Updating eZ Platform

This page explains how to update between minor or patch versions of eZ Platform.

> If you are looking for information on how to upgrade to eZ Platform from eZ Publish (4.x or 5.x), take look at the Upgrade doc.

> This instruction reflects a proposed git-based workflow for handling updates; feedback on how this works in practice and input on how to further improve/simplify it is welcome.

In the instructions below, replace `<version>` with the version of eZ Platform you are updating to (for example: `v1.2.0`).

If you are testing a release candidate, use the latest rc tag (for example: `v1.2.1-rc1`).

Update procedure

Existing projects can easily be updated using Composer. From the project's root, create a new branch from the project's master, or from the branch you're upgrading on:

From your master branch

```
git checkout -b <branch_name>
```

If it's not there, add `ezsystems/ezplatform` as an upstream remote:

From your new update branch

```
git remote add ezplatform http://github.com/ezsystems/ezplatform.git
```

Then pull the tag into your branch:

From your new update branch

```
git pull ezplatform <version>
```

You will get conflicts, and it is perfectly normal. The most common ones will be on `composer.json` and `composer.lock`.
The latter can be ignored, as it will be regenerated when we execute composer update later. The easiest is to checkout the version from the tag, and add it to the changes:

> If you get a **lot** of conflicts (on the `doc` folder for instance), and eZ Platform was installed from the share.ez.no tarball, it might be because of incomplete history. You will have to run `git fetch ezplatform --unshallow` to load the full history, and run the merge again.

From your new update branch

```
git checkout --theirs composer.lock && git add composer.lock
```

You may also run `git remove composer.lock` if you do not keep a copy of it in the branch.

Merging composer.json

Manual merging

Conflicts in `composer.json` need to be fixed manually. If you're not familiar with the diff output, you may checkout the tag's version, and inspect the changes. It should be readable for most:

From your new update branch

```
git checkout --theirs composer.json && git diff composer.json
```

You should see what was changed, as compared to your own version, in the diff output. The ezplatform update changes the requirements for all of the `ezsystems/` packages. Those changes should be left untouched. All of the other changes will be removals of what you added for your own project. Use `git checkout -p` to selectively cancel those changes:

```
git checkout -p composer.json
```

Answer no (do not discard) to the requirement changes of `ezsystems` dependencies. Answer `yes` (discard) to removals of your changes.

Once you are done, inspect the file, either using an editor or by running `git diff composer.json`. You may also test the file's sanity with `composer validate`, and test the dependencies by running `composer update --dry-run`. (will output what it would do to dependencies, without applying the changes.

Once finished, run `git add composer.json.`

Fixing other conflicts (if any)

Depending on the local changes you have done, you may get other conflicts: configuration files, kernel...

There shouldn't be many, and you should be able to figure out which value is the right one for all of them:

- Edit the file, and identify the conflicting changes. If a setting you have modified has also been changed by us, you should be able to figure out which value is the right one.
- Run `git add conflicting-file` to add the changes

Updating composer.lock

At this point, you should have a composer.json file with the correct requirements. Run `composer update` to update the dependencies.

```
composer update --with-dependencies ezsystems/ezpublish-kernel
ezsystems/platform-ui-bundle ezsystems/behatbundle
```

In order to restrict the possibility of unforeseen updates of 3rd party packages, we recommend by default that `composer update` is restricted to the list of packages we have tested the update for. You may remove this restriction, but be aware that you might get a package combination we have not tested.

When updating from 15.12.1 or earlier to 16.02 or later

On PHP conflict
Because from release 16.02 onwards eZ Platform is compatible only with PHP 5.5 and higher, the update command above will fail if you use an older PHP version. Please update PHP to proceed.

Database update

The 16.02 release required an update to the database. Import `vendor/ezsystems/ezpublish-kernel/data/update/mysql/dbupdate-6.1.0-to-6.2.0.sql` into your database:

```
mysql -u<username> -p<password> <database_name> <
vendor/ezsystems/ezpublish-kernel/data/update/mysql/dbupdate-6.1.0-to-6.2.0.sql
```

Dump assets

The web assets must be dumped again for the prod environment:

```
php app/console assetic:dump --env=prod web
```

Commit, test and merge

Once all the conflicts have been resolved, and `composer.lock` updated, the merge can be committed. Note that you may or may not keep `composer.lock`, depending on your version management workflow. If you do not wish to keep it, run `git reset HEAD <file>` to remove it from the changes. Run `git commit`, and adapt the message if necessary. You can now test the project, run integration tests... once the upgrade has been approved, go back to `master`, and merge your new update branch:

```
git checkout master
git merge <branch_name>
```