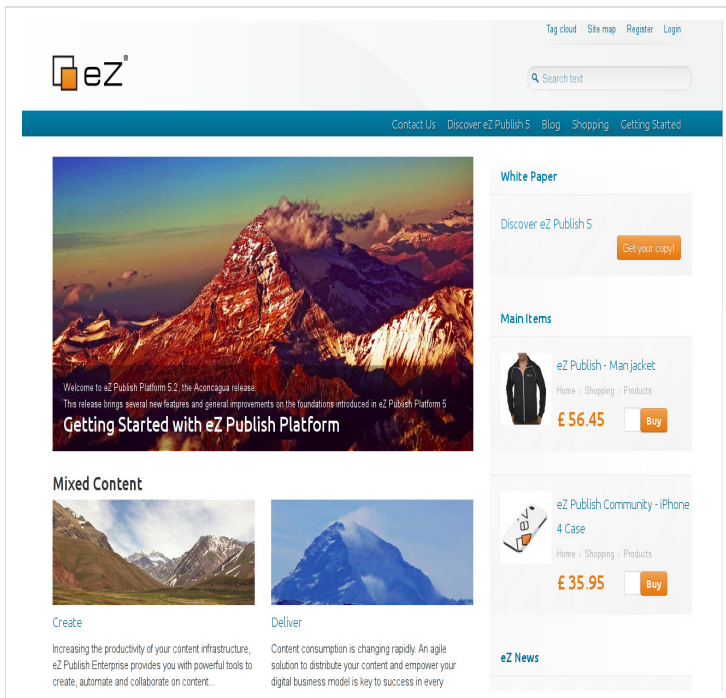


## 5.2 Release Notes

- Meet the updated eZ Publish Platform 5.2
  - 5.2 is on a direct path to 5.3LTS
  - Quick links
- 5.2 Features
  - Context-aware HTTP cache with UserHash
  - Symfony 2.3.\*
  - Content Controller override support
  - API Signals listenable, also via Symfony event system
  - Multi repository persistence cache support
  - Support for more than 30 languages
  - Extended support for comments with CommentsBundle
  - Pagerfanta support for pagination
  - Custom tags for XMLText field type
  - Extensible REST API
  - DFS table split
  - Javascript REST Client
  - Configurable front controller
  - Miscellaneous helpers
- 5.2 Contributors
- 5.2 Full Changelog
- 5.2 Known issues

### Meet the updated eZ Publish Platform 5.2

*Tag line: We highly recommend 5.2.x as a baseline for creating new Symfony / eZ Publish Platform 5.x based web sites, as it will make your life easier.*



eZ Publish Platform 5.2 is a STS (Short term supported) release towards 5.3 LTS (Long term supported) which enhances several aspects of the 5.x kernel for usability and extensibility, in addition to additional performance improvements on top of what 5.1 delivered. The main topic of 5.2 has been to improve a lot in terms of building websites, DemoBundle should now serve as a good example.

### 5.2 is on a direct path to 5.3LTS

Even if you plan to launch on 5.3 LTS, no worries, we plan to add richer composer support in the next couple of months providing you a more streamlined update process allowing you to start using 5.2 right now.

### Quick links

- [Install 5.2](#)
- [Upgrade from 5.1 to 5.2](#)

### 5.2 Features

5.2 introduces several new features which makes it easier to use, faster to develop on and also improves performance across the board. Performance tests using jMeter\* internally have shown up to 25% improvement (from 361 to 452) to requests per seconds when testing on pure new stack using eZ Demo, while also improving legacy fallback performance 18% on ezflow (from 37 to 43 requests per second).

#### Context-aware HTTP cache with *UserHash*

With eZ Publish Platform 5.2, new stack's HTTP cache is finally on par and beyond what was possible in eZ Publish 4.x/legacy. With *UserHash*, by default all internal controllers, including the ViewController, will return cacheable responses, based on user access rules. This behavior is also applicable to custom controllers by making their response *vary* on *X-User-Hash* header. Moreover, UserHash generation, initially based on current user's assigned roles and policies, is extensible.

Purging is furthermore done in a **stale cache** like way, making sure stale cache gets served while the new one is being generated.

- [Context-aware HTTP cache documentation.](#)
- [Extend UserHash generation.](#)
- [Original pull-request.](#)

## Symfony 2.3.\*

Updated our web framework foundation to Symfony 2.3 LTS, more info on this release can be found [here](#).

## Content Controller override support

Have you ever been missing a way to extend the provided content controller? Had needs to do custom cache logic and been missing the good old `cache_ttl = 0` hack?

5.2 brings with it a powerful way to use the existing override system to, instead of overriding just a template, **override the whole controller**. This allows you to extend the built-in controller and for instance get full access to tune the cache headers for a specific content type that has different use than normal content, or just enrich the final template.

- [How to use a custom controller to display a content item or location.](#)
- [Original pull-request.](#)

## API Signals listenable, also via Symfony event system

Signal slot events are now fully *listenable*, which makes it possible to react on core API signals. These events are now also dispatched over Symfony's event system so you can listen to them there synchronously, while custom slots (listeners for signals) can in the future be dispatched asynchronously.

This enhancement makes it possible to define micro-workflows.

- [How to listen to Core events.](#)
- [Signals reference.](#)
- [Original pull-request.](#)

## Multi repository persistence cache support

This release resolves a limitation with persistence caching introduced in 5.1, a new caching service "ezpublish.cache\_pool" takes a siteaccess group setting "cache\_pool" into consideration for selecting cache service, and hence can be configured with multi repository setups in mind. In addition you can now safely reuse this cache pool as the persistence cache is prefixed within "spi" cache key and will not clear your data if you reuse it.

- [Updated documentation on reusing the cache service](#)
- [Updated documentation on configuring multi repository setup](#)
- [Original pull-request](#)

## Support for more than 30 languages

With 64-bits servers now being common, this release takes advantage of that if present and provides support for **up to 62 languages on 64-bits environments** and **30 languages on 32-bits platforms**. This feature is supported across both *Symfony stack* and *legacy stack*, and is accomplished by increasing the size of the integer field used for language bit masks. Improved storage engines still has the possibility to provide even more languages on pure *Symfony stack* in the future.

- [JIRA issue](#)
- [Original Pull requests: legacy, 5.x](#)

## Extended support for comments with CommentsBundle

Brand new CommentsBundle provides an abstracted way to display comments. It is **provider based**, which means that it is open to **any kind of commenting system**.

It has native support for **Disqus** and **Facebook comments**.

- [CommentsBundle](#)
- [Documentation](#)

## Pagerfanta support for pagination

Pagination can easily become cumbersome when done by hand. **Pagerfanta** is a powerful library for object oriented pagination, standard in the Symfony world. eZ Publish Platform 5.2 comes with native support of PagerFanta, with dedicated adapters.

- [How to paginate API search results](#)
- [Initial pull-request](#)
- External links
  - [Pagerfanta](#)
  - [WhiteOctoberPagerfantaBundle](#)

## Custom tags for XMLText field type

3rd party custom tags support have been greatly improved and implementation is now much easier. It is now possible add a custom XSL stylesheet to the main one and to assign a priority. This also makes it possible to easily override existing eZXML tags rendering.

- [How to implement a Custom Tag for XMLText FieldType](#)
- [Initial pull-request](#)

## Extensible REST API

The eZ Publish Platform REST API can now easily be extended: custom routes, media-types... Our framework will easily handle input parsing, error handling and output generation (to XML or JSON) for you.

- See [Extending the REST API](#).

## DFS table split

The DFS MySQLi implementation now uses two distinct tables for storage and cache. The main benefit is that clearing cache is as simple as truncating the cache table without affecting storage in any way. This feature is enabled by default, and can easily be disabled.

- Pull request: <https://github.com/ezsystems/ezpublish-legacy/pull/750>
- [Technical documentation](#)
- Initial feature suggestion by [@omh](#)

## Javascript REST Client

As part of this release we have made the Javascript REST client which we aim to use in our future user interface open, this can be found on github and can be used for your own REST needs.

- [github.com/ezsystems/ez-js-rest-client](https://github.com/ezsystems/ez-js-rest-client)

## Configurable front controller

The standard symfony like index\*.php files can now be configured using environment variables, no need to change the file anymore.

- [Apache Virtual host setup](#)
- For all possibilities check at any time, check your [web/index.php](#) and [web/index\\_dev.php](#) (note: links goes to "master" and might include features not found in your version)

## Miscellaneous helpers

Various Twig helpers were added to ease daily tasks, such as:

- [ez\\_content\\_name](#), to display a Content name in the current language.
- [ez\\_field\\_value](#), to get a Field value in the current language.
- [ez\\_is\\_field\\_empty](#), to check if a given field is empty for the current language.

These helpers are also available in PHP code through [FieldHelper](#) (service ID: `ezpublish.field_helper`) and [TranslationHelper](#) (service ID: `ezpublish.translation_helper`) services.

## 5.2 Contributors

For 5.2 we want to reach out a special thank you to those that have contributed to this release, especially the brave few who have helped out on the "new stack":

- [bchoquet-heliopsis](#)
- [masev](#)
- [pborreli](#)
- [paradoxe](#)
- [tharkun](#)
- [crevillo](#)
- [jeromegamez](#)
- [Plopix](#)
- [itagent](#)
- [mmaury-heliopsis](#)

In addition `ezpublish-legacy` is still receiving a lot of pull requests, contributions to this part of the release can be found [here](#).

## 5.2 Full Changelog

For the full 5.2 changelog please refer to the [5.2 Changelog](#) dedicated page.

## 5.2 Known issues

The complete list of know issues is available in the [5.2 Known issues](#) page.

*\* jMeter numbers reflect a load of 800 users.*