

# LEVEL UP YOUR DRUPAL 8 CONFIGURATION MANAGEMENT

**Scott Weston, Principal Architect**

Spring 2019

**bounteous**



**Scott Weston**

Principal Architect

**Hi, I'm Scott.**

**Principal Drupal Architect at Bounteous**

**Acquia Certified Grand Master**

**Working with Drupal since 2007**

Slides will be on the site after this presentation!

My pronouns: he, him, his

# WHO IS BOUNTEOUS

We create big-picture digital solutions that help leading companies deliver transformational brand experiences.

The logo for Wilson, featuring the word "Wilson" in a red, serif font.The logo for the Commonwealth of Massachusetts, featuring a blue shield with a white cross and a Native American figure, with the word "MASSACHUSETTS" in a blue, sans-serif font below it.The logo for Tesla, featuring the word "TESLA" in a black, sans-serif font with wide letter spacing.The logo for Ballad Health, featuring the word "Ballad" in a black, sans-serif font above the word "Health" in a smaller, black, sans-serif font, with a colorful, abstract graphic to the right.The logo for CHS Community Health Systems, featuring the letters "CHS" in a large, green, sans-serif font, with "Community Health Systems" in a smaller, black, sans-serif font to the right.The logo for Zipcar, featuring a green circle with a white "Z" inside, followed by the word "zipcar" in a black, sans-serif font.The logo for Advance Auto Parts, featuring the words "Advance" and "Auto Parts" in a red, sans-serif font, with a black and white checkered flag graphic to the right.The logo for Centura Health, featuring a blue cross icon to the left of the words "Centura Health" in a black, sans-serif font.The logo for Caesars Entertainment, featuring a laurel wreath to the left of the words "CAESARS ENTERTAINMENT" in a black, sans-serif font.The logo for PBS, featuring a black circle with a white silhouette of a human head and a brain, followed by the letters "PBS" in a black, sans-serif font.

## **THE PROBLEM**

How can you operate environments (Development, Stage, Production) with some different configuration; and maintain sanity?

# AGENDA

- 1. The Scenario**
- 2. The Tools**
- 3. The Config MISO**
- 4. The Solution**

# THE SCENARIO



# EXAMPLE ENVIRONMENTAL DIFFERENCES

	Development	Staging	Production
Devel, Webprofiler	ON	OFF	OFF
Caching	OFF	ON	ON
Errors	ALL	Fatal	None
Robots.txt	Disallow *	Disallow *	Authorable
Logging	Dblog	Dblog	Syslog
Env Indicator	Green	Yellow	Red
Content Blocks	Ignore	Ignore	Ignore



**Tip: Make a spreadsheet to think through/document your config management approach.**

# THE TOOLS



# CONTRIBUTED MODULES

## Required

- Configuration Ignore (config\_ignore)
- Configuration Split (config\_split)
- Configuration Filter (config\_filter) \*

\* Split & Ignore need the Config Filter module.

## Recommended



Tip: Chosen (chosen) – Makes select boxes on config forms easier to use. Needs Chosen jQuery library.

- Webform Config Ignore (webform\_config\_ignore) – Does not revert config for existing webforms.

# DETERMINE THE ENVIRONMENT

Acquia: `$_ENV['AH_SITE_ENVIRONMENT']` (dev, test, prod)

Pantheon: `$_ENV['PANTHEON_ENVIRONMENT']` (dev, test, live)

Other: Look for similar value in `$_ENV`, docs, support

# THE CONFIG MISO



**M**



**MANAGEMENT**

**I**



**IGNORE**

**S**



**SPLIT**

**O**



**OVERRIDES**



# THE CONFIG MISO

## CORE CONFIGURATION MANAGEMENT

If it's the same across all environments, let Core Config Management handle it.

```
drush config:export
```

```
drush config:import
```

# THE CONFIG MISO

## CONFIGURATION IGNORE

- Use when you don't want `drush config:import` to modify specific configurations.
- Can be very specific or use wildcards.
- Config will still be exported, but will not be imported.



**Tip: NEVER ignore `core.extension` or `config.ignore`.**

# THE CONFIG MISO

## CONFIGURATION IGNORE EXAMPLES

Block instances used as content components

`block.block.*`

Robotstxt module settings

`robotstxt.settings`

The body copy of the password recovery email

`user.mail.password_reset.body`

# THE CONFIG MISO

## CONFIGURATION SPLIT

Use when you have different configurations in one or more environment AND you want to manage via config.

Config Split stores the config yml files of the split in a different directory when running **drush config-split:export**.

A split can either be Active or Inactive.

If Active, the split configuration is import on **drush config:import**.

Leverage the `$_ENV` environment indicator to activate specific splits per environment.

# TWO KINDS OF SPLITS

## **Complete**

Full module's configuration is split into its own directory.

Example: Devel OFF or ON depending on environment.

## **Conditional**

Only certain settings of a module are split into its own directory.

Example: Error/Notice display settings. Full on Dev, hidden on Prod.

# THE CONFIG MIS

## CONFIGURATION OVERRIDES

Set configuration values in `settings.php`.

These configuration values will be used no matter what active configuration indicates.

# THE CONFIG MISO

## CONFIGURATION OVERRIDES EXAMPLES

Robotstxt value for Dev and Stage environments

```
$config['robotstxt.settings']['content'] = "User-agent: *\nDisallow: /\n";
```

API Keys or IDs which you never want modified:

```
$config['google_tag.settings.container_id'] = "GTM-ABC1234";
```

Config Split should be active:

```
$config['config_split.config_split.stg']['status'] = TRUE;
```

# THE SOLUTION



# EXAMPLE ENVIRONMENTAL DIFFERENCES

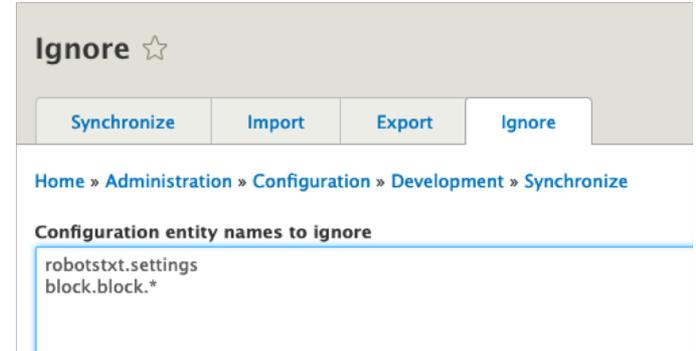
		Development	Staging	Production
<b>S</b>	Devel, Webprofiler	ON	OFF	OFF
<b>S</b>	Caching	OFF	ON	ON
<b>O</b>	Errors	ALL	Fatal	None
<b>O</b> <b>I</b>	Robots.txt	Disallow *	Disallow *	Authorable
<b>S</b>	Logging	Dblog	Dblog	Syslog
<b>O</b>	Env Indicator	Green	Yellow	Red
<b>I</b>	Content Blocks	Ignore	Ignore	Ignore

# STEPS

- 1. Configure ignores**
- 2. Add `$_ENV` detection and switch to `settings.php`**
- 3. Create split directories**
- 4. Create config split for environments**
- 5. Create overrides**

# CONFIGURE IGNORES

1. Enable Config Ignore
2. Navigate to /admin/config/development/configuration/ignore (a tab on the Config Synchronize pages)
3. Input config to be ignored.



**Tip: The config file name and array structure in the file help identify the string to use for config ignore.**

# \$\_ENV DETECT & SWITCH

```
757
758 // Check for $_ENV['AH_SITE_ENVIRONMENT'] on Acquia.
759 if (isset($_ENV['AH_SITE_ENVIRONMENT'])) {
760     $detected_environment = $_ENV['AH_SITE_ENVIRONMENT'];
761 }
762 else {
763     $detected_environment = 'dev';
764 }
765
```

In `settings.php`, add detection of environment and a switch for environment-specific settings.

```
766
767 // Environment-specific settings.
768 switch ($detected_environment) {
769     case 'dev':
770         // Dev-specific settings.
771         break;
772
773     case 'test':
774         // Stage-specific settings.
775         break;
776
777     case 'prod':
778         // Production-specific settings.
779         break;
780 }
781
782
```

# CREATE CONFIG SPLIT DIRECTORIES

- Sibling to main sync directory.
- Named for easy identification.
- Make directories before creating the splits.
- Config Split will make **.htaccess** files.



```
scott:/$ cd /path/to/config
```

```
scott:config$ mkdir split_dev split_stg split_prod
```

# CREATE CONFIG SPLITS (SHOWING PROD HERE)

1. Make sure current config is exported.
2. Artificially set the `$detected_environment` in `settings.php` (`$detected_environment = 'prod' ;`)
3. Cache rebuild (`drush cr`)
4. Enable modules that you want ON in production (syslog)
5. Make config changes that you want in production (performance->caching, etc.)
6. Create the split using the config form.
7. Export the config form values to main config directory using Single Export (no drush here, sorry!)
8. Export the split config items (`drush csex production`)
9. Remove `$detected_environment` from Step 1 and then Cache Rebuild
10. Run `drush cim` to get your 'dev' config.

# CREATE OVERRIDES

1. Add the hard-coded values to **settings.php**
2. Colors for Environment Indicator
3. Split turned on in each environment
4. Overrides for robotstxt in Dev and Stage, but not prod

```
// Environment-specific settings.
switch ($detected_environment) {
    case 'dev':
        // Dev-specific settings.

        // Indicator
        $config['environment_indicator.indicator']['bg_color'] = 'green';
        $config['environment_indicator.indicator']['fg_color'] = 'yellow';
        $config['environment_indicator.indicator']['name'] = 'Dev';

        // Split
        $config['config_split.config_split.development']['status'] = TRUE;

        // Override
        $config['robotstxt.settings']['content'] = "User-agent: *\nDisallow: /\n";
        break;

    case 'test':
        // Stage-specific settings.

        // Indicator
        $config['environment_indicator.indicator']['bg_color'] = 'yellow';
        $config['environment_indicator.indicator']['fg_color'] = '#000000';
        $config['environment_indicator.indicator']['name'] = 'Stage';

        // Split
        $config['config_split.config_split.stage']['status'] = TRUE;

        // Override
        $config['robotstxt.settings']['content'] = "User-agent: *\nDisallow: /\n";
        break;

    case 'prod':
        // Production-specific settings.

        // Indicator
        $config['environment_indicator.indicator']['bg_color'] = 'red';
        $config['environment_indicator.indicator']['fg_color'] = '#000000';
        $config['environment_indicator.indicator']['name'] = 'Production';

        // Split
        $config['config_split.config_split.production']['status'] = TRUE;

        // No Override for robotstxt as that is managed on config form!
        break;
}
```

# Please provide your feedback.

<https://www.midcamp.org/2019/topic-proposal/level-your-drupal-8-configuration-management>

**Scott Weston**

Principal Architect

**Email:** [scott.weston@bounteous.com](mailto:scott.weston@bounteous.com)



# Contribution Day

Saturday 10am to 4pm

You don't have to know code to give back!

New Contributor training 10am to Noon  
with **AmyJune Hinline**

