Drush Make Driven Development

DrupalCampNJ, Feb 2014
Intro

• Leonid Makarov
  (leonid.makarov@blinkreaction.com)
  – Sr. Solutions Architect
  – BlinkReaction is where I have been doing Drupal for the past 7 years
HOW DO WE START BUILDING A NEW DRUPAL SITE?
Starting up from scratch

1. Open project page
2. Download required version
3. Unpack into docroot
Maybe use Drush?

drush dl <project name>
Starting up from a distribution

Download distribution → Done ?
Not really...

1. Open project page
2. Download required version
3. Unpack into docroot
Even with “drush dl”...
Custom distribution?

• How easy is that?
DRUPAL DISTRIBUTIONS
What are distributions?

• Distributions are full copies of Drupal that include:
  – Drupal Core
  – themes, modules, libraries
  – installation profiles
What are installation profiles?

• They define installation steps (such as enabling modules, defining content types, etc.) that run after Drupal's base installation when you first install Drupal.
• Drupal core comes bundled with 3 installation profiles (standard, minimal, testing)
• May require non-core dependencies (modules, themes, or libraries)
Distributions vs installation profiles

**Installation profile**
- themes
- modules
  - README.md
  - build-starter.make
  - drupal-org-core.make
  - starter.profile
  - starter.install
  - drupal-org.make
  - starter.info

**Distribution**
- includes
- misc
- modules
  - profiles
    - minimal
    - standard
    - starter
      - libraries
      - modules
      - themes
        - drupal-org-core.make
        - starter.install
        - build-starter.make
        - starter.info
        - README.md
        - starter.profile
        - drupal-org.make
- testing
- scripts
- sites
  - themes
    - PATCHES.txt
    - xmlrpc.php
    - install.php
How does an installation profile become a distribution?
Build and packaging scripts

(test)

build

package

(powered by Drush Make)
DRUSH MAKE
What is Drush Make?

• An extension to Drush included as of Drush version 5.
• A tool for assembling all the dependencies needed for an installation profile to work as a full Drupal website.
• The central tool used for packaging distributions on Drupal.org.
What can Drush Make do?

• Can put together a complete Drupal docroot by pulling sources from various locations.
• It does this by parsing a flat text file (similar to a drupal `.info` file) and downloading the sources the file describes.
• It is possible to distribute a complicated Drupal distribution as a single text file.
Drush Make features

• Downloading Drupal core, as well as contrib project from drupal.org.
• Checking code out from SVN, Git, and Bzr repositories.
• Getting plain .tar.gz and .zip files (particularly useful for libraries that can not be distributed directly with Drupal core or modules).
• Fetching and applying patches.
• Fetching modules, themes, and installation profiles, but also external libraries.
MAKE FILES
.make files

- distro make file
- core make file
- contrib make file
drupal-org-core.make

• Used to define core version and any patches to Drupal core.
• Used to point to a -dev release, or a specific Git revision of the core.
• This file is optional.
drupal-org-core.make

api = 2
core = 7.x

; Drupal Core
projects[drupal][version] = 7.23

; Patches for Core
drupal-org.make

- Defines all of the contributed modules, themes, and 3rd party libraries required by the installation profile.
api = 2
core = 7.x

; Modules
projects[libraries][version] = 2.1
projects[wysiwyg][version] = ""

; Themes
projects[shiny][version] = ""

; Libraries
libraries[ckeditor][download][type] = get
libraries[ckeditor][download][url] = http://download.cksource.com/CKEditor/CKEditor/CKEditor%204.2/ckeditor_4.2_standard.zip
libraries[ckeditor][destination] = libraries
build-[distro].make

• This one is a little tricky...
build-[distro].make

• This file is used to build a fully functional docroot that includes:
  – core
  – the installation profile code itself
  – all of the contrib dependencies

• It utilizes the recursive nature of Drush Make.
build-[distro].make

api = 2
core = 7.x

; Drupal Core
includes[] = https://raw.github.com/blinkreaction/drupal-starter-distro/master/drupal-org-core.make

; Installation Profile
projects[starter][type] = profile
projects[starter][subdir] = ""
projects[starter][download][type] = git
projects[starter][download][url] = https://github.com/blinkreaction/drupal-starter-distro.git
projects[starter][download][branch] = master
Huh?

• Yeah, I know...
BUILD PROCESS
Build process

- Reference to `drupal-org-core.make` => Processing...
- Reference to installation profile (self) => Downloading...
  => Recursively building `drupal-org.make`...

- Reference to a particular core version => Downloading...
- References to core patches => Downloading and applying...

- References to contribs => Downloading...
- References to contrib patches => Downloading and applying...
Build results

Fully functional docroot:
- core
- installation profile
- contrib dependencies
DRUSH MAKE EXAMPLES
Drush Make examples

• A complete distro build
  drush make build-[distro].make ./docroot

• A core only build
  drush make drupal-org-core.make ./docroot

• A contribs only build
  drush make --no-core build-[distro].make ./docroot
Questions so far?
Drush Make driven development

DMDD
DMDD process

• Site is developed as an installation profile
  – any custom code goes into the profile.
• All external dependencies and patches are defined in the make files.
• Installation profile is the only thing stored in the project repo (no external dependencies are committed).
How to DMDD (tools)

• Drush (do-it-yourself)
  https://github.com/drush-ops/drush

• Kraftwagen
  http://kraftwagen.org/

• Drush DMDD extension (WIP)
DMDD pre-requirements

• A structured repository
  – https://github.com/blinkreaction/drupal-starter-repo

• Tools
  – Drush 6
  – Drush DMDD extension
Drush DMDD extension

• Provides Drush command wrappers to Drush Make.
• Can initialize a docroot (clean build).
• Can build and rebuild core and contribs.
• Makes certain assumptions about the repo structure (template repo).
DMDD workflow

• Clone the template repo
• Initialize docroot (drush dmdd init)
• Do something. E.g.:
  – add a new contrib in drupal-org.make
  – update core version in drupal-org-core.make
  – apply a patch
• Build (drush dmdd core|contrib|all).
• Watch Drush Make working.
DMDD extension in action

DEMO TIME
Who is DMDD for?

• Anyone!
Who is DMDD for?

• Individual developers:
  – Manage core and contrib dependencies in an organized fashion
  – Manage patches in a single place, versioned and documented
  – Have a quick high level overview of the dependencies and versions
  – Simplify the upgrade process
Who is DMDD for?

• Enterprise:
  – Have a single starting point for every site - the Platform.
  – Maintain a hierarchy of platforms and sub-platforms.
  – Perform platform updates in a managed fashion.
  – Do Continuous Integration.
     • (you have to anyway)
What’s next?

• DMDD adoption as a package/dependency management best practice for any Drupal project (similar to what npm does for nodejs or Composer does for PHP).
• Better tools integrated into Drush and Drush Make for making this happen.
• A possible alternative - switch Drupal contribs to Composer. (?)
Questions?
Thank you

See you at the After Party!