Getting Content into Drupal Using Migrate

Version 0.4 (Montréal 26 October 2013)

Ryan Weal
Kafei Interactive Inc.
Montréal QC

ryan@kafei.ca
Twitter : http://twitter.com/ryan_weal
Pump.io : http://comn.ca/ryanweal
Git tip : https://www.gittip.com/ryan_weal/

What is Migrate?

- A pre-built framework for porting content to new Drupal sites from any system
- A review structure that allows content administrators to see progress
- An alternative to upgrading old Drupal sites

Thinking of Upgrading Drupal?

- A long process, involving taking backup copies and occasionally *rolling back the entire site*
- Disable all of the modules
- Removing the theme, set to Garland
- Swap out the filesystem for D7
- Run the upgrade.php script, truncate your sessions and clear the caches (cross fingers)
- Unlucky? Delete or fix problematic variables, one by one
- Now all you have is Drupal Core (field data still exists, but no UI)

Upgrading the rest of Drupal

- Get the CCK module, enable content_migrate
- Find replacements to your field modules
- Run update.php every time you get a module, then run content_migrate for each field
- Be prepared to run custom queries to cleanup after bugs
- Now all you have is an upgraded site that is two weeks old

Upgrading post-snapshot data

- Feeds
  - Rejects duplicates, FUN!
  - Can mutate input with Feeds Tamper, with only 2700 clicks!
  - No way to easily capture errors
- Add XML : now you have two problems!
  - How about we use Regular Expressions to fix the XML?
  - Did anybody write down all the regexes we did?
  - Click-based mappings
- Temptation of regexes and the lost steps
- Oops, we screwed up. Let's use node_convert
  - does it support $x field?

How do we keep track of all of that?

- Upgrade core
- Upgrading all the fields
- New modules and themes
- Fixing bugs
- Extra cruft in the database
- Dealing with feature changes between versions

Skip upgrading, use Migrate!

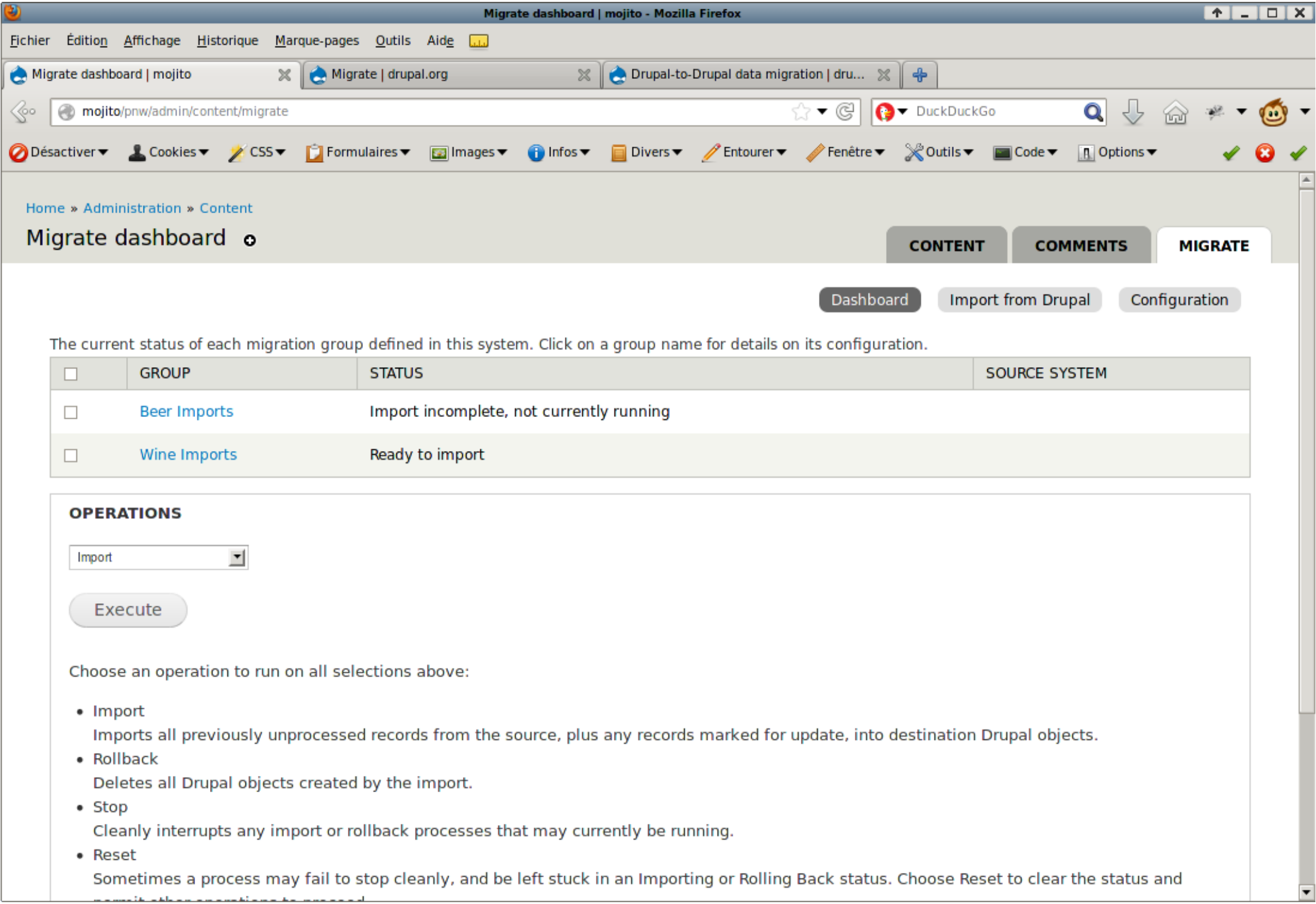
- Migrate will do the dirty work
  - Developer will map the fields
  - Edge cases can be fixed in code
  - Migrate will do the rest
- New site development can start fresh (can use upgraded site)
- Rearchitect problematic areas from the old implementation
- Review your migrations with your team and content administrators

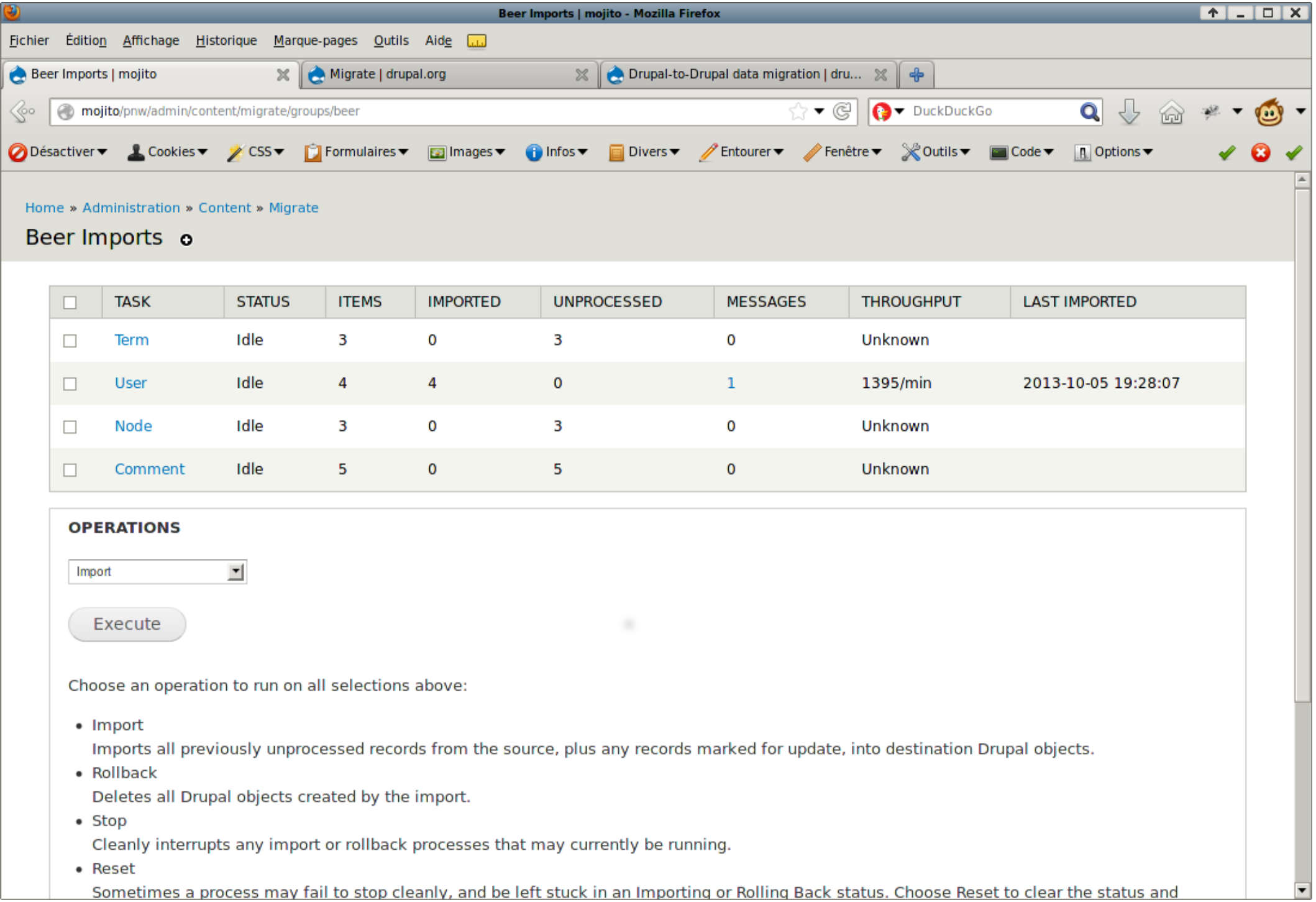
Or migrate from a non-Drupal source :

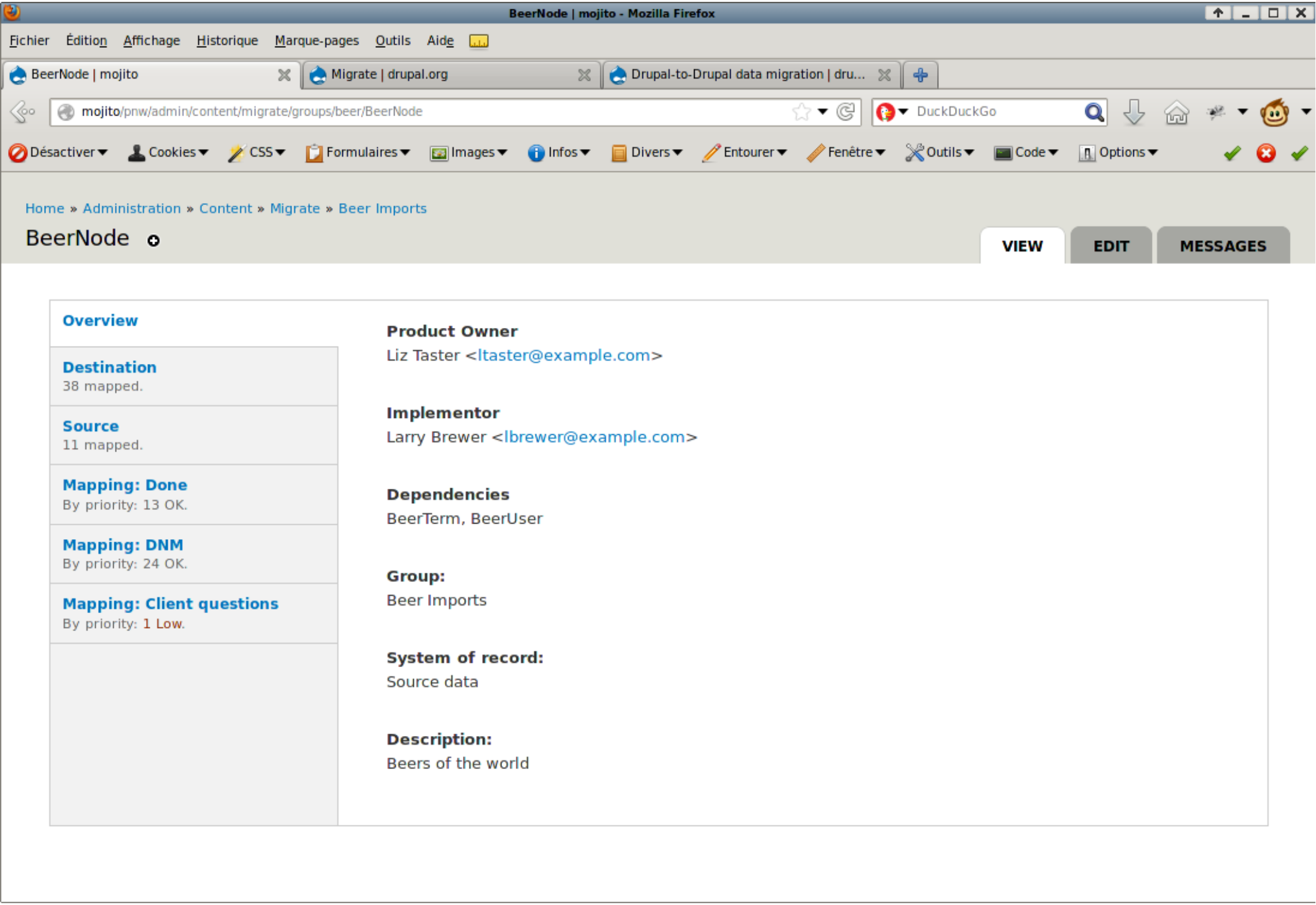
- Other content management systems, using pre-built migrations (Wordpress, Typo3)
- Files! Such as XML, CSV
  - Can be as multiple files
- Different databases / legacy systems using any supported databse driver (Postgres, MS SQL, Oracle)

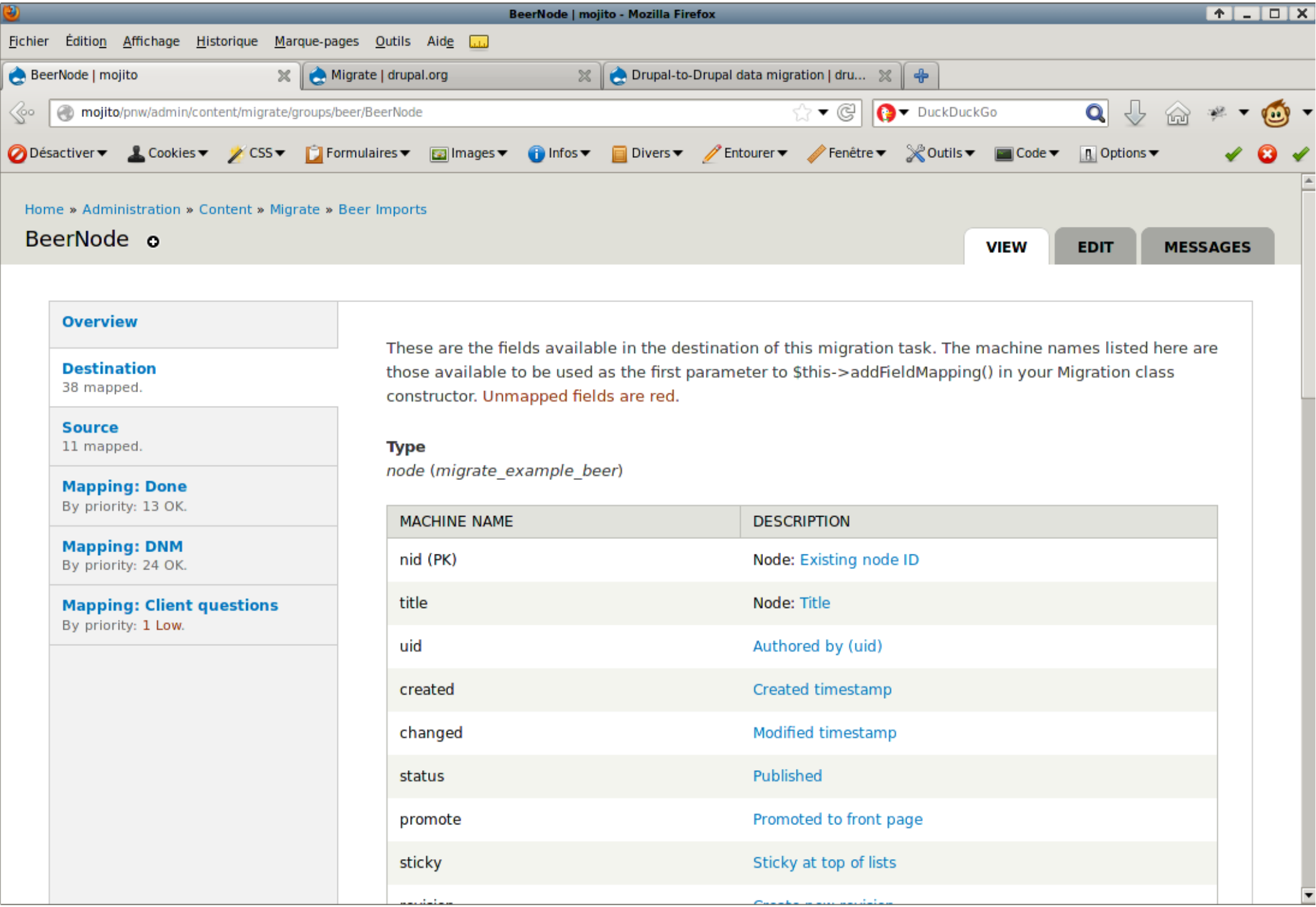
Migration in a Nutshell

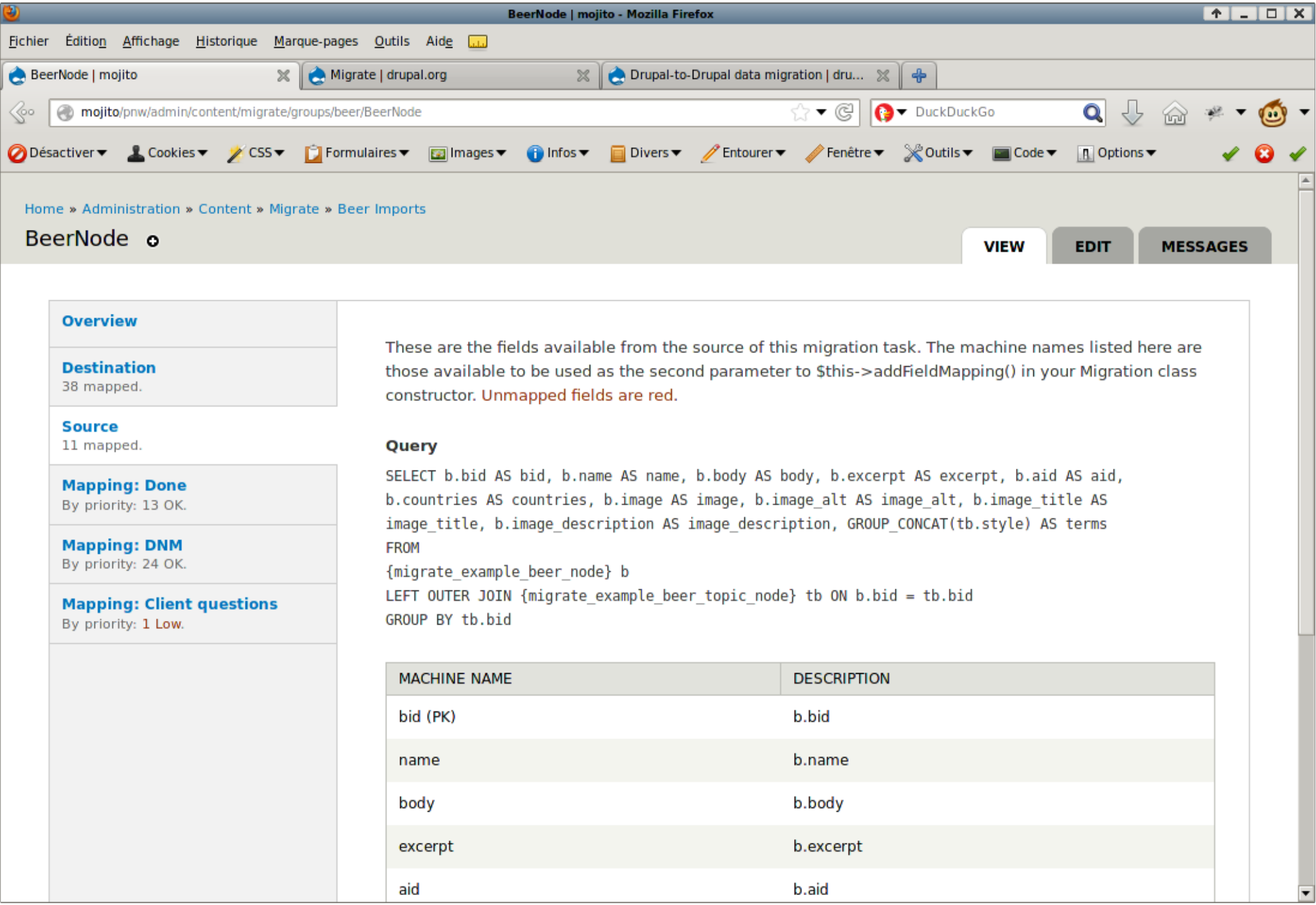
- The process is much simplified compared to an upgrade :
  - Install Drupal
  - Add the migrate module
  - Add modules that contain base classes you want
  - Create your custom-coded migration classes to extend the templates
  - Connect to your old database(s)
  - Run the migration templates in bulk


Migrate dashboard | mojito - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils Aide
Migrate dashboard | mojito
Migrate | drupal.org
Drupal-to-Drupal data migration | dru...
mojito/pnw/admin/content/migrate
DuckDuckGo
Désactiver Cookies CSS Formulaires Images Infos Divers Entourer Fenêtre Outils Code Options
Home » Administration » Content
Migrate dashboard
CONTENT
COMMENTS
MIGRATE
Dashboard
Import from Drupal
Configuration
The current status of each migration group defined in this system. Click on a group name for details on its configuration.
GROUP
STATUS
SOURCE SYSTEM
Beer Imports
Import incomplete, not currently running
Wine Imports
Ready to import
OPERATIONS
Import
Execute
Choose an operation to run on all selections above:
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


Beer Imports | mojito - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils Aide
Beer Imports | mojito
Migrate | drupal.org
Drupal-to-Drupal data migration | dru...
mojito/pnw/admin/content/migrate/groups/beer
DuckDuckGo
Désactiver Cookies CSS Formulaires Images Infos Divers Entourer Fenêtre Outils Code Options
Home » Administration » Content » Migrate
Beer Imports
TASK STATUS ITEMS IMPORTED UNPROCESSED MESSAGES THROUGHPUT LAST IMPORTED
Term Idle 3 0 3 0 Unknown
User Idle 4 4 0 1 1395/min 2013-10-05 19:28:07
Node Idle 3 0 3 0 Unknown
Comment Idle 5 0 5 0 Unknown
OPERATIONS
Import
Execute
Choose an operation to run on all selections above:
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

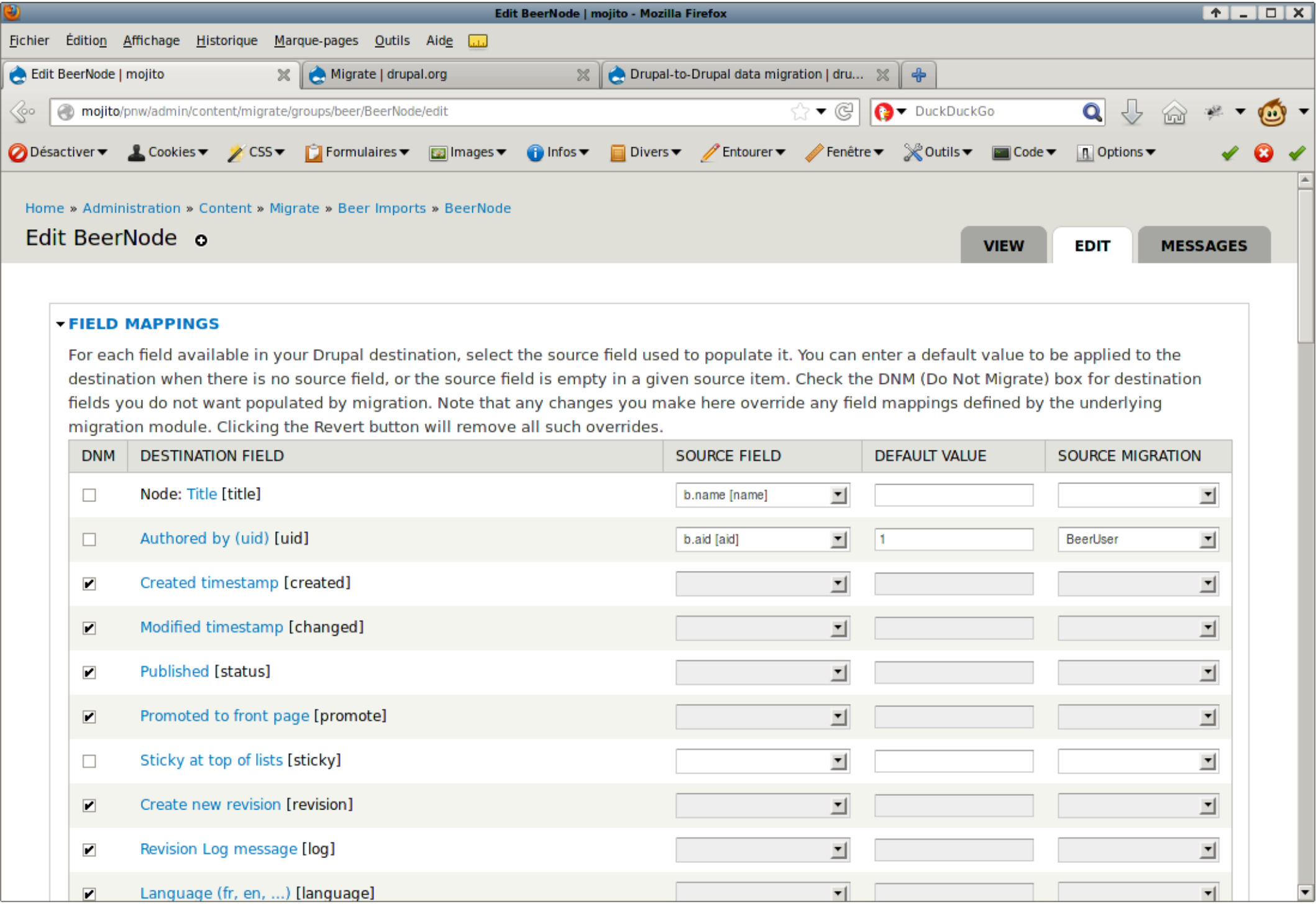

BeerNode | mojito - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils Aide
BeerNode | mojito
Migrate | drupal.org
Drupal-to-Drupal data migration | dru...
mojito/pnw/admin/content/migrate/groups/beer/BeerNode
DuckDuckGo
Désactiver Cookies CSS Formulaires Images Infos Divers Entourer Fenêtre Outils Code Options
Home » Administration » Content » Migrate » Beer Imports
BeerNode
VIEW
EDIT
MESSAGES
Overview
Destination
38 mapped.
Source
11 mapped.
Mapping: Done
By priority: 13 OK.
Mapping: DNM
By priority: 24 OK.
Mapping: Client questions
By priority: 1 Low.
Product Owner
Liz Taster <ltaster@example.com>
Implementor
Larry Brewer <lbrewer@example.com>
Dependencies
BeerTerm, BeerUser
Group:
Beer Imports
System of record:
Source data
Description:
Beers of the world


BeerNode | mojito - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils Aide
BeerNode | mojito
Migrate | drupal.org
Drupal-to-Drupal data migration | dru...
mojito/pnw/admin/content/migrate/groups/beer/BeerNode
DuckDuckGo
Désactiver Cookies CSS Formulaires Images Infos Divers Entourer Fenêtre Outils Code Options
Home » Administration » Content » Migrate » Beer Imports
BeerNode
VIEW
EDIT
MESSAGES
Overview
Destination
38 mapped.
Source
11 mapped.
Mapping: Done
By priority: 13 OK.
Mapping: DNM
By priority: 24 OK.
Mapping: Client questions
By priority: 1 Low.
These are the fields available in the destination of this migration task. The machine names listed here are those available to be used as the first parameter to $this->addFieldMapping() in your Migration class constructor. Unmapped fields are red.
Type
node (migrate_example_beer)
MACHINE NAME
DESCRIPTION
nid (PK)
Node: Existing node ID
title
Node: Title
uid
Authored by (uid)
created
Created timestamp
changed
Modified timestamp
status
Published
promote
Promoted to front page
sticky
Sticky at top of lists


BeerNode | mojito - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils Aide
BeerNode | mojito
Migrate | drupal.org
Drupal-to-Drupal data migration | dru...
mojito/pnw/admin/content/migrate/groups/beer/BeerNode
DuckDuckGo
Désactiver Cookies CSS Formulaires Images Infos Divers Entourer Fenêtre Outils Code Options
Home » Administration » Content » Migrate » Beer Imports
BeerNode
VIEW
EDIT
MESSAGES
Overview
Destination
38 mapped.
Source
11 mapped.
Mapping: Done
By priority: 13 OK.
Mapping: DNM
By priority: 24 OK.
Mapping: Client questions
By priority: 1 Low.
These are the fields available from the source of this migration task. The machine names listed here are those available to be used as the second parameter to $this->addFieldMapping() in your Migration class constructor. Unmapped fields are red.
Query
SELECT b.bid AS bid, b.name AS name, b.body AS body, b.excerpt AS excerpt, b.aid AS aid,
b.countries AS countries, b.image AS image, b.image_alt AS image_alt, b.image_title AS
image_title, b.image_description AS image_description, GROUP_CONCAT(tb.style) AS terms
FROM
{migrate_example_beer_node} b
LEFT OUTER JOIN {migrate_example_beer_topic_node} tb ON b.bid = tb.bid
GROUP BY tb.bid
MACHINE NAME
DESCRIPTION
bid (PK)
b.bid
name
b.name
body
b.body
excerpt
b.excerpt
aid
b.aid

Drupal 2 Drupal migration!

- Collection of templates for Drupal Core, versions 5, 6 and 7
- Nodes, taxonomies, users, roles, comments, etc.
  - Basic settings already mapped
  - Published, sticky, etc...
- The dev version offers a UI based approach!
  - You can edit custom coded modules also!


Edit BeerNode | mojito - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils Aide
Edit BeerNode | mojito
Migrate | drupal.org
Drupal-to-Drupal data migration | dru...
mojito/pnw/admin/content/migrate/groups/beer/BeerNode/edit
DuckDuckGo
Désactiver Cookies CSS Formulaires Images Infos Divers Entourer Fenêtre Outils Code Options
Home » Administration » Content » Migrate » Beer Imports » BeerNode
Edit BeerNode
VIEW
EDIT
MESSAGES
FIELD MAPPINGS
For each field available in your Drupal destination, select the source field used to populate it. You can enter a default value to be applied to the destination when there is no source field, or the source field is empty in a given source item. Check the DNM (Do Not Migrate) box for destination fields you do not want populated by migration. Note that any changes you make here override any field mappings defined by the underlying migration module. Clicking the Revert button will remove all such overrides.
DNM
DESTINATION FIELD
SOURCE FIELD
DEFAULT VALUE
SOURCE MIGRATION
Node: Title [title]
b.name [name]
Authored by (uid) [uid]
b.aid [aid]
1
BeerUser
Created timestamp [created]
Modified timestamp [changed]
Published [status]
Promoted to front page [promote]
Sticky at top of lists [sticky]
Create new revision [revision]
Revision Log message [log]
Language (fr, en, ...) [language]

Things Migrate is really good at

- Managed relationship between old and new content
  - Keeps track of NIDs, TIDs, UIDs, etc. between old and new systems
  - YES, you can preserve NIDs and UIDs!!!!! OMG Yesss
- Runtime features :
  - Rollbacks
  - Incremental
  - Offsets / start points*
  - Memory management
  - Timeout prevention

Possibilities

- Automate like crazy using Drush
  - Grouped migrations
  - Ordered migrations
- Deploy fresh content on a schedule
- Build around real data, demonstrate real sites
- Delete the migrate modules when you are done (not dependant)

How Do We Do This?

- Get the migrate module
- If upgrading Drupal, get migrate_d2d also
  - If you want UI, use 2.6x branch of migrate.
  - The Commerce Übercart Migration now supports 2.6 (patched ;)
- Create a new custom module
- Put your OOP hat on, each migration will be a « class »
- Copy source DB and files to the same host as D7

Running migrations

- Use drush, apache/nginx gets in the way, can lose connection
  - Won't stop running? Lost connection. Use Reset!
  - UI is ok too but timeouts common. Reset. Import.
- Migrate manages memory and long running processes automatically
- Show what is running with devel's dpm() and/or drush_print_r();

Your Custom Module

- kafei_migrate.info
  - List out all of your files[] here
- kafei_migrate.migrate.inc
  - Register your groups and classes here
- kafei_migrate.module
  - Helper functions (no migrate logic)
- user.inc, node.inc
  - Mapping of old fields to new fields
  - Each source+destination pairing will probably get a separate class
- file.inc, menu_links.inc
  - More mappings, if needing customization (d2d mostly automatic)
  - Different handlers for public/private

Variable Path – Registration to Row

- kafei.migrate.inc :
  - $arguments array defined for each migration
  - hook_migrate_api() loads $arguments for each migration when « re-register statically-defined classes » is run
  - hook_migrate_api_alter() is possible to integrate/override migrations
- user.inc :
  - file is loaded when migration is run
  - the __construct function passes the $arguments into $this
  - prepareRow and prepare load a single piece of content into the variable $row
    - when you are working on $row you always have access to $this
    - only one $row is processed at one time to conserve memory

Registration – What Version?

- Can be defined in 3 different places (only two supported, one recommended) :
  - Best way is to put everything into kafei_migrate.migration.inc (2.5, 2.6)
    - hook_migrate_api()
    - Find content > migrate > config > register
    - call register function in hook_install?
  - Second way is to instantiate a new instance of a class (2.5)
    - Find content > migrate > config > register
    - Clear all caches (need custom hook for this from d2d docs)
  - Third way is depreciated and should be avoided at all costs. Involves $migrations[] array from hell. Only works in 2.4 and 2.5.
    - « auto-registration » *old code! refactor!*

Class Registration – hook_migrate_api

```
function kafei_migrate_api() {
  $common_arguments = array(
    'source_connection' => 'd6', // name of secondary DB in settings.php
    'source_version'=> 6,
    'group_name' => 'kafei',
    'paths' => $paths, // CUSTOM (only gets run once)
    'format_mappings' => array(
      '1' => 'filtered_html', // site manager
    ),
  );
  $user_arguments = $common_arguments + array(
    'machine_name' => 'KafeiUser',
    'description' => t('Import Drupal 6 users'),
    'class_name' => 'KafeiUser',
    //'role_migration' => 'KafeiRole',
  );
```

Class Registration - hook_migrate_api

```
  /**
   * Class registration!
   */
  $api = array(
    'api' => 2,
    'groups' => array(
      'kafei' => array(
        'title' => t('Kafei Migrations'),
      ),
    ),
    'migrations' => array(
      'KafeiUser' => $user_arguments,
      'KafeiFile' => $public_file_arguments,
      'KafeiMenu' => $menu_arguments,
      'KafeiMenuLinks' => $menu_link_arguments,
    ),
  );
} // function kafei_migrate_api
```

Class Registration Notes

- Groups may need to be instantiated first, sometimes 2x might be needed :
  - Migrate > Configuration > Re-register statically-defined classes
- Extending migrate_d2d may require many $arguments, read docs on drupal.org for migrate_d2d
- Rollback, then « Remove migration settings » to delete previous mapping, then re-register
- Safe to truncate migrate_status and re-register.

Custom Classes - user.inc file

```
class KafeiUser extends DrupalUser6Migration {
  public function __construct(array $arguments) {
    parent::__construct($arguments); // make use of extending that class!
  }
  public function prepareRow($row) {
    // skip this $row?
    if ($row->thing = 'ugly') {
      return FALSE;
    }
  }
  public function prepare($entity, $row) {
  }
}
```

Class Construct : Mapping

```
$this->map = new MigrateSQLMap($this->machineName,
    array(
      'order_id' => array(
        'type' => 'varchar',      // or use 'type' => 'int',
        'length' => 255,          // for int, use 'unsigned' => TRUE,
        'not null' => TRUE,
        'description' => "Order ID",
      ),
    ),   MigrateDestinationEntityAPI::getKeySchema('commerce_order',
'commerce_order')
  );
```

Consruct : Sources & Destinations

- If you use migrate_d2d your source and destination are configured in the registration array, so you can skip this section
- You can find all the supported sources and destinations on DrupalContrib
- File-based imports (XML, CSV) allow *multiple* input files so you can batch things
- MigrateDestinationTable($table) is awesome
- Count query is a source parameter

Construct : Source Query (DBTNG style)

```
$query = db_select('migrate_example_beer_topic', 'met')
        ->fields('met', array('style', 'details', 'style_parent', 'region', 'hoppiness'))
        ->orderBy('style_parent', 'ASC');
$this->source = new MigrateSourceSQL($query);
```

Construct : Source Query Joins (DBTNG)

```
$query->leftJoin('content_field_location', 'l', 'n.nid = l.nid');
$query->fields('l', array('field_location_value'));

// in the next section we will do this with the field to establish the mapping:
$this->addFieldMapping('field_location', 'field_location_value')->defaultValue(0);
$this->addFieldMapping('field_location:language')
  ->defaultvalue('UND');
```

Construct : Source Xpath parser – XML binding

```
// Set up our destination - terms in the migrate_example_beer_styles vocabulary
$this->destination = new MigrateDestinationTerm('migrate_example_beer_styles');

$xml_folder = '/usr/share/beer/';
$item_url = $xml_folder . 'beers.xml';
$item_xpath = '/Beers/Beer'; // relative to doc
$item_id_xpath = 'BEER_ID'; // relative to item_xpath
$this->source = new MigrateSourceList(new MigrateListXML($list_url),
  new MigrateItemXML($item_url), $fields);
$this->source =  new MigrateSourceXML($item_url, $item_xpath, $item_id_xpath, $fields);
$this->destination = new MigrateDestinationEntityAPI('beer_entity', 'beer_entity');
```

Construct : Field Mapping

- You need to define an association between source field and destinaton field using addFieldMapping or addSimpleMappings
- You will also need an array of $fields (names) for your query (can be same if using addSimpleMappings).
- If you set the mapping with addFieldMapping you can set :
  - $this->addFieldMapping('thing', 'thing')->defaultValue(TRUE);
- Set sourceMigration to establish links between different migration classes (nid, uid references)
  - $this->addFieldMapping('uid', 'uid')->sourceMigration('KafeiUser'); // can also be an array of multiple
- Visit migration page in UI to see if you are done :
  - Review these pages with your clients. Red is bad!
  - Once you are sure you do not need a source/dest field, you can set it to DNM (do not map)

Construct :Field Mapping

```
$this->addFieldMapping($DESTINATION, $SOURCE); // source optional!
$this->addFieldMapping(NULL, $SOURCE); // source temporary for processing
$this->addFieldMapping($DESTINATION, 'variable_defined_later'); // we will define later in prepareRow function

// LDAP mappings (note : not queried yet)
$this->addFieldMapping('ldap_user_puid_sid', 'ldap_sid');
$this->addFieldMapping('ldap_user_puid_sid:language')
        ->defaultvalue('UND'); // subfield

$this->addFieldMapping('order_number')
        ->xpath('ORDER_ID'); // XML source format
```

Construct : Field Mapping – Preserve nid, uid

```
// Keep old UIDs. Same syntax for node NIDs.

// Do not do this since we do not login as user 1!

$this->addFieldMapping('uid', 'uid');

$this->removeFieldMapping('is_new');  // migrate_d2d has value pre-set, so we unset

$this->addFieldMapping('is_new')->defaultvalue(TRUE); // set to false to overwrite
```

Time to process the $rows!

- We can do any query we want in the later stages of processing.
- The « classic » style (non-DBTNG) queries are ok here, you can use db_set_active() to connect to other database.
  - db_set_active('d6'); // d6 is name of second DB in settings.php
  - Remember to switch back to D7 database db_set_active();
- You now have a construct that will create your migration class, load each row, and store it in Drupal!
- Now we can do more... fix the bad data.

Modifying Data

- prepareRow function is a preprocessor function that allows you to modify the content that is being migrated
- prepareRow($row) function runs THEN prepare($entity, $row) function will run
- prepareRow will not work for files
- prepareRow allows you to reject specific nodes/users/etc. (return FALSE);

function PrepareRow($row)

- Uses field handlers so the arrays are not expanded
- $row->body should have real data (not an array)
- Try just throwing data at it and see if it works!
  - $row->thing = thing;
- Look at the row here and reject it if you want
  - if ($row->title == 'test') {
  - return FALSE;
  - } // goodbye test node!
- Provide the source argument to one of your previous addFieldMapping sources :
  - $row->variable_defined_later = 'combine' . 'some' . 'things';
- Get stuff ready for the next stage
  - $row->tmp_junk = 'random things';

function prepare($entity, $row)

- $entity is fully expanded Drupal object that you know and love
  - Deal with all your unsupported fields here
  - Date field not working? Who cares?! I can deal with an expanded Drupal array!
- Do all your XML or CSV changes here

function complete

- Usually unnecessary
- Dirty post-processing work
  - calculating commerce order totals from the migrated items in the order
- Affecting unrelated things in the system, live dangerously!
  - Commerce needs this to calculate order* total after items*.
  - *orders and items are separate classes

function rollback

- Usually unnecessary
- Runs globally by default, not per-row

migrate_extras module

- This module has some field handlers that you might want to make use of
- Just install it and things magically work (allowing you to work with things in prepareRow rather than prepare)
- This module will eventually go away

FAQ

- Webforms : https://drupal.org/project/migrate_webform
- Ubercart → Commerce :
- https://drupal.org/node/1832736#comment-8001541
- Node revisions are possible at https://drupal.org/node/1298724
- Dealing with files (use migrate_d2d + docs) :
  - migrate : https://drupal.org/node/1540106
  - migrate_d2d : https://drupal.org/node/1819704

Cleaning HTML - simplehtmldom

- If you need to edit HTML on the fly, do it in a jQuery-like object-oriented way after installing simplehtmldom module :

```
// Load and parse string into objects
$html = str_get_html($text);
// Apply new path pattern to any links
foreach($html->find('a') as $element) {
  // remap private file paths
  $element->href = str_replace('system/private_files', 'system/files', $element->href);
}

// Send the updated HTML text back
return $html->outertext;
```

Homework

- *Beer*. Shows you the basics of writing a migration. (migrate_example module)
- *Wine*. More advanced demo using multiple XML files with Xpath parsing (migrate_example module).
- Migrate_d2d has an example module – this one is GOOD (for reading)
- The examples module (always a good resource).

Migrate resources

- The migrate example module has Beer
- When you are done with Beer, then try Wine
- Documentation on drupal.orghttps://drupal.org/node/415260
  - Print version grabs sub-pages ;)
- D2D documentation https://drupal.org/node/1813498
- DrupalContrib summaries (find source/dest connectors) : http://drupalcontrib.org/api/search/7/migrate
- Migrate_extras adds field support in some cases https://drupal.org/project/migrate_extras