

# Tes - Custom Drupal 8 Migration

## About me

- Drupal developer ("Software Engineer")
- Working at Tes for about 2.5 years.
- Working with Drupal for about 5 years.
- Mainly working in D7 although I did have a year of D6 (never want go back to that)
- Recently started working with D8.

## About Tes

- Foremost education news site for teachers and school leaders
- World's longest running education magazine
- Nearly 8M registered users
- Dynamic global marketplace for education resources
- UK Teaching recruitment services provider
- See <https://www.tes.com/> and <https://www.tes.com/about>

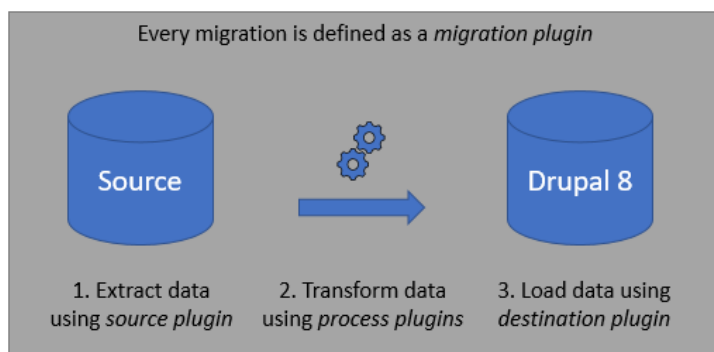
## What's this presentation about?

So, I kind of didn't know what to call this presentation the sort of ideas I had were: *"first tentative steps into D8 migration"*, *"a beginner's guide to Drupal 8 migrations"*, *"my migration headaches"* ... but I ended up with the catchy title above.

So, what this presentation is:

- a migration outline strategy - live & test
- discussion of some different migration sources
- example code - config & custom plugins
- some of the issues along the way

## Extract, Transform, Load processes



As part of this presentation we will be looking at:

- Source plugins (extract phase)
- Process plugins (transform phase)
- Destination plugins (load phase)

## Module dependencies

---

Modules required include:

- migrate (core)
- migrate\_plus (contrib)
- migrate\_tools (contrib)
- source **SQLBase** migration type part of the core migrate source plugins
- if non-sql data source then various contrib modules available to (e.g. migrate\_source\_csv)
- migrate\_plus includes example migration modules
- migrate\_tools gives us the UI and some useful drush utilities

## Background to the migration

---

- Existing D7 site
- 250K+ nodes (215K+ are news\_articles)
- 16K+ authors
- 55K+ managed files
- Proposed redesign of news part of site
- Want to move to D8 and the redesign provides the rationale for that
- Still a *work in progress*

## Data analysis

---

- Before starting coding look at existing data
- Is it worth migrating?
- Automated vs manual migration?
- Manual migration has benefits => user familiarity, focus on what's important (less likely to ask for unnecessary content if they have to enter it)
- Talk to existing users (editors) - get feedback about what works, what doesn't, any content ambiguity
- Drives migration strategy, helps build target content types etc

## Test - CSV migration

---

- Need test content for themers (so need a sample migration for dev)
- Export => CSV extracts (via views and custom code)
- Import => using source CSV, `BaseCSV` class (migrate\_source\_csv contrib)

## Source Content

---

- Taxonomy terms - various "News Article" vocabularies
- Author nodes (node.type = byline), minimal details

- News article nodes (node.type = news\_article), more complex content
- Managed (and unmanaged) files - teaser/hero images, attachments, author photos, embedded media
- Data analysis & feedback helped decisions on what to migrate (e.g. ambiguous fields removed -> teaser vs hero image)
- News article target content type simplified

## Source "News Article" content type

<div>+ Content</div>	group_main_content	<div>Vertical tab</div>	<div>tab closed</div> <div>classes group-main-content field-group-tab</div> <div>required_fields yes</div>
<div>+ Headline</div>	title	Node module element	
<div>+ Byline</div>	field_news_article_byline	Entity Reference	View
<div>+ Standfirst</div>	field_news_article_standfirst	Long text	Text area (multiple rows)
<div>+ Body</div>	body	Long text and summary	Text area with a summary
<div>+ Post script</div>	field_news_article_post_script	Long text	Text area (multiple rows)
<div>+ Publication Date</div>	field_publication_date	Date	Pop-up calendar
<div>+ Storycode</div>	field_news_article_storycode	Text	Text field
<div>+ Media</div>	group_media	<div>Vertical tab</div>	<div>tab closed</div> <div>classes group-media field-group-tab</div> <div>required_fields yes</div>
<div>+ Images</div>	field_news_article_images	Image	Media browser
<div>+ Teaser Image</div>	field_news_article_teaser_image	Image	Media browser
<div>+ Brightcove</div>	field_news_article_brightcove	Brightcove	Video (upload/browse)
<div>+ Files</div>	field_news_article_files	File	Media browser
<div>+ Charts</div>	group_charts	<div>Vertical tab</div>	<div>tab closed</div> <div>classes group-charts field-group-tab</div> <div>required_fields yes</div>
<div>+ Chart</div>	field_chart	Chart field (TES)	Chart field widget (TES)
<div>+ Category</div>	group_category	<div>Vertical tab</div>	<div>tab closed</div> <div>classes group-category field-group-tab</div> <div>required_fields yes</div>
<div>+ Category</div>	field_news_category	Term reference	Select list
<div>+ Section</div>	field_news_section	Term reference	Check boxes/radio buttons
<div>+ News Article Tags</div>	field_news_article_tags	Term reference	Select list
<div>+ Related content</div>	field_news_article_related_conte	Entity Reference	View
<div>+ Exam Results</div>	field_exam_results_ref	Term reference	Check boxes/radio buttons
<div>+ Exam Results Section</div>	field_exam_results_section_ref	Term reference	Select list

## Target "News Article" content type

<div>Edit</div>	<div>Manage fields</div>	<div>Manage form display</div>	<div>Manage display</div>	<div>Devel</div>
<div>Home » Administration » Structure » Content types » News Article</div>				
<div>+ Add field</div>				
LABEL	MACHINE NAME	FIELD TYPE		
Article author	field_news_author	Entity reference		
Article body	field_news_body	Text (formatted, long, with summary)		
Attached File	field_news_attached_file	File		
Category	field_news_category	Entity reference		
Image	field_news_image	Image		
Publication date	field_news_publication_date	Date		
Related content	field_news_related_content	Entity reference		
Subtitle	field_news_subtitle	Text (plain, long)		
Title (Mobile)	field_news_title_mobile	Text (plain)		
Workflow	field_news_workflow	Workflow state		

## Defining database connection

Settings file (settings.local.php) entry

```
$databases['migrate']['default'] = array (
  'database' => 'tes_cms',
  'username' => 'xxx',
  'password' => 'xxxxxxxx',
  'prefix' => '',
  'host' => 'hostname'
  'port' => '3306',
  'namespace' => 'Drupal\\Core\\Database\\Driver\\mysql',
  'driver' => 'mysql',
);
```

Define **source**  **connection-name** in config

```
source:
  key: migrate # source database connection
```

## Migration group "news2"

---

- File migrate\_plus.migration\_group.news2.yml
- Group related migrations together
- Define **shared** configuration
- Define shared **constants**
- Run migrations as a group `drush mim --group=news2`
- Use migrate UI

```
1 id: news2
2
3 label: News Full Imports
4
5 description: News migration import group
6
7 source_type: Database
8
9 shared_configuration:
10   source:
11     key: migrate # source database
12
13   # custom selection - for teasers and articles etc
14   selection:
15     from: 375000
16     # to: 99999999
17     live: true
18
19   # tags xref
20   tags_xref:
21     - regions
22     - phases
23     - subjects
24     - tags
25
26   # file folders, overridden on per basis
27   folders:
28     source_domain: 'https://tes.com'
29     source_public_folder_uri: '/sites/default/files'
30     target_folder: 'public://'
31     # public_folder_uri: '/sites/default/files'
32     urldecode: true
33
34   # required refdata
35   required:
36     vids:
37       - regions
38       - phases
39       - subjects
40
41   constants:
42     admin_user: 1
43     format_tes: tg_html
44     permanent_file_status: true
45
46 process:
47   langcode:
48     plugin: default value
49     source: language
50     default_value: 'und'
51
52 dependencies:
53   enforced:
54     module:
55       - tg_news2_migration
```

## Taxonomy term mapping

---

- Pre-populated several vocabularies
- regions, phases, subjects will be used along with tags to provide News Article tagging
- Pulling in 3 taxonomy fields from source to a single field on new content type
- Source field\_news\_article\_tags, field\_news\_category, and field\_news\_section
- If it exists in regions, phases, or subjects then term not migrated
- Migration requirement -> certain vocabularies populated, **checkRequirements()**

## Author images, attachments, teaser (or hero) images

---

- Create file entities for these
- Use core destination plugin `entity:file`
- Want to select target folder for each file source
- Want to "niceify" source file names (remove spaces, uppercase etc)
- Choose single image from teaser or hero fields

## Authors

---

- Create node for each author (`node.type = "author"`)
- Use core destination plugin `entity:node`
- Use `migration_lookup` plugin to map image field to file entity
- **no\_stub: true** - dont create stubs for missing file entities
- Optional dependency - author images migrated first

```
1 id: news2_authors_migration
2
3 label: News (live) authors migration
4
5 migration_group: news2
6
7 migration_tags:
8   - news migration
9   - news authors
10
11 source:
12   plugin: d7_authors
13
14 # override custom selection
15 selection:
16   from: 0
17   to: 99999999
18   live: true
19
20 keys:
21   - nid
22
23
24 destination:
25   plugin: entity:node
26
27 process:
28   title:
29     -
30       plugin: skip_on_empty
31       source: nid
32       method: row
33     -
34       plugin: callback
35       callable: ucwords
36       source: title
37
38   field_author_description: body
39
40   field_author_image/target_id:
41     plugin: migration_lookup
42     migration: news2_author_images_migration
43     source: photo_fid
44     no_stub: true # if not found then dont create "stub"
45
46   field_author_image/alt: image_alt
47
48 type:
49   plugin: default_value
50   default_value: author
51
52 uid: constants/admin_user
53
54 migration_dependencies:
55   optional:
56     - news2_author_images_migration
57
58 dependencies:
59   enforced:
60     modules:
61       - tg_news2_migration
62
63
```

## News articles

---

- Create node for each published news article (`node.type = "news_article"`)
- Destination plugin `entity:node`
- Use `migration_lookup` for category, teaser image, and attachments
- Use `migration_lookup` (to current migration) with **no\_stub:false** to create related content link
- Stub entries will be replaced when converted
- Optional migration dependency on tags, teasers, authors and attachments

- Embedded (inline) files and media downloaded as unmanaged file to target folder
- Existing (site) body images repointed to absolute uri's
- Embedded media tags converted to <img> tags

```
25 process:
26 # had an issue with "stub entries" when I was using the skip functionality, and they were being thrown out!
27 title: title
28
29 # its expected that this is plain text already
30 field_news_subtitle: standfirst
31
32 field_news_body/value: body
33
34 # field_news_body/summary: summary # @todo: looks like we want plain text, will discuss with Scott
35
36 field_news_body/format: constants/format_tes
37
38 field_news_title/mobile: mobile_title # @todo: tba - see prepare row
39
40 field_news_publication_date: publication_date
41
42 field_news_attached_file:
43 -
44   plugin: skip on empty
45   method: process
46   source: attached_files
47
48 -
49   plugin: explode
50   delimiter: ', '
51
52 -
53   plugin: migration lookup
54   migration: news2_attachments_migration
55   no_stub: true
56
57 field_news_image/target_id:
58   plugin: migration lookup
59   migration: news2_leaser_images_migration
60   source: nid # image is a 1:1 mapping, so have done it on NID
61   no_stub: true
62
63 field_news_image/alt: image.alt
64
65 field_news_category:
66 -
67   plugin: skip on empty
68   source: tags
69   method: process
70
71 -
72   plugin: explode
73   delimiter: ', '
74
75 # xref is now done in the source plugin "prepare row" bit
76 #
77 #   plugin: custom migration lookup
78 #   migration: news_tags_migration
79 #   no_stub: true # if not found then dont create "stub"
80
81 field_news_author:
82 -
83   plugin: skip on empty
84   method: process
85   source: author
86
87 -
88   plugin: explode
89   delimiter: ', '
90
91 -
92   plugin: migration lookup
93   migration: news2_authors_migration
94   no_stub: true # if not found then dont create "stub"
95
96 field_news_related_content:
97 -
98   plugin: skip on empty
99   method: process
100   source: related
101
102 -
103   plugin: explode
104   delimiter: ', '
105
106 -
107   plugin: migration lookup
108   migration: news2_articles_migration
109   # no_stub: defaults to false, as its based on the current migration we do want to create a stub for re-population
110
111 # field_news_workflow: @todo
112
113 type:
114   plugin: default value
115   default_value: news_article
116
117 # @todo: use actual user once users extracted - tba?
118 uid: constants/admin_user
119
```

# Migration UI

- provided by migration\_tools module
- migration groups page
- migration group overview page
- individual migration execute page
- individual migration overview, source, process & destination pages

## Migration groups page

### Migrations

[Home](#) » [Administration](#) » [Structure](#)

+ Add migration group

MIGRATION GROUP	MACHINE NAME	DESCRIPTION	SOURCE TYPE	OPERATIONS
Default	default	A container for any migrations not explicitly assigned to a group.		List migrations
News Full Imports	news2	Work in progress News migration import group	Database	List migrations
News Imports	news	Work in progress News migration import group	CSV File	List migrations

## Migration group overview page

- displayed in label (title) order, not in migration dependency order

## Migrations

[Home](#) » [Administration](#) » [Structure](#) » [Migrations](#) » [Edit migration group](#)

MIGRATION	MACHINE NAME	STATUS	TOTAL	IMPORTED	UNPROCESSED	MESSAGES	LAST IMPORTED	OPERATIONS
<a href="#">News (live) articles, and inline files, migration</a>	news2_articles_migration	Idle	124	0	124	0		<a href="#">Execute</a>
<a href="#">News (live) attached files migration</a>	news2_attachments_migration	Idle	107	107	0	0	2018-03-02 11:25:10	<a href="#">Execute</a>
<a href="#">News (live) author images migration</a>	news2_author_images_migration	Idle	18	18	0	0	2018-03-02 10:48:36	<a href="#">Execute</a>
<a href="#">News (live) authors migration</a>	news2_authors_migration	Idle	16735	16734	1	0	2018-03-02 11:40:54	<a href="#">Execute</a>
<a href="#">News (live) tags migration</a>	news2_tags_migration	Idle	284	0	284	0		<a href="#">Execute</a>
<a href="#">News (live) teaser images migration</a>	news2_teaser_images_migration	Idle	124	124	0	0	2018-03-02 11:42:49	<a href="#">Execute</a>

## Migration ID Execute page

- Operations - import, rollback, stop, reset
- Update checkbox
- Ignore dependencies checkbox

## Execute migration

[Home](#) » [Administration](#) » [Structure](#) » [Migrations](#) » [Edit migration group](#) » [Migrations](#) » [Migration overview](#)

### OPERATIONS

Choose an operation to run \*

Import ▼

[Execute](#)

### Definitions

- Import: Imports all previously unprocessed records from the source, plus any records marked for update, into destination Drupal objects.
- Rollback: Deletes all Drupal objects created by the import.
- Stop: Cleanly interrupts any import or rollback processes that may currently be running.
- Reset: Sometimes a process may fail to stop cleanly, and be left stuck in an Importing or Rolling Back status. Choose Reset to clear the status and start a new process.

### OPTIONS

- ☐ Update  
Check this box to update all previously-imported content in addition to importing new content. Leave unchecked to only import new content
- ☐ Ignore dependencies  
Check this box to ignore dependencies when running imports - all tasks will run whether or not their dependent tasks have completed.

## Migration ID Overview page

# Migration overview

View

Edit

Messages

Delete

Overview

Source

Process

Destination

Home » Administration » Structure » Migrations » Edit migration group » Migrations

## OVERVIEW

**Group:**  
news2

**Description:**  
News (live) articles, and inline files, migration

**Soft Migration Dependencies**  
news2\_tags\_migration, news2\_teaser\_images\_migration, news2\_attachments\_migration

## Migration ID Source page

### Source

View

Edit

Messages

Delete

Overview

Source

Process

Destination

Home » Administration » Structure » Migrations » Edit migration group » Migrations » Migration overview

### SOURCE

#### Query

```
SELECT n.nid AS nid, n.title AS title, body.body_value AS body, standfirst.field_news_article_standfirst_value AS standfirst, post_script.field_news_article_post_script_value AS post_script, publication_date.field_publication_date_value AS publication_date, author.field_news_article_byline_target_id AS author
FROM
  (node) n
LEFT OUTER JOIN (field_data_body) body ON n.nid = body.entity_id AND body.entity_type = :type AND body.bundle = :n.type AND body.deleted = 0 AND body.delta = 0
LEFT OUTER JOIN (field_data_field_news_article_standfirst) standfirst ON n.nid = standfirst.entity_id AND standfirst.entity_type = :type AND standfirst.bundle = :n.type AND standfirst.deleted = 0 AND standfirst.delta = 0
LEFT OUTER JOIN (field_data_field_news_article_post_script) post_script ON n.nid = post_script.entity_id AND post_script.entity_type = :type AND post_script.bundle = :n.type AND post_script.deleted = 0 AND post_script.delta = 0
LEFT OUTER JOIN (field_data_field_publication_date) publication_date ON n.nid = publication_date.entity_id AND publication_date.entity_type = :type AND publication_date.bundle = :n.type AND publication_date.deleted = 0 AND publication_date.delta = 0
LEFT OUTER JOIN (field_data_field_news_article_byline) author ON n.nid = author.entity_id AND author.entity_type = :type AND author.bundle = :n.type AND author.deleted = 0 AND author.delta = 0
WHERE (n.type = :db_condition_placeholder_0) AND (n.nid BETWEEN :db_condition_placeholder_1 AND :db_condition_placeholder_2) AND (n.status = :db_condition_placeholder_3)
```

MACHINE NAME	DESCRIPTION
nid	Node id
title	Article Title
body	Article Body
standfirst	Standfirst content
post_script	Postscript content
publication_date	Publication date
author	Author
attached_files	Attached files
tags	News taxonomy terms
related	Related content
mobile_title	Derived field
image_alt	Derived field

These are the fields available from the source of this migration task. The machine names listed here may be used as sources in the process pipeline.

## Migration ID Process page

View

Edit

Messages

Delete

Overview

Source

Process

Destination

Home » Administration » Structure » Migrations » Edit migration group » Migrations » Migration overview

### PROCESS

DESTINATION	SOURCE	PROCESS PLUGIN	DEFAULT
language	language	default_value	und
title	nid	skip_on_empty	
field_news_subtitle	standfirst	get	
field_news_body/value	body	get	
field_news_body/format	constants/format_tes	get	
field_news_title_mobile	mobile_title	get	
field_news_publication_date	publication_date	get	
field_news_attached_file	attached_files	skip_on_empty	
field_news_image/target_id	nid	migration_lookup	
field_news_image/alt	image_alt	get	
field_news_category		skip_on_empty	
field_news_author	author	skip_on_empty	
field_news_related_content	related	skip_on_empty	
type		default_value	news_article
uid	constants/admin_user	get	

Run

## Migration ID Destination page



## Destination

[View](#)[Edit](#)[Messages](#)[Delete](#)[Overview](#)[Source](#)[Process](#)[Destination](#)

[Home](#) » [Administration](#) » [Structure](#) » [Migrations](#) » [Edit migration group](#) » [Migrations](#) » [Migration overview](#)

### DESTINATION

#### Type

entity:node

#### MACHINE NAME

#### DESCRIPTION

No fields

These are the fields available in the destination plugin of this migration task. The machine names are those available to be used as the keys in the process pipeline.

## Source Plugins

- separate source plugin for each migration
- `d7_news_files` migration used for multiple file type migrations, just config (selection & target folder) different
- source plugin classes are extending migrate's `SqlBase` or `File` classes
- source plugins will generally implement: **`query()`**, **`fields()`**, and **`prepareRow()`** methods
- we are also implementing **`checkRequirements()`**
- source data transformation (and possible rejection - `return FALSE` ) is done in **`prepareRow()`**

For details on the use source plugins have a look at <https://www.drupal.org/docs/8/api/migrate-api/migrate-source-plugins/overview-of-migrate-source-plugins>

## Process Plugins

- minimal use of process plugins
- source plugins have already transformed much of the data (see earlier)
- core process plugins make our life easier
- use "out of the box" functionality wherever possible
- only one custom process plugin **`d7_file_download`**

Core or Custom	Name	Notes
Core	callback	We used this one in the authors node title (just to run a <i>ucwords</i> against the supplied Author Title field)
Core	download	File download plugin - creates the target uri to our files (guzzle get)
Core	default_value	Supply a default value for a field process, e.g. content type field in News or Authors
Core	skip_on_empty	Skip process (or skip row) in a field process
Core	explode	Turn a delimited string into an array, thereby allowing a 1:M field explosion (e.g. news tags)
Core	migration_lookup	Standard mechanism of mapping entity ids (xref for entities created in other migrations or maybe even in this migration)
Custom	d7_news_file_download	A custom file download (extends core Download) allowing for revised guzzle options.

For more on the core process plugins see <https://www.drupal.org/docs/8/api/migrate-api/migrate-process-plugins/list-of-core-process-plugins>

## Custom process plugin: d7\_news\_file\_download

- Extending existing migrate Download class
- Provides ability to override default guzzle\_options

### Show me the code

```
namespace Drupal\tg_news2_migration\Plugin\migrate\process;

use Drupal\Core\File\FileSystemInterface;
use Drupal\migrate\Plugin\migrate\process\Download;
use GuzzleHttp\Client;

/**
 * @see Core Download process plugin for more details.
 *
 * Example:
 *
 * @code
 * process:
 *   plugin: d7_news_file_download
 *   source:
 *     - source_url
 *     - destination_uri
 *   guzzle:
 *     # ability to override guzzle options
 *     timeout: 60 # default is 30
 *
 * @MigrateProcessPlugin(
 *   id = "d7_news_file_download"
 * )
 */
class D7NewsFileDownload extends Download {

  /**
   * {@inheritdoc}
   */
  public function __construct(array $configuration, $plugin_id, array $plugin_definition, FileSystemInterface $file_system, Client $http_client) {
    if (isset($this->configuration['guzzle']) && is_array($this->configuration['guzzle'])) {
      foreach ($this->configuration['guzzle'] as $option => $value) {
        // overwrite any existing config option
        $configuration['guzzle_options'][$option] = $value;
      }
    }
    parent::__construct($configuration, $plugin_id, $plugin_definition, $file_system, $http_client);
  }
}
```

## Destination Plugins

- This migration is using no custom destination plugins
- All "out of the box"
- entity:file
- entity:taxonomy\_term
- entity:node
- Possibly entity:user if we decide to migrate users

For more on destination plugins have a look at <https://www.drupal.org/docs/8/api/migrate-api/migrate-destination-plugins-examples>

## Useful drush commands

- Provided by migrate\_tools module
- migrate-status (ms)
- migrate-import (mim)
- migrate-rollback (mr)
- migrate-stop (mst)
- migrate-reset-status (mrs)
- migrate-message (mmsg)
- most useful: **migrate-status**, **migrate-import**, **migrate-reset-status**
- use --help option to find out more

## drush migrate-status (ms)

- drush ms --group=news2
- same information as displayed in UI
- but, displayed in migration dependency order

```
cmd: drush ms --group=news2
```

Group	Migration ID	Status	Total	Imported	Unprocessed	Last Imported
News Full Imports (news2)	news2_author_images_migration	Idle	18	18	0	2018-02-26 20:15:38
News Full Imports (news2)	news2_tags_migration	Idle	284	245	39	2018-02-26 20:15:39
News Full Imports (news2)	news2_authors_migration	Idle	16735	16734	1	2018-02-26 20:23:34
News Full Imports (news2)	news2_teaser_images_migration	Idle	124	124	0	2018-02-26 20:24:14
News Full Imports (news2)	news2_attachments_migration	Idle	107	107	0	2018-03-02 06:48:21
News Full Imports (news2)	news2_articles_migration	Idle	124	124	0	2018-02-26 20:51:33

```
-> github-cms-tes-platform-v2 : news_migration | ../modules/custom>
```

## drush migrate-import (mim)

```
# run all migrations in named group (in dependency order)
drush mim --group=news2

# will run just the migraton IDs listed - unprocessed items only
drush mim news2_tags_migration,news2_authors_migration

# run the named migration with update flag
drush mim news2_tags_migration --update
```

Options available:

```
cmd: drush mim --help
Perform one or more migration processes.

Examples:
migrate:import --all                Perform all migrations
migrate:import --group=beer         Import all migrations in the beer group
migrate:import --tag=user           Import all migrations with the user tag
migrate:import --group=beer --tag=user Import all migrations in the beer group and with the user tag
migrate:import beer_term,beer_node  Import new terms and nodes
migrate:import beer_user --limit=2  Import no more than 2 users
migrate:import beer_user --idlist=5 Import the user record with source ID 5

Arguments:
[migration_names] ID of migration(s) to import. Delimit multiple using commas.

Options:
--all[=ALL]                Process all migrations.
--group[=GROUP]            A comma-separated list of migration groups to import
--tag[=TAG]                Name of the migration tag to import
--limit[=LIMIT]            Limit on the number of items to process in each migration
--feedback[=FEEDBACK]     Frequency of progress messages, in items processed
--idlist[=IDLIST]          Comma-separated list of IDs to import
--update[=UPDATE]          In addition to processing unprocessed items from the source, update previously-imported items with the current data
--force[=FORCE]            Force an operation to run, even if all dependencies are not satisfied
--execute-dependencies[=EXECUTE-DEPENDENCIES] Execute all dependent migrations first.

Aliases: mim, migrate-import
-> github-cms-tes-platform-v2 : news_migration | ../modules/custom>
```

## drush migrate-reset-status (mrs)

- useful if your migration has failed and is left in a non "Idle" status.

```
drush mrs news2_articles_migration
```

## Dealing with migration errors

- strategy is to code around them (from our testing)
- however errors will occur

```
cmd: drush mim --group=news2
[notice] Processed 18 items (18 created, 0 updated, 0 failed, 0 ignored) - done with 'news2_author_images_migration'
[notice] Processed 0 items (0 created, 0 updated, 0 failed, 0 ignored) - done with 'news2_tags_migration'
[notice] Processed 16734 items (16734 created, 0 updated, 0 failed, 0 ignored) - done with 'news2_authors_migration'
[notice] Processed 124 items (124 created, 0 updated, 0 failed, 0 ignored) - done with 'news2_teaser_images_migration'
[notice] Processed 107 items (104 created, 0 updated, 3 failed, 0 ignored) - done with 'news2_attachments_migration'
[error] news2_attachments_migration Migration - 3 failed.
```

- because run via drush, optional dependencies honoured and migration stop at failure
- in this case errors were visible in migrate\_map\_[ID] table

```
source_ids_hash: 6de7e724dab38c14560a1ab7e3d64573654d0791c2db6836d152a558a66df2c1
sourceid1: 87142
destid1: null
source_row_status: 3
rollback_action: 0
last_imported: 0
hash: null
```

- also available in migrate messages table

```
msgid: 1 source_ids_hash: 6de7e724dab38c14560a1ab7e3d64573654d0791c2db6836d152a558a66df2c1 level: 1
message: cURL error 28: Operation timed out after 29998 milliseconds with 57946981 out of 64995621 bytes received
(see ...)
```

- can run a partial import `drush mim news2_attachments_migration --idlist="87142,87421,87394" --update`
- may need to remove error entries from map table first

```
DELETE FROM migrate_map_news2_attachments_migration WHERE destid1 IS NULL AND source_row_status != 0;
```

## Gotchas

- config changes only imported on install
- config left over after module uninstall - resolved by adding an **enforced dependency** to the migration config files
- loads of migration tables
- unexpected warning messages when running `drush ms`
- errors and the (not so) white screen of death
- cryptic error messages
- for requirements exceptions look at the log
- stub entry creation
- redirect to install.php if non-existent source db

## migrate\_plus "Beer Example"

- Very useful example

- Variety of entity creation methods - user, taxonomy\_term, node
- Config setup
- Migration groups
- Use of shared configuration and constants
- Use of migration dependencies
- Stub entity creation
- Custom migration plugins (source plugins)
- Use of (some of the) core process plugins - including: **callback**, **default\_value**, **static\_map**, **migration\_lookup**
- Loads of in-code documentation

See <https://github.com/jigarius/drupal-migration-example> for an example of a config only migration

## Useful links, in no specific order

---

- [https://www.drupal.org/project/migrate\\_plus](https://www.drupal.org/project/migrate_plus)
- <https://www.drupal.org/docs/8/api/migrate-api/migrate-api-overview>
- <https://www.drupal.org/project/migrate/issues/2728233>
- <https://github.com/jigarius/drupal-migration-example>
- <https://agencychief.com/blog/drupal-8-csv-migration>
- [https://github.com/wunderio/migrate\\_source\\_example](https://github.com/wunderio/migrate_source_example)
- <https://evolvingweb.ca/blog/drupal-8-migration-migrating-basic-data-part-1>
- <https://deninet.com/blog/2017/06/07/building-custom-migration-drupal-8-part-3-users-and-roles>
- <https://www.colorado.edu/webcentral/2017/04/04/writing-node-migrations-drupal-7-drupal-8>
- <https://www.sitepoint.com/your-first-drupal-8-migration/>
- <https://www.metaltoad.com/blog/drupal-8-migrations-part-3-migrating-taxonomies-drupal-7>
- <https://www.metaltoad.com/blog/drupal-8-migrations-part-4-migrating-nodes-drupal-7>
- <https://www.drupaleasy.com/blogs/ultimike/2016/04/drupal-6-drupal-81x-custom-content-migration>