

Migrating a Site from Stage To Live

Session Outline

A. Assumptions: This session assumes that you have a drupal site in a development environment or other environment that needs to be moved.

B. Considerations: It is important to consider the target environment (where you want to move your site to) in relationship to your development environment.

1. Does your development environment match your target environment?
 - a. Linux
 - b. Apache
 - c. PHP the same version
 - d. MySQL
 - e. Same drupal version

C. The Backup and Migrate Module:

1. Download the module from
http://drupal.org/project/backup_migrate
2. Install it onto your development site.
 - a. Unzip the file and place it in your `/sites/all/modules` folder
or place the zipped file in your `/sites/all/modules` folder
and unzip it there. Discard zip file.
3. Turn on the module in your modules directory. Find it at Administer → Site Building → Modules (`admin/build/modules`) under the “other” tab
4. Save

D. Configuring The Module in your Development Environment

1. Find the module configuration settings at Administer → Content Management → Backup and Migrate
(`admin/content/backup_migrate`)

2. Backup/Export database
 - a. Default settings for table exclusions and data exclusions are usually sufficient. If you have errors, you may want to exclude certain tables. The data exclusions streamline your file size by eliminating unnecessary data. It is not recommended that you exclude any tables. You might, however, consider excluding data from all tables that are “logs,” “cache” data, “devel” data (if you are using the devel module) and “watchdog” if you are looking to decrease size or clean up excess development data for the first migration/launch.
 - b. Backup File Name can be changed. You can use different tokens. You are provided with a list of tokens that can be used under the “replacement patterns” tab.
 - c. Compression: You can compress the files to make them smaller or not. Your preference.
 - d. Destination: You can choose to have it downloaded to your computer or saved to the files directory. I would recommend doing both.
 - e. Timestamp Format is provided. If you would like a different format, it must be a PHP format string and a link to the date page for PHP is provided if you want to look it up.
<http://us.php.net/manual/en/function.date.php>
3. Restore/Import database allows you to browse your computer for a database file and use it to restore your database. You can also choose a backup from your “saved backups directory” by clicking the link at the bottom of the page. You will be able to choose from manual or scheduled backups. The administration screen looks like this:

Backup/Export DB	Restore/Import DB	Saved Backups
Backup Schedule		
manual backups	scheduled backups	
Backup and Migrate		
PoliceOfficersAssociation-2009-07-25T00-00-22.sql.gz	Fri, 07/24/2009 - 21:00	285.47 KB Download Restore Delete
PoliceOfficersAssociation-2009-07-23T00-00-27.sql.gz	Wed, 07/22/2009 - 21:00	279.04 KB Download Restore Delete
PoliceOfficersAssociation-2009-07-21T00-00-27.sql.gz	Mon, 07/20/2009 - 21:00	271.76 KB Download Restore Delete

4. Saved Backups shows you your manual backups and your scheduled backups. You can download them, delete or restore.
5. Backup Schedule allows you to choose how often to back up your database and also how many backup files to keep. These backups will live in your live files directory, so it is recommended that you also do manual backups that you keep locally (in case your entire site goes down, including all of your live files).

E. Copy your sites folder from your development environment

F. Set Up Your Target Environment

1. Place a fresh Drupal download in your target environment
2. Replace the “sites” folder with your copy of your sites folder from your development environment.
3. In phpMyAdmin (on your live server) Create a new database
 - a. type in a name in the create a new database field and click create
 - b. Click privileges tab and create a new user with all privileges if necessary. Make a password for the user and add log in information. Click “go”
4. Import your database
 - a. Click the Import tab in your phpMyAdmin.
 - b. Under Files to import, browse for your database backup
 - c. Click “Go” in the bottom right corner of phpMyAdmin
5. Alter The settings.php File
 - a. Line #92 put in your new information in this format exactly

```
$db_url = 'mysql://username:password@localhost/databasename';
```

 - i. NOTE: localhost may not be accurate for your specific database server. You need to know what your database server URL is, as well as if it uses a specific port. For example, your database server might be `mysql.yoursitename.com`

– in which case “localhost” would be replaced with that name.

b. Save

G. Navigate to your live drupal site at <http://www.yoursitenamehere.com>

1. You should see your site up live!
2. If you don't and see a maintenance page, double check your `settings.php` file and make sure:
 - a. You saved your changes
 - b. Your changes are correct and you have the correct user name, password and database name.
 - c. You have the right syntax (meaning: all the quotation marks, colons, semicolons, etc. are accurately placed and completed).

H. Set up Cron!

This is extremely important and often forgotten. Many of the things that Drupal does so well (site search, actions triggered by events, full use of taxonomy, and even mailings through SimpleNews or other options) depend on cron. Even backup and migrate requires Cron in order for the backups to be fully reliable.

Read about Cron here: <http://en.wikipedia.org/wiki/Cron>

Setting up cron properly is different on different hosts. If you cannot find easy help files for how to set cron up on your site, contact your host and they should be able to let you know what to do.

If you have no other way of setting up cron, there is also a simple drupal module to help you: Poormanscron

<http://drupal.org/project/poormanscron>

I. So, you want your site in a live environment but you do not want it available to the world or search engines?

There may be reasons for you to put a site into a live environment, but want to protect it from the general public and from search engines. One good reason: the site is nearly ready but your client needs to start using the awesome CMS you've configured to enter the content into the site!

- a. The primary way to do this is to use `.htaccess` and `.htpasswd`. The advantage to this method is that it is simple and does not add another module to your site. The disadvantage is that now your site editors/client need two sets of usernames/passwords – one to access the site at all, and the other to log into their account on the drupal site you have created.
- b. Alternatively, there is another simple drupal module that can help you with this: Secure Site

<http://drupal.org/project/securesite>

The advantage to this method is that your client/site editor only has to log in once, directly into their account on the site. By using Secure Site, you essentially set it up so that only users with roles of your choosing have access to the site at all. This also blocks search engine crawlers until you release the site from Secure Site. The other advantage: if your client/site editor forgets their log in information, the module gives them a field where they can request it be resent.

J. If you prefer to zip your files before downloading/uploading them:

You can use shell (Terminal, for example) to zip your site files up so that you only move them as one chunk instead of as many files at a time.

- a. type `ssh FTPusername@domain.tld` and press return
- b. Answer “yes” to “do you wish to continue connecting” and press return
- c. Type in your `FTPpassword` and press return
- d. Type `ls` and press return

- e. Type `tar -czf name_of_file_you_want_to_create domain_directory_wanted` and press return
- f. Type `ls` again and press return, just to check in the list and make sure your new zipped file is there.

Modules to take away from this session:

Backup & Migrate - http://drupal.org/project/backup_migrate

Automate backups, create an easy administrative environment for backing up and restoring your database.

Secure Site - <http://drupal.org/project/securesite>

Protect your website with browser-based authentication that is tied directly to user accounts in your drupal site.

Poormanscron - <http://drupal.org/project/poormanscron>

A module that runs cron on your site during pre-set intervals without requiring you to set up cron on your web host.

Boost - <http://drupal.org/project/boost>

Static page caching for performance and scalability boost, which can also then be used to offer a static version of your site during maintenance “offline” time.

Software to take away from this session:

MAMP – <http://www.mamp.info>

Mac-based apache/mysql/php setup used for running a “localhost” web environment, enabling you to build and test a site on your local machine.

XAMPP - <http://www.apachefriends.org/en/xampp.html>

An apache/mysql/php + option for Windows, Linux and Mac operating systems.

How to find us:

Nicole Bluto –

Twitter: @nbluto

Online: www.nicolebluto.com

Rain Breaw –

Twitter: @sunrainprods

Online: www.sunrainproductions.com

Takeaways from this talk can be downloaded here:

www.sunrainproductions.com/drupalcampla