palantir.net

Understanding migration development in Drupal 8

Strategies and tools to see what's happening

Midcamp 2019

Dan Montgomery

Dan Montgomery

+ Senior Engineer and Technical Architect, Palantir.net

palantir.net

#midcamp @palantir

About Palantir.net

palantir.net

Table of Contents

Overview of migration

Automated testing

Building a migration

How to run migrations

Planning for manual clean up

palantir.net

Overview of automated content migration

Source data

Destination entities

Migration

Source data

Database

- MySQL
  - Drupal 6
  - Drupal 7

Flat files

- YAML
- CSV
- JSON

Destination entities

Nodes

Node Translations

Files

Media

Paragraphs

Etc

Migration

Custom migration module

Supported by Drupal 8 modules

- Migrate (Drupal core)
- Migrate Plus
- Contrib modules

palantir.net

Automated testing

Automated testing use cases

- Test driven development
    - Create tests first using example content
    - Write migration to make tests pass
- Reduce regressions
    - Confirm plugins are working as expected

palantir.net

Automated testing strategy

Behat

Static source data example

Test data directly on the new site

Palantir Behat Extension

`https://github.com/palantirnet/palantir-behat-extension`

Automated testing example

```
@api @migration
Feature: Article migration
  Given I run the article migration
  When I load a migrated article
  Then I should see that the content is the same

  Scenario: Simple fields
    When I examine the "article" node with title "My Example Article"
    Then entity field "title" should contain "My Example Article"
    And entity field "langcode" should contain "und"
    And entity field "uid" should contain "admin"
    And entity field "status" should contain "1"
    And entity field "field_date_published" should contain
"2018-01-01T19:30:00"
```

palantir.net

Automated testing paragraph example

```
@api @migration
Feature: Example paragraph
  Given I run the paragraph migration
  And I run the node migrations
  When I load an example paragraph
  Then I should see that the content is the same

  Scenario: Example on article
    When I examine the "article" node with title "My Example Article"
    Then paragraph field "field_paragraphs" should be of type "example"

  Scenario: Heading field
    When I examine the "article" node with title "My Example Article"
    And I examine paragraph "2" on the "field_paragraphs" field
    Then entity field "field_heading" should contain "Example heading"
```

palantir.net

Building out the migration

- Scaffolding
- What does a completed migration look like?
- Create a migration starting point
- Explore the data
- Map source to destination
- Managing relationships

palantir.net

Scaffolding

Migration module components

Configuration (YAML)

- Source
- Destination
- Process

Custom Plugins

- Source
- Destination
- Process

Create a custom module

`.info.yml` file

Configuration files

    config/install/

Plugins

    src/Plugin/migrate/

Migration configuration: Uninstall hook

```
/**
 * Implements hook_uninstall().
 */
function example_migration_uninstall() {
  $config_files = [
    'migrate_plus.migration.example_node_page',
    'migrate_plus.migration_group.example_node',
  ];

  foreach ($config_files as $config_file) {
    \Drupal::configFactory()->getEditable($config_file)->delete();
  }
}
```

palantir.net

Migration groups

Shared details, including connection details

Add a database to `settings.php`

Define a group that uses that database

`migrate_plus.migration_group.example_migration.yml`

```
shared_configuration:
  source:
    key: migration_source
```

Add `` `migration_group: example_migration` `` to your migrations

What does a completed migration look like?

Migration configuration

example_migration/config/install/**migrate_plus.migration.example_article**.yml

```yaml
label: Example Article
id: example_article

migration_group:
example_group

migration_dependencies: {}
```

```yaml
destination:
  plugin: entity:node

source:
  plugin: d7_node
  node_type: article

process:
  title: title

  type:
    plugin: default_value
    default_value: article
```

palantir.net

palantir.net

Create a migration starting point

Identify the destination

Copy an example (if possible)

Identify the source

Identify the destination: Classes

Destination classes in core / contrib modules

- `drupal_migrate` module has many of them
- `src/Plugin/migrate/destination/`
- `@MigrateDestination` annotation

Identify the destination: IDs

Simple class

```
*    id = "url_alias"
```
(in annotation)

palantir.net

Derivative classes

Very common

Dynamically create source and destination classes based on a deriver class and parameters

```
    *   id = "entity_revision",
    *   deriver =
"Drupal\migrate\Plugin\Derivative\MigrateEntityRevision”
```

Identify the destination: IDs

Derivative class

```
 * destination:
 *   plugin: entity_revision:node
 *   default_bundle: article
```

Copy an example

Annotation documentation

`core/modules/migrate/src/Plugin/migrate/destination/EntityRevision.php`

Example migration

Search for `plugin: url_alias` (plugin: [destination ID])

`core/modules/path/migrations/d7_url_alias.yml`

Find an article online

Identify the source

From example

```
source:
    plugin: d7_url_alias
```

Look up the id

```
`id = "d7_url_alias"`
```

Search for sources

`` `drupal_migrate` `` module

```
`src/Plugin/migrate/source/`
```

`` `@MigrateSource` `` annotation

palantir.net

palantir.net

Running migrations

Running migrations with Drush

List migrations

`drush ms`

List migrations in a group

`drush ms --group=[name of group]`

Run a full migration

`drush mim [name of migration]`

Number of rows

`drush mim [name of migration] --limit=10`

Selected rows

`drush mim [name of migration] --idlist=1,2,3,99`

Roll back a migration

`drush mr [name of migration]`

Reset a migration if an error occurs

`drush mrs [name of migration]`

How to re-run migrations

Uninstall and reinstall the migration module

```
drush pm-uninstall
example_migration -y && drush
en example_migration -y
```

Use config_devel

Add your migration files to Auto Import

```
/admin/config/development/config
_devel
```

Load a page in the browser to reload all config

How to run migrations on sample data

Limit with drush options

```
`drush mi example_migration --idlist=123 --limit=1 `
```

Can be slow because all the other IDs are read, just not fully processed

How to run migrations on sample data

Limit with source plugin query

Add a filter to your query so that only that record is read in

May need to create your own source plugin wrapper that extends the base class

```
/**
  * {@inheritdoc}
  */
public function query() {
  $query = parent::query();
  $query->condition('n.nid', 123);

  return $query;
}
```

palantir.net

Explore the data

Explore the destination

Explore the source

Explore the destination

View the field configuration

- `/admin/structure/types/manage/[CONTENT TYPE]/fields`

Devel

- Create an example of the content
- View the node on the devel tab
    - `/devel/node/123`
    - `` `#values` `` field
- (optional) use `vardumper` module to make it easier

Explore the destination: Devel example

```
     "body" => array:1 [▼
       "x-default" => array:1 [▼
         0 => array:3 [▼
           "value" => """
             <p>There's nothing like having your own supply of fresh herbs, 
readily available and close at hand to use while cooking. Whether you have a 
large garden or a small kitchen window sill, there's always enough room for 
something home grown.</p>
             """
           "summary" => null
           "format" => "basic_html"
         ]
       ]
     ]
```

Using Devel with page-less entities

Go to `/devel/php`.

Example of loading paragraphs and paragraph translations.

```
// Load a node.
$node = \Drupal::entityManager()->getStorage('node')->load(123);

// Load the paragraphs on a node.
$field = $node->get('field_paragraphs');

$paragraphs = $field->referencedEntities();

// Load the first paragraph.
$paragraph = $paragraphs[1];

dpm($paragraph);

// Load a paragraph translation.
$paragraph_es = $paragraph->getTranslation('es');

dpm($paragraph_es);
```

Explore the source

Drupal site

- Use the methods listed for the destination

All source

- Create a debug process plugin

Debug process plugin

```
namespace Drupal\example_migration\Plugin\migrate\process;
use Drupal\migrate\ProcessPluginBase;
use Drupal\migrate\MigrateExecutableInterface;
use Drupal\migrate\Row;

/**
 * @MigrateProcessPlugin(
 *   id = "debug",
 *   handle_multiples = true
 * )
 */
class Debug extends ProcessPluginBase {

  /**
   * {@inheritdoc}
   */
  public function  transform($value, MigrateExecutableInterface $migrate_executable, Row $row,
$destination_property) {
    print_r(PHP_EOL . 'Debug: ' . PHP_EOL);

    if (isset($this->configuration['method']) && $this->configuration['method'] == 'row') {
      print_r($row->getSource());
    }
    else {
      print_r($value);
    }

    print PHP_EOL;
    return $value;
  }
}
```

src/Plugin/migrate/process/Debug.php

Debug process plugin: Command line

```
process:
  field_name:
    -
      plugin: get
      source: name
    -
      plugin: debug
```

```
process:
  my_field:
    plugin: debug
    method: row
```

Debug process plugin: Xdebug

Add a processing step

Add a breakpoint to your process plugin

Run the migration through the browser

```
admin/structure/migrate/manage/example_group/migrations
```

palantir.net

Map source to destination

Map source to destination

Create one entry per field in the destination under `process`

Identify an example from the source data

Run and re-run the migration on that example

Process plugins

Find examples in existing migrations

`` `src/Plugin/migrate/process/` ``

`` `@MigrateProcessPlugin` `` annotation

Reference online documentation

Online references

Migrate API process plugin overview

`https://www.drupal.org/docs/8/api/migrate-api/migrate-process-plugins`

Migrate API process configuration syntax

`https://www.drupal.org/docs/8/api/migrate-api/migrate-process-plugins/migrate-process-overview`

Drupal core process plugin reference

`https://api.drupal.org/api/drupal/namespace/Drupal%21migrate%21Plugin%21migrate%21process`

Process plugins: Custom

If you need to perform complex logic, you can create your own plugins

```
public function transform($value,
MigrateExecutableInterface $migrate_executable, Row
$row, $destination_property) {

    $value

    $row->getSource()
```
(all values)

palantir.net

Managing relationships

Entity relationships

Dependencies

- Run migrations in an order

Stubs

- Create empty content as needed

Entity relationships: Repeat runs

May want to run migrations twice if you want to avoid stubs

- For example: articles that reference other articles
- The system doesn't know what type of node to create
- Avoid broken references
- `drush mim [migration] --update`

palantir.net

palantir.net

What content needs manual cleanup?

Migration tags vocabulary

- Clean up content post-automated migration
    - Vocabulary of migration tags
        - Example: Has Embedded HTML
    - Add term reference fields to content types
    - Tag during the process stage

palantir.net

Thank You.

622 Davis Street, Suite 400
Evanston, IL 60201

773.645.4100

Dan Montgomery
Senior Engineer and Technical Architect

Questions?

PLEASE PROVIDE YOUR FEEDBACK!

mid.camp/262

The top rated sessions will be captioned, courtesy of Clarity Partners

CONTRIBUTION DAY

Saturday 10am to 4pm

You don't have to know code to give back!

New Contributor training 10am to Noon
with **AmyJune Hineline** of Kanopi Studios

Temporary process fields

```
_nid:
  -
    plugin: migration_lookup
    source: nid
    migration:
      - example_article
    no_stub: true
  -
    plugin: skip_on_empty
    method: row

_redirect:
  plugin: concat
  source:
    - constants/redirect_prefix
    - '@_nid'

redirect_redirect/uri:
  plugin: d7_path_redirect
  source:
    - '@_redirect'
```

Where to add data?

Source plugin

`query()`

Runs once

`prepareRow(Row $row)`

Good to add fields when data is only available from the source dataset

Once per row

Process plugin

`transform()`

Good for transforming data

Once per row

Access to the Drupal 8 site

Translations

Create a default and translations migration

Look up the ID of the default during the translation migration

```
process:
  nid:
    plugin: migration
    source: tnid
    migration: example_node_blog_post
```