

# Introduction to Drupal 9 migrations

Agaric.coop online webinars



**agaric**

**USA | Nicaragua | México | Germany**



<https://agoric.coop/31-days-drupal-migrations>

# Upcoming Full-day Trainings



- July 13 – Getting started with Drupal 9
- July 21 – Drupal 8/9 content migrations
- July 23 – Upgrading to Drupal 8/9 using the Migrate API

<https://agoric.coop/training>



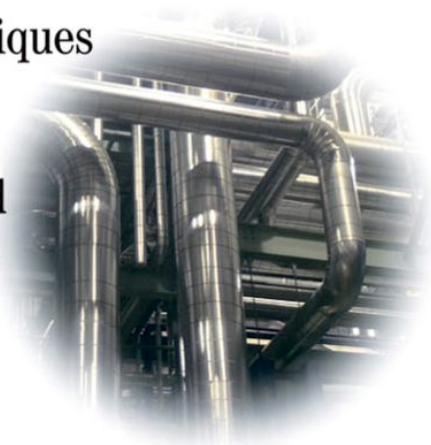
 WILEY

TIMELY. PRACTICAL. RELIABLE.

# The Data Warehouse ETL Toolkit

Practical Techniques  
for Extracting,  
Cleaning,  
Conforming, and  
Delivering Data

**Ralph Kimball**  
Joe Caserta



Extract  
Transform  
Load

# Extract (Source plugin)



# Transform (Process plugin)





# Load (Destination plugin)



**MY FIRST MIGRATION**



**LET'S DO THIS**



Google Sheet

files

get

nodes

users

static map

JSON file

SQL database

default\_value

CSV file

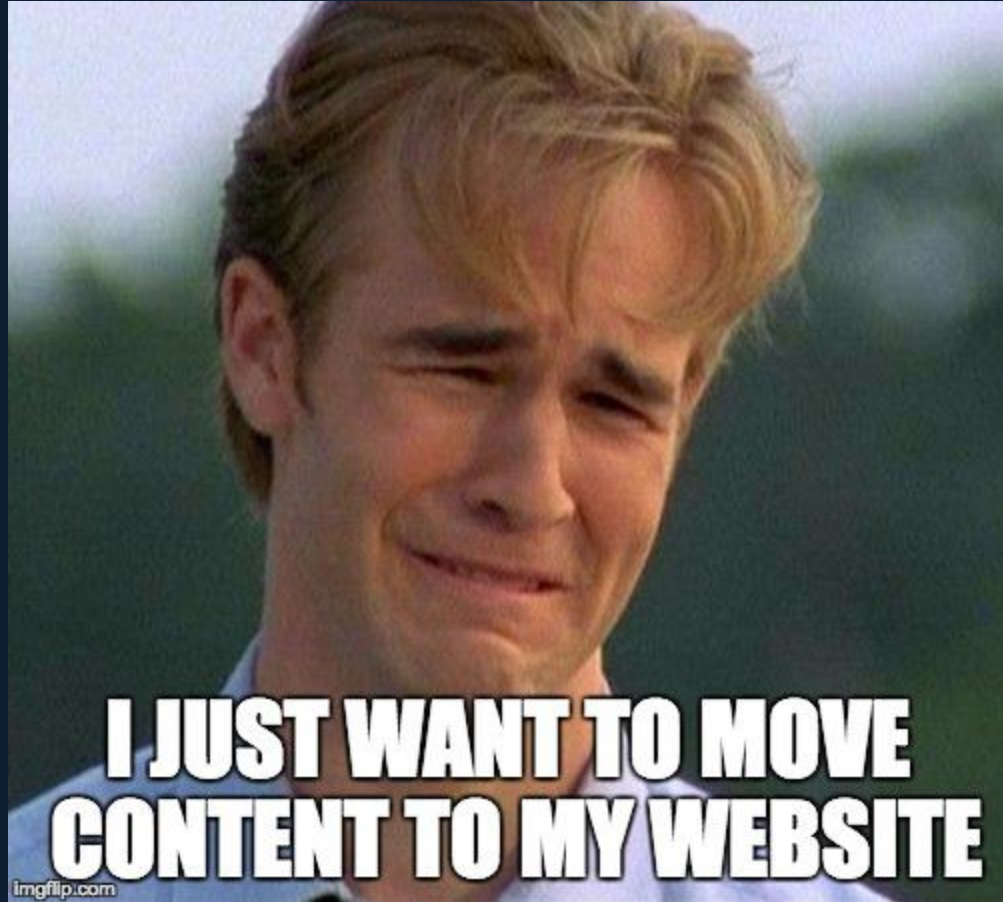
concat

iterator

paragraphs

XML file

migration\_lookup





## SOURCE

SQL database

CSV file

XML file

JSON file

Google Sheet

## PROCESS

default\_value

concat

iterator

get

migration\_lookup

## DESTINATION

nodes

users

images

files

paragraphs

# Source



	A	B	C
1	<b>First Name</b>	<b>Last Name</b>	<b>Department</b>
2	Michele	Metts	Biology
3	Benjamin	Melançon	Anthropology
4	Stefan	Freudenberg	Physics



# Process



# Destination

	A	B
1	<b>Name</b>	<b>Department</b>
2	Michele Metts	Biology
3	Benjamin Melançon	Anthropology
4	Stefan Freudenberg	Physics

# Common modules



```
composer require \  
    'drush/drush' \  
    'drupal/migrate_plus:^5.1' \  
    'drupal/migrate_tools:^5.0' \  
    'drupal/migrate_source_csv:^3.4'
```

# Example migration



```
id: ud_staff_csv_node_init
label: 'Import staff nodes from CSV'
source:
  plugin: csv
  path: modules/custom/ud_staff/sources/ud_staff.csv
  ids: ['Staff ID']
process:
  title: 'First Name'
destination:
  plugin: 'entity:node'
  default_bundle: ud_staff
```



# A two step process



- Write the migration YAML definition files.
- Execute the migrations.
  - From the command line (code and config).
  - From the user interface (config).



# Two ways to define migrations

- Code
  - Works out the box with Drupal core.
  - Files are placed on /migrations folder.
  - Rebuild caches to detect changes.
  - Run from the command line.
- Configuration entity
  - Require the Migrate Plus module.
  - Files are placed on /config/install folder.
  - Import (sync) configuration to detect changes.
  - Run from the user interface with Migrate Tools module.

# Tips for writing migrations



- Start from an existing migration. Look for an example online that does something close to what you need and modify it to your requirements.
- Pay close attention to the syntax of the YAML file. An extraneous space or wrong indentation level can break the whole migration.
- Read the documentation to know which source, process, and destination plugins are available. One might exist already that does exactly what you need.

# Tips for writing migrations



- Make sure to read the documentation for the specific plugins you are using. Many times a plugin offer optional configurations.
- Look for contributed modules that might offer more plugins or upgrade paths from previous versions of Drupal.
- When writing the migration pipeline, map one field at a time. Problems are easier to isolate if there is only one thing that could break at a time.

# Tips for writing migrations



- When mapping a field, work on one subfield at a time if possible. Some field types like images and addresses offer many subfields.
- There is no need to do every data transformation using the Migrate API. When there are edge cases, you can manually update those after the automated migration is completed.
- Commit to your code repository any and every change that produces right results.

# Tips for writing migrations



- Learn about debugging migrations.
- Seek help from the community. Migrate maintainers and enthusiasts are very active and responsive in the #migrate channel of Drupal slack.
- If you feel stuck, take a break and come back to it later.



# Live demo (what can go wrong?)





# Field mapping syntax



```
process:
  destination_field/0/subfield_1:
    plugin: plugin_name
    config_1: value_1
    config_2: value_2
  destination_field/0/subfield_2:
    plugin: plugin_name
    config_1: value_1
    config_2: value_2
  destination_field/1/subfield_1:
    plugin: plugin_name
    config_1: value_1
    config_2: value_2
  destination_field/1/subfield_2:
    plugin: plugin_name
    config_1: value_1
    config_2: value_2
```

Visit <https://agaric.coop/blog/understanding-syntax-drupal-migrations> for more information.

# Process pipeline syntax



```
process:
  destination_field:
    - plugin: plugin_name
      source: source_column_name
      config_1: value_1
      config_2: value_2
    - plugin: plugin_name
      config_1: value_1
      config_2: value_2
    - plugin: plugin_name
      config_1: value_1
      config_2: value_2
```

Visit <https://agaric.coop/blog/understanding-syntax-drupal-migrations> for more information.

# Date field



Expected destination format for date only field

SUBFIELD	TYPE	EXAMPLE *
value	string	2020-03-08

Expected destination format for date and time field

SUBFIELD	TYPE	EXAMPLE *
value	string	2020-03-08T07:03:21

\* Visit <https://agaric.coop/blog/migrating-dates-drupal> for format characters.

# Date field transformation



Expected destination format for date only field

	PATTERN *	EXAMPLE
source	F j, Y	March 8, 2020
destination	Y-m-d	2020-03-08

Expected destination format for date and time field

	PATTERN *	EXAMPLE
source	F j, Y H:i:s	March 8, 2020 07:03:21
destination	Y-m-d\TH:i:s	2020-03-08T07:03:21

\* Visit <https://www.php.net/manual/en/function.date.php> for format characters.



# Modules

## **Migrate Plus**

[drupal.org/project/migrate\\_plus](https://drupal.org/project/migrate_plus)

## **Migrate Tools**

[drupal.org/project/migrate\\_tools](https://drupal.org/project/migrate_tools)

## **Migrate Source CSV**

[drupal.org/project/migrate\\_source\\_csv](https://drupal.org/project/migrate_source_csv)

<https://agaric.coop/blog/list-migration-related-drupal-modules>

# Thanks



- **Lucas Hedding (heddn)**
- **Adam Globus-Hoenich (phenaproxima)**
- **Vicki Spagnolo (quietone)**
- **Michael Lutz (mikelutz)**
- **Markus Sipilä (masipila)**
- + **All migrate contributors**







# Upcoming Full-day Trainings



- July 13 – Getting started with Drupal 9
- July 21 – Drupal 8/9 content migrations
- July 23 – Upgrading to Drupal 8/9 using the Migrate API

<https://agoric.coop/training>

# Please Provide Feedback



<https://agaric.coop/r/drupal-migrations-feedback>

@dinarcon | #D9MigrateExample | mauricio@agaric.com