Drupal 7 to 8 Migration: Step by Step Notes

Kevin Pittman, Ivan Allen College of Liberal Arts, Georgia Institute of Technology

https://webdev.iac.gatech.edu/

Table of Contents

Quick Introduction to Drupal 8

Drupal 8, released in November of 2015, is a major rewrite of the Drupal core engine and APIs. It is to earlier versions of Drupal what Mac OS X was to all earlier versions of Mac OS. Because Drupal 8 has changed so much, there is no direct in-place upgrade path to go from any earlier version of Drupal to Drupal 8. Direct upgrade paths should be available from Drupal 8 to 9 and between other future major versions, but for now, we all have to get over the hump into Drupal 8.

According to Dries Buytaert, the key developer behind Drupal, users should not work on the premise of waiting for Drupal 9 before upgrading. The way the new development process works, you are best off upgrading to Drupal 8 as soon as it is feasible for your site. This is because Drupal 9 is not expected to be a painful upgrade like previous major version upgrades. Drupal 8 introduces a concept of incremental development and deployment, such that new features will be added to each point release of Drupal 8, but nothing old will be removed until Drupal 9. During this time, users can run add-on components that work against old or new APIs, thus allowing them to test out new features while also running a few legacy modules whose maintainers haven't had a chance to upgrade them yet. Hopefully, most legacy code will get updated before Drupal 9 comes out, and if site administrator have upgraded all other components on their sites first, the upgrade to Drupal 9 should be seamless – in theory, at least.

Note: You can migrate to Drupal 8 from Drupal 7 or 6. Since, hopefully, no one is still running on anything earlier than 7 (no Drupal before 7 receives any official support any more), the notes below are assuming you are on Drupal 7. All of the same concepts apply to Drupal 6, but the location of control panels will vary with those versions. If you are (heaven forbid) running anything before Drupal 6, you'll need to upgrade to Drupal 6 (or 7) first, then do a migration to Drupal 8 (though, you might be better off just rebuilding such a site from scratch in Drupal 8).

Migration Preparation: Clone Your Site

If you don't feel comfortable with cleaning up your live site, and the live site doesn't change frequently, you can clone the site and work with the clone instead. It is much safer to work with a cloned site, as if

you mess anything up too badly, you can just recreate the clone and start over again. The only downside is that if your live site does get changed, you'll have to copy over those changes manually.

A few tips for making a clone:

1. Remember that you need to create backups of both your live **filesystem** and your live **database**.
2. It's a good idea to set up your clone in such a way that it can't be accessed publicly, by either putting it on a development server that is firewalled, or setting firewall/access rules on the site itself (via your `.htaccess` file) to prevent access by the public.
3. You may need to modify the `.htaccess` file before **you** can access the site. If the live site has redirection rules to redirect all traffic to the www.* version of the site or the SSL / HTTPS version of the site, you'll need to remove those rules before you can access the site on your development server. If you run into too many problems, you can try replacing the `.htaccess` file with a clean version from a fresh recent copy of Drupal 7.
4. You may need to disable certain security features on the cloned site, such as single sign-on (SSO). For example, if you are using Central Authentication Service (CAS) and have users forced to log in via CAS, you may have to disable that on your clone site if your organization's CAS server won't authorize users on a non-standard web server. If you find this to be the case, you can do the following:
    1. On the live site, before cloning, set your CAS (or other SSO) configuration to allow authentication by either CAS or local accounts (you can set this back to "CAS only" after you've created your backups of the site).
    2. On the live site, also check for the username of user #1 (the full administration user). You can find this by browsing to the path `/user/1` on that site. If you don't know the password for this account, go ahead and reset it now, and make a note of the username and new password for later.
    3. Now, make your backups of the site, and then if desired, reset your CAS / SSO settings.
5. You may want to put the cloned site into "Maintenance Mode" as a reminder to yourself that it's the cloned site and not the live one (or add a colorful warning block that displays at the top of every page). It may sound silly, but more than a few problems have occurred due to a developer mixing up a live site and a development site when s/he had too many windows open at once.

Migration Preparation: Clean and Catalog Your Site

Before you attempt a migration, you'll want to clean up your existing Drupal 7 site and catalog its add-on components and customizations. This will help you to visualize what needs to be done and how much effort it will take to get everything working in Drupal 8. You can catalog with any tool you like (plain text file, word processing document, etc.), but the most efficient method is probably to create a spreadsheet in your favorite office application. Regardless of how you make your catalog, the easiest way to make sure you capture everything important is to just go through Drupal's Administrative Toolbar menu from left to right:

Content

In your spreadsheet, take note of how many nodes you have for each content type. You can use the

filtering options of the Content page to help with this (Tip for faster counting: every page in a content listing other than the last one has exactly 50 items), but it's still a pain. A better option is to write your own scripts, or try a third party module such as Node Type Count. You could also just get the data you need directly from the database with a SQL query (NOTE: If you used a prefix for your database tables, add the prefix before 'node' in the line below):

```
SELECT type, COUNT(*) FROM `node` GROUP BY type
```

Knowing your node counts will help you later on in determining which approach you want to take to migrating your site.

Structure

Go through all sections and remove anything that you can determine you no longer need. In addition:

- **Content Types:** If a content type doesn't have a "Delete" option, it was created by a module. If you can determine that the associated module is no longer in use and has been disabled and uninstalled, **and** you have removed all of the content using that content type, then you can go into the database and enable deletion of the type. Look in the `node_type` table and set "locked" and "disabled" to "0" (zero) for the type. Flush all of your caches, and with luck the "Delete" option should now be available on the Content Types page. (Don't try to delete the content type directly from the database, as there may be bits and pieces in other tables that need to be removed as well.)
- **Blocks:** Take note in your spreadsheet of the blocks being used and where they are being used. You'll need to make sure that your blocks still show up in the right places after you've migrated everything to Drupal 8
- **Books:** Take note of each book and it's base page NID (you'll need this later, as Book migration is missing from Drupal 8 core)
- **Menus:** Take note in your spreadsheet of which menu is your primary navigation menu (Main Menu)
- **Taxonomy:** Determine how and where each taxonomy is being used. You'll need to dig through your Field List report and check on each Term Reference field.
- **Views:** Take note of the views you are actively using, as they will have to be recreated by hand in Drupal 8 (there is currently no migration path for Views configurations)

Appearance

Disable any themes not being actively used and uninstall any unused non-core themes. Take note in your spreadsheet of the theme(s) in use and where they came from.

Chances are good that you will have a custom theme. If so, then you'll have to figure out who can rebuild that theme to work in Drupal 8. I have put together some quick reference notes on rebuilding a theme in Drupal 8 that might help.

People

- Do a full audit of all user accounts and block ones for people who are no longer in your organization. If you can determine who has edited what (you'll need to write some custom scripts or do some database diving to get this information), you could delete any old accounts

that never edited anything.

The following SQL code lets you see who has made revisions to content (NOTE: If you used a prefix for your data tables, add that prefix to the appropriate places in the SQL below):

```
  SELECT u.name, count(*) FROM `node_revision` nr LEFT JOIN
`users` u ON (u.uid=nr.uid) GROUP BY nr.uid
```

- Do a full audit of your User Roles (found under `Users -> Permissions`) and remove any that aren't needed while making sure the permissions for the rest make sense.

Modules

- Disable and (if not part of Drupal core) remove any modules that you don't truly need to keep active
- Catalog the remaining modules in your spreadsheet: Module name, machine name (found in parenthesis beside or below the human friendly name), and version number.
- Go through that list of modules and check each module's project page (`https://drupal.org/project/`*`machine-name`*) to check the module's Drupal 8 status. Take note of what you'll need to do to make that module's functionality available in your Drupal 8 site, taking into account the following:
    - **If a module has no Drupal 8 version and there are no known alternatives**, then you'll have to figure out how to live without that particular functionality or pay someone to write a custom module for you
        - Example: ***Blog*** and ***PHP Filter*** have been removed from Drupal 8 core, and while third-party versions were started, neither is in a usable state (and trying to use PHP Filter is **highly** discouraged due to the potential security issues). However, you can recreate the basic functionality of Blog with a custom View, and you can create a module to dynamically generate content pages or blocks via custom PHP code.
    - **If a module has no Drupal 8 version, but there is an alternative module**, then you'll have to learn how the alternative works and evaluate it to see if will work for your unit's needs
        - Example: ***Admin Menu*** functionality for Drupal 7 is now provided by ***Admin Toolbar*** for Drupal 8
    - **If a module has an unstable (Beta, Alpha, Dev) release for Drupal 8**, then you probably want to wait on these until a stable release comes out. You can look for alternatives, or see if you can just wait to add that functionality back into your site. Bear in mind that just because development work began on a module, there is no guarantee that there will ever be a stable release.
        - Example: ***Webform*** is being completely rewritten, and even today is still in a beta stage
    - **If a module has a stable (Release Candidate or Production) release for Drupal 8, but no migration path**, then you will have to look to see if the module has notable data or configuration that needs to be brought over to Drupal 8, and if so, figure out how to

do that by hand or by custom scripts that you write.

    - Example: ***Views*** (now part of Drupal 8 core), still does not have any migration paths, so you'll have to recreate your views in Drupal 8 by hand

- **If a module has a stable release for Drupal 8 and a migration path - Hooray!** These rare beauties should just work with little or no special effort, though you'll still need to test them carefully and learn about any changes or new features in the Drupal 8 version.

- Some of your modules may be **custom modules** - ones that you, your staff, or a developer you hired wrote specifically for your website. You'll have to figure out how to get these rewritten for Drupal 8, either doing it yourself or hiring someone to do the work for you. I've put together some quick reference notes on rebuilding modules in Drupal 8 that might help.

Configuration

The main area of concern here is `Configuration -> Content authoring -> Text formats`, where you need to clean out any text formats that are not in use (once again, this can be hard to determine for certain without a little database diving), and take note in your spreadsheet of the remaining formats and their filters.

Due to migration issues with filters, the recommended approach is to disable **all filters** from all text formats before migration. Once you've completed the automated migration process, you can go back and enable the filters you want active on your new Drupal 8 site's text formats. Please note that filters simply modify a node's data before it is displayed on the screen for a visitor, so turning off filters has no effect on the node data that generates your pages.

Other Components

You may have other components that have data elements, such as Paragraphs, Panels, etc. If so, you'll need to review them as well and take notes in your spreadsheet on the content managed by those components.

Migration Preparation: Testing and Plan Building

To Migrate or Not To Migrate: That is the Question

There are two popular methods for getting a Drupal 7 (or 6) site into Drupal 8:

- Use the experimental **Migrate** tool provided in Drupal 8 core.
- Use the tried-and-true **copy-and-paste** by hand method.

Which method is right for you? There are a few factors that can help you make a decision:

- If your site was built to best practice standards and has few special customizations and limited use of special layout systems (particularly ones that don't migrate directly to Drupal 8), then the Drupal 8 Migrate Tool is a good option. Just bear in mind that for components that do not have a migration path, you'll still have to do some form of copy-and-paste or write your own migration scripts.

- If your site has a small number of nodes and no complex layouts, then a cut-and-paste approach is a good option, as it lets you end up with a really clean Drupal 8 site. As for how much is a "small number", that depends on the size of your web support staff. If it's just you, then you may find twenty to twenty five pages to be a reasonable maximum for copy-and-paste. On the other hand, if you have four assistants, then it may be just as feasible to do cut-and-paste on a site with over a hundred pages.
- If your site has a lot of complex layouts, and the layout system doesn't translate to Drupal 8, then you may **have** to use copy-and paste to get your complex pages into a supported layout system.
- If your existing site is royally hosed from multiple previous developers with little true Drupal experience messing with it over the years, then you may not be able to get Migrate to work correctly, in which case copy-and-paste may be your only option.

What Exactly is Migrate, and How Does it Work?

Migrate is a tool is provided in the core of Drupal 8 to aide site administrators with importing content and settings from a Drupal 6 or 7 website. It was a contributed (third-party) tool in previous versions of Drupal.

The way Migrate works is that it utilizes a set of rules, called a "migration path" to determine how to retrieve important data and/or configuration from a Drupal 6 or 7 site, rewrite it as needed, and store it in a Drupal 8 site. Migration paths are normally provided by each individual module (core, third-party, or custom), and thus are the responsibility of whomever maintains a particular module.

As of Drupal 8.4, Migrate is still considered an experimental module, and does not have full support for all parts of Drupal 8 core. Some pieces that are still missing as of Drupal 8.4 include:

- **Book**: Your book nodes will all migrate over to Drupal 8, but your hierarchy information does not come over automatically. It is possible to bring this over by hand or by a script, as it's mainly just a matter of copying data from the Drupal 7 `menu_links` database table to the Drupal 8 `book` database table.
- **Views:** While the previously contributed (third-party) Views module is now in Drupal 8 core, as of Drupal 8.4 there still was no migration path available to bring over your existing views from earlier versions of Drupal. Thus, your only choice is to recreate those views by hand.

As a side note, the **PHP Code** text filter was removed after Drupal 7. If you're still using it (and you **really, really shouldn't be**), you'll need to move that code into some kind of custom Drupal 8 module if you need to keep using it. Any text fields set to text format using that filter will be set to 'null' and will not display to visitors, but can be edited by site administrators to retrieve any needed PHP code from them.

Test, Test, Test

Before you embark on any kind of Drupal 8 migration, it is best to set up a test site and get very familiar with how things work in Drupal 8. Install any contrib (third party) modules that you plan to use and test them out as well. Use this sandbox site to learn how (and how not) to do things, and then keep it around even after you start your migration, as you may want to refer to it from time to time.

It's also a good idea to do a migration practice run (or two, or three). Run through the process knowing that you aren't going to keep the Drupal 8 site you create, just to see what does and doesn't migrate

over, and to look for any "gotchas" that you aren't expecting.

A Final Plan

After testing everything out, go over any pain points with your stakeholders and figure out how each will be addressed. Work up a timeline and set dates for when migration will occur. Keep in mind that either method is going to take some time to work through, so there is no such thing as an instant upgrade in going to Drupal 8.

When possible, it's best to freeze your Drupal 7 site (i.e. disallow any changes) once you start the process, but there will always be certain types of sites where being up-to-date is so important that it's not feasible to freeze the site. In that case, you'll need to figure out how to pick up any last minute changes and bring them over to the new site.

An important factor here is that you **cannot** run the automated Migrate tool more than once, so any last minute changes would **have** to be brought over by hand. This may be reason enough to convince your stakeholders to allow the site to be frozen, even if that is inconvenient for the unit (or, it may just mean you have to do the migration over a long holiday weekend).

In any case, make sure you have a solid plan and have addressed everything in your spreadsheet before you begin the real migration.

Migration With the Migrate Tool

Migrate Tool - Preparation

1. Identify a development web hosting space where you can host both a copy of your Drupal 7 site and your Drupal 8 site while it is under development. Keep in mind that Drupal 8 requires a minimum of PHP 5.6 to run, and prefers PHP 7. If you wish to set up an full clone of your Drupal 7 site, Drupal 7 should run under either version of PHP, but it is possible that some third-party or custom modules may not like PHP 7.
2. At a minimum, clone your Drupal 7 database to your development space. Optionally clone the filesystem as well and get the site accessible in your development space. Keep in mind that you may need to adjust `the .htaccess` file and/or the `sites/default/settings.php` file to get the site to work under a development DNS hostname.
    - If you do not use a private filesystem, then you can get away with just cloning your Drupal 7 database and pointing to your live filesystem (this won't hurt your live site at all). You do need your Drupal 7 database local, as most web servers are configured to disallow remote MySQL database access.
3. Install a fresh copy of Drupal 8 to a different part of your development web hosting space, but **do not configure anything yet!**
4. On the Drupal 8 site install and enable the Drupal 8 equivalent modules for those used on the Drupal 7 site (including any needed core modules that are not enabled by default), but **do not configure them!**
5. Enable all three **Migrate** modules found under the CORE - EXPERIMENTAL heading:

- Migrate, Migrate - Drupal, Migrate - Drupal UI

Migrate Tool - Operation

1. **Make sure you are logged in as user #1** (other accounts, even ones with full administrative privileges, do not work)
    - This shouldn't be a problem if you followed the directions and didn't do any custom configuration yet, but if you did create a second user account and then try to use it to run Migrate, you'll get an access denied error with no clear explanation.
2. Navigate to **/upgrade** on your Drupal 8 site
3. Enter the MySQL database username and password that can access your Drupal 7 database
4. Enter the local or remote path to your Drupal 7 filesystem
5. Review upgrade paths available. Nothing can actually be changed here, but you can check to make sure all the paths you are expecting to be used have been found.
6. If all looks good, start the migration process, then go get a cup of coffee (or two or three ...)

Post Migration Clean Up

- Reset username and password of user #1 (will be copied from old Drupal 7 site), *before* you log out!
- Review and repair "Basic site settings" in the "System" configuration area
    - Make sure system email address still makes sense
    - Delete paths from "Default 403 (access denied) page" and "Default 404 (not found) page" fields
    - Validate the front page path
- Review all Block placements (ones assigned to the administration theme 'Seven' in particular seem to get scrambled up a bit)
- Review all components mentioned earlier (Content Types, Menus, Taxonomies, User Roles, Text Formats) and adjust as needed. You may wish to rename the built-in system entities of each component back to their Drupal 8 defaults (Migrate will copy display names and descriptions from your Drupal 7 site).
- If any nodes show up with missing content, but you can see the content when you edit the node, chances are that one or more of your Text Formats didn't migrate over.
    - You can edit each node and set the format of each text field to a proper value, but this could take a while if you have a lot of nodes where this happened.
    - For bulk updates, you can modify the database, doing a search-and-replace in the `node__body` table's "body_format" field. Search for "null" and replace with the machine name of the text format you want to use instead. You may also need to do this on the data table for other formatted text fields. The field name will vary by table, but will always end with the suffix "_format".

- Most images on pages will not be editable in CKEditor, as the Image widget has changed. However, there are no longer configurable settings for individual images, so this is mostly a moot point. The only thing for your Content Editors to be aware of is that if they can't double-click on an image to change it, they just need to delete it and then place a new image using the Image widget.
- Right before you push your new site to your live production hosting, clear the Drupal logs, and go into the database and delete all of the tables with a prefix of 'migrate_', as none of that is needed when you go live, and it's easier to debug later problems if you clean that stuff out now.