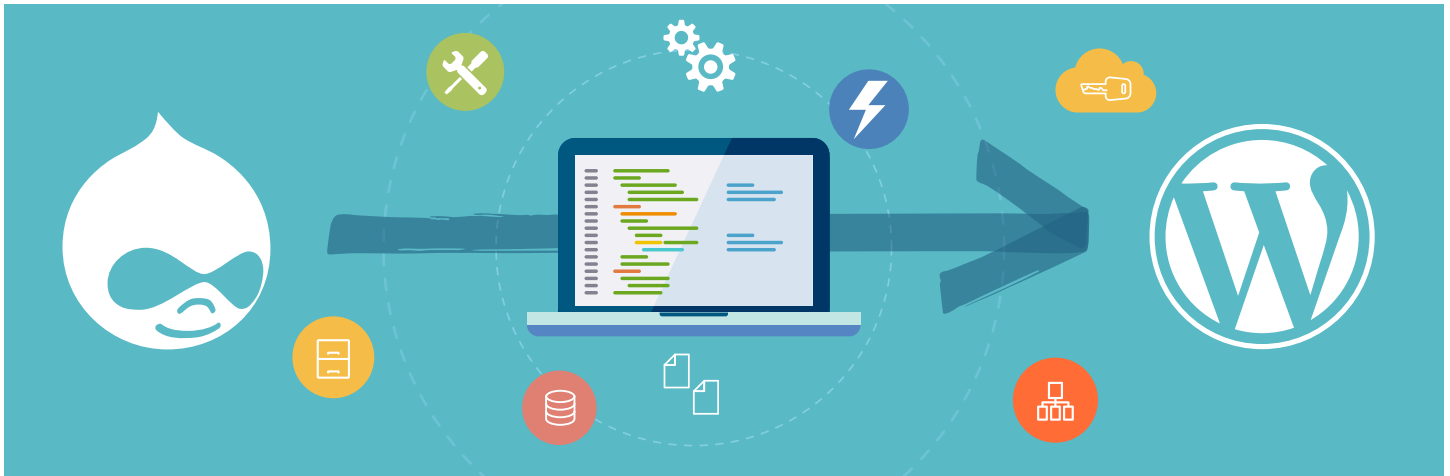




WPengine®

MIGRATING FROM DRUPAL TO WORDPRESS: A GUIDE FOR ALL TECHNICAL SKILL LEVELS

WHITE PAPER



Migrating your Drupal site to WordPress might seem like a daunting task, however rest assured as there are a number of solutions that can make the process a bit less intimidating, whether you migrate the site yourself or access professional services to do so.

If you wish to migrate content from Drupal to WordPress yourself, it will require an intermediate to advanced level of technical skill, along with enough time set aside to run the migration and do post-migration clean-up.

For those with less technical skill, it's advisable to seek out a service that can properly migrate your Drupal site to WordPress for you. Although these services come with a price-tag, it'll save you time and effort in the long run.

First, let's get started with some reasons you might want to migrate your website to WordPress from Drupal.

Why Migrate from Drupal to WordPress?

Here are a few reasons to consider taking advantage of migrating your Drupal site to WordPress...

WordPress is the No.1 CMS.

Reigning king of all content management systems, WordPress powers more than half the CMS market share and **26 percent** of the total Internet.

Out of the top 1 million websites using a CMS, [WordPress powers nearly half of those](#).

It is no wonder then that enterprises such as UBER, Siemens, Hallmark, and National Geographic all run their sites on WordPress. Because the platform is such a powerhouse, it offers an immense amount of thoroughly developed user resources, like 40,000-plus plugins and thousands of themes to choose from, each of which have been refined by WordPress' rather large and advanced community of developers, bloggers, and other end users.

WordPress is built to support high traffic, complex sites.

Some of the world's biggest sites use WordPress, including Forbes, CNN, and [more](#). Among the top 1 million sites in the world, in terms of traffic, nearly [24 percent](#) are on WordPress. Meanwhile, of the top 100,000 sites, more than 21 percent are on WordPress. And, 26 percent of the 10,000 highest traffic sites in the world trust WordPress, which has been proven to scale under the most intense traffic pressures and spikes, even for the most complex of websites, like multi-million dollar ecommerce sites.

WordPress' massive community provides incredible support.

One benefit to being the world's most popular CMS is the large community that follows. In 2015 alone, there were 89 WordCamps that included 21,000 attendees and 1,600 speakers across 34 countries.

Since WordPress is an open-source software, millions of people around the globe can refine the WordPress programming as well as its online documentation, continuously improving the platform and educating those who use it.

WordPress' large ecosystem of designers, developers, and specialists can also troubleshoot any issue or question you might have through its [forums](#) and other forms of support, like [developer resources](#), [online manuals](#), and much more.

WordPress is easier to use.

You don't need to be a web designer or developer to use WordPress. Even if you're lacking in coding expertise, it's still easy to use the platform's intuitive front-end facing qualities to immediately start producing and publishing content (as opposed to Drupal, in which strong technical expertise is required in order to publish content).

WordPress will endure into the future.

Modern development, backwards compatibility and the constant improvement of WordPress allows the platform to grow with your needs. WordPress is also a leader in new technologies, like [WP REST API](#), which extends WordPress' capabilities as a development platform and allows for greater CMS customization.

(While the list continues, for additional information on the benefits of developing on WordPress, check out this white paper, [5 Reasons to Develop Your Web Empire on WordPress](#) and these [key benefits of powering your site through WordPress](#).)

Now that we've discussed some key reasons to use WordPress, here's a few terminology differences between the two.

Drupal vs WordPress: Usability Differences

Many decide to move their site over to WordPress for its ease of use and flexibility, however there are some technical discrepancies between the two. Here are a few different front-end user element differences to know about between the two platforms, and the equivalent of certain functionalities.

DRUPAL		WORDPRESS
Nodes	=	Posts
Fields	=	Custom fields
Blocks	=	Widgets
Modules	=	Plugins

Nodes are how content is stored on a Drupal website and can be in the form of pages, polls, articles, forum topics, blog entries, and so on. This translates to **posts** within WordPress, which serve to deliver content as well.

Custom fields can be added to a post within WordPress, which is the metadata that includes extraneous information. The same goes for WordPress **user fields**, which includes the metadata table that belongs to users. This is the equivalent to Drupal's **fields**, which allow for data to be added to an element, such as the Title or Image.

Blocks are the boxes that can be made to appear in various regions of the Drupal site. WordPress **widgets** are most similar to blocks as they also allow for site customization without having to code anything. Widgets add content and features to your sidebars and allow for greater design control.

To extend functionality Drupal uses **modules**. WordPress uses **plugins** to add on additional features.

Before You Migrate Checklist

Before you migrate your site, there are certain things you need to take care of to ensure it goes smoothly. Use this checklist as a handy guide:

- Back up your site
- Take inventory of your content
- Establish a plan
- Prepare for SEO preservation
- Allot time for the migration

Here are a few key things to keep in mind before starting your migration from Drupal to WordPress.

Back up your site. Be sure to make a backup copy of your Drupal site before you start the migration. Not backing up your site could result in a loss of data should you run into any issues along the way. Be sure to backup your WordPress site too, if applicable.

Take inventory of your content. Evaluate your site and make a spreadsheet of all the content it holds. You might find that a lot of content is outdated and can be left behind. There are many services out there that can help with creating a [content inventory](#).

Establish a plan. Start by asking yourself questions about what exactly it is you're looking to accomplish with the migration process. Do you want to keep your current theme? Or are you looking to revamp the design of your site, and only move content? Will the URL structure change? Know exactly what you want to migrate, including domain name, design, site architecture, content, hosting, and so on. By establishing a plan, you'll be organized enough to ensure a smooth migration process.

Prepare for SEO preservation.

This is perhaps the most crucial thing you do if you wish to retain site rankings, as migrating your site can often result in SEO damage.

To ensure SEO preservation, you can use services like Xenu Link Sleuth to crawl your site and export URLs to help you understand your current site's architecture. If your technical expertise is only beginner level, then you'll want to contact one of the services mentioned later to ask them what measures they take to preserve SEO. (See [here](#) and [here](#) for more info on SEO preservation.)

Allot time for the migration. Migrating a site can consume a hefty chunk of your time, depending on the size of the site and the amount of content being moved over. Make sure that you are conducting the migration during a non-busy time of year for your business.

How to Migrate from Drupal to WordPress

While there is no entirely correct path for migrating your site from Drupal to WordPress, here are some suggested methods to consider, based on your level of technical expertise.

Your Technical Expertise: Beginner

Solution #1: Migration Services

If your coding skills are slim to none, rather than struggle with the levels of technical knowledge covered later on, you might want

to seek out a service to perform the migration for you. There are numerous online services that can provide seamless migration from Drupal to WordPress.

Before you decide on a service, it's important to have a migration plan formed that the services can address, as well as any questions you want answered, like how they are able to retain SEO rankings and URL structures.

Here are a few services worth checking out:

- **CMS2CMS** can migrate your site from Drupal to WordPress in just 15 minutes. Prices are fairly affordable and they offer insurance in case something goes awry during the process. They also offer a free demo as well as live chat assistance. www.cms2cms.com
- **Another Cup of Coffee** offers a Drupal to WordPress migration tool and additional migration services to help move your site from Drupal to WordPress. Services starting at \$1,000, and they'll migrate over pages, stories, taxonomies, users, and comments; the service price increases for additional migration features. anothercoffee.net/drupal-to-wordpress-migration-tool
- **Migrate to WP** will migrate your content in two to three days starting at a little less than \$1,000. For full content and theme migration, expect to pay around \$4,500 for a two to three week migration. www.migratetowp.com
- **gConverter** provides experienced developers that'll migrate your Drupal site to WordPress starting at \$75. Services include Drupal to WordPress conversion of content, modules, and templates. www.gconverters.com
- **CMSHelpLive** has helped websites such as The University of Texas and Harvard migrate to WordPress from Drupal. This service boasts the ability to preserve your SEO rankings and promises no loss of content or functionality. www.wordpress.cmshelplive.com/wordpress-migration.html
- **WordPrax** will migrate your Drupal site to WordPress starting at \$18 per hour. The services claim to provide responsive templates, SEO optimization, widget ready frameworks, and versatility in themes. www.wordprax.com/services/drupal-to-wordpress
- **WordSuccor** can help you migrate your site and claim to provide responsive templates, SEO optimization, and more. Hiring a developer starts at \$15/hr or you can inquire about getting a quote: www.wordsuccor.com

Solution #2: Teach Yourself

If you don't have the money and lack the technical expertise to migrate from Drupal to WordPress, you may want to teach yourself. This [step-by-step video](#) shows how to migrate Drupal to WordPress using a migration script.

For just \$10 it may help to get a visual idea of how to accomplish this task sans coding skills. Although it must be noted that one of the user ratings mentioned that there is no support for images, taxonomy, users, or custom content types. Others however, have [positive things](#) to say about the process.

Your Technical Expertise: Intermediate to Strong

If you have a solid understanding of code and working with SQL queries, then manually migrating your Drupal site to WordPress is a relatively costless way to move your site.

Solution: Database Migration with Queries

If you find running scripts, executing queries, and exporting an SQL file to be a cinch, then you might try converting Drupal to WordPress through database migration.

Here's a standard solution, although you may need to adjust the code depending on your specific needs and end-goals. Listed below you'll find the steps and applicable queries to migrate your Drupal database to WordPress:

1. Make a backup of both your Drupal and WordPress databases.
2. Before converting Drupal to WordPress, make sure that in your original Drupal install the taxonomies are correctly labeled. Step 11 will further discuss fixing taxonomy.
3. Create a new WordPress installation in a different database from your Drupal installation. Name the different databases 'drupal' and 'wordpress'.
4. Clear out previous content from the WordPress database by running this command in your database's SQL queries tab in phpmyadmin.

```
TRUNCATE TABLE wordpress.wp_comments;
TRUNCATE TABLE wordpress.wp_links;
TRUNCATE TABLE wordpress.wp_postmeta;
TRUNCATE TABLE wordpress.wp_posts;
TRUNCATE TABLE wordpress.wp_term_relationships;
TRUNCATE TABLE wordpress.wp_term_taxonomy;
TRUNCATE TABLE wordpress.wp_terms;
```

5. Apply this code to convert over multiple users.

```
DELETE FROM wordpress.wp_users WHERE ID > 1;
DELETE FROM wordpress.wp_usermeta WHERE user_id > 1;
```

6. To migrate over tags, use the following code...
(To ensure duplicate names don't get lost, make sure that the Drupal term_data table has been cleaned of all duplicate names.)

```
REPLACE INTO wordpress.wp_terms
(term_id, `name`, slug, term_group)
SELECT DISTINCT
    d.tid,
    d.name,
    REPLACE(LOWER(d.name), ' ', '_'), 0
```

```
FROM drupal.term_data d
INNER JOIN drupal.term_hierarchy h
USING(tid)
WHERE (1
)
)
;
INSERT INTO wordpress.wp_term_taxonomy
(term_id, taxonomy, description, parent)
SELECT DISTINCT
d.tid `term_id`,
'post_tag' `taxonomy`,
d.description `description`,
h.parent `parent`
FROM drupal.term_data d
INNER JOIN drupal.term_hierarchy h
USING(tid)
INNER JOIN drupal.term_node n
USING(tid)
WHERE (1
)
)
;
```

7. To convert over posts, apply the following query:

```
INSERT INTO wordpress.wp_posts
(id, post_author, post_date, post_content, post_title,
post_excerpt, post_name, post_modified, post_type,
`post_status`)
SELECT DISTINCT
n.nid `id`,
n.uid `post_author`,
FROM_UNIXTIME(n.created) `post_date`,
r.body `post_content`,
n.title `post_title`,
r.teaser `post_excerpt`,
IF(SUBSTR(a.dst, 11, 1) = '/', SUBSTR(a.dst, 12),
a.dst) `post_name`,
FROM_UNIXTIME(n.changed) `post_modified`,
n.type `post_type`,
IF(n.status = 1, 'publish', 'private') `post_status` FROM
drupal.node n
INNER JOIN drupal.node_revisions r
USING(vid)
LEFT OUTER JOIN drupal.url_alias a
ON a.src = CONCAT('node/', n.nid)
# If applicable, add more Drupal content types below.
WHERE n.type IN ('post', 'page', 'blog')
;
;
```

**If your Drupal installation has multiple post types, be sure to add the name of the post type into this line: WHERE n.type IN ('post', 'page', 'blog'). Failure to do so will result in not all posts types being converted over.*

8. To combine post types in WordPress, run this script:

```
UPDATE wordpress.wp_posts
SET post_type = 'post' WHERE
post_type IN ('blog');
```

9. To define the post/tag relationship, apply the following:

```
INSERT INTO wordpress.wp_term_relationships (object_
id, term_taxonomy_id)
SELECT DISTINCT nid, tid FROM drupal.term_node
;
# Update tag counts.
UPDATE wp_term_taxonomy tt
SET `count` = (
SELECT COUNT(tr.object_id)
FROM wp_term_relationships tr
WHERE tr.term_taxonomy_id = tt.term_taxonomy_id
)
;
```

10. Apply this query to migrate comments:

```
INSERT INTO wordpress.wp_comments
(comment_post_ID, comment_date, comment_content,
comment_parent, comment_author,
comment_author_email, comment_author_url, comment_
approved)
SELECT DISTINCT
nid, FROM_UNIXTIME(timestamp), comment, thread, name,
mail, homepage, ((status + 1) % 2)
FROM drupal.comments
;
# Update comments count on wp_posts table.
UPDATE wordpress.wp_posts
SET `comment_count` = (
SELECT COUNT(`comment_post_id`)
FROM wordpress.wp_comments
WHERE wordpress.wp_posts.`id` = wordpress.wp_
comments.`comment_post_id`
)
;
```

11. You do not have to do anything if you want to keep your Drupal images and files in the same location, but if you are FTP-ing your files to the uploads folder in your WordPress wp-content folder, use the following code to fix the image URLs.

```
UPDATE wordpress.wp_posts SET post_content =
REPLACE(post_content, ,"/files/", ,"/wp-content/
uploads/");
```

12. To fix taxonomy (assuming you've set it up correctly in your original Drupal site), use the following code:

```
UPDATE IGNORE wordpress.wp_term_relationships,
wordpress.wp_term_taxonomy
```

```
SET wordpress.wp_term_relationships.term_taxonomy_id =
wordpress.wp_term_taxonomy.term_taxonomy_id
WHERE wordpress.wp_term_relationships.term_taxonomy_id
= wordpress.wp_term_taxonomy.term_id
;
```

13. To assign author roles for your users, here's the code:

```
INSERT IGNORE INTO wordpress.wp_users
(ID, user_login, user_pass, user_nickname, user_email,
user_registered, user_activation_key, user_status,
display_name)
SELECT DISTINCT
u.uid, u.mail, NULL, u.name, u.mail,
FROM_UNIXTIME(created), ,', @, u.name
FROM drupal.users u
INNER JOIN drupal.users_roles r
USING (uid)
WHERE (1
# Uncomment and enter any email addresses you want to
exclude below.
# AND u.mail NOT IN (,test@example.com')
)
;
```

14. Here's the code you can use to set the author role's permissions:

```
INSERT IGNORE INTO wordpress.wp_usermeta (user_id,
meta_key, meta_value)
SELECT DISTINCT
u.uid, 'wp_capabilities',
'a:1:{s:6:"author";s:1:"1";}'
FROM drupal.users u
INNER JOIN drupal.users_roles r
USING (uid)
WHERE (1
# Uncomment and enter any email addresses you want to
exclude below.
# AND u.mail NOT IN ('test@example.com')
)
;
INSERT IGNORE INTO wordpress.wp_usermeta (user_id,
meta_key, meta_value)
SELECT DISTINCT
u.uid, 'wp_user_level', '2'
FROM drupal.users u
INNER JOIN drupal.users_roles r
USING (uid)
WHERE (1
# Remove and enter any email addresses you want to
exclude below.
# AND u.mail NOT IN ('test@example.com')
)
;
```

You can remove the number sign before the line that has the email address and put your own in so that you remain the administrator.

15. Use this code to assign and give administrator status:

```
UPDATE wordpress.wp_usermeta
SET meta_value = 'a:1:{s:13:"administrator";s:1:"1";}'
WHERE user_id IN (1) AND meta_key = 'wp_capabilities';
UPDATE wordpress.wp_usermeta
SET meta_value = '10'
WHERE user_id IN (1) AND meta_key = 'wp_user_level'
;
```

16. This code will help assign authors to the posts they wrote:

```
UPDATE wordpress.wp_posts
SET post_author = NULL
WHERE post_author NOT IN (SELECT DISTINCT ID FROM
wordpress.wp_users)
;
```

17. You can then feed the following code for the editor to help clean it up so that your posts don't look bizarre after the conversion.

```
UPDATE wordpress.wp_posts
SET post_name =
REVERSE(SUBSTRING(REVERSE(post_
name),1,LOCATE('/',REVERSE(post_name))-1));
```

That's the basics of migrating your Drupal site to WordPress. While these instructions only show a scope of what you can migrate, for additional references check out these resources:

- [Blondish.net](#)
- [The Underdog of Perfection](#)
- [Another Cup of Coffee](#)
- [Github](#)

It's important to note that with the exception of the steps from Blondish.net, the other articles do not specifically say that they work with the latest versions of Drupal and WordPress. Therefore, make sure before copying any code that you tailor it to your specific needs before application.

Migration Problems Addressed

As with any undefined process, you might run into issues along the way. Here are some problems you might run into and how to address them.

Problem #1: The categories and tags are translated to the wrong taxonomy. Solution: make sure that before converting the original Drupal install that the taxonomies are labeled correctly.

Problem #2: You run into duplicate errors when feeding the SQL through the conversion process. Solution: check the last known table that was converted over, and only convert from that after the table.

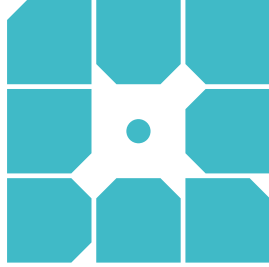
Problem #3: The individual posts and page links keep redirecting back to the home page. Solution: update the slugs by plugging in code into a wp-config.php file before the ending PHP tag. This code can be found at the very [end of Blondish.net's migration tutorial](#).

Final Thoughts

Although it will take time to plan a Drupal to Wordpress migration to ensure success, migrating your site can be a smooth process. While there is no defined path to perform a perfect migration, the right process for you will vary depending on the essential assets that must be brought over to WordPress from Drupal.

And if you lack the technical skill, there are countless services available to help you move your site based on your specific needs.

Planning to migrate from Drupal to WordPress? Give WP Engine a call and see why our managed WordPress platform delivers the enterprise-grade performance that your WordPress site and your business demand. We have solutions for businesses of all sizes. Learn more about our managed WordPress hosting plans.



About WP Engine

WP Engine powers amazing digital experiences for websites and applications built on WordPress. The company's premium managed hosting platform provides the performance, reliability and security required by the biggest brands in the world, while remaining affordable and intuitive enough for smaller businesses and individuals. Companies of all sizes rely on WP Engine's award-winning customer service team to quickly solve technical problems and create a world-class customer experience. Founded in 2010, WP Engine is headquartered in Austin, Texas and has offices in San Francisco, California, San Antonio, Texas, and London, England.

