to select the title of the node. So you want to look for "Node : Title". If you wish you can give it a label like "Newest Posts" or "Latest News".

Step 8 - Filtering

You may not want every thing showing in your latest posts. You can limit it. For that you have to go down the page to "Filters". Click on it to reveal the filtering options. This is where you can select the content type(s) to show in the block. Where it says "Add Filer" select "Node : Type" and then click on the "Add Filter" button. For the operator you want to select "Is One Of". Then you pick the content type from the "Value" drop down list.

Step 9 - Saving View

Once you are all done on this page, save your view.

Step 10 - Displaying the Block

Go to Site Building > Blocks. Here you will see all the blocks. Find your newly created one and decide what region to place it. Depending on your theme, you may have a lot of regions or very few few regions.

Once it is assigned to a region, set the weight. The larger the weight value, the farther down the block will be on the page.

Once the weight is set, configure the block. This will allow you to decide who see the block and on what pages it shows on. A good place to place a listing of newest posts is on the front page. To do that, you want to find "Pages" and put in the text area. Then right above select the "Show only on selected pages" option.

Once you are done making all your setting for this block, save this block.

Notes

You can use this method to show your content in other ways. You could show the highest rated content or most recent comments.

You may want to have a special page for listing your 25 highest rated or 25 newest posts. There are special modules that allow you to do this. You can use "Insert View" or "Viewfield" to put views in pages.

Check out the modules for Views at http://drupal.org/project/Modules/category/89

I. Working with the Menu

Introduction

This will not be an exhaustive piece on the subject of menus. My main point will be to start you on building your menu and giving you a flavor for how it works. A more complete guide to menus can be found at Creating a menu structure.

As we all know the menu, or navigation system, can make or break a web site. It must be easy to follow, often referred to as "intuitive." It must be complete, yet compact.

There is a lot of content on the Drupal site on menus. Most newbies have trouble understanding it all. I've been using Drupal for just over a month as I write this and I still have a long way to go on completely understanding menus.

For beginners it can be very confusing to understand the difference between menus and categories. The menu is a navigation system and categories is a system to order content data. So menus is to arrive at content and categories to order it. Initially to understand Drupal well, you have to see these as two separate things.

With a menu you can point directly to a node, like a page or a story, but you can also point to a term in a category, which would show you a summary list of stories or pages.

How To Menu

Drupal offers three primary ways, which may be combined, to provide your users with site navigation.

1. Textual menu - this is the "standard" line-by-line type of menu, like what you see on most sites, including mine. It can be enhanced in a few ways, such as using a CSS or separate book navigation (as I have done here).
2. Tabbed menu - becoming more popular because it's a little more "gee-whiz" in its presentation. It is debatable as to whether it is any more effective for your visitors. In

Drupal, it is divided into "Primary" (the tabs you always see) and "Secondary" (the part that drops down, or slides out). Not all themes support secondary links.
3. Books - Books are organized separately from menus, but have their own navigation, which you can see on this site. See my section Creating a Book Page.

Textual Menu

The textual navigation is the easiest to understand. As a matter of fact, I still don't understand how to make the "secondary" part of tabbed navigation work the way a lot of people think it should (drop down).

You may see the terms "primary links" and "secondary links" in many posts. This is one area where I find the Drupal documentation confusing (at best). While they sounded great, and I am now using them, they may not be the best thing for someone just starting out. Stick to the standard "Navigation" menu until you have a better feel; you can always go back and change this later.

For the most part, the "standard" menu is best built as you create content, but may require a little tweaking as you see how it lays out.

When you create a page, story, blog, or book page, one of the fields that you may (should) fill in before submitting, is the menu entry (if the node is to have one). You have the "title" (what is to appear in the menu as people see it) and a "description" (what they will see if they hover the cursor over that entry).

I rarely worry about the "weight" until I see how it shows up in the menu. At that point, you can either go back and edit the content you created or go to `Administer >> Site building >> Menus` and edit it there.

Okay, that was the easy part. Now let's say you want this particular content listed in the menu as a
child of some other page. No problem! Let's say you have a subject's introductory page already listed in the menu, for example "Family History." The page you're creating is "1860-1899." When you build the menu entry, you'll notice a selection box labeled "Parent Item." Scroll down the list until you find "Family History." Now when you submit this page, it will be a child of "Family History," making that item an expandable menu item. You just created a hierarchical menu!

Tabbed Menu

In those themes that support this technique, the "Primary" seems to get built automatically as you build the "Navigation" menu, unless you specify a different menu set. I have no idea yet how to make drop down "secondary" links part work - I think it requires a separate module. I do know how to make the secondary links appear in a block, if they exist. I like the way that works, but it may not be for everyone.

Books

The book "menu" is built automatically for you. The only thing you have to worry about is the order of the entries (hint: weight).

The only "complicated" part is turning on the book navigation block, which is done at

`Administer >> Site building >> Blocks`. All you really need to do is to tell Drupal which region to place it in and its relative weight. You can get fancy, if you want, with your style sheets.

More

Later you might consider using the taxonomy_menu module. It will add to your confusion, but it will be good when there are frequent changes in your vocabularies. It will make the difference between menu and categories almost completely disappear, because it allows you to make vocabularies appear as a menu. This way menus will be generated automatically.

If you want hierarchical drop down menus, the nice_menus module might come in handy.

I2. The Contact Form

For something as simple as a contact form, this is one of the most complicated things to get set up and operating.

Set Up

First, the contact module must be enabled. Go to `Administer >> Site building >> Modules` and locate it in the list of core modules. Click the check box and go to the bottom to save the change.

Go to `Administer >> Site building >> Contact form`. Here you can set up the "Categories" - or recipient name/office. [Don't confuse this with taxonomy categories.]

For example, email for the Sales Department might be given a category of "Sales." The email address that the form is sent to may be `sales@mycompany.com`. If you want a reply automatically sent to the person sending the contact email, you can specify that here. Don't worry if you don't know them all right away, you can come back and change this at any time. Click the "Submit button."

Now click on the "Settings" tab. Here you can limit how many contact emails an individual may send in an hour -- this helps limit spamming. You may also turn on personal contact forms here; this allows users to contact each other. Click on the "Save configuration" button.

Make It Accessible

To me this step seems totally unnecessary, but I suppose that some people want it.

Go to `Administer >> User management >> Access control`, locate the "Contact module" entry and enable it for the roles you want to be able to use Contact. Save your changes. The menu link (next step) will not be visible to any one not having access.

Add "Contact" to the Menu

Go to `Administer >> Site building >> Menus`. Under "Navigation," enable "Contact" as described in the instructions. Save your changes.

Using It In Content

To add a link to a content page use `<a href="/contact">Contact Us</a>`. Unfortunately, this does not give you the capability to specify which contact to send it to. Fortunately, there is help! Check out the Contact Forms module (more about it here).

Need a customized contact form? Check the WebForm or CCK modules. A recent change to the Contact_Forms module allows you to use it together with one of these two modules.

Contact Form Spam

If you find you're getting spam emails through your contact form, check out the Gotcha module.

J. URL Aliases

"URL" is an abbreviation of "Uniform Resource Locater." It is a fancy way of saying "my page's address (or name) on the web." It is the "name" by which a browser identifies a page to display. We've all seen advertisements that say "Check us out at abcxyz.com." Well, *abcxyz.com* is a URL for the home page of their web site (well, sort of - there is an implied add-on to that, such as `index.html`).

By default, Drupal calls your content "nodes" and identifies them by their position in your database. So your page on "The History of the Macadamia Nut - Part 1" might be known as "node/167." That's all fine and well for Drupal, because it understands that. But your visitor really doesn't care where the page is in the database; all they want to do is find the page again, or know what the entry in their bookmarks list is.

So Drupal has a feature called "URL Alias" that allows you to provide a more understandable name to the content. As far as browsers, servers, and search engines go, it is totally unnecessary. But for humans, it is nearly mandatory. This is why I tell people to always turn on the Path core module, which supports URL Aliasing. (I will mention another module shortly, called Pathauto.)

So, just before you submit that exhaustive treatise on macadamia history, and if you have the Path module enabled (I told you that you'd want it), then you'll see a section on the edit page that says "URL path settings." So let's say you want your visitors to see it as `http://www.mysite.com/MacadamiaHistory.htm`. In the URL field, you enter `MacadamiaHistory.htm`.

Now, some people will argue that putting that ".htm" on the end is not necessary. That's absolutely true. But then it's not necessary to do any of this. My opinion is that if you want your visitors to see a "normal" name, it should look like a normal WEB name. Web pages have some kind of type, such as ".htm" so we should too. But it's your preference.

Oops, forgot to do this before you submitted the page? Not to worry - Drupal to the rescue!

First, visit the page you created. In your browser's address field, you'll see its URL. On the end it will probably say "node/*xxx*" where *xxx* is some number. Write down that number. Now go to `Administer>>Site building>>URL Aliases`. There's an "Add Alias" tab at the top. In the top box (sorry, they seem to have dropped a label in 5.1), enter "node/*xxx*" from above. In the second box, enter "MacadamiaHistory.htm".

Now go back and visit that page and look at your browser's URL field.

If your site is going to have lots of content, particularly user-submitted content, you might want to looks at the PathAuto module. Not only will this module automatically generate URL aliases for new content (according to rules you can set up), but can even go back and change aliases in bulk.

K. Moving Entire Drupal Site with Databases

Start with Moving entire Drupal site with databases

This might help too: Backing Up Your Database

I know the "official" process is to load the "tar" file to the host and unpack it. As an alternative, one can copy the code directly from your test site on your PC. I've tried uploading code from my machine twice with disastrous effects. (Okay, so I was born blonde!) So, I've worked out my own process that works. [By the way, this also reduces the bandwidth usage for those who might be running short on that.]

What has always worked for me is to install Drupal with cPanel/Fantastico (I know there are some people out here who will speak ill of Fantastico, but it has always done me right). Now, to be fair, one needs to understand what Fantastico can and cannot do.

Fantastico only supports **core** functions. It does not support custom themes, contributed modules, or custom code (outside of content). Fantastico does not use "update.php" (as a matter of fact, doesn't even load it), so it is not a good idea to use it to upgrade your installation. As long as you understand this, you're much less likely to have problems.

Make sure that the correct versions of modules and themes are working on your test site.

Here's my process: (by the way, this should pretty much work for changing hosts as well)

1. Keep a pencil and paper handy to write down what changes you have to make. You can use this if you need to restart, or to think about changes to your site that slow down moving to new releases.
2. Use phpMyAdmin (on some hosts, it's hidden under MySql) to back up your live database. If you have a site that is actively receiving new content, you may have to put the site into maintenance mode to prevent losing new content.
3. It would also be a good idea to back up your test database, just in case...
4. Import the database into your test site.
5. Download any pictures you've uploaded and any folders that are created by the modules you have installed.
6. If your live site and test site are not at the same version, you will need to run

"update.php."
- Open your browser to your test site. Don't panic if everything looks strange, or even blank.
- In the URL bar, append "update.php" and press Enter. If it says you don't have the access rights, don't worry.
  - Using Notepad or similar text editor, open "update.php".
  - Near the top, you will see:
    ```
    // Enforce access checking?
    $access_check = TRUE;
    ```
  - Change it to `FALSE` and save it.
  - Try "update.php" again.
  - If you have any errors listed, search the Drupal site for fixes. There's a good chance that someone else has had the problem. If it appears to be a new error, post a request for help. Take your site out of maintenance mode and plan to start over when it's fixed.
  - Change `$access_check` back to TRUE.
- If you had a custom theme, you probably need to re-enable it for the site to look right.
- Now check out all your module settings. They'll probably be okay.
- Look through your content, especially the front page. Make sure your menus are correct.
- Check out any content that uses custom code, especially if you are changing Drupal versions.
- Does everything check out? Good, you've done all the real work. Now the easy(er) part.

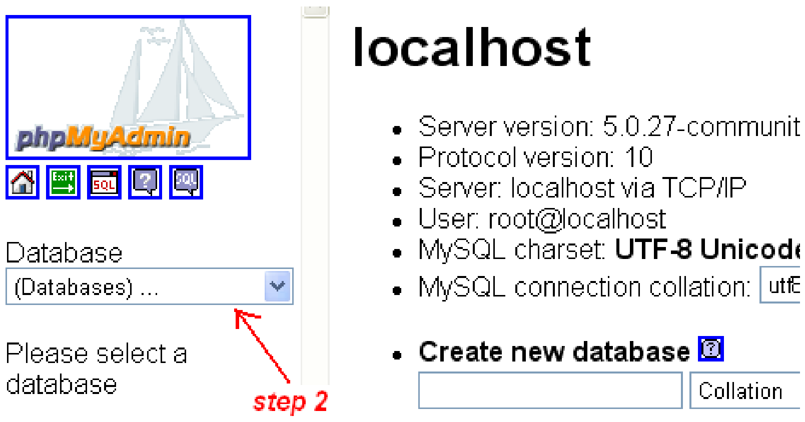
7. Okay, now it's time to do the damage. For me, the rest of this takes less than 15 minutes, so the outage to your users is minimal.
8. If your current site was installed with Fantastico, delete the current installation. If not, then you have to manually remove all the folders.
9. Now tell Fantastico to install the current version. Leave the directory field blank to put it in your "root" directory (probably "public_html"). The userid and password you supply will be your "super user" (user/1).
10. When it is done, I always tell it to email me the installation summary.
11. If you have any customized themes, non-core modules, or pictures, upload them (FTP) to the correct places on the server.
12. If there were pictures or module-related folders, upload them now.
13. When that's done, use phpMyAdmin on your test site and create a back up.
14. Then go to your cPanel again and invoke phpMyAdmin. When you get there, select your Drupal database and then "Import." There's a section to locate the back up on your local computer (Browse). Find it, and then Click on the "Go" button.
15. If you have a customized theme, go to `Administer>>Site Configuration>>Themes` and enable it. [***Hint***: Do not panic if your site looks weird at first. Remember you can always login with `http://www.mywebsite.com/?q=user` (yes, "user," not your ID.]
16. If your test and live directories are not identical, you may need to update IMG links, etc.
17. You should be in business.
18. One more little bit of business if you used Fantastico: go to the main Drupal directory on your PC and copy (FTP) the `update.php` script to your server. I don't know why they leave it out, other than they don't use it. You may need it later.

Backup Process with phpMyAdmin

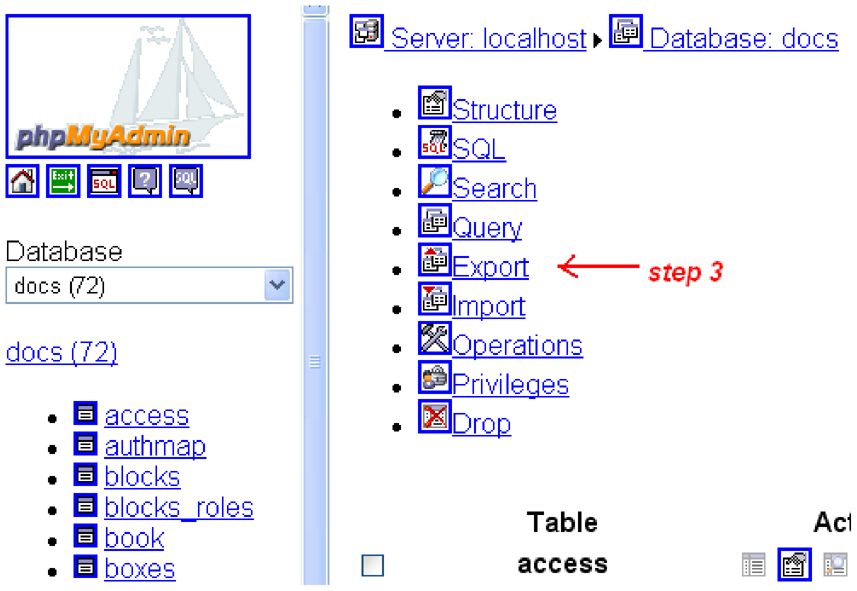
Backup Process with phpMyAdmin

The screen shots here are from phpMyAdmin 2.9.2, which I have on my local computer. My web hosts have 2.9.0.2 and look very similar. They certainly produce exchangeable results.

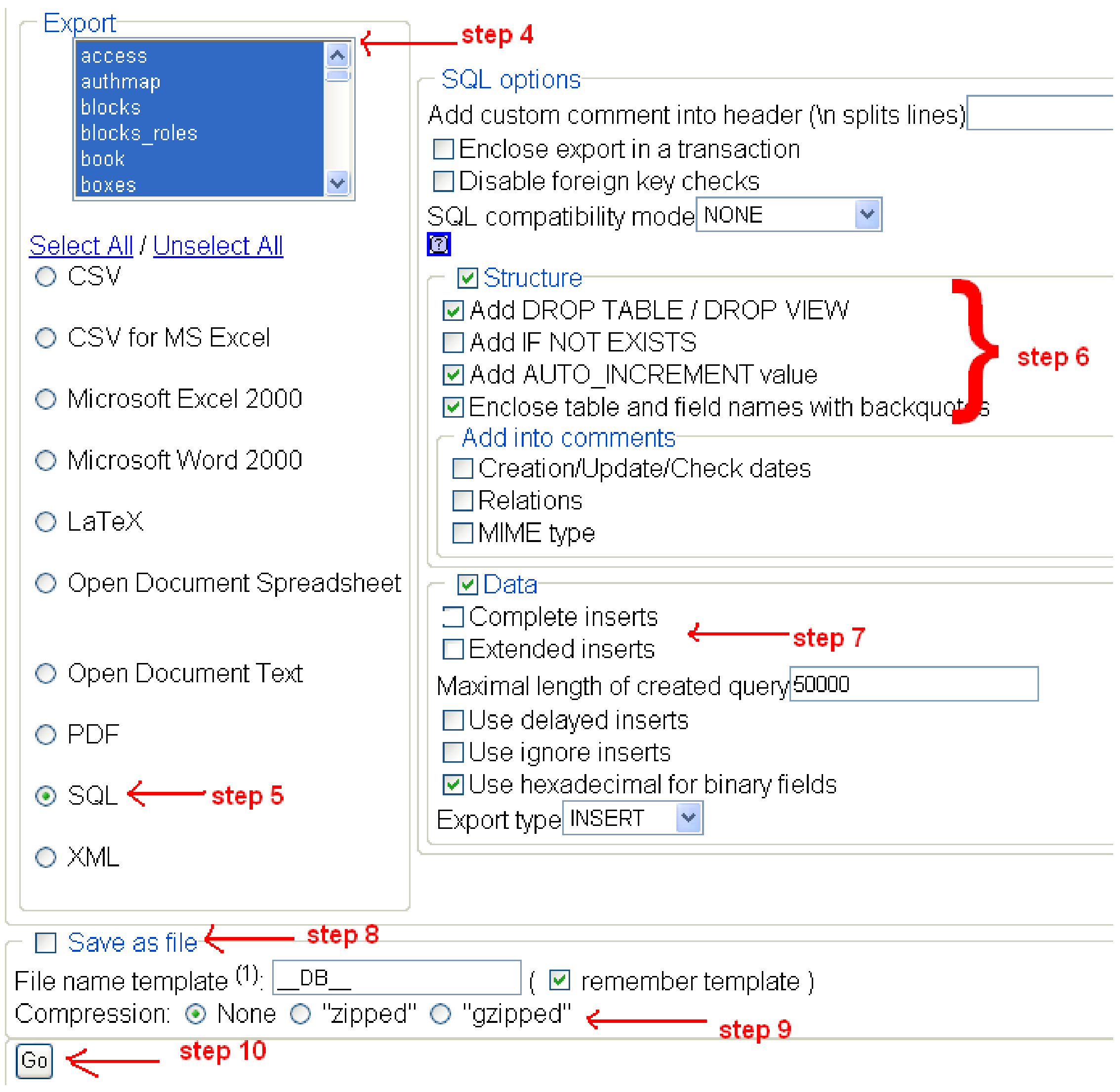
1. Log into phpMyAdmin on your server.
2. Select the name of your database - or your Drupal database if you have several databases.



3. The next screen will show you all the tables inside your Drupal database. Ignore those, and click the "Export" tab on the top set of tabs.



4. Look at the left box at the top of the Export section. All the tables in the database you selected are in that box.



Export
access
authmap
blocks
blocks_roles
book
boxes

step 4

Select All / Unselect All

CSV
CSV for MS Excel
Microsoft Excel 2000
Microsoft Word 2000
LaTeX
Open Document Spreadsheet
Open Document Text
PDF
[✓ SQL] step 5
XML

SQL options
Add custom comment into header (\n splits lines)
Enclose export in a transaction
Disable foreign key checks
SQL compatibility mode NONE

[✓ Structure]
[✓ Add DROP TABLE / DROP VIEW]
Add IF NOT EXISTS
[✓ Add AUTO_INCREMENT value]
[✓ Enclose table and field names with backquotes]
step 6

Add into comments
Creation/Update/Check dates
Relations
MIME type

[✓ Data]
Complete inserts
Extended inserts
step 7
Maximal length of created query 50000
Use delayed inserts
Use ignore inserts
[✓ Use hexadecimal for binary fields]
Export type INSERT

Save as file step 8
File name template (1): __DB__ ([✓ remember template])
Compression: [✓ None] "zipped" "gzipped" step 9

Go step 10

   - If you have other programs that use the database, then choose only those tables that correspond to your Drupal installation.
   - If you only have Drupal installed, in the left column, click "Select All."
5. Ensure that the SQL button is selected too.
6. In the Structure section tick the following boxes: Structure, "Add DROP TABLE," "Add AUTO_INCREMENT," and "Enclose table and field names with backquotes."
7. In the DATA section leave the boxes inside this section unticked, but make sure to keep the check box next to the "DATA" heading checked. If any of the boxes is pre-checked, uncheck them.
8. Tick the "Save as file" option, and leave the template name alone.
9. For now, select "None" for compression.
10. Now click "Go" and you should be prompted for a file to download. Save the file to your computer. Depending on the database size, this may take a few moments.

You have now backed up your database! (Not so bad, was it?)

Once that download is complete, you may check the "zipped" option, click "Go", and download the next file. If you want, you can download a backup in each of the compression formats. Your choice.

L. Moving Stuff to Your Web Site

Occasionally, you need to update something other than your content (database), like the style sheets, or new code levels.

FTP stands for File Transfer Protocol. It is pretty much the standard way to move stuff (files, code, style sheets, etc.) to your site. This is not how you will create content; that goes into the Drupal database.

Now, the postings on the Drupal site will tell you to get a fancy FTP program. If you are at least on Windows 2000, there is absolutely no need. You can do it with Windows itself and it will look just like another folder and you can drag and drop. I find this process to be easier if you use Internet Explorer than some of the other browsers.

In your "Address" bar for your browser, enter `ftp:username:password@ftp.domain.com` where *username* is your user name at your site (not Drupal), *password* is the password for that username, *domain.com* is your site's domain name (without "www").

This will take you to your site. In Linux, you'll probably have to click into the "public_html" folder to get to your Drupal stuff. Then just navigate to the correct folders from there.

M. Setting Up Cron

I could not get a decent answer from the Drupal site on setting up the Cron jobs that it keeps complaining about. Nor would my hosting support people help ("That's a user problem...").

My hosting provider does not allow me to have Shell access (probably a wise move). But they do provide the more-or-less-standard cPanel function. On my version, the "Cron" entry is in the lower left.

I also had WebCalendar installed on one of my sites. When I went to **cPanel**, I noticed that WebCalendar had a command already set up. Modifying it a bit, this is what I came up with to put in:

```
cd '/home/<i>username</i>/public_html/' ; php -q 'cron.php';
```

Note that *username* is my host's user ID for domain management and my Drupal installation is in my "root" folder (actually "public_html").

This worked for getting Cron run, but did generate some error messages.

I was happy that Cron ran, but a bit concerned about those messages. So I did some searching on the Drupal site and came up with several posts of the same messages, but no solutions. So I posted again. This time someone saw it through.

They suggested using WGET, but I don't have shell access. But I did, for some reason, check the "Advanced" mode on cPanel again. I noticed that there was now a helpful hint there (of course in a small font). It said to use `GET http://nanwich.info/cron.php` (obviously, use your own URL). I did and the error messages went away and Cron is working great!

For Cron jobs another possibility is http://drupal.org/project/poormanscron

> For every page view, this module checks to see if the last Cron run was more than 1 hour ago (this period is configurable). If so, the Cron hooks are executed, and Drupal is happy. These Cron hooks fire after all HTML is returned to the browser, so the user who kicks off the Cron jobs should not notice any delay.

Setting up Cron on Hostmonster through the cPanel interface

1. Logging into the cPanel interface for your hostmonster account, scroll to the bottom and look in the "Advanced" section
2. Select "Cron Jobs." It is represented with a calendar icon
3. Select "Standard"
    - Do not use an email address or you will be hit with a notification every time the cron job runs
4. Enter the following command: php /home/+++Your hostmonster login username+++/public_html/cron.php
    - This is only for your main domain, subdomains are in different folders
5. Select the number of times that you would like the cron job to run
6. See if it works -you can check whether or not a Cron Job has been run in the Logs menu in Drupal

N. Categories (Taxonomy)

You will see a lot of posts on the Drupal site about creating and using a Taxonomy (or "vocabulary" and "terms"). Most of those posts will be over your head - many are still way over my head.

While it is true that the more content you have the more obvious the need for a taxonomy becomes, there is certainly no reason why a smaller website that has things to classify can't use it.

But to try to help you get a slightly better idea of how to use them, I'll use a case study here.

The **Recipe** module is probably a good example. On one of my sites, in order to foster a bit more "community" feel and encourage visits to my site, I decided to add a group cookbook (a real one, not like this book). The recipe module does that.

It didn't take me very long to realize that entering a bunch of recipes without any organization would get messy pretty quick. Well, recipes fall into several categories: Appetizers, Entrées, Desserts, etc. So let's set up those things as a "vocabulary" with which we can organize the recipes.

1. Go to `Administer>>Content management>>Categories` and click on the "Add vocabulary" tab. [In Drupal 6, the menu path is `Administer>>Content management>>Taxonomy`]
2. Enter the name, for example "Recipes." Then a "Description" like "Our community

cookbook."
3. Select the type of content this applies to. The Recipe module introduces a "recipe" type.
4. I selected a "single" hierarchy. Later on, if the number of contributions gets large, I can always add sub-categories (like "Beef," "Poultry," and "Pork") and change to multiple level hierarchy.
5. I then selected "Required" to force the users to choose a category for any recipe they enter.

That's it for the "vocabulary," so click on the "Submit" button. You'll go back to the Categories list. You should see your new vocabulary listed.

Towards the right, you'll see a link to "**add terms**." Click on it.

1. Since this is a single level hierarchy, the "Parent" should say "<root>."
2. In "Term name" enter your first term, such as "Appetizers."
3. Enter a "Description" such as "Things for before the meal."
4. Don't worry about the rest yet, just click the "Submit" button.
5. Keep adding the rest of your terms ("Salads," "Soups," "Side dishes," etc.

Now when a user goes to `Create content` and selects "Recipe," they will be required to choose one of these categories for it. And if they go to the "Cookbook" menu item (which was created by the Recipes module) they'll see a list of categories that they can browse.

That wasn't as complicated as all those posts sound like, was it?

For another example, I run a web site for a group that has members submit articles for a monthly newsletter. They wanted a way to organize it so that they could go back and review it by date or topic. So I created two vocabularies, one for the issue date and one for the topic. Now they can, with one click, read all the book reports, movie reviews, or humorous articles. And, every month when they submit new articles, they automatically show up in those lists. [By the way this was done with a custom content module that is easily adapted to other uses. It is available through me.]

Another use is to categorize collections of FAQs, which I do on another site. They had, at the time I set it up, three hard-coded HTML pages with different types of FAQs; no one wanted to touch them. I installed the FAQ module (which is great), and set up the three terms in a single vocabulary. That allowed them to actually grow their FAQ library to now contain six categories, and it is trivial for them to maintain it.

Some newbies swear by these articles: Drupal and the New Paradigm and The Power of Drupal Categories

O. Common Problems

There are some problems we all seem to "find;" this section documents a few of them. [Remember to always search before posting on the forums.]

Tables messed up, images not showing, other "strange" problems with HTML - I think every Drupal user finds this problem. Drupal defaults to filtered HTML; that is, only certain tags are allowed. Further, that input format also breaks long lines of text. The fix is real easy: switch to the "Full HTML" input format. I make that the default for administrators (like me). Note that you may still want the URL Filter and Code Filter modules turned on for this input format; they are not defaults.

Help, I turned on Site Maintenance, now I can't login! - About 3 out of 4 Drupal users have done this to themselves (including me). You can still log in with `http://www.example.com/?q=user`. Note that "user" is exactly that - do not put your username there.

I don't want anonymous users to see "Create content." - "Create content" is actually a child menu item of "Content" which is usually disabled. Go to your menus administration screen and enable the "Content" parent above "Create Content." Then you will see a "Reset" link appear. Click on that. Once again, "Content" will be disabled, but "Create content" will not be shown to anyone who does not have the access permissions to do so (especially anonymous users).

I enabled the Contact module, but if I click on it as an anonymous user, I get the Access denied page. - You've missed a step. Go to `Administer >> User management >> Access control` and scroll down to the contact module section. There you need to click on the check box for "access site-wide contact form" for "anonymous users" (and, I would assume, "authenticated users").

Also read the handbook section Troubleshooting FAQ

P. Links and IMG

Yes, you can link between pages in Drupal. It's the same as not having Drupal, except the name may be weird if you're not using the Path module (URL Aliasing).

The biggest mistake people make is not knowing that there needs to be a leading slash ("/"). Omitting this will probably cause a "404:Page not found" error and, depending on which browser you're using, additional problems, like being logged out.

My home page on this site is node #4, so a link to it would look like this: `<a href="/node/4"&gt; Home</a>`, but with URL Aliasing turned on, I can also code it like this: `<a href="/home">Home</a>`

The picture on the "Accessing Your Site" page is created with this tag: `<img src="/files/pictures/Docs/WDP.jpg" align="right" hspace=1>`

Q. Additional Tips and Tricks

This section gives you some ideas for improving your site or making life easier for you.

Q1. Tracking Module Status

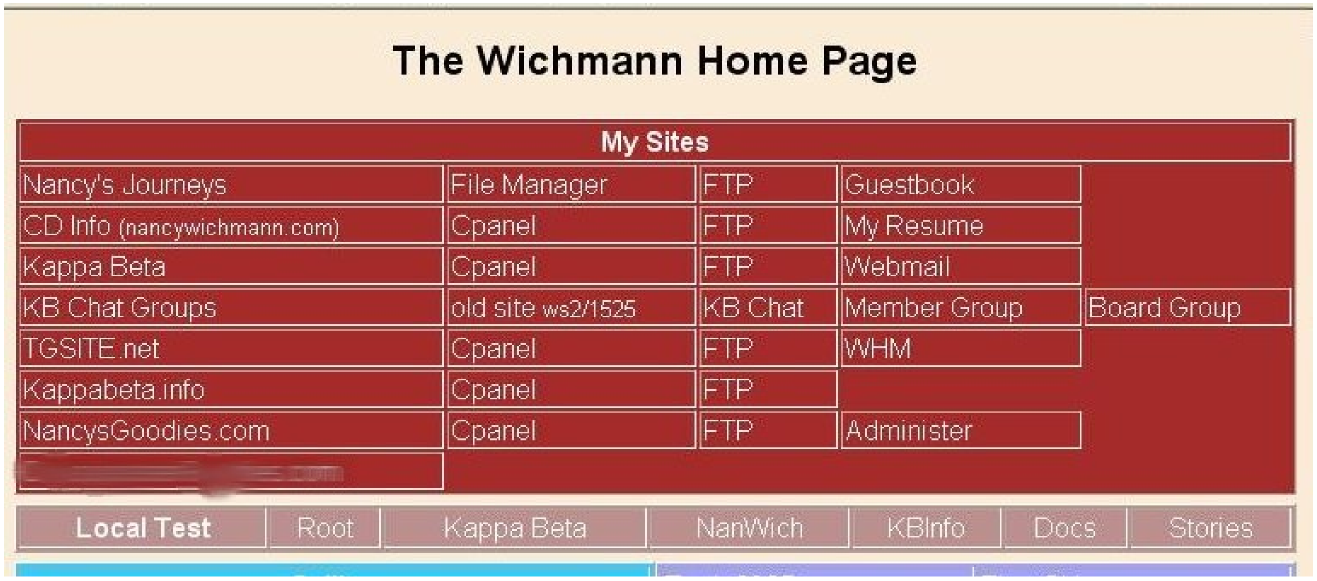
A "site" you might want to build on your PC is a "catch-all" site for monitoring module and theme status with the Update Status module.

This module will check with the Drupal site and let you know if newer releases of your modules are available.

Before you pooh-pooh this idea, let me suggest this additional tip: Many people eventually find that they need (want) to patch, tweak, or otherwise modify modules or themes. If you create a "catch-all" site, you can place unmodified code here so that you always have a "clean" copy to fall back on. You can then copy it to the correct site before changing it. Now, not only do you have the fall-back plan, but you get the added advantage of knowing if it needs to be updated.

Q2. Making Multiple Site Maintenance a Bit Easier

A lot of people don't realize that there browser home page does **not** have to be out on the web somewhere. It can be right on your computer.



The Wichmann Home Page

My Sites				
Nancy's Journeys	File Manager	FTP	Guestbook	
CD Info (nancywichmann.com)	Cpanel	FTP	My Resume	
Kappa Beta	Cpanel	FTP	Webmail	
KB Chat Groups	old site ws2/1525	KB Chat	Member Group	Board Group
TGSITE.net	Cpanel	FTP	WHM	
Kappabeta.info	Cpanel	FTP		
NancysGoodies.com	Cpanel	FTP	Administer	
[redacted]				

Local Test	Root	Kappa Beta	NanWich	KBInfo	Docs	Stories

Here's part of my home page. As you can see, I have links for all my maintenance tasks (cPanel, FTP, etc.) included, so I don't have to keep looking them up.

I also included links to my test sites, so they are easy to get to as well.

Q3. Controlling User Log In

How do I disable "Create New Account?"

It's in `Administer » User management >> User settings`. Click the radio button that says: Only site administrators can create new user accounts.

How do I disable User Log In entirely, and how would I get in if I do?

First, you can go to `Administer » Site building >> Blocks` and change the "Region" setting for the "User Login" block to `<none>`. Now your user login block will disappear.

In order for you to login, I offer two techniques:

- Enter `http://www.myexample.com/user` (or, if you don't use Clean URLs, `http://www.myexample.com/?q=user`). Yes, it is the word "user" -- not your user ID.
- Go to your Site information settings and stick a "login" link in the footer. `<a href="/user">login</a>` I do this on several sites.

R. Keeping Your Local and Remote Sites Synchronized

The point of a test site is to have a place to experiment or develop new content without impacting your users or giving away your latest gee-whiz before its time. But it is also important to make sure that you always have a good working version of your live site on your test site.

Let's ignore, for the moment, that in an "ideal" world you will have a staging site which is always identical to the live site and nothing goes live without being staged.

There are two basic techniques for keeping your sites synchronized:

- Manually making sure that the content of the two sites are the same.
- Frequent uploads or downloads of database back ups.

The first technique is quite time consuming and it's difficult to keep user comments on your test site. It also relies on your memory to know what's been changed, unless you update both sites at the same time. If there are things that are dependent on other things (sorry for the technical terminology), you have to be careful that you update them in the right order so that your random visitor doesn't break something.

The second technique allows for user comments to be downloaded to your test site. From there they can be incorporated into the content and deleted, or left as is. You just have to be careful that you don't lose new, experimental, or incomplete content that hasn't been put onto the production site yet.

Which is better? I can't answer that question for you. You need to weigh the risks and benefits and decide that for yourself. Personally, at my age and with my memory, the second technique is probably better for me, at the risk of losing something not yet uploaded. I also have the bad habit of updating things on the live site and not bring the change back to my test site.

To read more discussion on this topic you can read this thread in the Developer mailing list and this post by Larry Garfield (Crell).

S. More Reading

Now that you're a Drupal expert, there are some additional topics you might find useful:

- Adding Hidden Site Design Notes - how to put design or maintenance notes on your site that only admins can see.
- Core modules - what the many core modules do.
- Contributed modules - a starter list to get you to the right modules, if you need them.
- Site recipes - collection of tricks and techniques.
- Multi-site Installation
- Site configuration challenge: corporate brochure - many ways to get to a "Corporate Brochure" type site.
- Best practices guidelines - a guide to doing the right things.
- My favorite module or theme is outdated. What next? -- it happens!
- PHP and Javascript snippets -- useful code to use as is or adapt to your own needs.
- SQL snippets -- database stuff.
- CSS Tips, Tricks, and Techniques
- Theme developer's guide
- Special cases
- How to write automated tests
- The Road to Drupal Hell -- this should be required reading for anyone who wants to do something not in the core Drupal.
- Using Drupal in an Academic Environment

T. Glossary

A **node** is a container for stuff (sorry for the technical term). Some of that stuff is the content you create. Drupal itself creates a few nodes for its own stuff.

A **module** is a way to extend the functionality of Drupal. It is usually a lot of programmed code (usually in php) and, usually, a style sheet (CSS). For example, if you want to include meta tags to describe your content, you would add on the "Nodewords" module (also known as "Meta Tags").

A **Teaser** a short enticing phrase about the asset (page) to encourage readers to visit the full story. By default, the first paragraph or two of the page content, usually displayed above a "read more..." link.

A **theme** is a means of manipulating and describing how you want your content displayed to your visitors. This includes elements such as your header, icons, block layout, etc. It also includes programming and style sheets.

A **server** is (generally) a computer that provides services to the Internet. These services may be things like running the database or managing the gathering and dissemination of information.

A **browser** is the "program" that you use to display content from the Internet. In reality, it is usually a set of programs, not a single one; it is also a set of tables (e.g. settings) that are used to control its display. Examples are Internet Explorer, Netscape, and Firefox. This operates on the **client**, or user, side of the presentation.

A **URL** (Universal Resource Locater) is the "address" of a resource (such as a page of content) on the web. It is the way the web browser locates your content or site. You will see the URL listed in the address bar on your browser.

A **path** is generally site-specific and refers to the means by which a resource is located.

This could be a full URL (see above), or a relative location (such as "files/xyz/image.jpg" - where "files/xyz" would be the path to the file "image.jpg").

HyperText Markup Language (**HTML**) is the standardized language of the web. It has its own "vocabulary," consisting of tags, elements, and descriptors. A **tag** is the basic component and is used to say, "The following content is to be displayed according to these rules." An example of a tag is a level one heading (`<H1>`). Most tags can have additional information to tell the browser more specifically how you want it to render the content. This specification is called an **element**. Most elements require more information to make them work, this is the **descriptor**, which really should be called "value." For example, if you want that heading centered, you would use the "align" element and give it a descriptor (value) of "center." So , completely constructed it would look like this: `<H1 ALIGN="CENTER">`.

A **Taxonomy** is a way of characterizing stuff. It can be used for grouping, selecting, and protecting stuff. Many people who are new to Drupal think this is a very difficult subject (admittedly, we can make it so), however, virtually all of us had an introduction to taxonomy in school: classifying living creatures (i.e. the Linnaean taxonomy). In that taxonomy, we classified living things according to kingdoms (plant or animal), phylum, class, and so on, down to genus and species. In reality there is an additional classification below species; sub-species (animals) or varietal (plants). [Oh, yeah, I vaguely remember that! That's a taxonomy?]

In Drupal, the highest level of taxonomy description is the "**vocabulary**;" it is used for defining the terms, or tags, that actually end up on your stuff to be used for the various purposes. In the above example, think of "Living things" as the vocabulary. Each vocabulary has one or more "**terms**" that are used to tag (i.e. define, or describe) your stuff. Terms may be hierarchical; that is they may exist in levels. Genus and species would be hierarchical terms. The vocabulary is assigned to input types (e.g. stories, recipes); terms are assigned to a given piece of content (e.g. "Groundbreaking Research on Macadamia Nut Yields" or "My Fabulous Macadamia Brittle"). Notice that I said "terms" - plural - because an individual node may have more than one term associated with it; for example, the "Research" news article may be assigned to "Nuts," "Trees," and "Harvesting." It could then be viewed through any of those terms (or keywords).

Breadcrumbs is a term borrowed from Hansel and Gretel, who left crumbs of bread along their path so they could find their way back out of the forest. In current computer parlance, it refers to the section, usually near the top of the page, that shows the path you followed to locate the current page. For example, it might show `Home > Macadamia Nuts > Current Events > News Articles`, meaning that you started at the home page, clicked on "Macadamia Nuts" in the menu, then selected "Current Events" in the sub-menu, and finally selected, "News Articles."

A **database** is a container for containers. It is a collection of related "tables" that are generally used for a single application (such as Drupal). A **table** is a collection of data used for a specific purpose within that application, such as identifying users. Within a table, each individual grouping of data is referred to as a **row** (or in traditional terms, a "record"). Each row is identified by one or more **keys** that allow easy retrieving of the row. Each row is then broken down into **columns** (often called fields, although this is more appropriate for forms on which the data is displayed). A column holds a specific piece of information for the row, such as a user name or country.

Now, just to complicate things a bit, some times we describe the collection of **item**-related table rows with a collective term. One such term is "node." For example, information about a page on your site may exist in several tables; yet we describe all of this as a "node."

Still confused? Let's try to relate this to an example you're probably familiar with. Let's relate this to your windows computer.

Your [hard] disk (or disc) is sort of like a database; it is a collection of your data. On that disk, you have folders; they are analogous to tables within a database. Inside those folders, you have documents or programs; these relate to rows. Within the document (e.g. a Word document), you have paragraphs; these are much like columns.

Okay, let's add to the analogy a little. Word or Lotus 1-2-3 would be your theme, as they describe and manipulate the content before it is displayed to you. It's a bit of a stretch for several reasons, but you can then think of Windows itself as your browser, since it is responsible for the final rendering of the content to you.

Does that help a little?

The Structured Query Language (**SQL**) is a standard specification for how database engines locate data that you want. An example might be `SELECT country FROM user_profile WHERE username = "Nancy"`; this would get the value from the column "country" in the "user_profile" table using column "username" as the key.

See also:

- Terminology for a more complete list of Drupal terms