

Updating eZ Studio

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

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 Studio you are updating to (for example: `v1.2.0`).

Update procedure

When updating from 16.02 or earlier to 16.04 or later

Starting from 16.04, the main meta repository was split in two:

- **ezstudio** with clean installer
- **ezstudio-demo** with demo content and design

Before pulling the latest version you have to change your git remote repository. There are two ways to do this:

1. The quick way: simply replace repository name in config file for example using sed (this way is useful if you added any remotes):

```
sed -ie 's/ezsystems\/ezstudio/ezsystems\/ezstudio-demo/g' .git/config
```

2. Using git remote:

- list your remote repositories:

```
git remote -v
```

- copy the url containing ezsystems/ezstudio
- replace the url using git:

```
git remote set-url origin https://github.com/ezsystems/ezstudio-demo.git
```

Next, pull the latest changes:

```
git pull
```

Now, manually remove the contents of the `app/cache/` folder:

```
rm -rf app/cache/*
```

You can easily update your existing Studio project 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/ezstudio-demo` as an upstream remote:

From your new update branch

```
git remote add ezstudio http://github.com/ezsystems/ezstudio-demo.git
```

Then pull the tag into your branch:

From your new update branch

```
git pull ezstudio <version>
```

When updating from 15.12.1 or earlier to 16.02 or later

If you had custom siteaccesses or languages set up in your installation, a conflict may appear here. This is because new siteaccesses are introduced in Studio demo in this release: "fr" for French, "de" for German and "no" for Norwegian.

To avoid overriding your siteaccesses with the new ones, you need to accept your own changes in the `app/config/ezplatform.yml` file.

If you have no custom siteaccesses or languages and no conflict occurs, you can do one of the following things:

A) Add languages and permissions to use the newly introduced siteaccesses from the demo.

To do this, log in to the application and go to the Admin Panel.

Choose Languages and click Create a new language. Create a language for each of the new siteaccesses.

Then, click **Roles** and select the Anonymous Role. Click Edit limitations next to the following function:

user	login	SiteAccess(1766001124)	
------	-------	--------------------------	---

Select all available siteaccesses and click Save.

B) Remove the new siteaccesses.

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 Studio was installed from the [share.ez.no](#) tarball, it might be because of incomplete history. You will have to run `git fetch ezstudio --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
```

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 HEAD composer.json
```

You should see what was changed, as compared to your own version, in the diff output. This update changes the requirements for all of the `ezsystem/` 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

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/repository-forms eZsystems/studio-ui-bundle
eZsystems/eZstudio-demo-bundle eZsystems/landing-page-fieldtype-bundle
eZsystems/flex-workflow
```

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 this release onwards eZ Studio 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 requires 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 -p -u <database_user> <database_name> <
vendor/eZsystems/ezpublish-kernel/data/update/mysql/dbupdate-6.1.0-to-6.2.0.sql
```

To enable the new Flex workflow notification feature, import the following file:

```
mysql -p -u <database_user> <database_name> <
vendor/ezsystems/ezstudio-notifications/bundle/Resources/install/ezstudio-notificat
ions.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 update branch:

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