Advanced Configuration Management In Drupal 8

DrupalCon
NASHVILLE 2018

Phase2

Mike Potter
SOFTWARE ARCHITECT
mpotter@phase2technology.com
mpotter on drupal.org

- Primary maintainer of ***Features, Features Override, Config Actions*** modules
- Architect of ***Open Atrium 2*** distribution

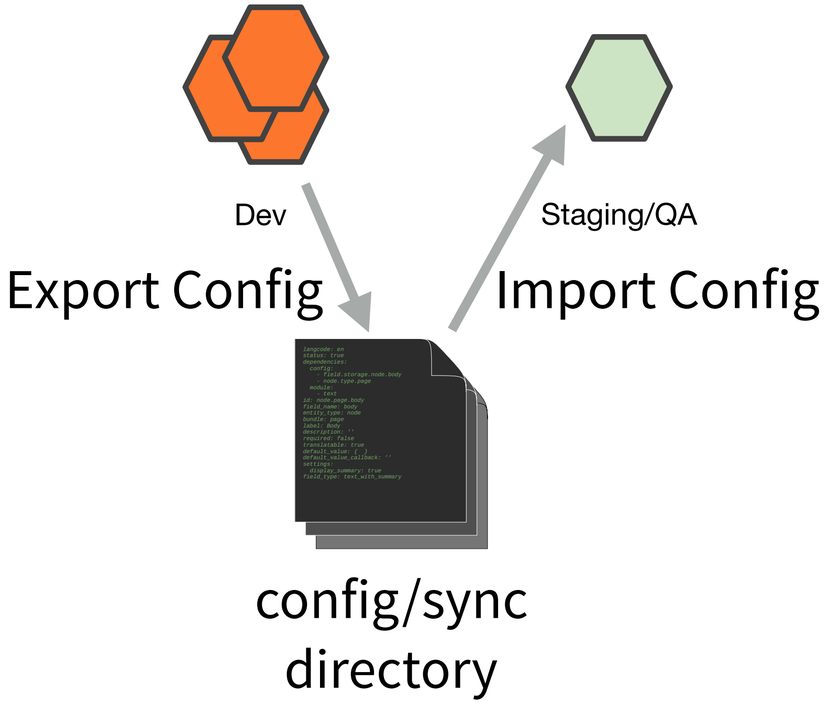
Agenda

1. Core Config Workflow
2. Environment-Specific Config (Config Split)
3. Installing with Config
4. Overriding Config
5. Features?
6. Templates & Actions
7. Summary

CORE D8 CONFIG WORKFLOW

D8 Core Config Deployment

- Modify location of config sync in settings.php
  `$config_directories['sync'] = 'config/default';`
- **Ensure config/default is added to your git repo**



- On **DEV**: Full Export config
  `drush config-export`
- On **DEV**: **Review Changes**,
  Commit and Push to your **develop** branch
- On **QA**: Pull develop, Review and import config
  `drush config-import`

Export and Import

- `drush config-export <label>`
  - Has several options for add, commit, push
  - Probably better to just use git commands.
  - Label is optional name of config location (defaults to "default")

- `drush config-import <label>`
  - `--preview=diff` to see changes
  - `--partial` ??

What about "config-import --partial"

- Will not delete config in DB that is missing from config/sync folder
- Be careful after doing a full config-export that any unwanted config doesn't get added to git repo.
- **DO NOT USE "git add -A"!!!!!**
- MUST **carefully review** changes from config-export

NOT BEST PRACTICE

Importing a single module

- Similar to "features-revert"
- Load config from a specific module /config/install

```
drush config-import --partial
   --source=modules/MODULENAME/config/install/
```

Headaches...

- Can only import into the same SITE UUID
  - Clean install creates new random uuid
- Config items have their own UUID
  - Creating new config creates random uuid
- config-import can delete config
  - will complain if content exists

ENVIRONMENT-SPECIFIC CONFIG

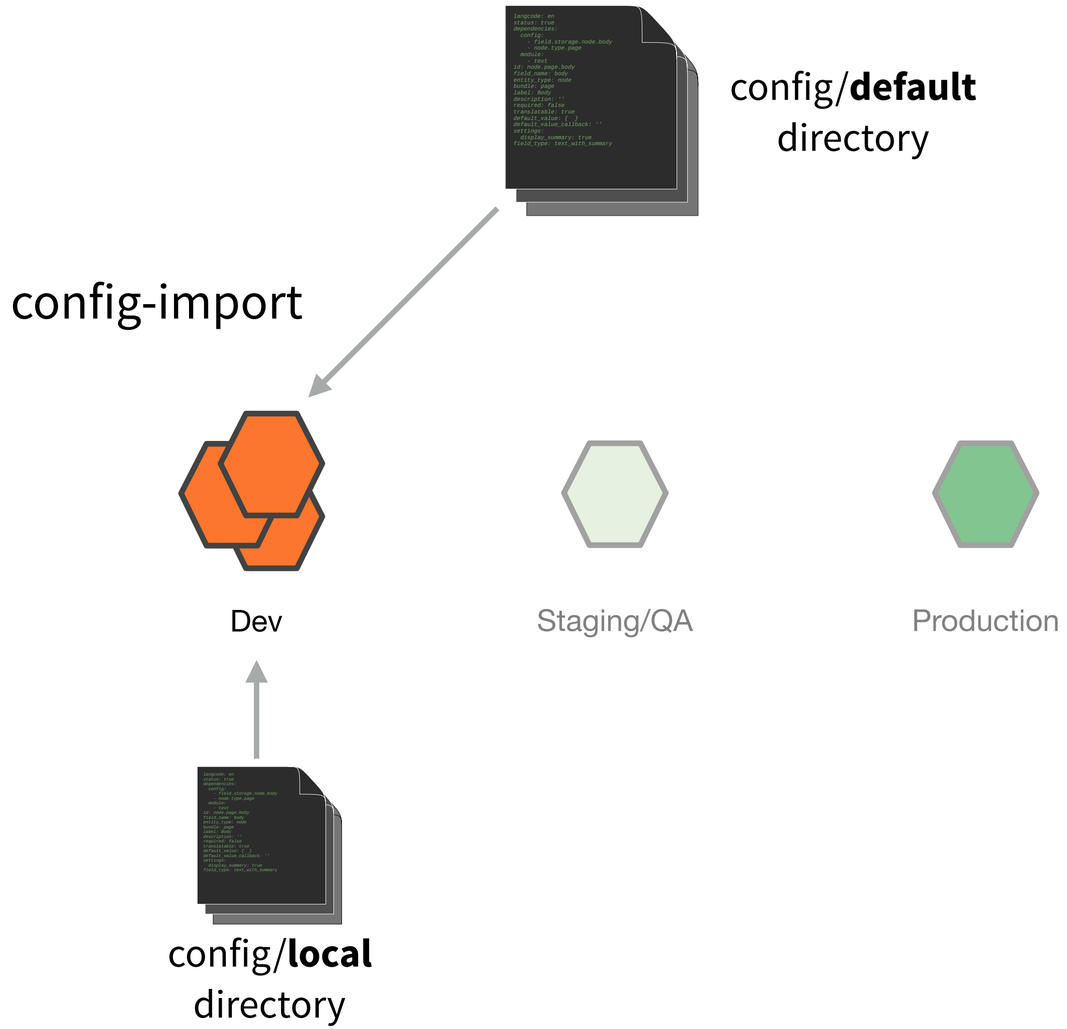
Environment overrides

- Different environments need different configuration
  - local
  - stage
  - prod
- Use **Config Split** module
  - splits different config into different config/sync directories
  - config-import merges directories

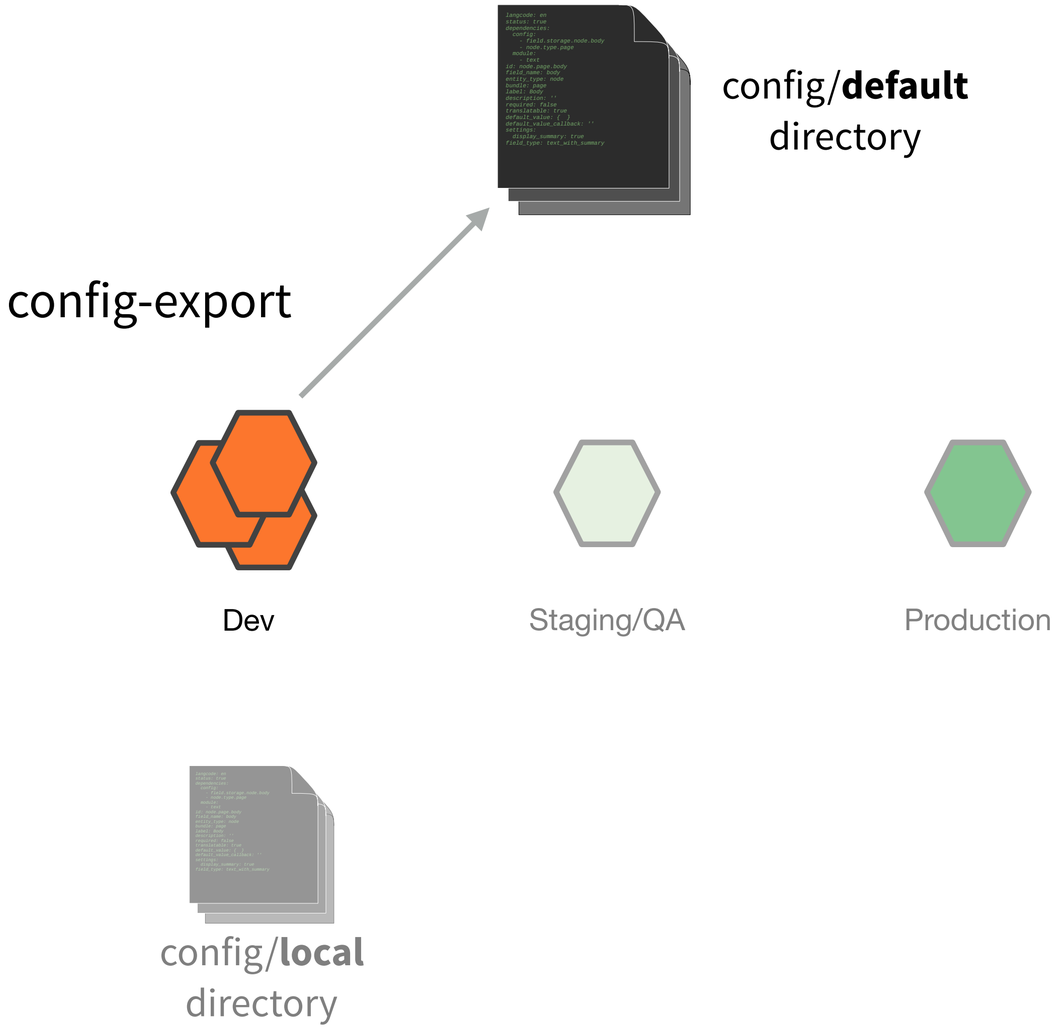
Config Split

1. Must create sync directory for each environment
2. Create splits in Config Split UI
   - Disable each split by default
3. Export the full config
4. Enable the Split Environment to use in settings.php
   - (e.g. based on ENV var)
5. config-import will merge the sync
6. To export the split, use
   `drush csex` instead of `drush cex`

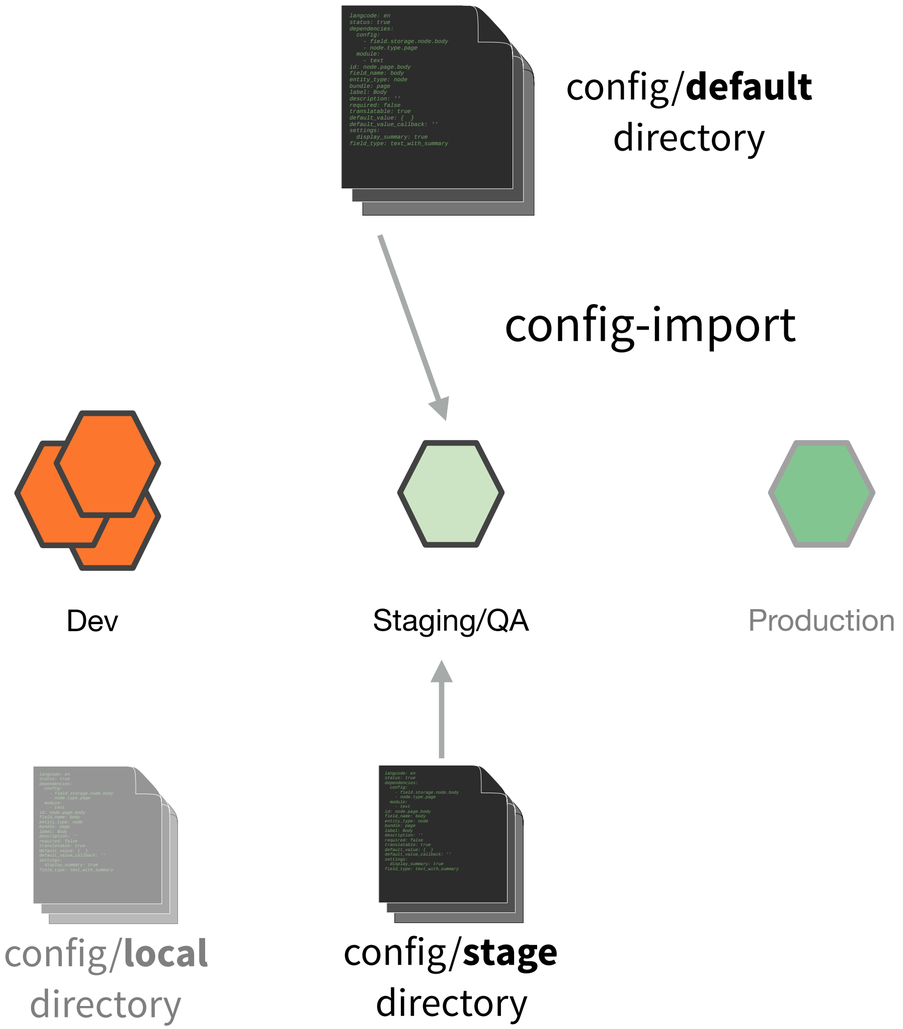
Deployment with Config Split



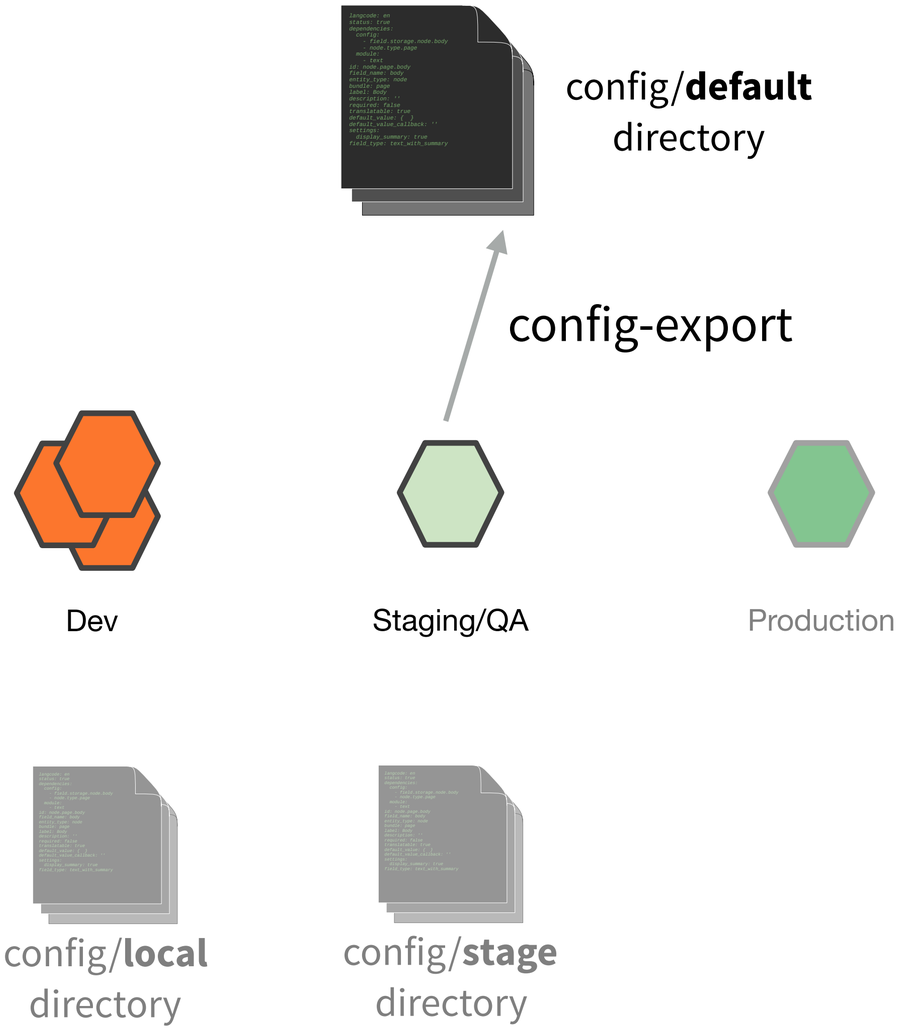
Deployment with Config Split



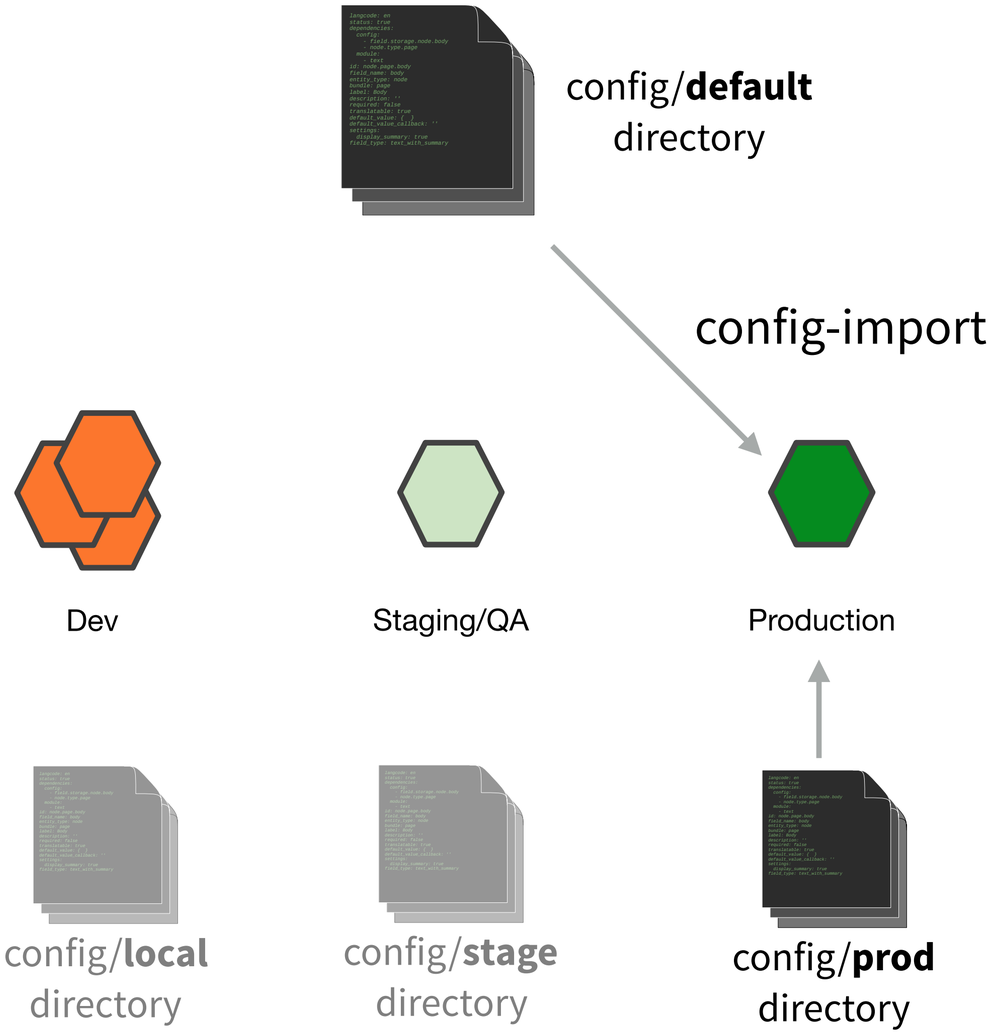
Deployment with Config Split



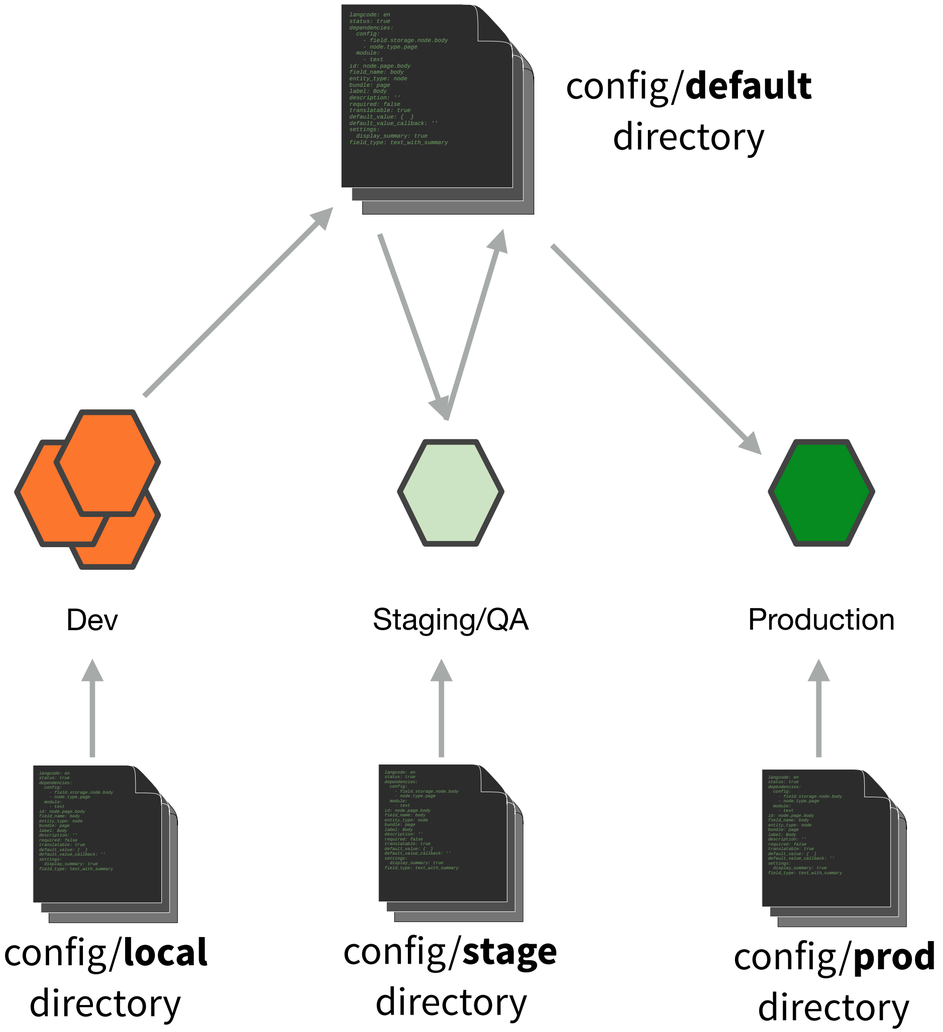
Deployment with Config Split



Deployment with Config Split



Deployment with Config Split



Blacklist vs Graylist

- **Blacklist** is: Module or Config you want **ONLY** in the Split
  - Does not exist in sync/default.
  - Config Split will enable module when the split is imported.
  - Don't need to mess with core.extensions.yml
  - Modules in Blacklist removed from core.extensions

Blacklist vs **Graylist**

- **Graylist** is: Config that exists in **BOTH** default and split
  - Config in Split will override what is in default
  - If no Split overrides, then default config is used.
  - Config *not deleted* when exported.
  - Only exported if config different from default.

Config Split Example

- **DEV**
  - Enable "devel" module
  - Dev SOLR config
  - Local "DEV" site name
- **STAGE**
  - Enable "shield" module
  - Shield config for user/password
  - Stage SOLR config
- **PROD**
  - Production SOLR config

DEV split

The configuration listed as part of a split are exported to the split directory rather than the usual sync directory when exporting the whole configuration. When importing the whole configuration, the configuration in the split directories is merged with the default sync directory and overrides the configuration. The configuration does not end up being overwritten in the sense of configuration overrides such as the overrides from settings.php.

Label *

dev — Machine name: dev [Edit]

Label for the Configuration Split Setting.

STATIC SETTINGS

Folder

config/dev

The directory, relative to the Drupal root, to which to save the filtered config. Recommended is a sibling directory of what you defined in `$config_directories[CONFIG_SYNC_DIRECTORY]` in settings.php, for more information consult the README. Configuration related to the "filtered" items below will be split from the main configuration and exported to this folder. Leave the folder empty to use a special database storage if you do not want to share the configuration.

Weight

0

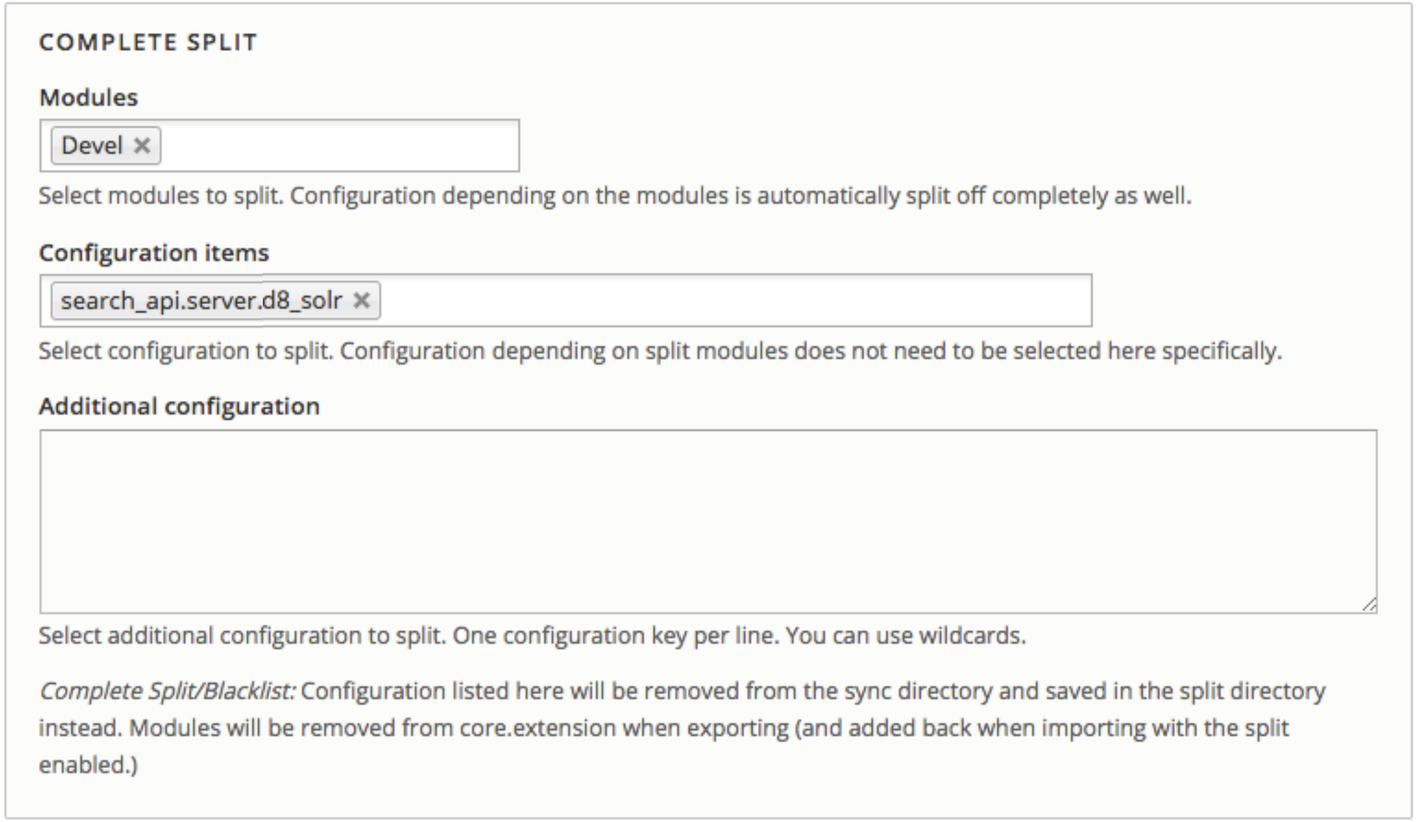
The weight to order the splits.

☐ Active

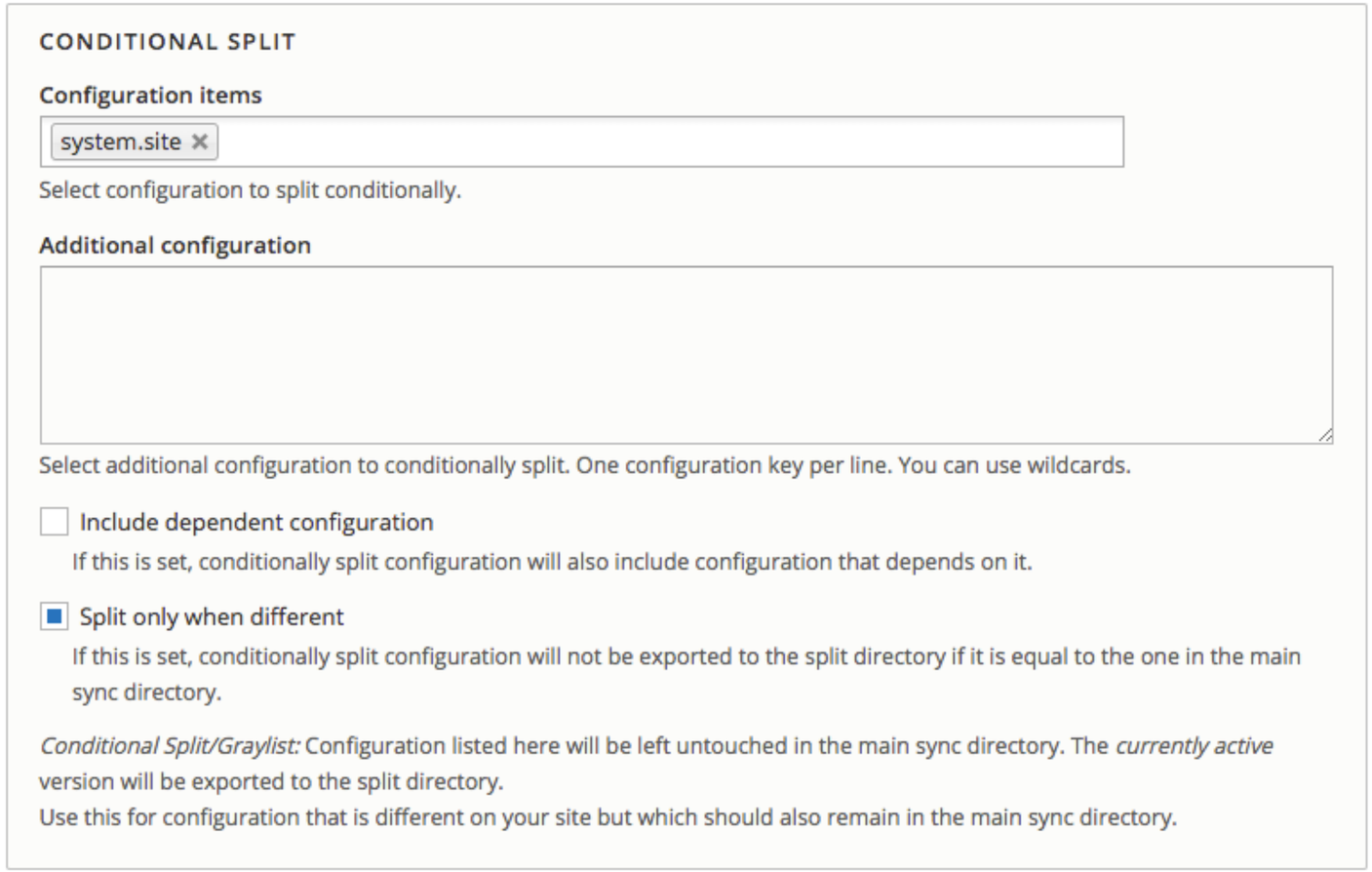
Active splits get used by default, this property can be overwritten like any other config entity in settings.php.

These settings need a cache clear when overridden in settings.php and the split needs to be single imported before the config import for new values to take effect.

DEV split ("black" list)



DEV split ("gray" list)

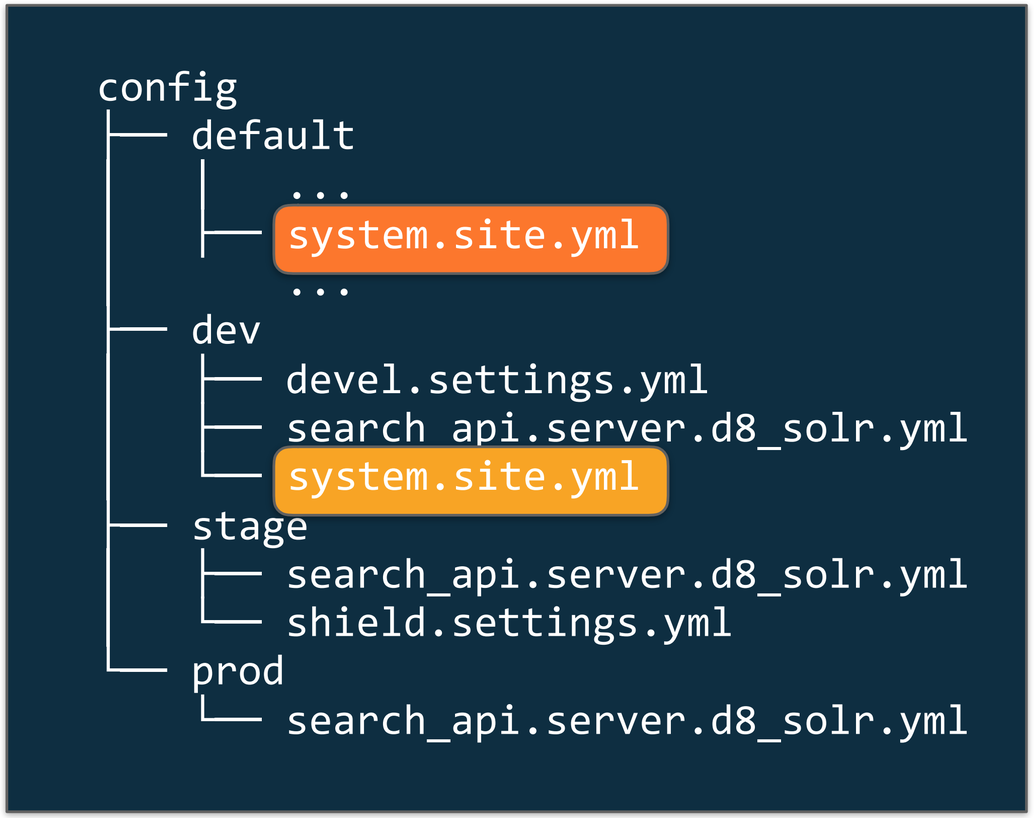


DEV split files

```
config
├── default
│   ├── ...
│   ├── system.site.yml
│   └── ...
├── dev
│   ├── devel.settings.yml
│   ├── search_api.server.d8_solr.yml
│   └── system.site.yml
├── stage
│   ├── search_api.server.d8_solr.yml
│   └── shield.settings.yml
└── prod
    └── search_api.server.d8_solr.yml
```

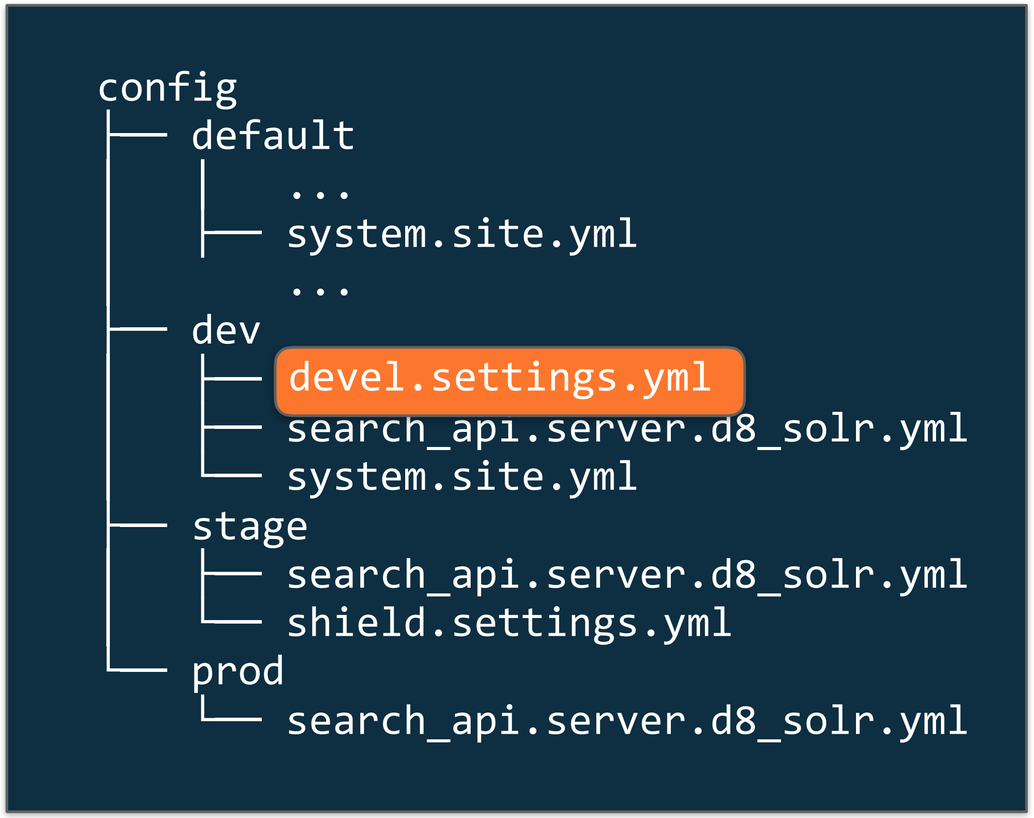
- No core.extensions in any split

DEV split files



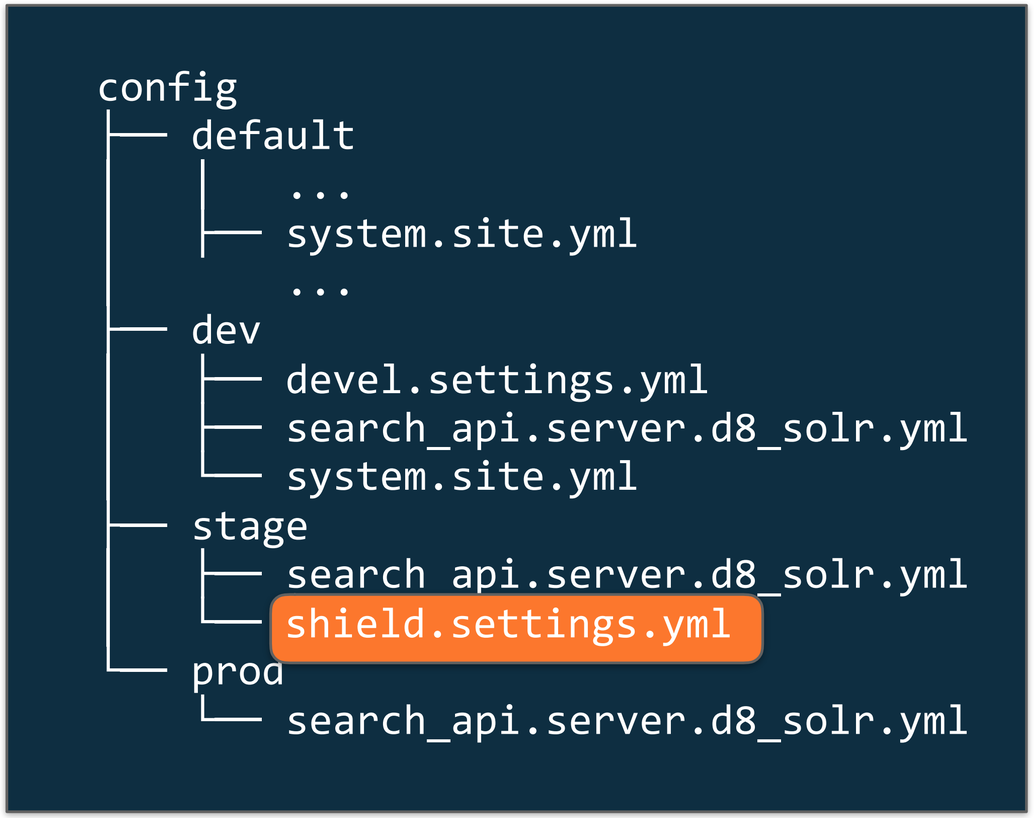
- No core.extensions in any split
- system.site is "gray listed" so included in "default" but also in "dev"

DEV split files



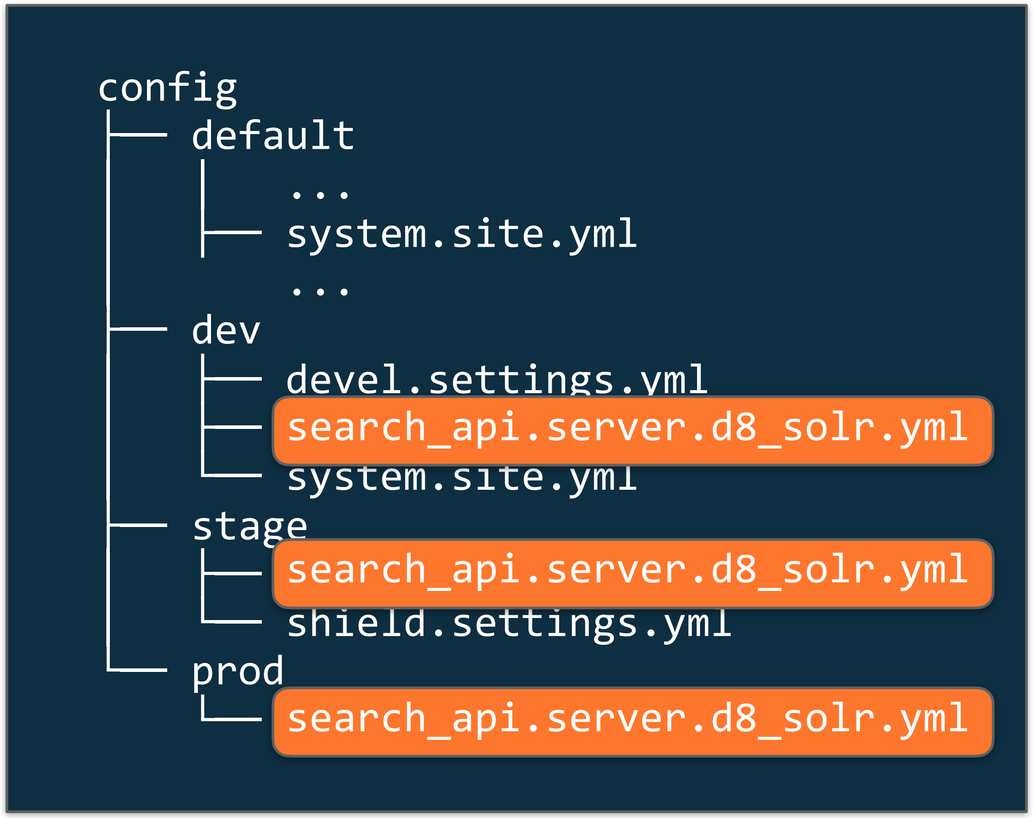
- No core.extensions in any split
- system.site is "gray listed" so included in "default" but also in "dev"
- devel just in dev

DEV split files



- No core.extensions in any split
- system.site is "gray listed" so included in "default" but also in "dev"
- devel just in dev
- shield just in stage

DEV split files



- No core.extensions in any split
- system.site is "gray listed" so included in "default" but also in "dev"
- devel just in dev
- shield just in stage
- all have search_api solr.

INSTALLING SITE USING CONFIG

When/Why to "Install from Config"

- To support clean config export/import workflow.
- Want all devs (and environments) to have same uuids.
- Installing Drupal normally and then doing config-import causes config created during install (from modules, etc) to have different uuids.
- Easy CI workflows without needing custom profile.

Config Installer Profile

- Download the config_installer profile (or via composer)
- **Ensure config/sync location is in settings.php**
- Via UI:
  Prompts for tar.gz or config/sync directory in installer UI.
- Via Drush:
  drush site-install config-installer
- Sets *actual* profile as part of the config import

https://www.drupal.org/project/config_installer

Using a custom profile

- Put config into profiles/ProfileName/config/sync
  or, specify location in profile.info:
  `config_install: config/sync`
- Add a core 8.x patch (below), then install site normally
  - `drush site-install profile-name`
- Will set the site uuid and config uuids the **same** on **every site** using the profile.
- This is likely deprecated in favor of Config Installer but might be useful for Distributions

https://www.drupal.org/node/2788777

Using drush

- drush site-install profile-name --config-dir=config/default

- DEPRECATED in Drush 9
- (also had trouble with config created by the installer)

Setting the Site ID

Shouldn't need this, but if you must:

- Look in config/default/system.site.yml

```
uuid: 0e1861a0-8e9a-469b-97d7-33317d99d785
name: My Site Name
mail: admin@example.com
...
```

- Copy the uuid value to clipboard
- Use drush config-set to set the site ID in the database:

```
# drush config-set system.site uuid <paste-uuid-here>
Do you want to update uuid key in system.site config? (y/n): y
```

When to Import/Update

- Run updates **BEFORE** import
  - git pull
  - composer install
  - drush updatedb
  - drush config-import
- Will be enforced in core
- Use update hooks for config schema changes

https://www.drupal.org/node/2762235

OVERRIDING CONFIG

"Runtime" Overrides

- Can override $config[] in settings.php
  - Cannot add new config, cannot delete config.
- **Config Override module** supports "runtime" overrides in custom modules
  - Extends core ConfigFactoryOverrideInterface
  - Same system used for Config Translations
- Overrides only affect Immutable config (doesn't affect import/export)

When do Overrides Apply?

```
\Drupal::config('system.site');
```

- Returns "Immutable" READ-ONLY config.
- Can be overridden.

```
\Drupal::configFactory()->getEditable('system.site');
```

- Returns "Mutable" READ/WRITE config.
- Overrides are ignored.

- UX challenge of editing config that is overridden

FEATURES ??

Features has Problems

- Tricky to figure out how to split up configuration
  - New assignment plugins tried to help, but are complex
- Features headaches with "unmet dependencies"
- Installing a Features export module will create new config uuids
  - causes issues with config-sync
- Features aren't actually reusable

STOP USING FEATURES!

Yes, MPotter just said:
"Stop using Features!"

- Core D8 Configuration already works!
  - Puts config into YAML files
  - Allows normal version control
  - Captures ALL the config
  - Clean export/import workflow
- Config Split is handling environment-specific config
- Get more comfortable seeing all the yml files in one place.

What about Distributions ?

What IS a Distribution ?

- Re-usable Profile
  - Used on many sites
  - Selection of functionality to enable/disable per site
  - Not just a starting point...need to update functionality
- Can't just add common config/sync to profile
  - Every site would have the same UUIDs
  - Create modules with config/install for functionality, but remove the UUID from each modules config yml.
- You can use Features to generate the config for module.

How to migrate away from Features ?

- Revert all Features to ensure DB is up-to-date
- Do full "config-export"
- *Optional*: Remove config/install files from custom modules.
- If certain modules are only enabled in certain environments, create Config Splits for that config.
- Ensure config/sync is committed/pushed to git repo.
- Test a fresh install using Config Installer
- Custom code can remain in custom modules.

TEMPLATES & ACTIONS

Re-usable Config

- Features were supposed to be re-usable
- But they are NOT!
  - Config contains machine names

field.field.node.project_blog.project_image.yml

```
langcode: en
status: true
dependencies:
  config:
    - field.storage.node.project_image
    - node.type.project_blog
  module:
    - text
id: node.project_blog.project_image
field_name: project_image
entity_type: node
bundle: project_blog
label: 'Image Field'
description: ''
required: false
translatable: true
default_value: {  }
default_value_callback: ''
settings:
  display_summary: true
field_type: text_with_summary
```

Templates

- How could we make a feature reusable?
- What if we created a "config template"

field_template.yml

```
langcode: en
status: true
dependencies:
  config:
    - field.storage.node.@field_name@
    - node.type.@bundle@
  module:
    - text
id: node.@bundle@.@field_name@
field_name: @field_name@
entity_type: node
bundle: @bundle@
label: 'Image Field'
description: ''
required: false
translatable: true
default_value: {  }
default_value_callback: ''
settings:
  display_summary: true
field_type: text_with_summary
```

Actions

- Then created a way to "use" the template

```
source: field_template.yml

@bundle@: "project_blog"
@field_name@: "project_image"

dest: field.field.node.@bundle@.@field_name@
```

Overrides

- Then override some values:

```
source: field_template.yml
@bundle@: "project_blog"
@field_name@: "project_image"
dest: field.field.node.@bundle@.@field_name@

change:
  label: "My Project Image Field"
  description: "Project-specific description"
  required: true
```

Config Actions Module

- drupal.org/project/config_actions
- Pluggable framework for manipulating configuration:
  - variable replacements
  - change values
  - add values
  - delete config
  - include templates
  - read/write yml files or config entities
- http://config-actions.readthedocs.io/
- Beta, release planned soon

TIPS & TRICKS

Config vs Content

- Not everything is config:
  - Block content, Menu links, etc
  - Default values for Entity Reference fields
- Default Content module can export entities as json
  https://www.drupal.org/project/default_content
  `drush default-content-import-all`
- Use Migrate module with migrate_source_csv
  to create default menu links, taxonomy terms, etc.

Config keeps exporting

- After config-import, some config still gets exported
- Often fields in entity_form_display or entity_view_mode
- If pulling config export from another developer:
  - Run config-export and examine the diffs
  - Might need to commit/push to update the other dev or might need to fix your local.
- To "fix" your local:
  - use "drush config-delete ..." to delete the problem config
  - then re-run config-import to create it from scratch.

Other Config Modules

- config_suite - auto export/import when config changes.
- config_readonly - useful on Production
- ...many others

SUMMARY

Summary

1. Stop using Features in D8
2. Use config-export / config-import
3. Use Config Split for environment-specific config
4. Use Config Installer to install from config
5. Manage config changes via git
6. Config Actions for generating config from templates
7. Don't be afraid of config yml files!
8. Stay tuned for CMI 2.0!

QUESTIONS?

THANK YOU

COME BY AND SAY HELLO AT BOOTH

205

MIKE POTTER

mpotter@phase2technology.com

mpotter on drupal.org

DrupalCon
NASHVILLE 2018
APRIL 9-13

Join us for contribution sprints

Friday, April 13, 2018

Mentored Core sprint

9:00-18:00
Room: 103

First time sprinter workshop

9:00-12:00
Room: 101

General sprint

9:00-18:00
Room: 104

#drupalsprint

What did you think?

Locate this session at the DrupalCon Nashville website:

http://nashville2018.drupal.org/schedule

Take the Survey!

https://www.surveymonkey.com/r/nashiville