Upgrading from 5.2 to 5.4

This section describes how to upgrade your existing eZ Publish 5.2 installation to version 5.4. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure that the installation you are performing the upgrade on is offline.

Note on Paths

- /<ezp5-root>/: The root directory where eZ Publish 5 is installed in, examples: "/home/myuser/www/" or "/var/sites/ezpublish/"
- /<ezp5-root>/ezpublish_legacy/: Root directory of "Legacy" (aka "Legacy Stack", refers to the eZ Publish 4.x installation which is bundled with eZ Publish 5) normally inside "ezpublish_legacy/", example: "/home/myuser/www/ezpublish_legacy/"

Check for requirements

PHP 5.4.4 and higher is needed. Further information regarding system requirements can be found on Requirements Documentation Page.

Upgrade steps

Step 1: Upgrade the legacy distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 5.4 to a separate directory and then copy the directories that contain site-specific files from the existing 5.2 installation into "/<ezp5-root>/". Make sure you copy the following directories:

- `ezpublish_legacy/design/<your designs>` (do NOT include built-in designs: admin, base, standard or admin2)
- `ezpublish_legacy/var/<your_var_dir>/storage`
- `ezpublish_legacy/var/storage/packages`
- `ezpublish_legacy/settings/siteaccess/<your_siteaccesses>`
- `ezpublish_legacy/settings/override/*`
- `ezpublish_legacy/extension/*` (not including the built-in / standalone ones: ezflow, ezjscore, ezoe, ezodf, ezie, ezmultiupload,

ezmbpaex, ez_network, ezprestapiprovider, ezscriptmonitor, ezsi, ezfind)
- src/
- `ezpublish_legacy/config.php` & `ezpublish_legacy/config.cluster.php`, if applicable
- `ezpublish/config/*`

NB: *Since writable directories and files have been replaced / copied, their permissions might have changed. You may have to reconfigure webserver user permissions on specific folders as explained in the file* [permissions chapter of the installation process](#)*.*

Step 2: upgrade custom legacy extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied from the existing 5.2 installation into "*<ezp5-root>/ezpublish_legacy/extension/*". However, make sure that you do not overwrite any extensions that are included in eZ Publish distribution, which currently are (***Note:*** *In eZ Publish Platform 5.4, extensions have version number 5.3 with exception of ezdemo, ezoe, ezformtoken, ezjscore and ezfind*):

Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

Important: *If you plan to upgrade your eZ Website Interface, eZ Flow or eZ Demo site package as well, then additional extensions will be updated and the step for re-generate the autoload arrays can be skipped until that is done (links to documentation for upgrading these site packages can be found in the last step of this page).*

Step 3: upgrade the database

> This step assumes use of the built in database drivers, mysql (incl mariadb) and PostgreSQL, for other databases supported via extension please use scripts and documentation provided by extension.

Import to your database the changes provided in

`/<ezp5-root>/ezpublish_legacy/update/database/<mysql|postgresql>/5.3/dbupdate-5.2.0-to-5.3.0.sql`

and

`/<ezp5-root>/ezpublish_legacy/update/database/<mysql|postgresql>/5.4/dbupdate-5.3.0-to-5.4.0.sql`

If the installation is using the DFS cluster, import the cluster database changes provided in
`/<ezp5-root>/ezpublish_legacy/update/database/mysql/5.4/dbupdate-cluster-5.3.0-to-5.4.0.sql`

Update the database/repository configuration

`ezpublish.system.<siteAccessName>.database` has been removed for defining database settings. You now need to:

- **Define a Doctrine connection**

  MySQL settings : ezpublish.yml or config.yml

  ```
  doctrine:
      dbal:
          connections:
              my_connection:
                  driver:   pdo_mysql
                  host:     localhost
                  port:     3306
                  dbname:   my_database
                  user:     my_user
                  password: my_password
                  charset:  UTF8
  ```

PostGreSQL : ezpublish.yml or config.yml

```
doctrine:
    dbal:
        connections:
            my_connection:
                driver:   pdo_pgsql
                host:     localhost
                port:     5432
                dbname:   my_database
                user:     my_user
                password: my_password
                charset:  UTF8
```

source

Pro Tip
Set your base DB params in your `parameters.yml/parameters.yml.dist` and refer them here.

parameters.yml

```
parameters:
    database_driver: pdo_mysql
    database_host: localhost
    database_port: 3306
    database_name: ezdemo
    database_user: my_user
    database_password: my_password
    database_charset: UTF8
```

ezpublish.yml / config.yml

```
doctrine:
    dbal:
        connections:
            my_connection:
                driver:   %database_driver%
                host:     %database_host%
                port:     %database_port%
                dbname:   %database_name%
                user:     %database_user%
                password: %database_password%
                charset:  %database_charset%
```

- **Define one or several repositories**

ezpublish.yml

```
ezpublish:
    repositories:
        main: { engine: legacy, connection: my_connection }
```

- **(Optional) Make your SiteAccess config point to the right repository**

ezpublish.yml   ❯ Expand

```
ezpublish:                                                                             source
    system:
        my_siteaccess_group:
            repository: main
```

Remove the old connection information
Note : to benefit from the new configuration, don't forget to remove the old configuration

Old database access to remove

```
ezpublish:
    system:
        my_siteaccess_group:
            database:
                type: mysql
                user: my_user
                password: my_password
                server: localhost
                database_name: ezdemo
```

Step 4: Apply 5.4 configuration changes

YAML files

Since default configuration files have been overwritten during step one, the few additions to those files that were made in 5.2 need to be applied manually to the configuration files.
All of those changes are **additions**, none of them replaces what you already have.

For most of them, at least one, if not all hierarchy elements (monolog, handler, framework, router...) will already be there. All you have to do is **add the missing bits in the existing configuration block**s.

In **ezpublish/config/config.yml**, you need to add a few default values for the framework

ezpublish/config/config.yml

trusted_hosts has been changed and *session.handler_id* is added to the *framework* block making Symfony pick what has been configured in PHP instead.

```
framework:
    trusted_hosts:    ~

    session:
        handler_id:  ~
```

Additionally configuration for *swift_mailer* has been added:

```
swiftmailer:
    transport: "%mailer_transport%"
    host:      "%mailer_host%"
    username:  "%mailer_user%"
    password:  "%mailer_password%"
    spool:     { type: memory }
```

ezpublish/config/config_dev.yml

In framework, set *router.strict_requirements* to true.

```
framework:
    router:
        resource: "%kernel.root_dir%/config/routing_dev.yml"
        strict_requirements: true
```

ezpublish/config/config_prod.yml

Remove *framework.router.strict_requirements*.

ezpublish/config/routing.yml

Add **new login routes,** _ezpublishRestOptionsRouter route loader, and and _liip_imagine:

```
_ezpublishRestOptionsRoutes:
    resource: "@EzPublishRestBundle/Resources/config/routing.yml"
    prefix: %ezpublish_rest.path_prefix%
    type: rest_options
_liip_imagine:
    resource: "@LiipImagineBundle/Resources/config/routing.xml"
login:
    path:   /login
    defaults:  { _controller: ezpublish.security.controller:loginAction }
login_check:
    path:   /login_check
logout:
    path:   /logout
```

ezpublish/config/security.yml

Under ezpublish_front firewall, update to fit the following (**be sure to remove ezpublish: true**):

```
security:
    firewalls:
        ezpublish_front:
            pattern: ^/
            anonymous: ~
            form_login:
                require_previous_session: false
            logout: ~
```

ezpublish/config/parameters.yml.dist

If you have added anything to `parameters.yml`, we suggest that you add your custom settings to `parameters.yml.dist`, so that the composer post-update script handles those, and generates their values correctly.

ezpublish_legacy.default.view_default_layout can be removed from this file, and the following added:

```
parameters:
    mailer_transport:  smtp
    mailer_host:       127.0.0.1
    mailer_user:       ~
    mailer_password:   ~
```

composer will ask for your own values when you run *composer update*.

ezpublish/config/ezpublish*.yml

In all ezpublish.yml files (prod, dev, etc), replace the "**handlers**" key in *stash.caches.** with "**drivers**".

```
stash:
    caches:
        default:
            drivers:
```

`ezpublish.system.<siteAccessName>.session_name` has been deprecated for defining session name. You now need to use `ezpublish.system.<siteAccessName>.session.name`.

Before:

```
ezpublish:
    system:
        my_siteaccess:
            session_name: SomeSessionName
```

After:

```
ezpublish:
    system:
        my_siteaccess:
            session:
                name: SomeSessionName
```

Templates

In your templates, change your links pointing to `/user/login` and `/user/logout` to appropriate `login` / `login_check` / `logout` routes:

> ### Before
>
> ```
> <a href="{{ path( 'ez_legacy', {'module_uri': '/user/login'} ) }}">Login</a>
> <form action="{{ path( 'ez_legacy', {'module_uri': '/user/login'} ) }}" method="post">
> <a href="{{ path( 'ez_legacy', {'module_uri': '/user/logout'} ) }}">Logout</a>
> ```

> ### After
>
> ```
> <a href="{{ path( 'login' ) }}">Login</a>
> <form action="{{ path( 'login_check' ) }}" method="post">
> <a href="{{ path( 'logout' ) }}">Logout</a>
> ```

ezpublish/EzPublishKernel.php

It is not possible to just copy your old `EzPublishKernel.php` file over from the previous installation, since quite a few changes were made to this file in this release. We suggest that you simply reflect in the new kernel file any changes you made in the previous version.

composer.json

If you had modified `composer.json` to add your own requirements, you must re-apply those changes to the new version.

Varnish (if applicable)

The recommended varnish (3 and 4) VCL configuration can now be found in the `doc/varnish` folder. See also the Using Varnish page.

Anonymous state of a user is not checked through presence of `is_logged_in` cookie any more. Therefore, when using Varnish, you must change the following in your VCL file:

> ### Before
>
> ```
> # ez_user_hash sub-routine
> if (req.http.Cookie !~ "is_logged_in=" ) {
>     # User don't have "is_logged_in" cookie => Set a hardcoded anonymous hash
>     set req.http.X-User-Hash = "38015b703d82206ebc01d17a39c727e5";
> }
> ```

After

```
# ez_user_hash sub-routine
if (req.http.Cookie !~ "eZSESSID" ) {
    # User don't have session cookie => Set a hardcoded anonymous hash
    set req.http.X-User-Hash = "38015b703d82206ebc01d17a39c727e5";
}
```

Update the DFS cluster configuration

If your legacy installation uses DFS clustering, you need to create the new stack configuration for it.

Web server configuration

The officially recommended virtual configuration is now shipped in the `doc` folder, for both apache2 (`doc/apache2`) and nginx (`doc/nginx`). Both h are built to be easy to understand and use, but aren't meant as drop-in replacements for your existing configuration.

One notable change is that `SetEnvIf` is now used to dynamically change rewrite rules depending on the Symfony environment. It is currently used for the assetic production rewrite rules.

Read more on the Web servers page.

Step 5: Run composer update

Run `composer update --no-dev --prefer-dist` to get the latest eZ Publish dependencies.

At the end of the process, you will be asked for values for the parameters previously added to `parameters.yml.dist`.

Step 6: Regenerate the autoload array for extensions

To regenerate the autoload array, execute the following script from the root of your eZ Publish Legacy directory:

```
cd ezpublish_legacy
php bin/php/ezpgenerateautoloads.php --extension
```

Step 7: Link assets

Assets from the various bundles need to be made available for the webserver through the web/ document root.

The following commands will first symlink eZ Publish 5 assets in "Bundles" and the second will symlink assets (design files like images, scripts and css, and files in var folder) from eZ Publish Legacy:

```
php ezpublish/console assets:install --symlink
php ezpublish/console ezpublish:legacy:assets_install --symlink
php ezpublish/console assetic:dump --env=prod
```

Step 8: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell: Navigate into the new eZ Publish directory.Run the script using the following shell command:cd /<ezp5-root>/ezpublish_legacy/php bin/php/ezcache.php --clear-all --purgePurging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.
Note: Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files

have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 9: Upgrade Extensions (site package)

Next, depending on if you originally installed eZ Flow, eZ Webin or eZ Demo site, follow the steps mentioned in the eZ Webin, eZ Flow or eZ Demo upgrade documentation.

Minor Versions Update Warning

Please have in mind that, after the upgrade to a major version of eZ Publish, you will need to perform regular updates with composer.

Before any minor update with composer, please execute all require steps described in the 5.4.x Update Instructions.