

Surviving Drupal 8 Migrations

How to stretch a two week project to three months

Paul Nijjar

Waterloo Region Drupal User Group

March 16, 2017

A Sad Story

KWLUG was switching web hosts. The new host did not want the (crufty, unsupported) D6 site in production on his site.

Thus “we” decided to upgrade D6 to D8.

“Let’s use the migration framework! It will be good experience!”

“How hard could it be?”

A Sad Story

KWLUG was switching web hosts. The new host did not want the (crufty, unsupported) D6 site in production on his site.

Thus “we” decided to upgrade D6 to D8.

“Let’s use the migration framework! It will be good experience!”

“How hard could it be?”

A Sad Story

KWLUG was switching web hosts. The new host did not want the (crufty, unsupported) D6 site in production on his site.

Thus “we” decided to upgrade D6 to D8.

“Let’s use the migration framework! It will be good experience!”

“How hard could it be?”

About KWLUG's website

Mostly simple, smallish information site (fewer than 2000 nodes).

Has meeting agendas, podcasts, videos, presentation topics, old initiatives.

The original goal was to be a highly interactive portal. Now it is mostly static.

Started in Drupal 4. Rewritten (?) in Drupal 5, then migrated to Drupal 6.

Design Decisions

- Reduce duplicated content types (pages, forums, wikipages, blogs...)
- Merge presentations and meeting agenda content types. Link podcasts and vidcasts to their meeting agendas.
- Preserve as much old history as possible.
- Clean up site navigation and UX.
- Clean out spam accounts, permission levels.

Resources

Giant blog post:

<http://pniijar.freeshell.org/2017/drupal8-migrate/>

Source code:

<https://github.com/pniijar/kwlug-drupal8-migration>

Migration Overview

- Plan the migration
- Set up D6 site (files, database)
- Set up D8 install site and configure D6 database
- `drush migrate-upgrade -configure-only`
- `drush config-export`
- Set up custom migrate modules
- Set up a migration group
- Customize migrations
- Refine until you are happy or have given up.

Migration Process Overview

Source plugins read from the D6 database and prepare fields.
`query()`, `prepareRow()`, `fields()`, `getIDs()`.

YAML files map source fields to destination fields.

Process plugins massage outputs from source plugins to populate destination inputs (concatenate, default value, static maps).

Destination plugins are deep magic.

What Works

- Configuration management!
- Migration templates
- Iterative development
- Most content types

What Doesn't Work (Well)

See: <https://drupal.org/node/2167633> (“Known issues when upgrading from Drupal 6 or 7 to Drupal 8”)

- Views, themes (who cares?)
- Input formats, display fields (so irritating)
- File fields, image fields

What Was Difficult

Everything was difficult.

Some things were difficulter:

- Merging **Agenda** and **Presentations**
- Figuring out attachments (and comments)
- Finding examples and help when I was stuck

What Was Difficult

Everything was difficult.

Some things were difficulter:

- Merging **Agenda** and **Presentations**
- Figuring out attachments (and comments)
- Finding examples and help when I was stuck

Tips

- Set up extra fields in the GUI and use configuration management to put them into the migration.
- Learn how to rollback and redo migrations.
- Learn how to migrate only a few records (especially on big installations).
- Use an SSD when testing migrations.

Example: Deleting Spam Accounts

- Specify `query()` in source plugin to pick out “contributing” users (ones who published nodes/comments).
- Change `migrate_plus.migration.upgrade_d6_user.yml` to use the new plugin.
- Disable all accounts (make their `status` zero)

Example: Mapping Text Formats

This addresses the irritating `filter_null` issue.

It depends upon filter formats being created in the initial install (so no filter formats are actually migrated).

- Figure out what formats exist in the D6 database.
- Figure out a mapping of old formats to new ones.
- Encode that mapping in a process plugin (`MapKWLUFormatFilter.php`).
- Change all YAML files that used `upgrade_d6_filter_format` to use this plugin. (Ugh.)

Example: Migrate attachments

- Migrate node types that have attachments.
- Migrate `migrate_plus.migration.upgrade_d6_file.yml`
- Add upload fields to content types with attachments (`UploadNode.php`)
- Add dependency on `upgrade_d6_file` migration.

Choose Your Own Presentation

What is interesting to you?

- Merging content types?
- Classifying content without taxonomies?
- Using CSV sources?
- TextField quirks?
- Migrating Flexinodes?
- Something else?
- Nothing? Maybe the presentation should end already.

The End

■ Stuff