



Upgrading / Migrating to Drupal 7

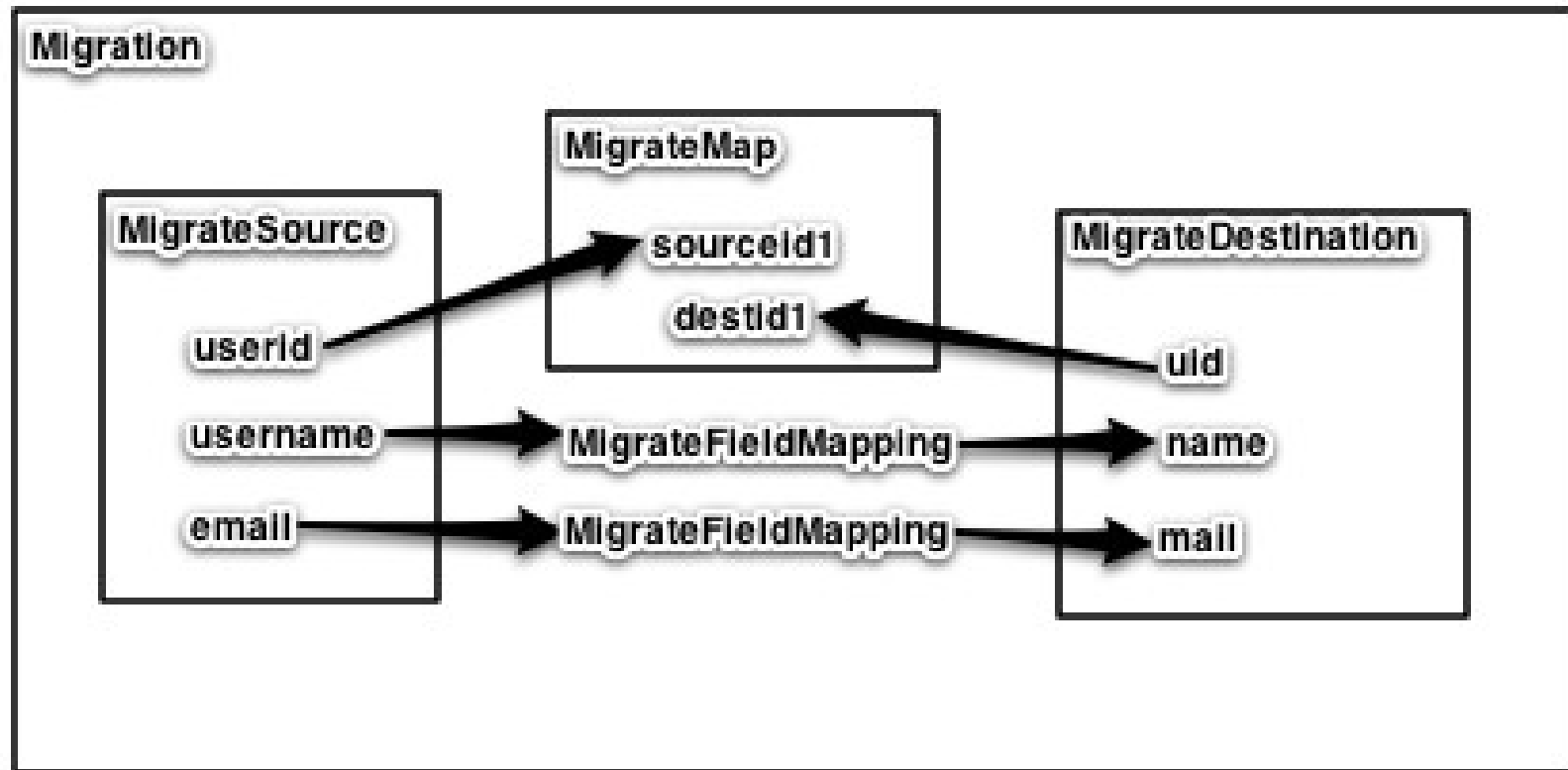
02.18.2013



I'm on D6. Let's do this the easy way:

drush sup

How Migrate works:





Ok, that didn't work. Now what?

- Migrate
<http://drupal.org/project/migrate>
- Migrate Extras
http://drupal.org/project/migrate_extras

Migrate Extras - Gotcha

	Drupal 6	Drupal 7
Address Field	N/A	✓
Automatic Nodetitles		Requested
CiviCRM		Requested
Commerce	N/A	Built-in
Countries	N/A	Built-in
Date	Built-in	Built-in
Content Taxonomy	✓	N/A
Domain Access		Requested
Email Field	Requested	Built-in
Embedded Media Field	✓	Requested
Entity API	N/A	✓
Entity Reference	N/A	Built-in
Field collection	N/A	Requested
Fivestar		Requested
Flag	✓	✓
Formatted Number CCK	✓	N/A
Interval	N/A	✓

Part of the Migrate Extras

✓	Migrate Extras	7.x-2.5	Adds migrate module Requires: Migrate (disabled) Required by: Migrate (disabled)
---	-----------------------	---------	--

Part of that Module

✓	Date Migration	7.x-2.6	Provides support for i Requires: Migrate (disabled) Required by: Date Migration
---	-----------------------	---------	---



How to use and write your first Migrate Module!

- Download / Enable Migrate, Migrate Extras, and module specific Migrate Extensions (Date)
 - migrate
 - migrate_extras
 - date_migrate
 - migrate_ui
- Create / Enable your migration module
- Enable your content migration types in the Migrate UI
- Migrate it!



Creating our first migrate module:

Pre-Reqs

- Migrate uses PHP Classes heavily:

```
class cats {  
    protected $type = "";  
    protected $adult = TRUE;  
    public function __construct($type) {  
        $this->type = $type;  
    }  
}
```

```
class kittens extends cats {  
    protected $adult = FALSE;  
}
```



Creating our first migrate module:

Pre-Reqs

- Migrate uses DBTNG:

```
$query = db_select('node', 'n')  
  ->fields('n', array('nid', 'uid', 'title', 'body'))  
  ->condition('n.created', '1361224788', '>')  
  ->orderBy('n.nid', 'DESC');
```




Creating our first migrate module: migrate_cats.info

```
name = "Migrate Cats"  
description = "Example Cat Migration."  
package = "Migration"  
core = 7.x
```

```
dependencies[] = taxonomy  
dependencies[] = image  
dependencies[] = comment  
dependencies[] = migrate  
dependencies[] = list  
dependencies[] = number
```

```
files[] = migrate_cats.module  
files[] = cats.inc
```



Creating our first migrate module: migrate_cats.module

```
<?php
```

```
/**
```

```
 * @file
```

```
 * THIS SPACE INTENTIONALLY LEFT BLANK.
```

```
 *
```

```
 * Yes, there is no code in the .module file. Migrate
```

```
 * operates almost entirely through classes.
```

```
 */
```



Creating our first migrate module: migrate_cats.migrate.inc

```
<?php
```

```
function migrate_cats_migrate_api() {  
    $api = array(  
        'api' => 2,  
        'migrations' => array(  
            'CatNode' => array('class_name' => 'catNodeMigration')  
        ),  
    );  
    return $api;  
}
```



Creating our first migrate module: cats.inc

```
<?php
class catNodeMigration extends Migration {
    /* Set up migration / mapping */
    public function __construct() {
        parent::__construct();
        $this->description = t('Migrate them cats.');
```

\$this->map – MigrateMap Class instance

\$this->source – MigrateSource Class instance

\$this->destination – MigrateDestination Class instance

\$this->addFieldMapping – Field Mappings

```
    }
    /* Finesse data / fix it before saving. */
    public function prepareRow($current_row) {
        $current_row->{property}
        return TRUE;
    }
}
```



Creating our first migrate module: Map the information

```
<?php
```

```
class catNodeMigration extends Migration {  
    /* Set up migration / mapping */  
    public function __construct() {  
        parent::__construct();  
        $this->map = new MigrateSQLMap($this->machineName,  
            array(  
                'fid' => array(  
                    'type' => 'int',  
                    'not null' => TRUE,  
                    'description' => 'Feline ID.',  
                    'alias' => 'c',  
                )  
            ),  
            MigrateDestinationNode::getKeySchema()  
        );  
    }  
}
```



Creating our first migrate module: Setting up the source (MySQL).

```
<?php
```

```
class catNodeMigration extends Migration {  
    /* Set up migration / mapping */  
  
    public function __construct() {  
        parent::__construct();  
        $query = db_select('cats', 'c')  
                ->fields('c', array('name', 'adult', 'age', 'fur'));  
        // Create a MigrateSource object, which manages retrieving the input  
data:  
        $this->source = new MigrateSourceSQL($query);
```

- MigrateSourceCSV
- MigrateSourceList
- MigrateSourceMSSQL
- MigrateSourceMultiltems

- MigrateSourceOracle
- MigrateSourceSQL
- MigrateSourceXML



Creating our first migrate module: XML Source.

```
<?xml version="1.0" encoding="UTF-8"?>
<cats>
  <cat>
    <felineid>0001</felineid>
    <name>Felix</name>
    <age>2</age>
  </cat>
  <cat>
    <felineid>0002</felineid>
    <name>Grumpy</name>
    <age>15</age>
  </cat>
</cats>
```



Creating our first migrate module: Setting up the source (XML).

```
<?php
```

```
class catNodeMigration extends Migration {  
  /* Set up migration / mapping */  
  
  public function __construct() {  
    parent::__construct();  
    $fields = array(  
      'name' => t('Cat name'),  
      'age'   => t('Actual Age')  
    );  
    $xml_folder = drupal_get_path('module', 'migrate_cats') . '/xml/';  
    $cats_url = $xml_folder . 'cats.xml';  
    $cat_xpath = '/cats/cat';  
    $cat_ID_xpath = 'felineid';  
    $this->source = new MigrateSourceXML($cats_url, $cat_xpath,  
    $cat_ID_xpath, $fields);  
  }  
}
```




Creating our first migrate module: Setting up the destination (Node).

```
<?php
```

```
class catNodeMigration extends Migration {  
    /* Set up migration / mapping */  
  
    public function __construct() {  
        parent::__construct();  
        // Set up our destination - nodes of type cats  
  
        $this->destination = new MigrateDestinationNode('cats');
```

- MigrateDestinationComment
- MigrateDestinationFile
- MigrateDestinationNode
- MigrateDestinationTerm
- MigrateDestinationUser

- MigrateDestinationProfile2
- MigrateDestinationPrivateMsg
- MigratePathautoHandler



Creating our first migrate module: Setting up the Field Mapping.

```
<?php
```

```
class catNodeMigration extends Migration {  
    /* Set up migration / mapping */  
  
    public function __construct() {  
        parent::__construct();  
        $this->addFieldMapping('title', 'name')  
            ->description(t('Mapping cat name in source to node title'));  
  
        // Unmapped destination fields  
        $this->addUnmigratedDestinations(array('age'));
```



Creating our first migrate module: Finesse the data.

```
<?php
```

```
class catNodeMigration extends Migration {  
    /* Finesse data / fix it before saving. */  
    public function prepareRow($current_row) {  
        $current_row->created = time();  
        // Make sure those pesky yard diggers don't get in!  
        if($current_row->name == 'Dog') {  
            return FALSE;  
        }  
        return TRUE;  
    }  
}
```

Running our first migrate module:

- admin/content/migrate

<input checked="" type="checkbox"/>	Idle	BeerTerm	3	0	3	0	Unknown
-------------------------------------	------	----------	---	---	---	---	---------

OPERATIONS

Import

Execute

Choose an operation to run on all migrations selected above:

- Import – Imports all previously unimported records from the source, plus any records marked for update, into destination Drupal objects.
- Rollback – Deletes all Drupal objects created by the migration.
- Rollback and import – Performs the Rollback operation, immediately followed by the Import operation.
- Stop – Cleanly interrupts any import or rollback processes that may currently be running.
- Reset – Sometimes a migration process may fail to stop cleanly, and be left stuck in an Importing or Rolling Back status. Choose Reset to clear the status and permit other operations to proceed.

► **OPTIONS**



Migrate Classes

- MigrateSource
 - <http://drupalcontrib.org/api/drupal/contributions%21migrate%21includes%21sour>
- MigrateDestinationEntity
 - <http://drupalcontrib.org/api/drupal/contributions%21migrate%21plugins%21destir>



Migrate Resources

- Migrate Examples:
 - migrate_example
 - migrate_example_baseball
 - migrate_example_oracle
- XDEB Migration (dated - 2011):
 - <http://xdeb.org/node/1539>
- RedCat Migration (dated - 2011):
 - <http://btmash.com/article/2011-02-25/migrating-content-using-migrate-module>