

Drush Make driven development

Presented by Steven Jones

Introduction

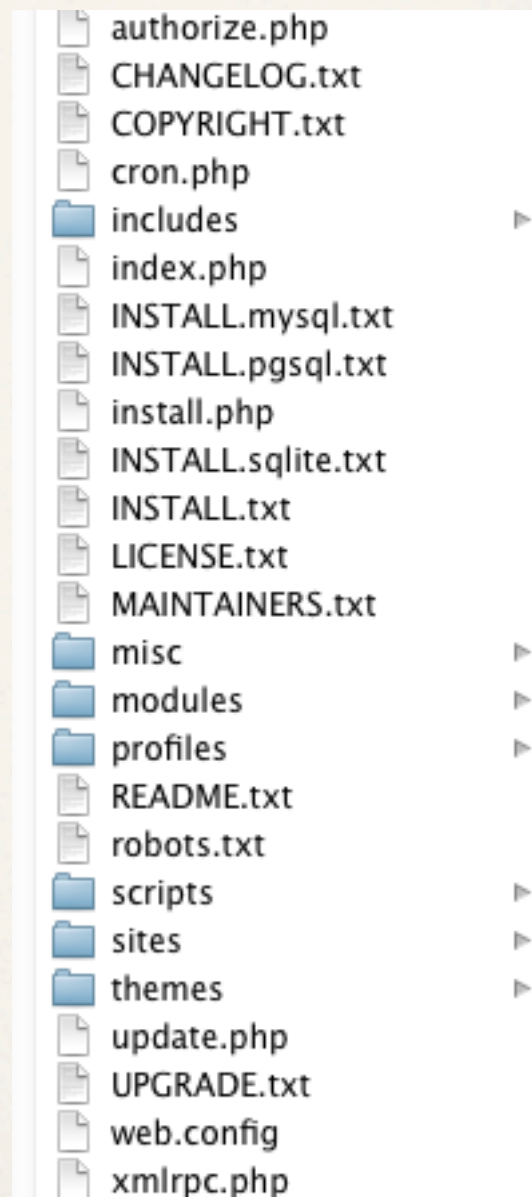
- About me
- Traditional approach
- Advantages/disadvantages
- Drush make approach
- Development & deployment
- Questions

About me

- Computerminds
- Drupal developer for 6+ years
- Lead developer
- twitter.com/darthsteven

How do most people develop Drupal sites?

- Version control system (hopefully)
- Contains entire Drupal codebase:
- Mixture of:
 - Other people's code
 - Their custom code



Whats good about that?

- Easy to download the codebase
- Single place to get the codebase from
- Easy to hack a fix into third party code
- Everything is versioned

What's bad about that?

- Patching third party code is hard
- Core/module upgrades are painful
- How do you know what you have?

Patching code is hard

- Apply the patch
- Commit the code
- Keep a record?

Module upgrades are faffy

- Download the new module
- Replace the old code
- Re-apply patches?

Core upgrades

- Like upgrading modules
- But way more painful
- Delete and copy?
- Merge in changes?
- Patches?

Got a new client with an existing site?

- What's in the repo?
- Is that *actually* Drupal 7.20?
- What patches were applied?
- What's been secretly hacked?

Drush Make!

- A manifest: a .make file
- Specify the names and versions of projects
- Specify patches
- Recursive

A make file (the manifest)

```
projects[field_display_label][subdir] = "contrib"  
projects[field_display_label][version] = "1.2"  
  
projects[field_formatter_class][subdir] = "contrib"  
projects[field_formatter_class][version] = "1.0"  
  
projects[field_formatter_settings][subdir] = "contrib"  
projects[field_formatter_settings][version] = "1.0"  
  
projects[field_group][subdir] = "contrib"  
projects[field_group][version] = "1.1"  
  
projects[field_multiple_limit][subdir] = "contrib"  
projects[field_multiple_limit][version] = "1.0-alpha3"  
  
projects[field_permissions][subdir] = "contrib"  
projects[field_permissions][version] = "1.0-beta2"  
  
projects[field_slideshow][subdir] = "contrib"  
projects[field_slideshow][version] = "1.7"
```


How does this work for developers?

- 'Stub' make file
 - Contains the Drupal core version
 - Contains the private git repo to download
- Private git repo contains
 - An install profile - with makefile
 - Custom code!

Stub make file?

```
core = 7.x
api = 2

projects[drupal][type] = core
projects[drupal][version] = "7.20"

projects[secret_project][type] = profile
projects[secret_project][download][type] = git
projects[secret_project][download][url] = git@github.com:computerminds/secret_project.git
projects[secret_project][download][branch] = "develop"
```


Install profile?

- Actual install profile
- Make file
- Custom code
- .gitignore
- Stub make file

Example Site Repo

- hawk_stub-master.make
- hawk.info
- hawk.make
- hawk.profile
- modules
- README.md

- custom

- hawk_twitter

- hawk_twitter.info
- hawk_twitter.module

Drush make

```
steven at Steven-Joness-MacBook-Pro in ~/Sites  
$ drush make 'https://raw.githubusercontent.com/computerminds/hawk/master/hawk_stub-master.make' hawk-platform  
--working-copy
```

Drush make

```
Checked out branch master. [ok]
Found makefile: hawk.make [ok]
>> Project bean contains 5 modules: bean_uuid, bean_usage, bean_all, bean_admin_ui, bean.
bean-7.x-1.0-rc6 downloaded. [ok]
bean patched with bean-1883232-entity-metadata-wrapper.patch. [ok]
bean patched with bean-1885958-setter-callbacks.patch. [ok]
Generated PATCHES.txt file for bean [ok]
>> Project views contains 2 modules: views, views_ui.
views-7.x-3.5 downloaded. [ok]
>> Project ctools contains 9 modules: views_content, stylizer, page_manager, ctools_plugin_example
, ctools_custom_content, ctools_ajax_sample, ctools_access_ruleset, bulk_export, ctools.
ctools-7.x-1.2 downloaded. [ok]
entity cloned from git://git.drupal.org/project/entity.git. [ok]
Checked out revision e7b054ffe9eca11641204ef0c69c603ff6a03394. [ok]
entity patched with entity-1815634-01.patch. [ok]
entity patched with 1788764-writable_text_format.patch. [ok]
Generated PATCHES.txt file for entity [ok]

steven at Steven-Joness-MacBook-Pro in ~/Sites
$ █
```


After 'making'

- hawk_stub-master.make
- hawk.info
- hawk.make
- hawk.profile
- modules
- README.md

- contrib
- custom

- bean
- ctools
- entity
- views

- hawk_twitter

- hawk_twitter.info
- hawk_twitter.module

hawk.make

```
core = 7.x
api = 2

projects[ctools][subdir] = "contrib"
projects[ctools][version] = "1.2"

projects[entity][subdir] = "contrib"
projects[entity][type] = "module"
projects[entity][download][type] = "git"
projects[entity][download][url] = "git://git.drupal.org/project/entity.git"
projects[entity][download][revision] = "e7b054ffe9eca11641204ef0c69c603ff6a03394"
projects[entity][patch][] = "http://drupal.org/files/entity-1815634-01.patch"
; Allow writing to text format.
projects[entity][patch][] = "http://drupal.org/files/1788764-writable_text_format.patch"

projects[bean][subdir] = "contrib"
projects[bean][version] = "1.0-rc6"
projects[bean][patch][] = "http://drupal.org/files/bean-1883232-entity-metadata-wrapper.patch"
projects[bean][patch][] = "http://drupal.org/files/bean-1885958-setter-callbacks.patch"

projects[views][subdir] = "contrib"
projects[views][version] = "3.5"
```


Really easy things

- New version of bean:

```
steven at Steven-Joness-MacBook-Pro in ~/Sites/hawk-platform/profiles/hawk (master)  
$ vim hawk.make
```

More easy things

- Applying a patch
- Updating Drupal core
- Adding your <insert feature here> modules
- Applying the same patches as another project
- Seeing what you have

How does this work for deployments?

- Build from the 'stub' make file
- Move the site specific directory
- Legacy tool?
- Drush make a 'full' codebase and add to another branch

Useful projects

- Drush make
 - Included in Drush
 - drupal.org/project/drush
- Drush remake
 - drupal.org/project/drush_remake
- Hawk install profile
 - github.com/computerminds/hawk

Questions

- Thank you for listening
- Any questions?