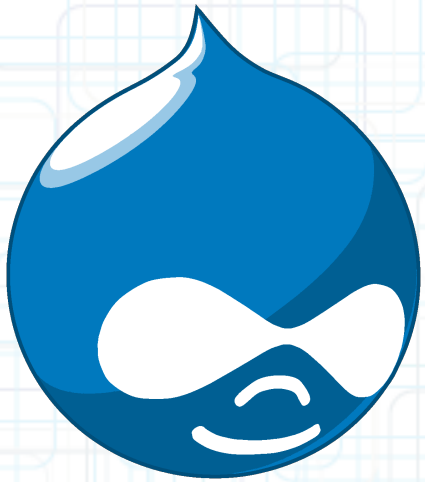
Get Me Outta Here! Converting From Drupal 6 To Wordpress

Presented by
Shawn DeWolfe

What's the Deal



You have a Drupal site and you want to go Wordpress

What's With Drupal?
(Chapter 1)

- Drupal has been around for about 16 years.
- Drupal is less popular than Wordpress, but still popular in its own right.
- Drupal is a CMS (a little like Wordpress)
- It's driven by PHP / MySQL
- It's free.

What's With Drupal?
(Chapter 1)

- The Drupal upgrades come often and they are big affairs
- You have to wait for all of the modules to catch-up.
- You often need to re-code big parts of your site.

- Under the hood, this is all about data in a MySQL Database.

What's good about Drupal

- Very pliable and multi-purpose
- Does things like RSS & Feed aggregation
- Complex content management have been figured out and done well
- Views is a great way to the site's data appear organized and dynamic. (but there is The Loops for Wordpress)
- Lots of corporate support
  (heck-- even Microsoft likes Drupal)

What's bad about Drupal?

- Memory consumption
- Modulitus.
- Modulitus Extremis.
- Drupal 7 has been called the "Vista of Drupal"-- a trade down in the overall quality of the experience from it's predecessor
- You gotta stay on the Upgrade Train

What's so good about Wordpress?

- Easy to install
- Easy to admin
- Easy to export a wite
- Easy to extend
- Easy to upgrade
- Easy to edit content and scripts
- Heck, if you don't have any money, you can even post your site on Wordpress.com

Do I *have* to do this?

There are online services that largely automate this process. You can pay them for what I'm going to describe.

gConverter

According to gConverter, for everyone considering a move to or from Wordpress:

- 81% are moving TO Wordpress
- 19% are using gConverter to *move away from* Wordpress

- 64% are moving to Drupal
- 36% are using it to get away from Drupal

READ THE LABEL!

Some caveats

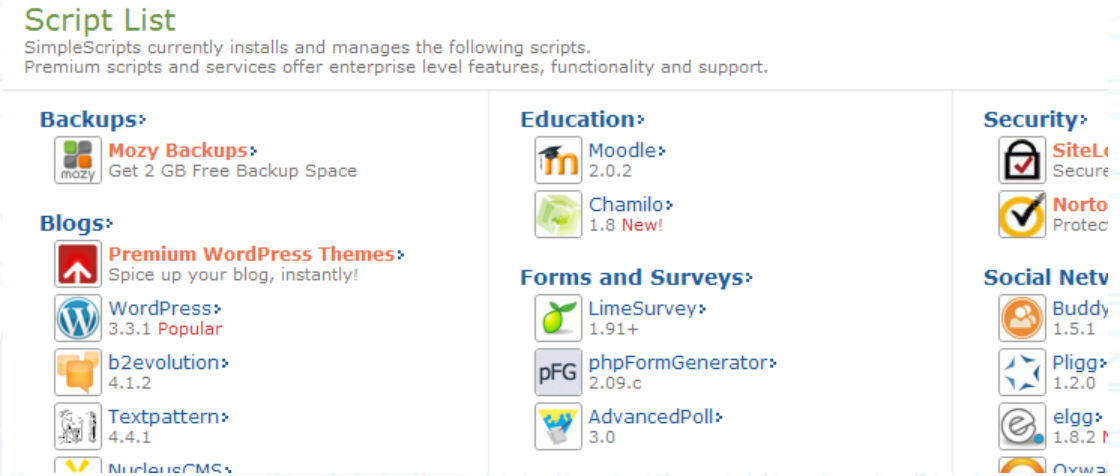
- This advice works for moving from Drupal 6.x to Wordpress 3.x
- It will handle the basic data moving from Drupal to Wordpress
  - nodes to posts/pages
  - taxonomy to tags
  - users will move over

Some more Caveats

- CCK fields *can* move over but it depends on which Wordpress plug-in you're using (custom content type manager or advanced custom fields).
- This won't move modules, you'll have to identify plug-ins to satisfy the roles of modules and install those
- That may require additional SQL import statements to be figured out and executed

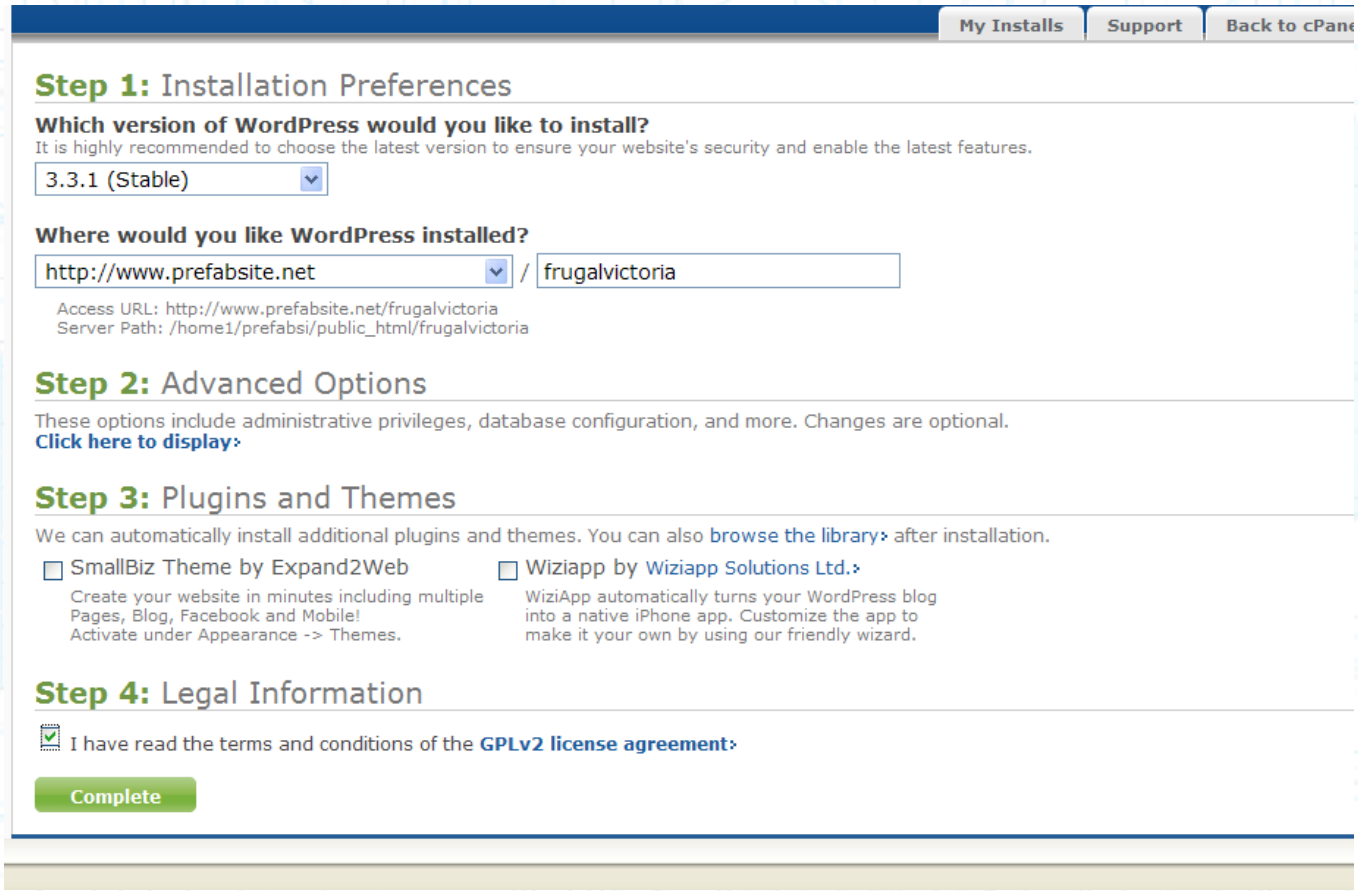
Getting Started

- Make a vanilla install of Wordpress
- DO NOT POST ON TOP OF AN EXISTING / POPULATED SITE (sorry for the all caps)
- The export process will overwrite the post id numbers with the numbering sequences of the Drupal site.



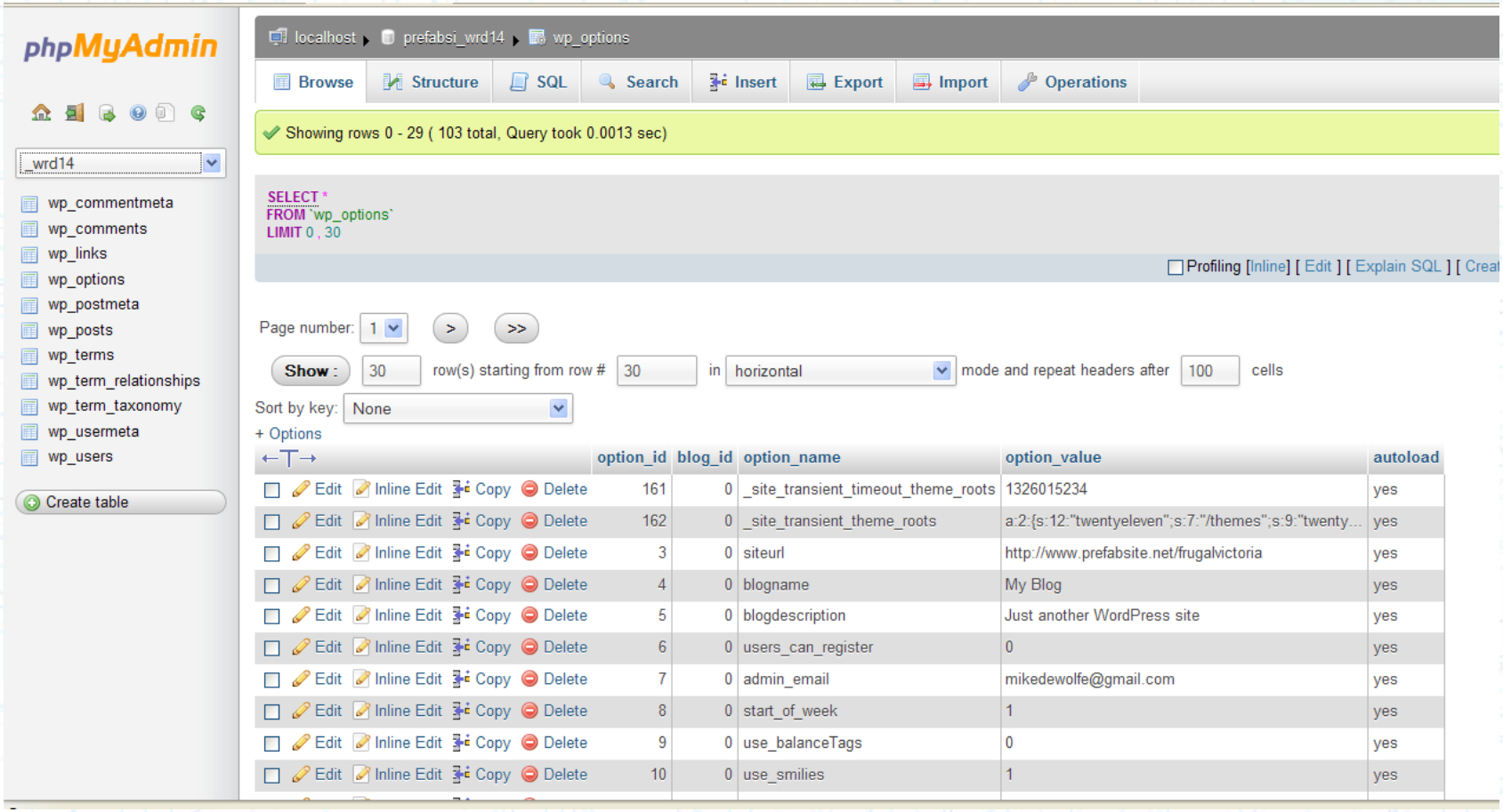
Our Installatron Overlords

- An example of just how turn-key Installatron is:



PhpMyAdmin

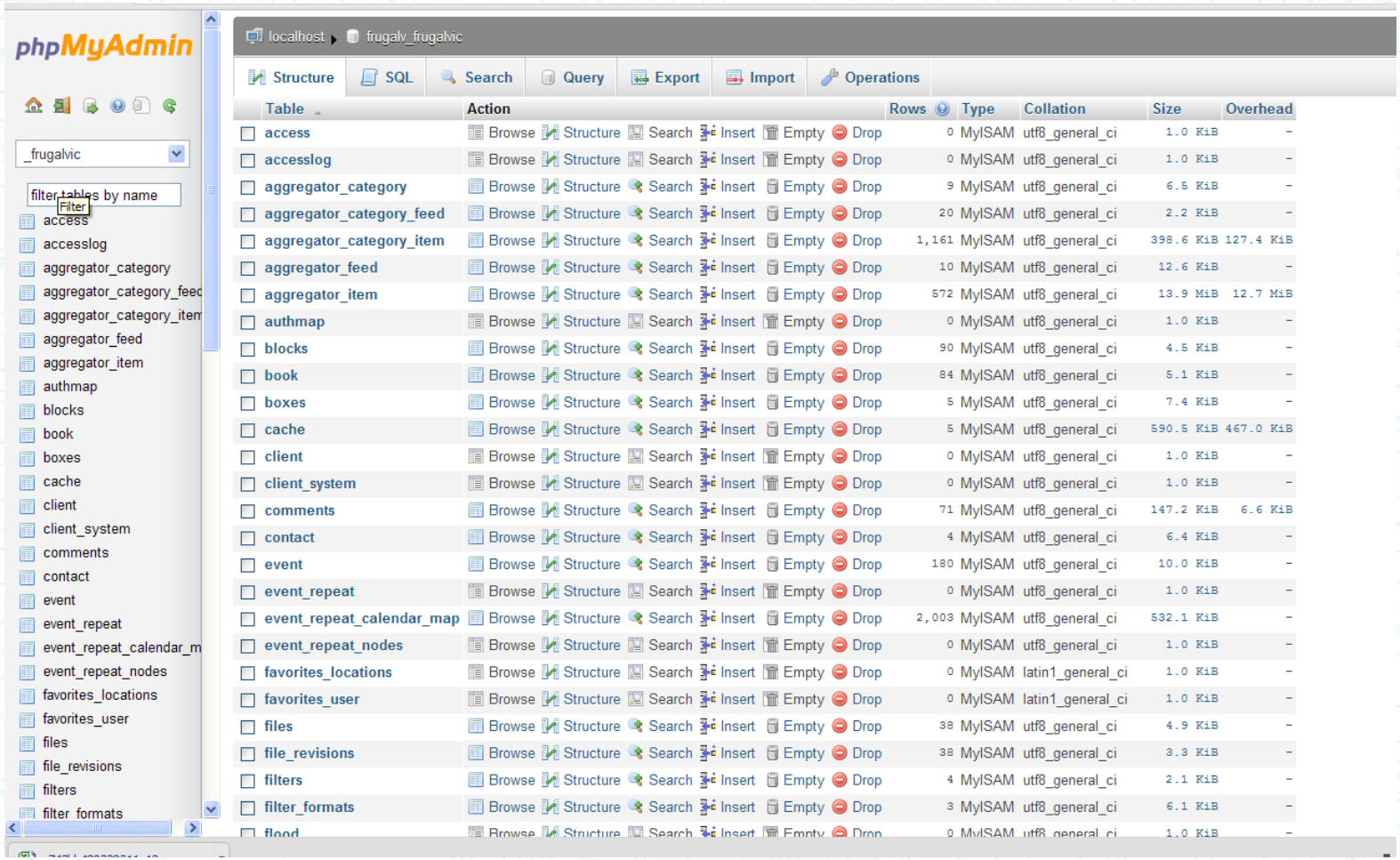
- I use PhpMyAdmin--
  command line was cool... in 1985



The Drupal Export

- Export a copy of your Drupal database. The database naming is very unlikely to mess / clash
- Clear the Drupal cache before you do the export (less data to shuffle)
- MAKE SURE YOU ARE WORKING WITH A COPY OF WORDPRESS RESIDING IN A NEW DIRECTORY AND A NEW DATABASE
- DO NOT DO THIS IN YOUR MAIN SITE (geez: what is with me and the all-caps)

Drupalled Up Database

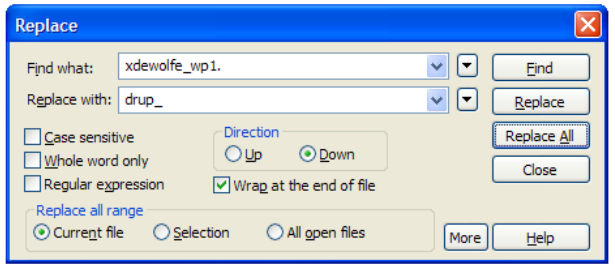


Getting the SQL

- download the Drupal-to-Wordpress SQL file of database statements
- edit the tables names in use so that naming is relevant (in case your Drupal or Wordpress tables have different prefixes)

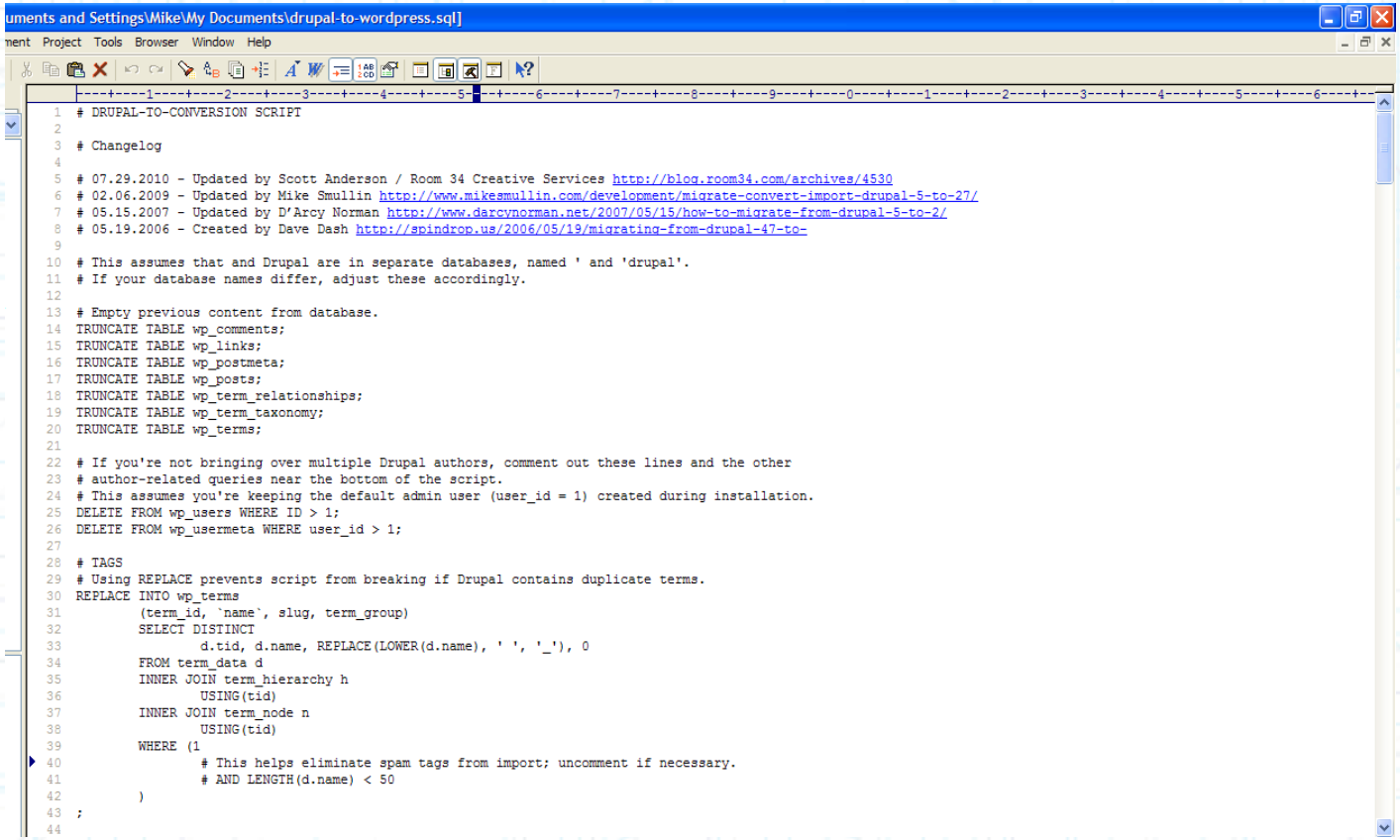
```
(term_id, `name`, slug, term_group)
SELECT DISTINCT
       d.tid, d.name, REPLACE(LOWER(d.name), ' ', '_'), 0
FROM xdewolfe_wp1.term_data d
INNER JOIN xdewolfe_wp1.term_hierarchy h
       USING(tid)
INNER JOIN xdewolfe_wp1.term_node n
       USING(tid)
WHERE (1
       # This helps eliminate spam tags from import; uncomment if necessary.
       # AND LENGTH(d.name) < 50
)


ERT INTO wp_term_taxonomy
     (term_id, taxonomy, description, parent)
     SELECT DISTINCT
            d.tid `term_id`,
            'post_tag' `taxonomy`,
            d.description `description`,
            h.parent `parent`
     FROM xdewolfe_wp1.term_data d
     INNER JOIN xdewolfe_wp1.term_hierarchy h
            USING(tid)
     INNER JOIN xdewolfe_wp1.term_node n
            USING(tid)
     WHERE (1
            # This helps eliminate spam tags from import; uncomment if necessary.
            # AND LENGTH(d.name) < 50
     )
```



The SQL statements

- Here's an idea of what's inside of the sausage



uments and Settings\Mike\My Documents\drupal-to-wordpress.sql]

ment Project Tools Browser Window Help

```
# DRUPAL-TO-CONVERSION SCRIPT

# Changelog

# 07.29.2010 - Updated by Scott Anderson / Room 34 Creative Services http://blog.room34.com/archives/4530
# 02.06.2009 - Updated by Mike Smullin http://www.mikesmullin.com/development/migrate-convert-import-drupal-5-to-27/
# 05.15.2007 - Updated by D'Arcy Norman http://www.darcynorman.net/2007/05/15/how-to-migrate-from-drupal-5-to-2/
# 05.19.2006 - Created by Dave Dash http://spindrop.us/2006/05/19/migrating-from-drupal-47-to-

# This assumes that and Drupal are in separate databases, named ' and 'drupal'.
# If your database names differ, adjust these accordingly.

# Empty previous content from database.
TRUNCATE TABLE wp_comments;
TRUNCATE TABLE wp_links;
TRUNCATE TABLE wp_postmeta;
TRUNCATE TABLE wp_posts;
TRUNCATE TABLE wp_term_relationships;
TRUNCATE TABLE wp_term_taxonomy;
TRUNCATE TABLE wp_terms;

# If you're not bringing over multiple Drupal authors, comment out these lines and the other
# author-related queries near the bottom of the script.
# This assumes you're keeping the default admin user (user_id = 1) created during installation.
DELETE FROM wp_users WHERE ID > 1;
DELETE FROM wp_usermeta WHERE user_id > 1;

# TAGS
# Using REPLACE prevents script from breaking if Drupal contains duplicate terms.
REPLACE INTO wp_terms
        (term_id, `name`, slug, term_group)
        SELECT DISTINCT
                d.tid, d.name, REPLACE(LOWER(d.name), ' ', '_'), 0
        FROM term_data d
        INNER JOIN term_hierarchy h
                USING(tid)
        INNER JOIN term_node n
                USING(tid)
        WHERE (1
                # This helps eliminate spam tags from import; uncomment if necessary.
                # AND LENGTH(d.name) < 50
        )
;
```

Here we... go!

- Here we go:
- The SQL will start by scrubbing out the existing Wordpress install:
- TRUNCATE TABLE wp_comments;
- TRUNCATE .... TRUNCATE some more...
- You will be left with the structure but most of data will be gone.

In With The New!

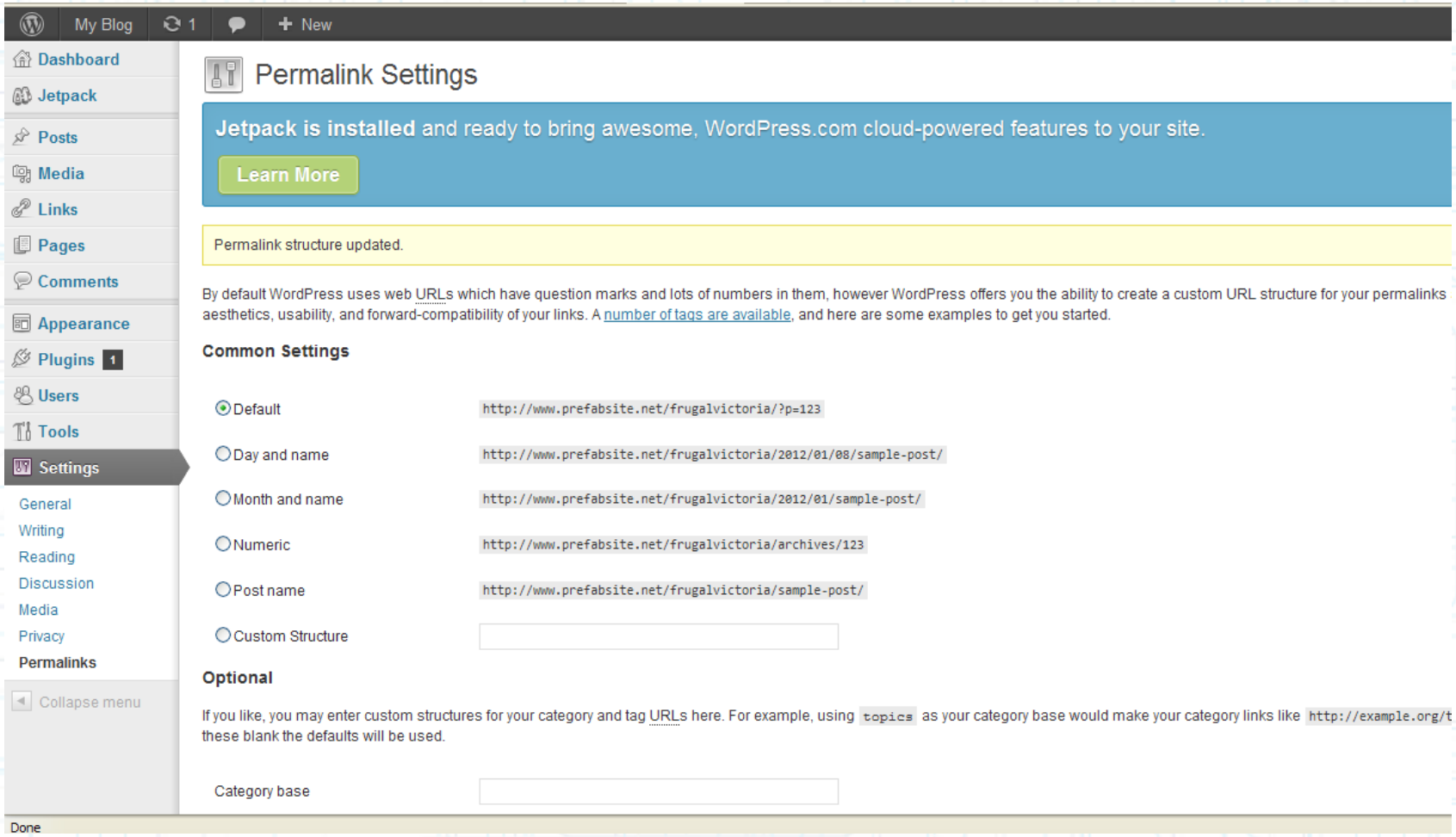
- It inserts tags
- It inserts posts
- It insert the tags-to-posts linkage
- It inserts user references
- this is a step where the users who posted can be imported into the Wordpress incarnation
- OR all user accounts can be imported regardless of activity.

URL aliases
(part 1)

- These are important because if the urls change, people can't find your content.
- Also, search engine will lose the links
- We need to hook the urls
- Install Custom Permalinks. As of Wordpress 3.3 it works okay enough (eg. Permalink amnesia)
- Go to the settings and click on the Default option for permalinks

URL aliases
(part 1)

- Lovin' the Permalinks



URL aliases
(Part 2: the Wrath of .htaccess)

- One gotcha for Permalinks-- the .htaccess file could write multiple copies of the url rewrite <IfModule mod_rewrite.c>
- And/or it can erase the .htaccess
- If it does that, then go in and edit the .htaccess file so that only one reference remains
- After you have it working, chmod it to REMOVE write access so that Wordpress cannot “fix it” later.

URL Tidying

- Put in alias references for all of the posts you have in your system:
- ```
  INSERT INTO `wp_postmeta` (`post_id`, `meta_key`, `meta_value`)
  ```
- ```
  SELECT DISTINCT `ID`, 'custom_permalink', CONCAT('node/',`ID`) FROM `wp_posts`;
  ```
- Then update those that have url aliases
- ```
  UPDATE `wp_postmeta` w, `drup_url_alias` d SET w.`meta_value` = d.dst WHERE w.`post_id` = REPLACE(d.`src`, 'node/', '');
  ```
  ```

Content Tidying
(part 1)

- You can add new content types
- Or, you can convert Drupal content-types into pages or posts
- If a post type ends up orphaned, it will be messed-up:
- You may not be able to edit it
- You won't be able to see the categories and tags.

Content Tidying
(Part 2)

- Here are some examples of clean up SQL statements you may want to run:
- ```
  UPDATE `wp_posts` SET `post_type` =
  'post' WHERE `post_type` = 'story';
  ```
- ```
  UPDATE `wp_posts` SET `post_type` =
  'page' WHERE `post_type` = 'book';
  ```
- ```
  UPDATE `wp_posts` SET `post_type` =
  'post' WHERE `post_type` = 'event';
  ```
  ```

Still More Reading

- Scurker.com - breaks it down http://scurker.com/blog/2010/02/migration-from-drupal-6-x-to-wordpress-2-9x/
- Tom Markiewicz http://www.tmarkiewicz.com/migration-from-drupal-to-wordpress-complete/
- Scott Anderson / Room 34 http://blog.room34.com/archives/4530

Even More Reading

- Mike Smullin http://www.mikesmullin.com/development /migrate-convert-import-drupal-5-to-27/
- D'Arcy Norman http://www.darcynorman.net/2007/05/15/how-to-migrate-from-drupal-5-to-2/
- Created by Dave Dash http://spindrop.us/2006/05/19/migrating-from-drupal-47-to-

Pay Someone Else

- gConverter - for a fee, they will transfer data from one CMS to another
- http://gconverter.com/drupal-to-wordpress/
  - Drupal to Wordpress - they also do this for a fee
    - http://migratetowp.com/

Web321 can also help:
https://web321.co/

- **THANK YOU FOR LISTENING**
-

???QUESTIONS???