DrupalCon

EUROPE2021

OCTOBER 4-7

DrupalCon
EUROPE2021
OCTOBER 4-7

Config Split 2.x

and the future of CMI2

Fabian Bircher

senior software engineer @ Nuvole

@fabianbircher @nuvoleweb

bircher

Nuvole

NUVOLE

Distributed team across Europe

International Organisations & Institutions

Pioneers with configuration management

About this presentation

Documentation

This session is an update to how Config Split works.

Basic introduction from Drupalcon Vienna 2017 is still valid, but there is a lot of new things to explore.

- How config to split is selected
- How config is split
- Improved concepts for workflows
- Improved UI

Roadmap to core inclusion of some features.

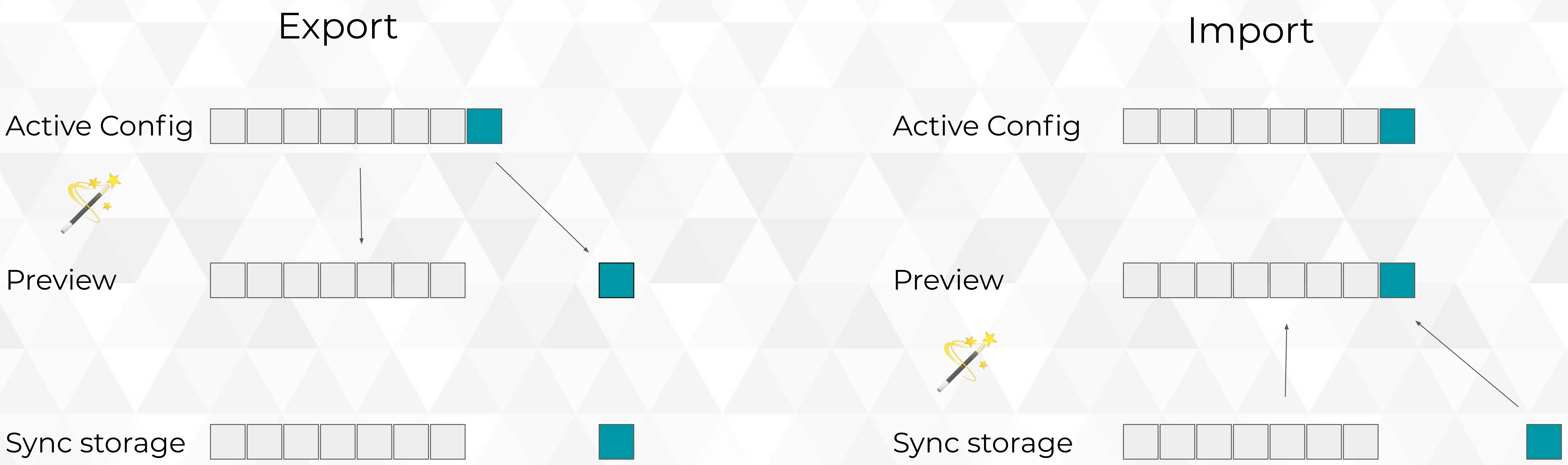
DRUPALCON VIENNA

Advanced Configuration Management with Config Split et al.

Fabian Bircher fabian@nuvole.org

web: nuvole.org **twitter:** @nuvoleweb

What is Config Split?



Export

Active Config

Preview

Sync storage

Import

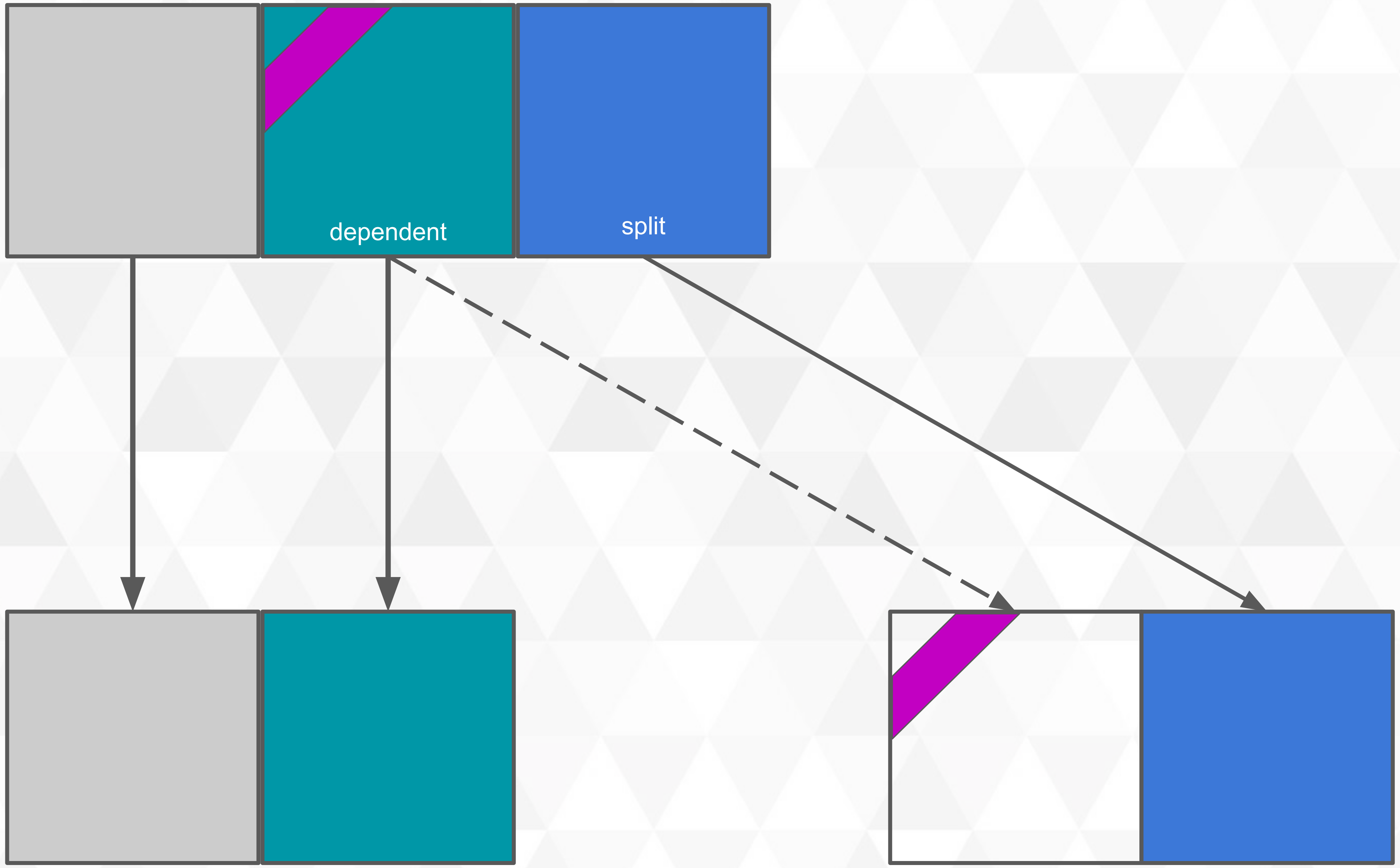
Active Config

Preview

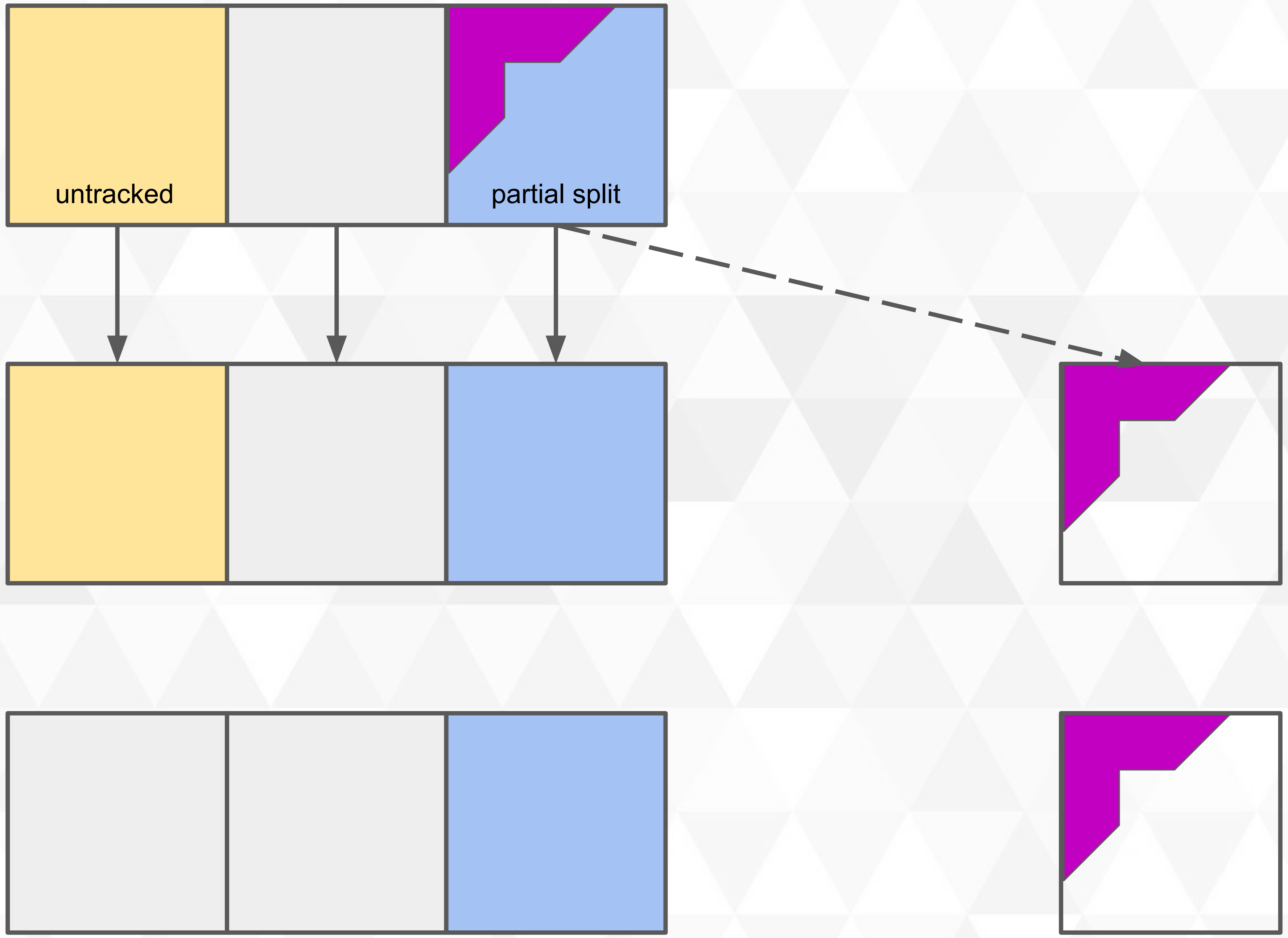
Sync storage

What is new in 2.x

Split modules as if they had been uninstalled
Split config as if it had been deleted

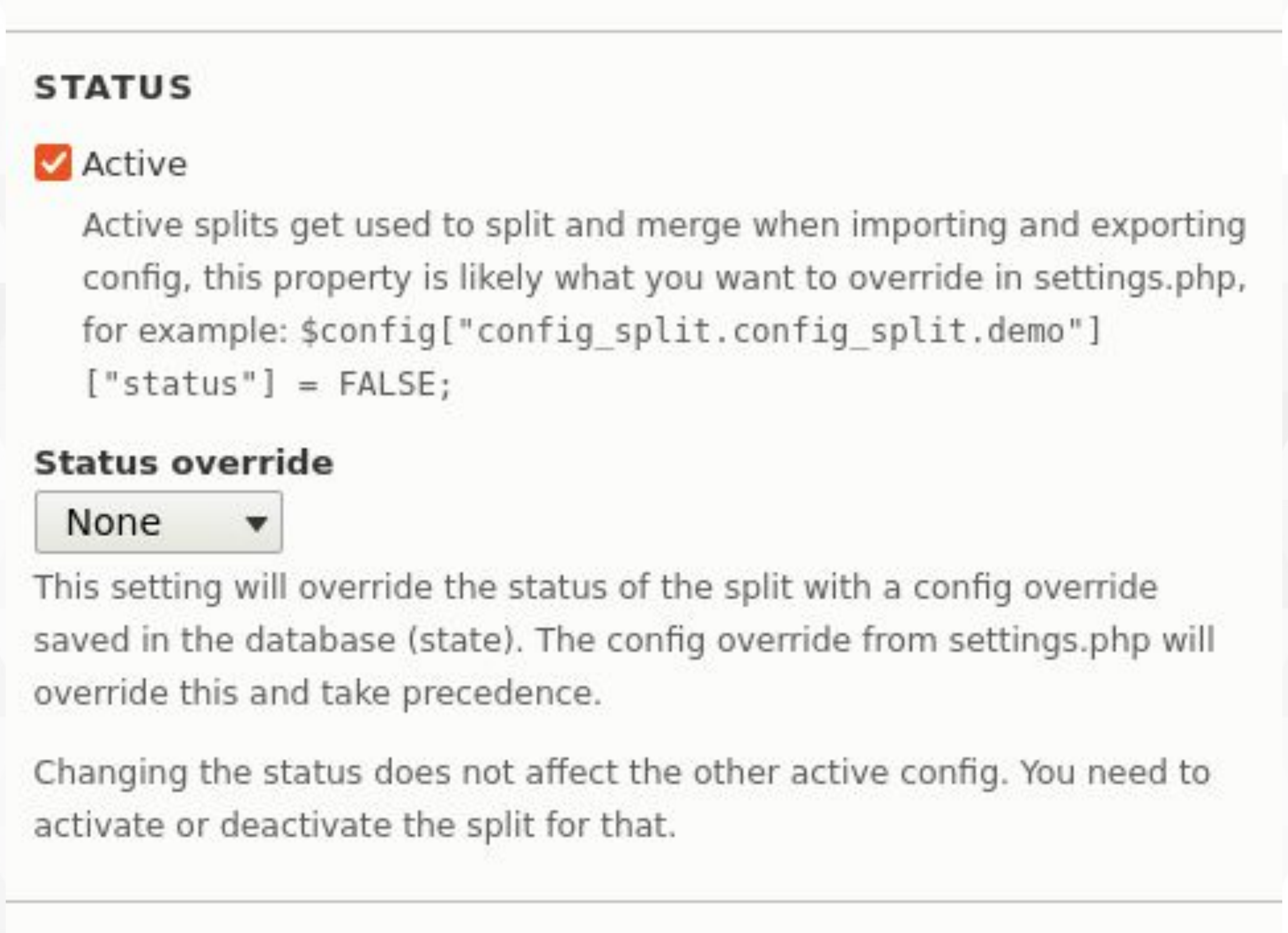


Partial split: only what is different in the sync directory.



Status override with state

In the database in addition to settings.php



STATUS

☑ Active

Active splits get used to split and merge when importing and exporting config, this property is likely what you want to override in settings.php, for example: `$config["config_split.config_split.demo"]["status"] = FALSE;`

Status override

None ▾

This setting will override the status of the split with a config override saved in the database (state). The config override from settings.php will override this and take precedence.

Changing the status does not affect the other active config. You need to activate or deactivate the split for that.

The status is normal configuration like before.

Status override is saved in the database and will not be exported when exporting config.

Status override is just a normal config override but with a UI in the same form as the normal split config.

The config override from settings.php override any other config override including this.

```
drush config-split:status-override
```

Split activation and deactivation

In 1.x this means: export configuration, change override in settings.php, import configuration

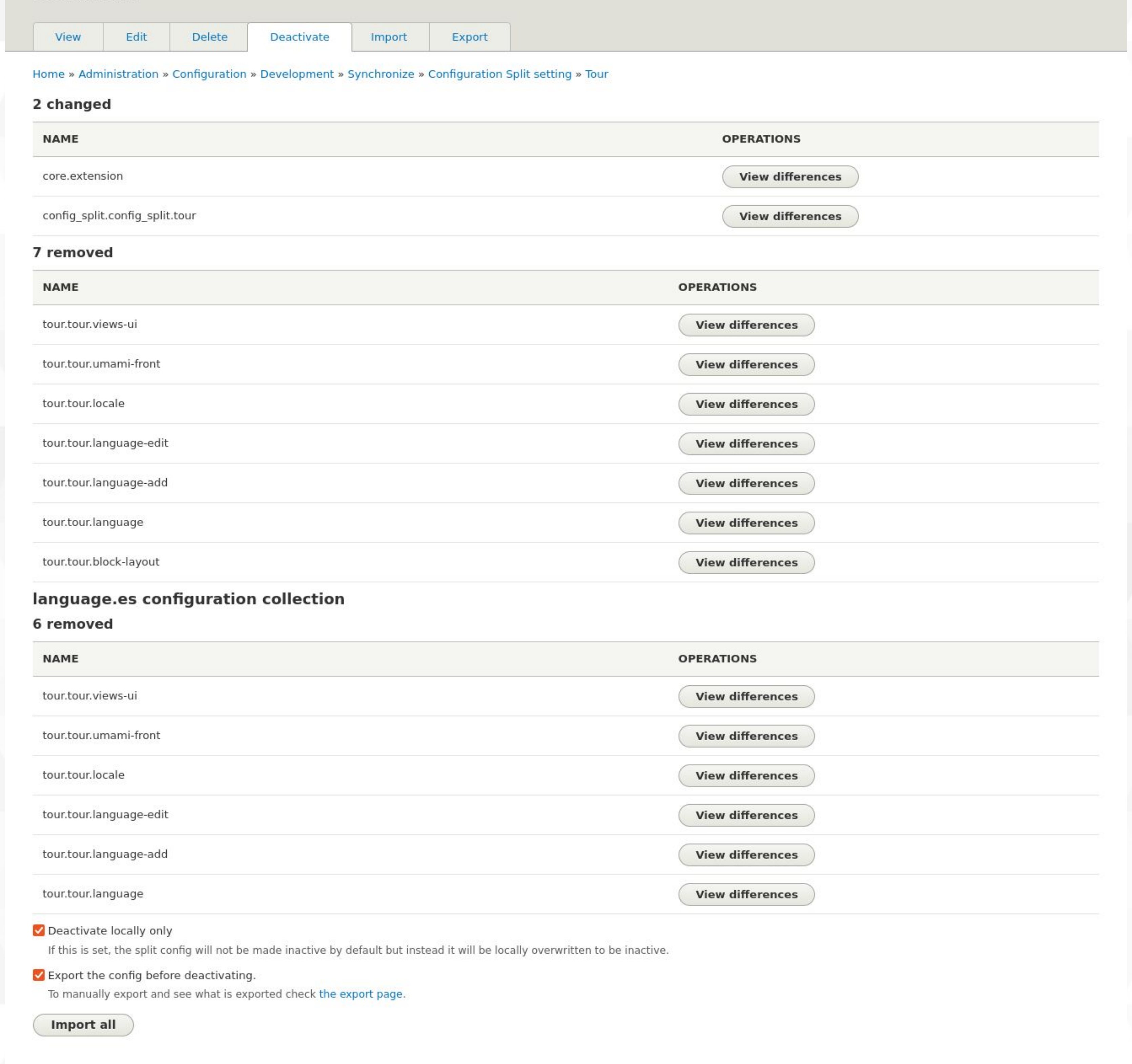
in 2.x this is one step: `drush config-split:activate` and `drush config-split:deactivate`

It also has a UI heavily inspired by the config import UI of drupal core.

Is related but distinct from importing and exporting an individual split.

DrupalCon
EUROPE2021
OCTOBER 4-7

UI for operations import, export, activate, deactivate



Home » Administration » Configuration » Development » Synchronize » Configuration Split setting » Tour

2 changed

NAME	OPERATIONS
core.extension	View differences
config_split.config_split.tour	View differences

7 removed

NAME	OPERATIONS
tour.tour.views-ui	View differences
tour.tour.umami-front	View differences
tour.tour.locale	View differences
tour.tour.language-edit	View differences
tour.tour.language-add	View differences
tour.tour.language	View differences
tour.tour.block-layout	View differences

language.es configuration collection

6 removed

NAME	OPERATIONS
tour.tour.views-ui	View differences
tour.tour.umami-front	View differences
tour.tour.locale	View differences
tour.tour.language-edit	View differences
tour.tour.language-add	View differences
tour.tour.language	View differences

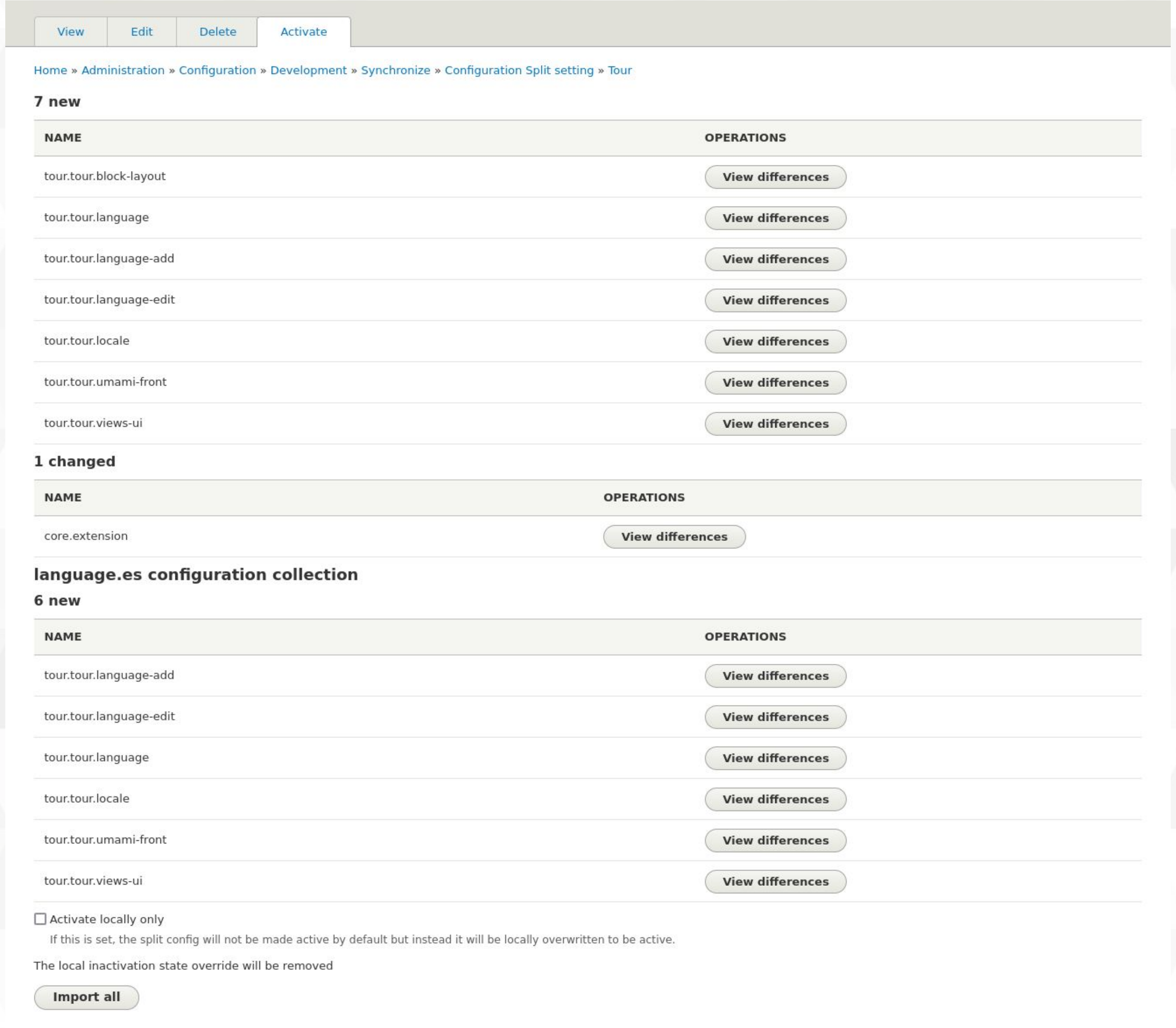
☑ Deactivate locally only

If this is set, the split config will not be made inactive by default but instead it will be locally overwritten to be inactive.

☑ Export the config before deactivating.

To manually export and see what is exported check the export page.

Import all



Home » Administration » Configuration » Development » Synchronize » Configuration Split setting » Tour

7 new

NAME	OPERATIONS
tour.tour.block-layout	View differences
tour.tour.language	View differences
tour.tour.language-add	View differences
tour.tour.language-edit	View differences
tour.tour.locale	View differences
tour.tour.umami-front	View differences
tour.tour.views-ui	View differences

1 changed

NAME	OPERATIONS
core.extension	View differences

language.es configuration collection

6 new

NAME	OPERATIONS
tour.tour.language-add	View differences
tour.tour.language-edit	View differences
tour.tour.language	View differences
tour.tour.locale	View differences
tour.tour.umami-front	View differences
tour.tour.views-ui	View differences

☐ Activate locally only

If this is set, the split config will not be made active by default but instead it will be locally overwritten to be active.

The local inactivation state override will be removed

Import all

Split as a collection

inside of sync directory and zip file

Example adding menu_ui to demo split.

```
$ drush config-export
 [notice] Differences of the active config to the export
directory:
+------------+-------------------------------------+-----------+
| Collection | Config                              | Operation |
+------------+-------------------------------------+-----------+
|            | core.extension                      | Update    |
|            | node.type.recipe                    | Update    |
|            | config_split.config_split.demo      | Update    |
|            | menu_ui.settings                    | Delete    |
| split.demo | config_split.patch.node.type.recipe | Create    |
| split.demo | menu_ui.settings                    | Create    |
+------------+-------------------------------------+-----------+
```

The standard drush config-export includes the split in the preview.

The split is included in the zip archive from the config export UI of drupal core.

Config of inactive splits are still in the active config, just in a collection.

Experimental: this should theoretically work and does for the limited examples I tried.

New splits are immediately available

During the config import config split considers all splits currently existing and new ones in the sync directory.

Config which is moved to a new active split will not be deleted.
You don’t have to single import the split first. A single config import should suffice.

Splits are more independent of each other

Using core services means the split is using the active config instead of the config altered by other splits.

This means it is possible to have two splits that säplit the same module at the same time.

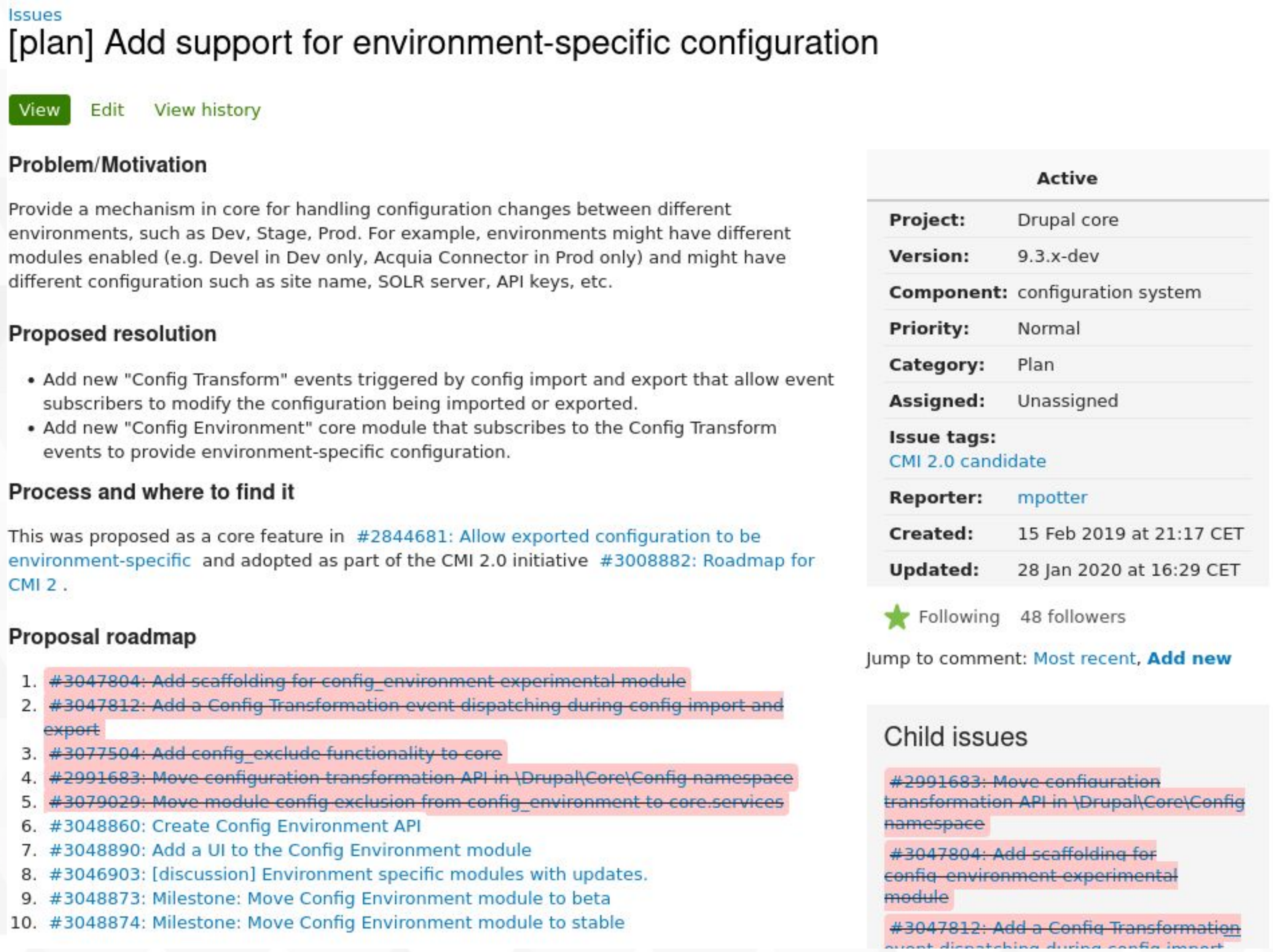
The weights still matter.

Weights can be adjusted in relation to other modules using the config transformation API in settings.php

CMI 2

Big plan

## [#3033427]



Issues

[plan] Add support for environment-specific configuration

Problem/Motivation

Provide a mechanism in core for handling configuration changes between different environments, such as Dev, Stage, Prod. For example, environments might have different modules enabled (e.g. Devel in Dev only, Acquia Connector in Prod only) and might have different configuration such as site name, SOLR server, API keys, etc.

Proposed resolution

- Add new "Config Transform" events triggered by config import and export that allow event subscribers to modify the configuration being imported or exported.
- Add new "Config Environment" core module that subscribes to the Config Transform events to provide environment-specific configuration.

Process and where to find it

This was proposed as a core feature in #2844681: Allow exported configuration to be environment-specific and adopted as part of the CMI 2.0 initiative #3008882: Roadmap for CMI 2 .

Proposal roadmap

1. ~~#3047804: Add scaffolding for config_environment experimental module~~
2. ~~#3047812: Add a Config Transformation event dispatching during config import and export~~
3. ~~#3077504: Add config_exclude functionality to core~~
4. ~~#2991683: Move configuration transformation API in \Drupal\Core\Config namespace~~
5. ~~#3079029: Move module config exclusion from config_environment to core.services~~
6. #3048860: Create Config Environment API
7. #3048890: Add a UI to the Config Environment module
8. #3046903: [discussion] Environment specific modules with updates.
9. #3048873: Milestone: Move Config Environment module to beta
10. #3048874: Milestone: Move Config Environment module to stable

Active	
Project:	Drupal core
Version:	9.3.x-dev
Component:	configuration system
Priority:	Normal
Category:	Plan
Assigned:	Unassigned
Issue tags: CMI 2.0 candidate	
Reporter:	mpotter
Created:	15 Feb 2019 at 21:17 CET
Updated:	28 Jan 2020 at 16:29 CET

Jump to comment: Most recent, **Add new**

Child issues

~~#2991683: Move configuration transformation API in \Drupal\Core\Config namespace~~

~~#3047804: Add scaffolding for config_environment experimental module~~

~~#3047812: Add a Config Transformation event dispatching during config import~~

Adding things to core is not easy.

Config Split solves the problem but is also used for many other use cases.

Config Split 2.x has new features which serve as a proof of concept for the new core module.

Mimic config_environment with config_split

- Only one split active at a time
- Use the collection as split storage
- Do not use the partial split feature

Difficulties!
config arrays are not sortable

It would have been too easy...

maybe more?
Config Split 2.x will find out

thank you for reporting bugs and contributing patches/MRs

Roadmap

- Use Config Split 2.x in many different projects
- Identify and fix bugs in Config Split 2.x
- Assess what is a hack/workaround for core issues
- Create new simplified contrib module
- Include it in core
- celebrate

“

Thank you!

”

DrupalCon
EUROPE2021
OCTOBER 4-7

Join us for contribution opportunities

4-7 October, 2021
Online on https://drupalcontributions.opensocial.site/

Mentored Contribution

7th of October: 14:00 - 17:00

First Time Contributor Workshop

5th of October: 14:00 - 17:00

General Contribution

4th of October: 9:00 - 12:00
5th of October: 14:00 - 17:00
6th of October: 9:00 - 12:00
7th of October: 14:00 - 17:00

#DrupalContributions

What did you think?

Have a look at the session description below
and take the survey for each session:

https://events.drupal.org/europe2021/schedule/all

Thank you!