

# **BUILDING, DEPLOYING AND MAINTAINING DRUPAL SITES LIKE A PRO**

Bob Kepford

[Mediacurrent.com](http://Mediacurrent.com)

[TheWeeklyDrop.com](http://TheWeeklyDrop.com)

# DEV -> STAGE -> PRODUCTION

A sane workflow.

# HOW?

Drush + Features + hook\_update\_N + Git + SSH

**HOLD ON.**

**WHY?**

# DRUPAL ROCKS.

Drupal Rocks.

**DRUPAL SUCKS.**

**DRUPAL + DEV WORKFLOW SUCKS LESS**

# DRUPAL 8 ROCKS!

Configuration Management

# WHAT WE WILL COVER

## PART 1

- Storing Drupal's configuration\* in code using the Features module.
- Using Drush to manage these features.

## PART 2

- Advanced Drush awesomeness
- Using Drush + SSH([keys](#)) to sync your databases.
- Using Git([Git Flow Workflow](#))to manage your code base.

# REQUIRED DRUPAL MODULES

- [Drush](#)
- [Features 2.0](#)
- [Strongarm](#)
- [DefaultConfig](#)
- [Diff](#)
- [Devel](#)

# OTHER REQUIREMENTS

- [Git](#) installed locally and on server
- How to use the commandline(ninja not required).
- SSH keys for Git and Drush interaction with servers.
- Git server or service like [Bitbucket](#) or [Github](#).

# FEATURES MODULE

The features module enables the capture and management of features in Drupal. A feature is a collection of Drupal entities which taken together satisfy a certain use-case.

# DRUSH

Drush is a command-line shell and scripting interface for Drupal, a veritable Swiss Army knife designed to make life easier for those who spend their working hours hacking away at the command prompt.

[drush.org](https://drush.org)

# GIT

Distributed version control system

<http://git-scm.com/>

# SSH

"Secure Shell (SSH) is a cryptographic network protocol for secure data communication."

[http://en.wikipedia.org/wiki/Secure\\_Shell](http://en.wikipedia.org/wiki/Secure_Shell)

# PACKAGING CONFIGURATION INTO MODULES

Or, clicky click to code.

# STEPS

- Build functionality in the UI
- Generate and enable new feature module
- Add/commit to Git repository

```
$ git add myfeature && git commit -m
```

- Edit functionality (UI)
- Check the status of your feature modules

```
$ drush fl
```

# STEPS

- Check your feature change with Drush

```
$ drush fd myfeature
```

- Update the feature module code to match the database change

```
$ drush fu featurename --y
```

- Check features module status again

```
$ drush fl
```

# DEMO

Building a "feature"

<http://fdug.dev>

# DRUSH COMMANDS

- `fl --` Lists all the feature modules and their status
- `fd my_module --` Inline diff between the code and DB
- `fu my_module --` Update the code with the changes from the DB
- `fr my_module --` Revert configuration in the DB to reflect the module's code.
- `fe my_module --` Export your feature from the commandline(I prefer the UI here.)

# MORE DEMO

Updating a feaure

<http://fdug.dev>

# CLEAN FEATURES

Preventing conflict Hell

# HOW????

- Each Content Type is a Feature
- Each View is a Feature
- Each Context is a Feature
- Yes I'm Serious
- A Common or Site Feature

# NO OVERRIDDEN FEATURES

Overridden features are dangerous. Review and commit the changes to code.

# FEATURES NAMING CONVENTION

- Content Type: ct\_mycontent\_type
- View: view\_my\_view
- Taxonomy: t\_my\_taxonomy\_vocab
- Context: c\_my\_context

# OTHER ADVICE

- Always change the feature package to a name appropriate for the project, do not leave the default “Features”.
- Use version numbers.
- Do not put dependencies on devel or UI modules into features.
- Features will not delete a field from the database.

# HANDLING QUIRKS WITH FEATURES

- `fr --force`
- Use a clean drupal install to debug

# HANDLING NON-EXPORTABLES

Nodes, Taxonomy terms, Commerce Products and [insert thing here].

```
hook_update_N( )
```

<http://befused.com/drupal/site-deployment-module>

# HANDLING NON-EXPORTABLES

```
/**  
 * Removes field_field_name.  
 */  
function FEATURE_NAME_update_7101() {  
    field_delete_field('FIELD_MACHINE_NAME');  
}
```

# HANDLING NON-EXPORTABLES

```
/**
 * Implements hook_update_N().
 * Add category terms
 */
function mymodule_update_7000() {
  $vocabulary_machine_name = 'category';
  $vocabulary = taxonomy_vocabulary_machine_name_load($vocabulary_machine_name);
  $names = array(
    'Awesome Term',
    'Drupal FTW',
  );

  $weight = 0;
  foreach($names as $name) {
    $term = (object) array(
      'name' => $name,
      'vid' => $vocabulary->vid,
      'weight' => $weight
    );
    taxonomy_insert_term($vocabulary->vid, $term->name, $term->weight);
    $weight++;
  }
}
```

# READING HOMEWORK

How to Write Drupal Update Scripts

Programmatically Revert Features

Programmatically remove a field from a node