



*Blog Basics: Porting your Blog to WordPress*

Copyright © 2008 by Shaun W. Taylor

[www.30daystotraffic.com](http://www.30daystotraffic.com)

[publisher@wordpressforbeginners.com](mailto:publisher@wordpressforbeginners.com)

Some rights reserved. You may republish excerpts from this book as long as they are accompanied by an attribution link back to [www.wordpressforbeginners.com](http://www.wordpressforbeginners.com).

Revision 2008-03 ([Check for updates](#))

## About This Book

This is a Quick Take Book. It is intended to offer quick information and advice on a very specific topic.

This book was produced in booklet form. For optimal readability, it will be printed in landscape view.

Finally, I'm here to help. Really. I want to help you. If you have questions, visit our products' companion sites—they are all newcomer and question friendly!

- [Blogging in Action](#): starting out with WordPress
- [30 Days to Traffic](#): explode traffic to your blog
- [30 Day Blog Makeover](#): take your blog to the next level

# 1

## Introduction

---

You're reading this book for one of several possible reasons:

1. You're looking into moving your blog to a self-hosted WordPress version, and you want to know what to expect.
2. You're in the process of moving your blog, and something isn't going how you expected.
3. You've already moved your blog, and you want to make sure that all of your bases have been covered.

This book will address each of those situations, but not explicitly. We are going to walk through the overall steps that must be completed in order to successfully migrate a blog to the WordPress platform.

Most of the information contained in this book is not specific to TypePad, even though it is used as the migration subject. I use TypePad as the subject because it requires more careful planning than several other platforms, and allows me to address difficulties that are shared across a few platforms.

## When Quick Migrations Get Ugly

I have good news and bad news for you. Migrating to WordPress is incredibly easy (for the most part). And for most people, it should be a relatively simple move, even if some parts can get rather tedious:

- If you've hacked your blog with any custom styles, stripping out the remnants of those styles can be painful
- If you have a large number of posts, error checking can be tedious
- If your posts are thoroughly cross-linked, you'll have to re-link them
- If you've used a lot of photos in your posts, you'll have to re-reference them and create your own thumbnails

That was the **good news**. The bad news is that to minimize downtime, to make sure your feed is available to your readers, and to ensure that you maintain your page rank after the move, you will have to jump through some hoops.

So the physical migration itself is easy — moving your content from one blog on one server to a new blog on another server — but attending to the other concerns is what could cause you some headaches. If you have a high page rank, and want to maintain it, you'll have to jump through some hoops. If you want to minimize site and feed downtime, be prepared for a little extra work.

There are many other smaller considerations, but the above are what will consume the lion's share of your effort. At the risk of stating the obvious, you probably noticed that characteristics of larger, more active blogs lend themselves to the more difficult migrations.

Even given the potential hiccups, I wouldn't expect the total migration time to exceed 16 to 24 hours for larger installations, and new or less active blogs can be migrated cleanly in under an hour.

## Domain Transfers

Another thing I'd like to point out before going further is that I'm assuming that you're going to be transferring a domain name along with porting your blog. That is, your blog is available at <http://yourblog.com>, and it should maintain that address when the migration is complete.

If you're moving from <http://your-account.typepad.com> or <http://your-account.blogspot.com> to <http://yourblog.com>, then the process is far simpler, because you won't need to worry about the DNS propagation lags. Of course, you'll also lose any page rank (at least temporarily) that you've worked to achieve, so understand that before you go in.

I'll address these considerations more throughout this book.

# 2

## Preparing for the Migration

---

**B**efore you begin moving any posts, there are a few things you should attend to. Most of these might seem obvious to you, but I mention them because ignoring them could leave the door open to situations that would be better left avoided.

The below are all related to troubleshooting that you might have to perform during the port. One of the first rules of troubleshooting is to narrow the suspect list.

So while there probably won't be an issue, if there is, you want to be able to quickly assign blame to the port and not another part of the system, which means eliminating any other potential causes of conflict.

### 1) Make sure the hosting account and platform are stable

Load some test HTML and PHP pages, call them in a browser, and make sure they work.

If you're moving your domain, you'll probably need to do this via a temporary URL or an IP address, because your domain is still pointing to your blog. This should have been provided to you in a welcome e-mail from your host; if not, you can request it from them.

Make sure you can access file system functions, like creating and moving folders and files, changing permissions, and so on.

### 2) Install WordPress on your hosting account

Make sure WordPress is installed and running smoothly. Again, if you're moving your domain, you'll probably need to call the WordPress installation script via a temporary URL or an IP address.

If you're looking for step-by-step instructions on WordPress installation, try the [Blogging in Action book and video course](#).

### 3) Apply your Themes, Plug-ins, and any Customizations

Make your blog look and behave as you want it to when the migration is complete. You want to make these changes now before your readers are looking at them. Apply your themes, modify your styles and pages, add your plug-ins, etc.

The *Your WordPress Blog in 1 Hour* book and video course has step-by-step instructions on WordPress themes, plug-ins, and administration, too.

I usually install Akismet, Wp-Cache, Simple Tags, All in One SEO, and Google XML Sitemaps right from the start.

### 4) Backup your WordPress Installation

Not just the database, but the entire installation. If there's a problem, you want to be able to roll back immediately.

If you're not sure how to complete a full backup, head to the WordPress for Beginner's companion site. We have a bonus backup book, video, and script to assist with your backups

### 5) Gather your Account Information

This includes any IP addresses, account logins, domain registrar logins, etc. that you'll need, or even might need if something goes wrong. Prepare for the worst and hope for the best.

## 6) Inventory your backlinks

Query Google and Yahoo! for any sites that link to you, and make note of those URL's. Search Google and Yahoo's indexes for links to your TypePad blog that you'll want to redirect.

You'll want to do more than just using the typical "link:" searches. Also search for yourdomain.com without the link. There are probably plenty of unlinked addresses out there that will eventually get followed by someone.

# 3

## Migrate the Data

---

**W**hile you might expect data migration, specifically moving all of your posts, categories, comments, trackbacks, and so on, to be difficult, this is actually the easiest part of the move.

Migrating your data requires two tasks: exporting the data from your old blog, and then importing it into WordPress.

WordPress has terrific support for importing from a variety of sources, and recent releases have hidden many of the import details from us. Whereas you were once required to call certain scripts, you can now handle import directly from the WordPress control panel.

### *Exporting Data from TypePad*

This used to require a ticket to the support desk, but now you export all of your content directly from your TypePad control panel.

1. Login to your TypePad control panel.
2. Go to Weblogs tab | Manage subtab.
3. Click the Import/Export menu option.
4. At the bottom of the resulting page, look for the "Export: Your-Blog-Name" link.

If you're using IE, right-click and perform a Save as to save the file to your desktop; otherwise, you can right-click and Save as to your desktop, or just open the link in your browser, and then perform a File | Save As.

5. Save the file to your desktop with a .txt extension.

## Exporting Data from Blogger

Blogger does not have an export function, so you'll need to work-around their system and create a custom blogging template that can be saved as an export file.

*These steps are from Blogger.com.*

1. Make a copy of your current blog template.
2. Replace your Blogger template with the following:

```
<Blogger>
AUTHOR: <${BlogItemAuthor$}>
DATE: <${BlogItemDateTime$}>
-----
BODY:
<${BlogItemBody$}>
-----
<BlogItemCommentsEnabled>
<BlogItemComments>
COMMENT-AUTHOR:<${BlogCommentAuthor$}>
COMMENT-DATE:<${BlogCommentDateTime$}>
COMMENT-BODY:<${BlogCommentBody$}>
-----
</BlogItemComments>
</BlogItemCommentsEnabled>
</Blogger>
```

3. Depending on whether your blog is from Blogger or BlogSpot, you will need to either overwrite your existing blog page (BlogSpot) or you'll be able to change your blog file-name to ensure that it isn't overwritten. Do this in Settings | Publishing.
4. In Settings | Formatting, change the number of days displayed on your front page to 5000.
5. In Settings | Formatting set Date/Time Format to the format MM/DD/YYYY HH:MM:SS AM|PM.
6. In Settings | Formatting set Convert Line Breaks to No.
7. In Settings | Archiving, set Archive Frequency to No Archive.
8. Republish your blog.
9. Open your blog in your web browser and save the file to your desktop.

Once all of the above steps are complete, you can go back in and restore your previous settings and template, and republish your blog, so that your readers still have access to familiarly formatted content during the migration.

## Importing your Data into WordPress

Your export is now finished. That one file contains all of your posts, categories, comments, etc. possibly spanning several years. Years of writing and creation. Amazing how so much work can be pared down into such a small file, isn't it?

Import is now even easier thanks to a wizard that's available directly from the WordPress control panel.

1. Make sure the wp-content directory is writable.
2. Open your WordPress administrative panel, and go to Manage | Import.
3. On this page, click the link representing your data's source. You'll see that WordPress has built-in support for importing from most popular blogging and content management platforms.
4. Select the export file you downloaded, and click the button to upload it; or upload it with FTP and name it as described on this screen.
5. Follow the screen prompts to complete your import. You'll need to map the author names on the old posts to users on the new WordPress installation, but the wizard is self-explanatory.

That's it for the import. For simple cases (new or inactive blogs, blogs without a transferring domain), all that would be left to do is redirect the traffic to the new location, but I'm going to continue on with the more complex scenario.

---

## Cleaning Up the Migration

Once the data migration is complete, you need to finish the process of getting your blog to look and act as you want it. Regardless of the style you've chosen, chances are you've found some display anomalies as you browse your new blog. These are the types of things you want to cleanup. You also want to take advantage of any WordPress features that you might not have had in your previous blog.

It's important to note that while you're doing this your traffic should still be pointed to your old blog!

### *Reorganize your Categories*

When I moved to WordPress from TypePad, I took the opportunity to reorganize my categories from the flat list I was using before, to a nested list. Doing so allowed me to better organize my posts, present a more logical structure to readers, and take advantage of various themes that automatically create drop-down or fly-out menus.

An example of the logical structure is the Java category on the blog in question. With TypePad, I had the option of lumping all Java posts into one overall Java category, or creating three categories and assigning a post to one or more.

With WordPress subcategories, I now have a parent Java category, with ME, EE, and SE subcategories. So all Java posts can be found in the Java category, but readers can further drill down to only read posts

assigned to the specific Java edition that they want to see. Of course, I could do this with TypePad, but it would require four separate top-level categories, instead of just one. It's neat and logical on WordPress.

You might find many situations where your categories can be trimmed, expanded, or nested, and it's a good idea to take care of these now so that traffic can be properly redirected on cutover.

## *Replace Blog-specific URL's*

Cross-linking your posts is a recommended blogging tactic. It keeps your readers engaged in your posts, and encourages more browsing. But all of those cross-linked posts now point to old URL's. You can just redirect them, of course, but it's a better idea to change the links that you control.

So scan all of your posts, look for any URL's that point to pages on your old platform, and update them to point to your new blog.

## *Re-reference your Images*

All of your images are stored on your old blog's server, and they were not part of the data export, so you'll need to manually move them to your WordPress installation. If you have them all on your PC, great. If not, grab the thumbnails and full-size images from your old blog.

There are several ways to proceed from here. If you're the fullsize-image-in-the-post sort, you can just upload your images in bulk to the content folder, and then manually replace the image references in your posts.

If you're the thumbnail-and-link-to-the-full-size-image sort, you're probably better off using a photo plug-in like [YAPB](#) (Yet Another Photo Blog). It will automatically create and link thumbnails for you.

You can also use WordPress's built-in thumbnail facility, but I find it a bit too inflexible to be useful.

Choose your poison and get it done.

When you reference your images, replace any absolute references with a relative reference from your server's root, e.g. /wp-content/uploads/year/month/image.gif. This will assist with any changes in the future, and it will allow you to view the images as they'll appear after you make your DNS changes.

## *Update Your Hacked styles*

If you hacked any styles in your old blog to work around its limitations, you'll probably need to remove those hacks, and move them to the overall stylesheet in WordPress.

## *Update your 404 Page*

**This is an important step that I think gets ignored far too often.**

Regardless of your efforts, expect that pesky 404 page to be getting at least a little bit of traffic over the coming weeks and months. Take this opportunity to make your 404 page very descriptive and helpful.

At the very least, tell people about the server move and link to your anchor articles. Informing them of the move doesn't make it any less inconvenient for them, but you're more likely to encourage the readers to stick around if you acknowledge and explain the error.

Your anchor posts are probably driving most of your traffic, so it only makes sense to make sure that they are featured prominently on your 404 page. You can list their titles verbatim, or if they're not very descriptive, add some keywords that describe their content to make it easy on people who were looking for something specific.

If you're feeling adventurous, you can also list all of the new categories, and map the old categories to new ones, as well as provide links to your chronological archives.

### *Tag your Posts*

This isn't a necessary step, but I love the concept of tags, so I always recommend tagging all posts that aren't frivolous in nature. You never know when these tags could come in handy in the future. You can use a plug-in like [Simple Tags](#) for this purpose.

### *SEO Anchor Pages and Posts*

Like the previous cleanup task, this is not necessary, but I recommend you take advantage of the SEO capabilities in WordPress. I wouldn't worry about any frivolous posts, but for any anchor post, or any post that gets Googled a lot, you want to make sure that you're taking advantage of facilities that are accessible to you. I use the [SEO All in One plug-in](#) for this purpose.

### *Replace Widgets*

You might have gotten fond of various widgets in your old blogging platform. I can almost guarantee that you'll find similar widgets and plug-ins in WordPress.

If you had plug-ins that you found essential to your blogging efforts, or that improved your readers' experience while visiting your blog, find comparable ones and add them.

---

## Redirecting Traffic

**M**ost of the migration work is complete, but these next few steps are usually the ones that cause the most stress, and it's because these steps are visible to the public. If you're going to drop any traffic, it's going to happen right here!

### *Step-by-Step Redirection*

Follow these steps in the order prescribed, and you shouldn't run into any problems. If you're not transferring a domain, you're in luck, you won't need to perform these steps; but maintaining your page rank is well worth the effort required if you are transferring a domain.

1. Make sure the receiving server is ready for traffic.
2. Go to your domain registrar, and point your domain to the nameservers on your new hosting account or server. You'll probably also have to delete the MX records that still exist.
3. Wait. Maybe 30 seconds. Maybe 3 hours. Just keep checking.
4. Ping your domain name to see which IP address responds, your old blog or your new server. As soon as the responding IP address is your new server, continue.

You do this by going to Start | Run, and typing "command" in the run box. This opens a command prompt. From there, type "ping yourdomain.com" and see what response you get.

5. In your WordPress administrative panel, go to Options | General, and make sure your blog is pointing to your domain name, and not the temporary URL or IP address you used to setup your WordPress blog.

**Important:** don't perform this step until you know your domain is pointed to your server or you'll lose access to your administrative panel. If this does happen to you, just open up your MySQL database, and update the entry in the wp\_options table.

Typing your domain name in a browser should now point you to your new WordPress blog, but complete propagation might take another day, meaning that people in other parts of the world might still be pointed to your old blog.

DNS is painful. You need to coordinate different servers in different locations, and you can never be sure if what you see is what someone else will see (until propagation completes). So when something goes wrong, it's often hard to pinpoint where the problem is. If you do things in the above order, you shouldn't run into any problems.

## Update your Feeds

If you're lucky, you've been using FeedBurner all along. This will make cutover a snap. FeedBurner only needs to know where to find your new feed.

Login to your FeedBurner account, click on the "Edit feed details" link at the top of the page, and enter your new feed address.

If you get messages like "The domain does not seem to exist" or "There was a problem retrieving the feed: java.net. UnknownHostException" a possible cause is DNS propagation. FeedBurner's help recommends pinging, resynching, validating your feed, and so on, but usually it just takes a little bit of time for these problems to clear up.

If you're not using FeedBurner, you'll need to redirect all feed requests to the new location using URL redirection (below).

## URL Redirection

There are plenty of ways to redirect, including plenty of ways to do it right and plenty of ways to do it wrong.

## Wrong Ways to Redirect Traffic

### Meta Tags, Bad Idea

```
<meta http-equiv="refresh" content="0; url=yourblog.com/your-new-post/">
```

### JavaScript, Another Bad Idea

```
window.location="yourblog.com/your-new-post/";
```

## Correct Ways to Redirect Traffic

In short, you want to use a 301 redirect. This tells all traffic to the old URL that the page has permanently moved, and is a search engine safe method of redirection.

If you've spent some time building page rank, and you want to maintain it, 301 redirects using the same domain should preserve that rank.

But note that page rank is assigned to a specific address, so if you were using yourblog.typepad.com/your-post/, and now you're redirecting to yourblog.com/your-new-post/, this is treated as a new domain and you will drop your page rank, at least temporarily. On the other hand, if you redirect from yourblog.com/your-post/ to your-

blog.com/your-new-post/, a 301 redirect should transfer page rank from the old address to the new.

### Server Side Scripting

The easiest way to create a 301 redirect (with a high success rate) is to use any server-side scripting language to respond with the appropriate HTTP headers. The problem is that individual posts on many blogging platforms usually end with a “.html,” which makes this method useless for redirecting individual post URL’s, but you can sometimes redirect old categories to new ones.

The process is the same for all scripting languages, but because this method is not consistently reliable, I would advise against using it, unless you’re certain that you won’t drop any traffic.

1. Create the corresponding directory structure. If the category you want to redirect is yourblog.com/your-category/, and you want to redirect it to yourblog.com/category/your-category/, you need to create a “your-category” folder in your blog’s root.
2. Add an index page with your scripting language’s extension, e.g. index.asp, index.jsp, index.php.
3. Use the scripting languages to respond with 301 redirect headers to yourblog.com/category/your-category/.

#### ASP

```
Response.Status="301 Moved Permanently"
Response.AddHeader "Location", "http://yourblog.com/category/your-category/"
```

#### JSP

```
response.setStatus(301);
response.setHeader("Location", "http://yourlog.com/category/your-category/");
response.setHeader("Connection", "close");
```

#### PHP

```
Header("HTTP/1.1 301 Moved Permanently");
Header("Location: http://yourblog.com/category/your-category");
```

### Using mod\_rewrite with .htaccess

mod\_rewrite is a powerful and complex Apache module. It is really a Swiss Army knife, and can do so many things if you know how to use it properly.

But mod\_rewrite can be very temperamental if you don’t know how to use it precisely. I will always try to get mod\_rewrite to listen to my commands, but I plan for the worst. If I can’t get it to bend to my will within a short while, I move on.

mod\_rewrite is a very clean, neat, and search engine safe method of redirection. You can initially redirect all of your URL’s, then after a month or so gradually drop your redirections as your URL’s are updated in indexes.

Add all of your mod\_rewrite rules to the .htaccess file in your blog root.

Here’s an example of redirection using mod\_rewrite:

```
RewriteCond %{HTTP_HOST} ^.*$
RewriteRule ^2008/03/my-old-post.html$
http://www.mydomain.com/2008/03/my-new-post/
```

### Using .htaccess

Using .htaccess should be a simple way of redirecting traffic. It's a very neat solution — all of your redirects are stored in one file — and it's a search engine safe way of implementation. Additionally, it works for all URL's.

For each URL to redirect, add a line to your .htaccess file (use the one in your blog's root, if your blog root is different from your site root, e.g. if you blog is located at yourblog.com/blog/).

```
Redirect 301 /old-post.html http://www.yourblog.com/your-new-post/
```

This method has many advantages and is the second method I would try, right after mod\_rewrite.

### Objection Redirection

When mod\_rewrite and .htaccess fail you, you can always fall back on the Objection Redirection plug-in.

1. Download [Objection Redirection](#), install it, and activate it. If you're unsure of how to use plug-ins in WordPress, check out the [Your WordPress Blog in 1 Hour book and video course](#).
2. Go to Options | Redirection in your WordPress administrative panel.
3. In the second pane, add the old URL in the "URL requested" field, and the URL to redirect to in the "Redirect their ass" field.

You'll need to repeat this for each URL to redirect. If you have quite a few, it's a better idea to create a SQL INSERT statement, and just add the redirections directly to the database.

## Final Tasks

---

### *Clean up Your Old Blog*

You might want to keep your old blog around for a while, just in case, but you'll want to tidy things up a bit. Remove any domain mappings, add redirection messages, delete old content, and so on.

I recommend keeping the account open for some time until the page rank has completely moved to your new blog, so that page rank poachers don't earn a victory by snatching a valuable domain or account name. It happens far too often, and I find it entirely unethical.

### *Update the Engines*

This step couldn't be easier, thanks to the [Google XML Sitemaps plug-in](#). Install it, generate your sitemap, and let the plug-in submit it automatically to Google, Yahoo! and Ask.

### *Backup*

I'm probably a bit too cautious when it comes to backups, but at the very least, now that your blog is populated and traffic has been redirected, you should generate a complete backup. A backup of your data and all of your WordPress files.

I usually back up every hour or so when I'm doing something important. My test is always to ask myself: "do I want to do this again?" If the answer is no, I do a quick backup.

## *Have a Drink*

A nice IPA would hit the spot right now.

# 7

---

## Conclusion

All-in-all, migrating to WordPress from another blogging platform can be made simple, though there are many pieces that need to be organized.

My biggest tip would be to **prepare**. Think through all of the steps that you'll need to take before you take them, so you make the process go as smoothly as possible. The last thing you want is to change your DNS information, only to find that you've forgotten to set up something on your target server, resulting in hours of unnecessary downtime.

Follow the steps in this book, and you should come out of the migration just fine. The move to WordPress is well worth the effort required in the migration.

To recap, here are the steps to follow to migrate your blog:

1. Prepare for the Migration
  - Make sure your new hosting account and platform are stable
  - Install WordPress on your hosting account
  - Apply your themes, plug-ins, and any customizations
  - Backup your WordPress installation
  - Gather your account information

## *Conclusion*

- Inventory your backlinks
2. Export your data from your old blog
3. Import your data into WordPress
  - Clean up the migration
  - Reorganize your categories
  - Replace blog-specific URL's
  - Re-reference your images
  - Clean up your hacked styles
  - Update your 404 page
  - Tag your posts
  - SEO anchor pages and posts
  - Replace widgets
4. Redirect Traffic to your blog
  - DNS redirect traffic to your new blog
  - Update your feeds
  - Redirect all requests to old URL's
5. Update the engines
6. Backup
7. Have one or many drinks

Good luck with your migration! And if you run into any problems, check out one of our companion sites (listed at the beginning of this book). We're always happy to help or answer questions.