Migrating to Drupal 11

Tips, tricks, and lessons learned

Introductions

Introductions

- Name: Michael Brewer
- Employer: Franklin College Office of Information Technology
  The University of Georgia
- Title: Web Developer Principal
- Drupal Experience: Approximately 14 years of Drupal experience
  Over 25 years of web development experience
- Board Member: United States PostgreSQL Association

Introductions

- We have over 130 Drupal sites
  (Colleges, Departments, Centers, Institutes, Research Labs, etc.)

- Hosting on Pantheon

Web Hosting

Web Hosting

NOTE: This is **not a recommendation** – UGA doesn’t do those.

This is just a description of what we use and how it works.

Web Hosting

Benefits of Pantheon, from our perspective:

- Custom upstream
- Good administrative dashboard
- Good command-line tools
- Good local development environment
- Good customer support (as containers can and do fail)

Upgrading to Drupal 11

Upgrading to Drupal 11

- This is the process we used
- There is no One True Method; this is just our method
- This was our smoothest upgrade by far
  (once we escaped the Composer Maelstrom)
- Pantheon’s server architecture helped,
  in that we were easily able to create d11 platforms/git branches for testing
- This is a process; your mileage may vary
- We’re still shaking out bugs

Preparation

Preparation

- Make sure to use Upgrade Status: https://www.drupal.org/project/upgrade_status
- Very important for server support
- Contains links to Drupal 11 issue pages for modules that aren’t ready yet
- This will get you close
- At some point, you will decide that you’re close enough (or that the maintainers for the modules you need aren’t going to upgrade in your needed timeframe)

Here We Go:
The Upgrade Process

The Upgrade Process

Always remember to brush and floss:

- **Back up** servers before any changes
- Back them up **again**
- Make a list of the modules currently enabled on your sites
- If you’re going to remove any modules (incompatible with D11, or just cleaning them out), back up your sites, then remove them **before updating**
- If CAS user: Set default timeout (under advanced options) **before updating**

The Upgrade Process

Okay, you’re ready to begin.

(Back the sites up again.)

(Test your ability to restore from backup.)

(Okay; now you’re **really** ready to begin…)

The Upgrade Process

Pantheon has documentation for how to upgrade.

This documentation led to

COMPOSER MAELSTROM

The Upgrade Process

Drupal has documentation for how to upgrade.

This documentation led to

COMPOSER MAELSTROM

The Upgrade Process

The only thing that worked for us?

- Pantheon had a blank D11 template (“create a new Drupal 11 site”)
- We created the site
- We deleted the various folders in our most complex Drupal 10 site (because git), keeping .git, .gitattributes, .gitignore
- We copied over the files and folders we needed (*.md; *.yml; composer.*; config/ ; patches/ ; recipes/ ; upstream-configuration/ ; web/)

The Upgrade Process

The only thing that worked for us?

- Then, the fun process:
  Adding the modules via composer require, one at a time.
- Note what modules failed (and how), then move on
- Eventually, you get a D11 instance for which composer update works.
- This is old-school iterative software development:
  Find something that works, add things that work.

The Upgrade Process

- What about those missing modules?

Patching

Patching

You have two basic choices when patching:

- Writing patch files (that composer handles for you)
- Updating module files in amber
(putting them outside composer-managed directories)

Patching

Writing patch files (that composer handles for you)
Positives:

- Keeps module code up-to-date as much as possible
- Composer handles the installation for you

Patching

Writing patch files (that composer handles for you)
Challenges:

- Patch files can be tricky to write (in terms of paths and syntax)
- Sometimes, you can do multiple files, but sometimes you can’t
- Removing files is tricky
- If a module update breaks the patch, Pantheon won’t let you apply the update (and doesn’t tell you why it failed)

Patching

Updating module files in amber
(putting them outside composer-managed directories)

Positives:

- Easier to write and test
- You never need worry that a composer install will fail silently and not work

Patching

Updating module files in amber
(putting them outside composer-managed directories)

Challenges:

- These clutter the repo
- You may need to modify the .gitignore rules in certain situations
- Your module code never changes; you will need to monitor for updates

Patching

Most common patch: .info.yml:

```
-core_version_requirement: ^9.5 || ^10
+core_version_requirement: ^10.1 || ^11
```

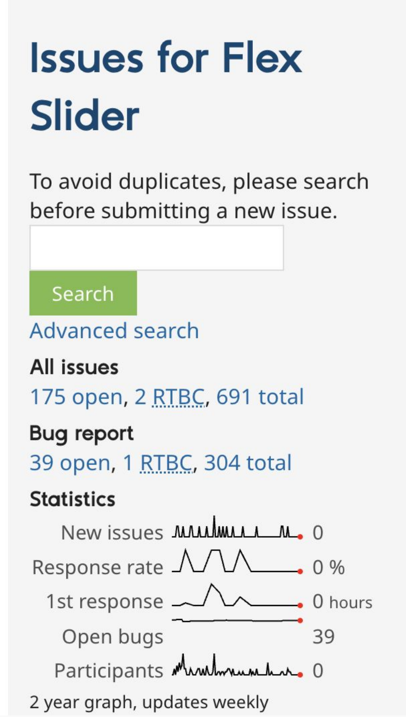
This takes care of most installation issues if you're using composer patching (in that composer will probably let you install it at this point)

This will also let it pass on the status page.

Patching

However, your modules may not work (although they might!).

- Visit the module project page on Drupal.
- Search for “Drupal 11” on their Issues page —>
- Look at the entries for
  “Automated Drupal 11 compatibility fixes”.
  (If they don’t have one, go to the Drupal 10 version)
- Look at the patches there and apply the latest working one
- Sometimes, the discussion threads will lead to a solution



Issues for Flex Slider

To avoid duplicates, please search before submitting a new issue.

Search

Advanced search

All issues
175 open, 2 RTBC, 691 total

Bug report
39 open, 1 RTBC, 304 total

Statistics

New issues	0
Response rate	0 %
1st response	0 hours
Open bugs	39
Participants	0

2 year graph, updates weekly

Patching

If those don’t work (or don’t exist):

- Go to the github project page for the module
- It is likely that there will be issues/patches there. Apply those.

Patching

Failing all those:

- Update the .info.yml file to let the installation proceed
- Try to go to the configuration page for the module; if it whitescreens:
  - Visit the log
  - Find the error
  - Google that error (and “Drupal”); this will probably suggest a fix
  - Implement a patch/amber solution
- Sometimes there are module solutions: https://www.drupal.org/project/jquery_deprecated_functions

Patching

Modules we have patched include:

- block_content_permissions
- Clarity
- CKEditor (4)
- collapsible_dnd
- color
- fb_likebox
- Flexslider
- lytics
- seven
  - (this wound up not working; just use Claro)
- tac_lite
- tagadelic
- tagclouds
- video_embed_wysiwyg

Your Tips/Tricks

Your Tips/Tricks

What has worked for you?

Questions?

Thanks!

Michael Brewer

Email: mbrewer@gmail.com
IG: https://www.instagram.com/operatic/
LI: https://www.linkedin.com/in/michaelalanbrewer/