



Please wait while we install a system update

# Updating Neos – Why, When and How

Karsten Dambekalns



# Karsten Dambekalns

Developer and consultant at Flownative GmbH

Working on & with Neos since 2005... for 17 years...

More than a third of my life! 🤨



# Why, When and How

## Why?

Because you get new features and bug fixes.

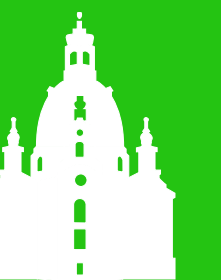
## When?

Whenever a new version is released.

## How?

Read & follow instructions, test & deploy.

Done!

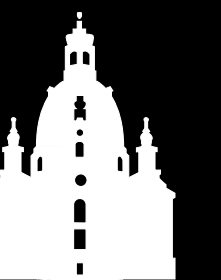




# Thanks for listening

Any questions?

Just kidding...



# Why should you update?

You benefit from fewer bugs, new features and better performance most of the time

Outdated versions will no longer receive support

Security fixes will no longer be done on old releases





# Features to gain...

## Greatly speed up node move actions

Due to an issue in how Doctrine 2. entities to commit) it recomputed  $n^2$  change computations.

## Atomic Fusion added to the core

Two years ago, Wilhelm and Martin introduced their JSX inspired packages around "Atomic Fusion" in a [blog post](#)

## Node Presets

Many developers are familiar with the [Sitegeist.Silhouettes](#) package that allows integrators to specify presets that can be used across multiple node type definitions. This feature is now part of the core functionality, and it can be used to centralize recurring configuration:

## Fusion m

This change ad  
used just to re

The idea is based on the new [match m](#)

```
1 rowClass = Neos.Fusion:Match {  
2   @subject = ${q(node).property('co  
3  
4  
5  
6  
7 }
```

## PSR 18 Http\Client

PSR 18 specifies a http client needs. With Flow 8 we support PSR 18 in the `Neos\Flow\Http\Browser` class and ensured that it can be injected as dependency. You can also inject the `Psr\Http\Client\Interfaces` directly into your code.

## Lazy loading images

Image tags rendered with the `Neos.Neos:ImageTag` fusion prototype (or the `ImageViewHelper` from the `Neos.Media` package) will now be rendered with the loading attribute set to "lazy" by default leveraging modern browsers' power to defer loading of images until they are visible.

Read more about the loading attribute [on MDN](#).

meantime and the [Neos.Demo](#) or duplicate possibilities of AFX. and we added a `backend`, version 7 marks an m.

be flushed at for duplicate and we added a izations for the ster publishing resulted in 3 times

us to low 8 ts more

also

in Flow 8 we also added support for





# Bugs to get rid of...

## 5.3 and up

- BUGFIX: Render labels of authentication providers <sup>1</sup>

## 5.3 and up

- BUGFIX: Exclude “/sites”-node from dimension-migration <sup>3</sup>
- BUGFIX: prevent failure for disabled superTypes <sup>5</sup> - a special thanks to Dominik Jansen for your first contribution - thank you so much! ❤️
- BUGFIX: Prevent disconnected Nodes from being migrated <sup>5</sup>

## 5.3 and up contain

- BUGFIX: Only emit thumbnailCreated when successful <sup>3</sup>
- BUGFIX: Persist Thumbnail using DBAL to avoid unique constraint errors <sup>7</sup>
- BUGFIX: Check for the native PHP extension instead of the ImagineService, since ImageService is not used further in this generator. <sup>2</sup>
- BUGFIX: backend message partials not found <sup>1</sup>

## 7.0

## 7.0 and up also contain

- TASK: Address PHP 8 deprecation warnings <sup>1</sup>
- BUGFIX: Remove unused formatting options <sup>1</sup>

## 7.2

## 7.1 and up also contain

- TASK: Allow Neos.Fusion:Match to return a default value with a type other than string <sup>3</sup>

## 7.3 also contains

- BUGFIX: Allow adding first asset collection again <sup>5</sup>

-n-drop upload of assets in media browser <sup>1</sup>

copying nodes from another dimension, you are only allowed to re-use the dimension

variable to search for identifier patterns by

-development-collection/pull/3429 <sup>4</sup>

anually by @jonnitto in <https://github.com>

BUGFIX: Don't skip childnodes when discarding nodes <sup>1</sup>

BUGFIX: Prevent repeated queries for non existing nodes

new shortcut nodes to "parent" by

ent node in breadcrumb by @jonnitto in #3450

on-migration by @Nikdro in #3110

variable to search for identifier patterns by

anually by @jonnitto in #3526

third parameter in sort operation by @jonnitto in

reValidation from ContentRepository context" by

ion tag by @naase-talman in #3541 <sup>2</sup>





# SECURITY



DRESDEN 2022

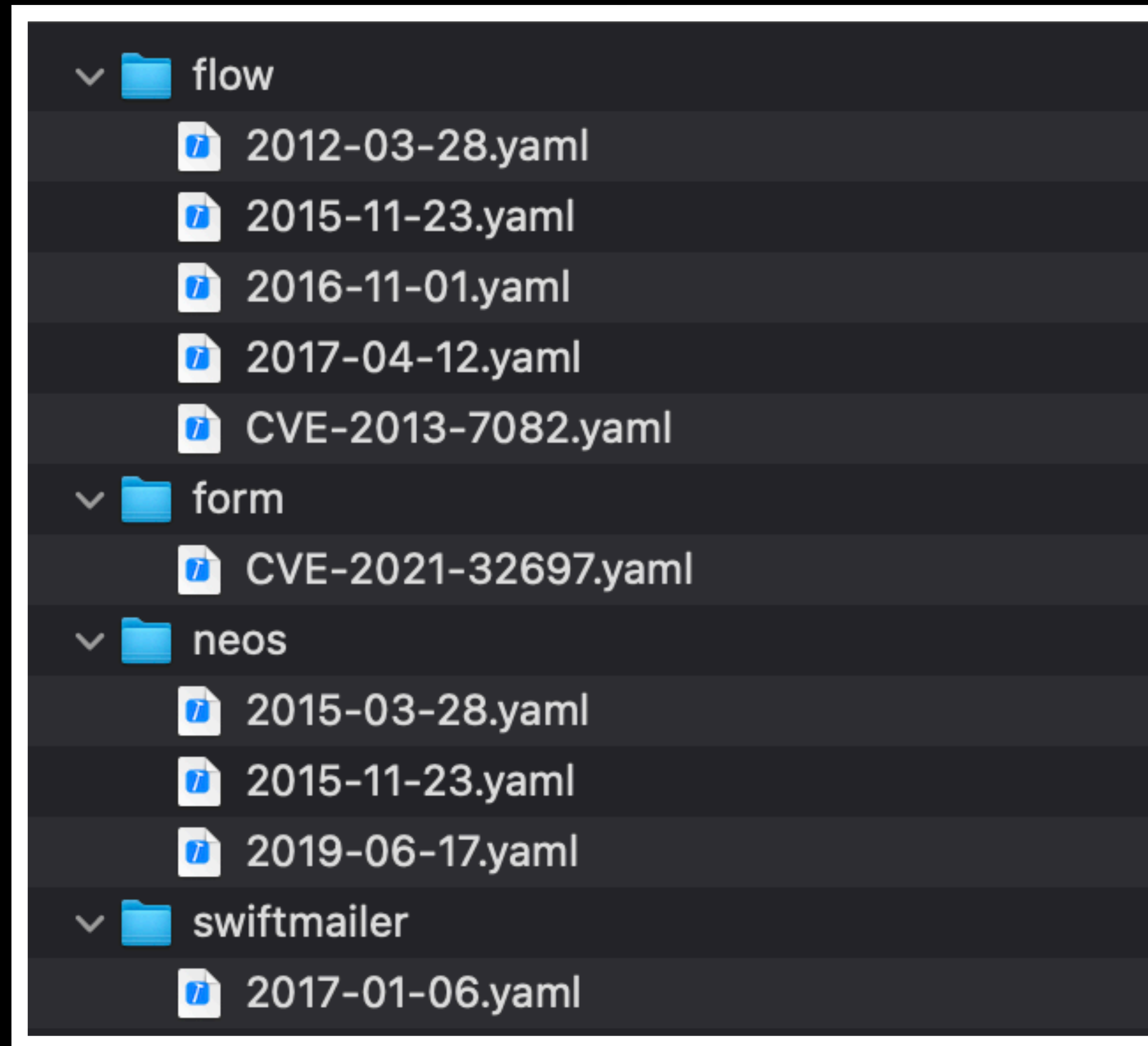


DRESDEN 2022









<https://github.com/FriendsOfPHP/security-advisories/tree/master/neos>



DRESDEN 2022



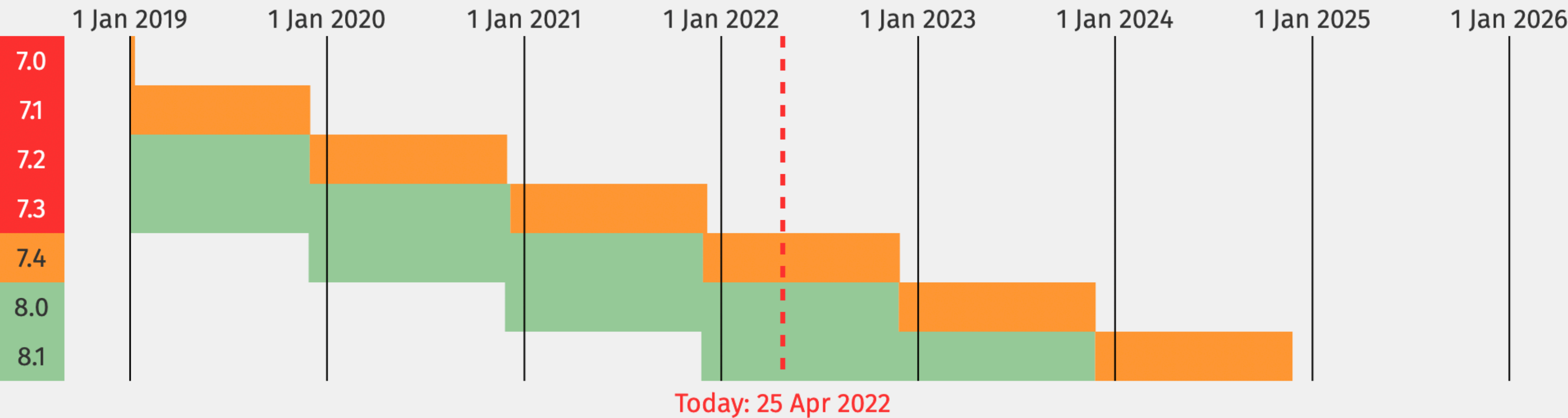
# Other tools in your stack



# Currently Supported Versions

Branch	Initial Release		Active Support Until		Security Support Until	
<a href="#">7.4</a>	28 Nov 2019	2 years, 4 months ago	28 Nov 2021	4 months ago	28 Nov 2022	in 7 months
<a href="#">8.0</a>	26 Nov 2020	1 year, 4 months ago	26 Nov 2022	in 7 months	26 Nov 2023	in 1 year, 7 months
<a href="#">8.1</a>	25 Nov 2021	5 months ago	25 Nov 2023	in 1 year, 6 months	25 Nov 2024	in 2 years, 6 months

Or, visualised as a calendar:



## Key

Active support	A release that is being actively supported. Reported bugs and security issues are fixed and regular point releases are made.
Security fixes only	A release that is supported for critical security issues only. Releases are only made on an as-needed basis.
End of life	A release that is no longer supported. Users of this release should upgrade as soon as possible, as they may be exposed to unpatched security vulnerabilities.



Elasticsearch	EOL date	Maintained until
5.6.x	2019-03-11	7.0.0
6.0.x	2019-05-14	6.1.0
6.1.x	2019-06-12	6.2.0
6.7.x	2020-09-26	6.8.0
6.8.x	2022-02-10	8.0.0
7.0.x	2020-10-10	7.1.0
7.1.x	2020-11-20	7.2.0
7.16.x	2023-06-07	7.17.0
7.17.x	2023-08-01	9.0.0
8.0.x	2023-08-10	8.1.0
8.1.x	2023-09-08	8.2.0

# When should you update?

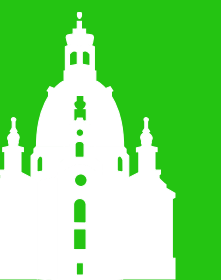
As often and as fast as possible!

Updating often keeps the routine drilled...

Staying up to date keeps the steps to take smaller!

Smaller steps mean lower risk when updating!

*An update a day, keeps  
the downtime away*





# But... we updated once, and it broke our site!

As mentioned, in case something goes wrong:

- do test your updates on a staging instance
- have a current backup
- have a rollback-strategy

If things still break on production deployment, the cause is almost never the new version...



# But... we don't need new stuff and things work fine

That is the „our small site works well“ reasoning, which is fine until:

- Your hosting provider emails you to inform you about the EOL of PHP 5.6, so your Neos 1.0 site will stop working
- Some security issue requires you to update PHP
- Related technology must be updated
- New features **must** be added (GD 2, ...)
- One wants to work with such old code



# But... we run a mission-critical site!

Some sites cannot afford a downtime caused by a bug introduced by an update

But of course you don't deploy the update to your mission-critical site untested, do you?

And you do have a backup and a rollback strategy in place, don't you?

If you can't have a staging instance for testing, your mission can't be that critical...



# But... we only use LTS versions!

Sometimes people claim they must only use LTS versions, because they are more reliable or receive better support

## **Reliability**

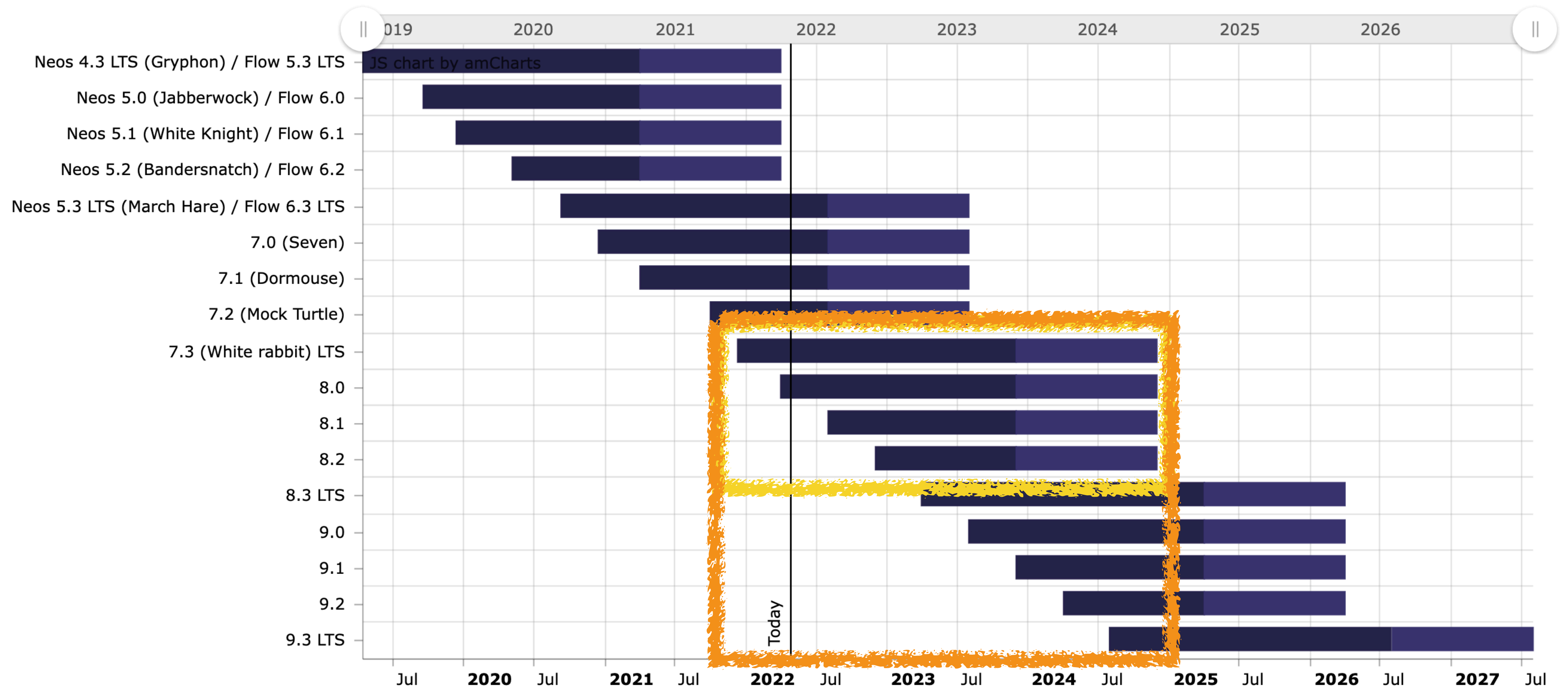
Granted, new features may introduce bugs. But all supported branches receive all fixes – due to how we manage the source code

## **Support lifetime**

Any versions „after“ an LTS receive support for as long as the LTS version – due to how we manage the source code



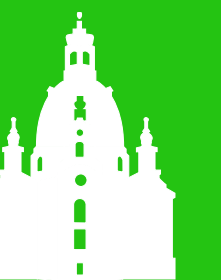




# How should you update?

The generic recipe is:

- read release notes
- adjust composer.json, if needed
- do a dry-run of composer update, check results
- do composer update, commit result
- run core migrations, check result, adjust as needed
- run database migrations
- test, test again, commit, deploy to staging, test, deploy to production



# Types of releases



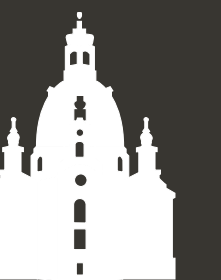
# Patch-level releases

Must never be breaking, only contain bugfixes

Might add new issues due to a buggy bugfix, happens rarely, though

Usually about an hour of work

**x.y.z+1**





# Minor releases

Add features and contain all the features and fixes of the previous minor release(s)

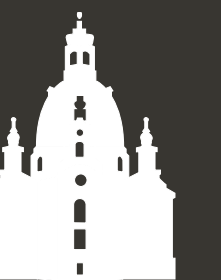
Must never be breaking, but new features may still have bugs (more probably in a x.y.0 release)

Could take longer. If new features need to be enabled or configured, even a few hours, depending on the complexity of your site

Might need a database migration and/or new configuration

A bit more testing than for a patch-level release is advisable

x.y+1.z



# Major releases

Add features, contain features and fixes of the previous minor release(s) and may have breaking changes

Probably need adjustments to your site's code and new (major) versions of used extensions

Probably need a database migration and configuration changes

Could require updated infrastructure (PHP or Elasticsearch version, ...)

Needs the most testing of all types of releases

**x+1.y.z**



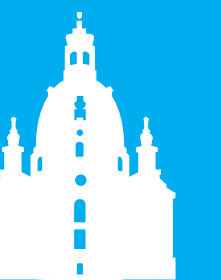
# Approaching an update

Updating to a new version can always be done “in one go”, you do not need to upgrade to each version in between

But you always need to read the release notes of the full range of versions you cover in your update

Check your dependencies, if needed adjust to allow newer packages

Collect a list of (breaking) changes that need attention



# Packages needing a dependency adjustment

Sometimes you must raise the allowed version of dependencies, e.g.

- flownative/google-cloudstorage 5.2 vs 5.3
- psmb/splitadd 0.3 vs 0.4

Usually not a problem, if you use constraints like ^5.2 – then 5.3 is allowed, too. And patch-level releases should always be allowed...

It's different for major versions (no surprise) and for 0.x versions!

Hint: use <https://semver.madewithlove.com/> to check constraints!





```
$ composer update
```

```
Loading composer repositories with package information
```

```
Updating dependencies
```

```
Your requirements could not be resolved to an installable set of packages.
```

#### Problem 1

- neos/flow[4.3.0, ..., 4.3.x-dev] require php ~7.0 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[5.0.0, ..., 5.2.x-dev] require php ~7.1 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[5.3.0, ..., 5.3.x-dev] require php ^7.1 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[6.0.0, ..., 6.3.x-dev] require php ^7.2 -> your php version (8.0.17) does not satisfy that requirement.
- neos/neos[7.0.0, ..., 7.1.0] require php ^7.3 -> your php version (8.0.17) does not satisfy that requirement.
- Root composer.json requires acmecon/site \* -> satisfiable by acmecon/site[dev-main].
- Conclusion: don't install neos/neos 8.0.1 (conflict analysis result)
- acmecon/site dev-main requires psmb/splitadd ^0.3 -> satisfiable by psmb/splitadd[0.3].
- neos/neos[3.3.0, ..., 3.3.x-dev] require neos/flow ~4.3.0 -> satisfiable by neos/flow[4.3.0, ..., 4.3.x-dev].
- neos/neos[4.0.0, ..., 4.0.x-dev] require neos/flow ~5.0.0 -> satisfiable by neos/flow[5.0.0, ..., 5.0.x-dev].
- neos/neos[4.1.0, ..., 4.1.x-dev] require neos/flow ~5.1.0 -> satisfiable by neos/flow[5.1.0, ..., 5.1.x-dev].
- neos/neos[4.2.0, ..., 4.2.x-dev] require neos/flow ~5.2.0 -> satisfiable by neos/flow[5.2.0, ..., 5.2.x-dev].
- neos/neos[4.3.0, ..., 4.3.x-dev] require neos/flow ~5.3.0 -> satisfiable by neos/flow[5.3.0, ..., 5.3.x-dev].
- neos/neos[5.0.0, ..., 5.0.x-dev] require neos/flow ~6.0.0 -> satisfiable by neos/flow[6.0.0, ..., 6.0.x-dev].
- neos/neos[5.1.0, ..., 5.1.x-dev] require neos/flow ~6.1.0 -> satisfiable by neos/flow[6.1.0, ..., 6.1.x-dev].
- neos/neos[5.2.0, ..., 5.2.x-dev] require neos/flow ~6.2.0 -> satisfiable by neos/flow[6.2.0, ..., 6.2.x-dev].
- neos/neos[5.3.0, ..., 5.3.x-dev] require neos/flow ~6.3.0 -> satisfiable by neos/flow[6.3.0, ..., 6.3.x-dev].
- psmb/splitadd 0.3 requires neos/neos ^7.0 || ^5.0 || ^4.0 || ^3.3 -> satisfiable by neos/neos[3.3.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev].
- Only one of these can be installed: typo3/neos[dev-master, 1.0.0-beta1, ..., 1.2.x-dev, 2.0.0-beta1, ..., 2.3.x-dev, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev], neos/neos[dev-master, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev]. neos/neos replaces typo3/neos and thus cannot coexist with it.
- acmecon/site dev-main requires neos/neos ^8.0 -> satisfiable by neos/neos[8.0.0, 8.0.1, 8.0.x-dev].
- Conclusion: don't install neos/neos 8.0.0 (conflict analysis result)

Use the option `--with-all-dependencies (-W)` to allow upgrades, downgrades and removals for packages currently locked to specific versions.



```
$ composer update
```

```
Loading composer repositories with package information
```

```
Updating dependencies
```

```
Your requirements could not be resolved to an installable set of packages.
```

```
Problem 1
```

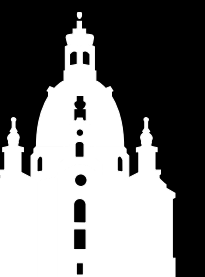
- neos/flow[4.3.0, ..., 4.3.x-dev] require php ~7.0 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[5.0.0, ..., 5.2.x-dev] require php ~7.1 -> your php version (8.0.17) does not satisfy that requirement.

# Your requirements could not be resolved to an installable set of packages.

## Problem 1

- neos/neos[5.2.0, ..., 5.2.x-dev] require neos/flow ~6.2.0 -> satisfiable by neos/flow[6.2.0, ..., 6.2.x-dev].
- neos/neos[5.3.0, ..., 5.3.x-dev] require neos/flow ~6.3.0 -> satisfiable by neos/flow[6.3.0, ..., 6.3.x-dev].
- psmb/splitadd 0.3 requires neos/neos ^7.0 || ^5.0 || ^4.0 || ^3.3 -> satisfiable by neos/neos[3.3.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev].
- Only one of these can be installed: typo3/neos[dev-master, 1.0.0-beta1, ..., 1.2.x-dev, 2.0.0-beta1, ..., 2.3.x-dev, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev], neos/neos[dev-master, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev]. neos/neos replaces typo3/neos and thus cannot coexist with it.
- acmecom/site dev-main requires neos/neos ^8.0 -> satisfiable by neos/neos[8.0.0, 8.0.1, 8.0.x-dev].
- Conclusion: don't install neos/neos 8.0.0 (conflict analysis result)

Use the option `--with-all-dependencies (-W)` to allow upgrades, downgrades and removals for packages currently locked to specific versions.





```
$ composer update
```

```
Loading composer repositories with package information
```

```
Updating dependencies
```

```
Your requirements could not be resolved to an installable set of packages.
```

#### Problem 1

```
- neos/flow[4.3.0, ..., 4.3.x-dev] require php ~7.0 -> your php version (8.0.17) does not satisfy that requirement.  
- neos/flow[5.0.0, ..., 5.2.x-dev] require php ~7.1 -> your php version (8.0.17) does not satisfy that requirement.  
- neos/flow[5.3.0, ..., 5.3.x-dev] require php ^7.1 -> your php version (8.0.17) does not satisfy that requirement.  
- neos/flow[6.0.0, ..., 6.3.x-dev] require php ^7.2 -> your php version (8.0.17) does not satisfy that requirement.  
- neos/neos[7.0.0, ..., 7.1.0] require php ^7.3 -> your php version (8.0.17) does not satisfy that requirement.
```

```
- Root composer.json requires acmecon/site * -> satisfiable by acmecon/site[dev-main].
```

```
Conclusion: don't install neos/neos 8.0.0 (conflict analysis result)
```

**- neos/flow[4.3.0, ..., 4.3.x-dev] require php ~7.0 -> your php version (8.0.17) does not satisfy that requirement.**

```
- psmb/splitadd 0.3 requires neos/neos ^7.0 || ^5.0 || ^4.0 || ^3.3 -> satisfiable by neos/neos[3.3.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev].
```

```
- Only one of these can be installed: typo3/neos[dev-master, 1.0.0-beta1, ..., 1.2.x-dev, 2.0.0-beta1, ..., 2.3.x-dev, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev], neos/neos[dev-master, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev]. neos/neos replaces typo3/neos and thus cannot coexist with it.
```

```
- acmecon/site dev-main requires neos/neos ^8.0 -> satisfiable by neos/neos[8.0.0, 8.0.1, 8.0.x-dev].
```

```
- Conclusion: don't install neos/neos 8.0.0 (conflict analysis result)
```

Use the option `--with-all-dependencies (-W)` to allow upgrades, downgrades and removals for packages currently locked to specific versions.



```
$ composer update
```

```
Loading composer repositories with package information
```

```
Updating dependencies
```

```
Your requirements could not be resolved to an installable set of packages.
```

```
Problem 1
```

- neos/flow[4.3.0, ..., 4.3.x-dev] require php ~7.0 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[5.0.0, ..., 5.2.x-dev] require php ~7.1 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[5.3.0, ..., 5.3.x-dev] require php ^7.1 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[6.0.0, ..., 6.3.x-dev] require php ^7.2 -> your php version (8.0.17) does not satisfy that requirement.

~~neos/neos[7.0.0, ..., 7.1.0] require php ^7.3 -> your php version (8.0.17) does not satisfy that requirement.~~

- Root composer.json requires acmecon/site \* -> satisfiable by acmecon/site[dev-main].

~~- Conclusion: don't install neos/neos 8.0.1 (conflict analysis result)~~

Root composer.json requires acmecon/site \*  
-> satisfiable by acmecon/site[dev-main].

- neos/neos[5.1.0, ..., 5.1.x-dev] require neos/flow ~6.1.0 -> satisfiable by neos/flow[6.1.0, ..., 6.1.x-dev].

- neos/neos[5.2.0, ..., 5.2.x-dev] require neos/flow ~6.2.0 -> satisfiable by neos/flow[6.2.0, ..., 6.2.x-dev].

- neos/neos[5.3.0, ..., 5.3.x-dev] require neos/flow ~6.3.0 -> satisfiable by neos/flow[6.3.0, ..., 6.3.x-dev].

- psmb/splitadd 0.3 requires neos/neos ^7.0 || ^5.0 || ^4.0 || ^3.3 -> satisfiable by neos/neos[3.3.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev].

- Only one of these can be installed: typo3/neos[dev-master, 1.0.0-beta1, ..., 1.2.x-dev, 2.0.0-beta1, ..., 2.3.x-dev, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev], neos/neos[dev-master, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev]. neos/neos replaces typo3/neos and thus cannot coexist with it.

- acmecon/site dev-main requires neos/neos ^8.0 -> satisfiable by neos/neos[8.0.0, 8.0.1, 8.0.x-dev].

- Conclusion: don't install neos/neos 8.0.0 (conflict analysis result)

Use the option --with-all-dependencies (-W) to allow upgrades, downgrades and removals for packages currently locked to specific versions.





```
$ composer update
```

```
Loading composer repositories with package information
```

```
Updating dependencies
```

```
Your requirements could not be resolved to an installable set of packages.
```

```
Problem 1
```

- neos/flow[4.3.0, ..., 4.3.x-dev] require php ~7.0 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[5.0.0, ..., 5.2.x-dev] require php ~7.1 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[5.3.0, ..., 5.3.x-dev] require php ^7.1 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[6.0.0, ..., 6.3.x-dev] require php ^7.2 -> your php version (8.0.17) does not satisfy that requirement.
- neos/neos[7.0.0, ..., 7.1.0] require php ^7.3 -> your php version (8.0.17) does not satisfy that requirement.
- Root composer.json requires acmecon/site \* -> satisfiable by acmecon/site[dev-main].
- Conclusion: don't install neos/neos 8.0.1 (conflict analysis result)
- acmecon/site dev-main requires psmb/splitadd ^0.3 -> satisfiable by psmb/splitadd[0.3].
- neos/neos[3.3.0, ..., 3.3.x-dev] require neos/flow ~4.3.0 -> satisfiable by neos/flow[4.3.0, ..., 4.3.x-dev].

```
- acmecon/site dev-main requires neos/neos ^8.0 ->
satisfiable by neos/neos[8.0.0, 8.0.1, 8.0.x-dev].
- Conclusion: don't install neos/neos 8.0.0
(conflict analysis result)
```

```
with it.
```

- acmecon/site dev-main requires neos/neos ^8.0 -> satisfiable by neos/neos[8.0.0, 8.0.1, 8.0.x-dev].
- Conclusion: don't install neos/neos 8.0.0 (conflict analysis result)

```
Use the option --with-all-dependencies (-W) to allow upgrades, downgrades and removals for packages currently locked to specific versions.
```



```
$ composer update
```

```
Loading composer repositories with package information
```

```
Updating dependencies
```

```
Your requirements could not be resolved to an installable set of packages.
```

```
Problem 1
```

- neos/flow[4.3.0, ..., 4.3.x-dev] require php ~7.0 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[5.0.0, ..., 5.2.x-dev] require php ~7.1 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[5.3.0, ..., 5.3.x-dev] require php ^7.1 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[6.0.0, ..., 6.3.x-dev] require php ^7.2 -> your php version (8.0.17) does not satisfy that requirement.

```
- psmb/splitadd 0.3 requires neos/neos ^7.0 || ^5.0 || ^4.0 || ^3.3 -> satisfiable by neos/neos[3.3.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev].
```

```
neos/neos[3.3.0, ..., 3.3.x-dev] require neos/flow ^6.3.0 -> satisfiable by neos/flow[6.3.0, ..., 6.3.x-dev].
```

```
- neos/neos[5.3.0, ..., 5.3.x-dev] require neos/flow ^6.3.0 -> satisfiable by neos/flow[6.3.0, ..., 6.3.x-dev].
```

```
- psmb/splitadd 0.3 requires neos/neos ^7.0 || ^5.0 || ^4.0 || ^3.3 -> satisfiable by neos/neos[3.3.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev].
```

```
- Only one of these can be installed: typo3/neos[dev-master, 1.0.0-beta1, ..., 1.2.x-dev, 2.0.0-beta1, ..., 2.3.x-dev, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev], neos/neos[dev-master, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev]. neos/neos replaces typo3/neos and thus cannot coexist with it.
```

```
- acmecom/site dev-main requires neos/neos ^8.0 -> satisfiable by neos/neos[8.0.0, 8.0.1, 8.0.x-dev].
```

```
- Conclusion: don't install neos/neos 8.0.0 (conflict analysis result)
```

```
Use the option --with-all-dependencies (-W) to allow upgrades, downgrades and removals for packages currently locked to specific versions.
```





```
$ composer update
```

```
Loading composer repositories with package information
```

```
Updating dependencies
```

```
Your requirements could not be resolved to an installable set of packages.
```

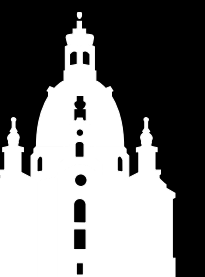
#### Problem 1

- neos/flow[4.3.0, ..., 4.3.x-dev] require php ~7.0 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[5.0.0, ..., 5.2.x-dev] require php ~7.1 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[5.3.0, ..., 5.3.x-dev] require php ^7.1 -> your php version (8.0.17) does not satisfy that requirement.
- neos/flow[6.0.0, ..., 6.3.x-dev] require php ^7.2 -> your php version (8.0.17) does not satisfy that requirement.
- neos/neos[7.0.0, ..., 7.1.0] require php ^7.3 -> your php version (8.0.17) does not satisfy that requirement.
- Root composer.json requires acmecon/site \* -> satisfiable by acmecon/site[dev-main].
- Conclusion: don't install neos/neos 8.0.1 (conflict analysis result)
- acmecon/site dev-main requires psmb/splitadd ^0.3 -> satisfiable by psmb/splitadd[0.3].
- neos/neos[3.3.0, ..., 3.3.x-dev] require neos/flow ^4.3.0 -> satisfiable by neos/flow[4.3.0, ..., 4.3.x-dev].
- neos/neos[4.0.0, ..., 4.0.x-dev] require neos/flow ^5.0.0 -> satisfiable by neos/flow[5.0.0, ..., 5.0.x-dev].

**- acmecon/site dev-main requires psmb/splitadd ^0.3 -> satisfiable by psmb/splitadd[0.3].**

- neos/neos[5.3.0, ..., 5.3.x-dev] require neos/flow ^6.3.0 -> satisfiable by neos/flow[6.3.0, ..., 6.3.x-dev].
- psmb/splitadd 0.3 requires neos/neos ^7.0 || ^5.0 || ^4.0 || ^3.3 -> satisfiable by neos/neos[3.3.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev].
- Only one of these can be installed: typo3/neos[dev-master, 1.0.0-beta1, ..., 1.2.x-dev, 2.0.0-beta1, ..., 2.3.x-dev, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev], neos/neos[dev-master, 3.0.0, ..., 3.3.x-dev, 4.0.0, ..., 4.3.x-dev, 5.0.0, ..., 5.3.x-dev, 7.0.0, ..., 7.3.x-dev, 8.0.0, 8.0.1, 8.0.x-dev]. neos/neos replaces typo3/neos and thus cannot coexist with it.
- acmecon/site dev-main requires neos/neos ^8.0 -> satisfiable by neos/neos[8.0.0, 8.0.1, 8.0.x-dev].
- Conclusion: don't install neos/neos 8.0.0 (conflict analysis result)

Use the option `--with-all-dependencies (-W)` to allow upgrades, downgrades and removals for packages currently locked to specific versions.





[Learn more about version constraints](#)

psmb/splitadd

^0.3

stable 

Results for **psmb/splitadd:^0.3**

0.4

0.3

0.2

0.1.1

0.1

dev-master

Satisfied?

```
composer require psmb/splitadd:"^0.3"
```





```
$ composer outdated
```

```
Info from https://repo.packagist.org: #StandWithUkraine
```

```
Color legend:
```

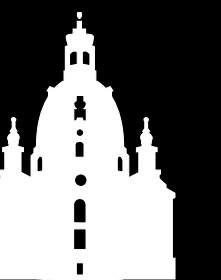
```
- patch or minor release available - update recommended
```

```
- major release available - update possible
```

carbon/condition	1.1.5	2.0.0	This package provides some fusion helper for mak...
carbon/includeassets	5.4.0	5.5.0	Include your assets (css, js) in an easy way int...
carbon/notification	2.3.0	2.3.1	This package provides a tiny fusion helper for n...
composer/composer	2.3.0	2.3.5	Composer helps you declare, manage and install d...
composer/semver	3.3.1	3.3.2	Semver library that offers utilities, version co...
doctrine/common	3.2.2	3.3.0	PHP Doctrine Common project is a library that pr...
doctrine/dbal	2.13.8	3.3.5	Powerful PHP database abstraction layer (DBAL) w...
doctrine/migrations	3.4.1	3.5.0	PHP Doctrine Migrations project offer additional...
doctrine/orm	2.11.2	2.12.1	Object-Relational-Mapper for PHP
doctrine/persistence	2.4.1	3.0.0	The Doctrine Persistence project is a set of sha...
firebase/php-jwt	v5.5.1	v6.1.2	A simple library to encode and decode JSON Web T...
flownative/google-cloudstorage	5.2.3	5.3.0	This Flow package allows you to store assets (re...
flowpack/cachebuster	1.1.0	1.2.0	Cache buster for static resources in Neos
flowpack/googleapiclient	1.2.0	1.3.0	Google Api Client provider for Neos Flow
flowpack/listable	3.5.0	3.5.1	Tiny extension for listing things
flowpack/neos-dimensionresolver	2.0	2.1	A support package for Neos CMS that allows for a...
google/apiclient	v2.12.1	v2.12.4	Client library for Google APIs
google/apiclient-services	v0.241.0	v0.246.0	Client library for Google APIs
google/auth	v1.19.0	v1.21.0	Google Auth Library for PHP
google/cloud-core	v1.44.1	v1.44.2	Google Cloud PHP shared dependency, providing fu...
google/cloud-storage	v1.1.5	v1.26.2	Cloud Storage Client for PHP
guzzlehttp/psr7	1.8.5	2.2.1	PSR-7 message implementation that also provides ...
imagine/imagine	1.3.1	1.3.2	Image processing for PHP 5.3
jcupitt/vips	v1.0.9	v2.0.2	A high-level interface to the libvips image proc...
jonnitto/plyr	2.7.0	3.0.0	Plyr.io for Neos.io
justinrainbow/json-schema	5.2.11	5.2.12	A library to validate a json schema.
acmecom/site	dev-development	Neos site package	for the Acmecom Site
monolog/monolog	2.4.0	2.5.0	Sends your logs to files, sockets, inboxes, data...
neos/behat	7.3.0	8.0.0	Behat support package for Neos Flow
neos/buildessentials	7.3.0	8.0.0	Neos Flow Build Toolchain Essentials



```
$ composer why psmb/splitadd  
acmecom/site dev-development requires psmb/splitadd (^0.3)
```



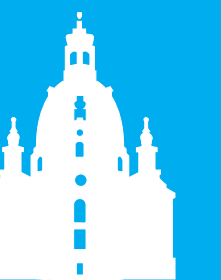


# Doing the update

As soon as your dry-run update works and gives reasonable results

Update without the dry-run flag and commit the changes to the manifest(s) and lockfile

Run core migrations on your own packages, check results, clean up as needed



# Clean up after core:migrate

Here's what I do after running core migrations:

- check the commits that were created
- soft-reset the repository to the latest upstream commit
- adjust changes as needed
- commit the changes in one commit

That way it's easier for me to keep things clean and clear





```
$ git log |head -n 37
commit 8c264b9dc62a5064facf2df66b65a239f89b8d84
Author: Karsten Dambekalns <karsten@dambekalns.de>
Date: Thu Apr 28 09:47:54 2022 +0200
```

TASK: Apply migration Neos.Fusion-20220326120900 to package "Acmecon.Site"

Expand Neos.Fusion-FusionObjects without namespace to fully qualified names

Note: This migration did not produce any changes, so the commit simply marks the migration as applied. This makes sure it will not be applied again.

```
commit c7d878be8f01e0c4045f3f5c8ee5d4080f03e2f6
Author: Karsten Dambekalns <karsten@dambekalns.de>
Date: Thu Apr 28 09:47:51 2022 +0200
```

TASK: Apply

Adjust "Set" (see <https://github.com/neos/flow-development-collection/issues/2742>)

Note: This migration did not produce any changes, so the commit simply marks the migration as applied. This makes sure it will not be applied again.

```
commit 744cfa0000b98d3de04dbf39e8670c0b9c292956
Author: Karsten Dambekalns <karsten@dambekalns.de>
Date: Thu Apr 28 09:47:46 2022 +0200
```

TASK: Apply migration Neos.Neos-20220318111600 to package "Acmecon.Site"

Replace legacy content cache tag definitions in Fusion

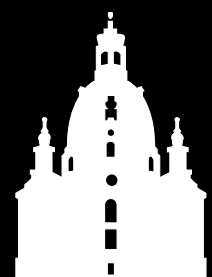
Note: This migration did not produce any changes, so the commit simply marks the migration as applied. This makes sure it will not be applied again.

```
$ git log |head -n 37
commit 978d876f6a5f616f35fa5ee185b6daa9cd5fc6c1
Author: Karsten Dambekalns <karsten@dambekalns.de>
Date: Thu Apr 28 09:47:46 2022 +0200
```

Apply core migrations to package "Acmecon.Site"

The migrations did not produce any changes, so the commit simply marks them as applied. This makes sure they will not be applied again.

**\$ git rebase -i origin/development**

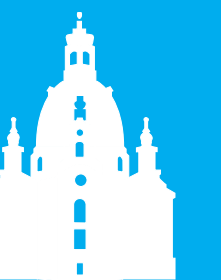


# Further adjustments

Core migrations usually adjust the „easy“ stuff, but some things cannot be automagically adjusted with reasonable effort

So some changes need to be done manually, examples for this are:

- **Fusion**  
removal of the default prototype generator
- **Logging**  
PSR-3 logger adjustments
- **Fluid ViewHelpers**  
no more `render()` arguments
- **HTTP components**  
use PSR middlewares instead



@@ -1,7 +1,13 @@

```
1 1 prototype(Neos.NeosIo:ButtonDimensionAware) < prototype(Neos.Neos:Content) {
2   templatePath = 'resource://Neos.NeosIo/Private/Templates/NodeTypes/ButtonDimensionAware.html'
3
4   @context.linkedNode = ${q(node).property('linkedNode')}
5   @context.setLanguage = ${q(node).property('language')}
6
7   buttonLabel = ${q(node).property('buttonLabel')}
8   openInNewTab = ${q(node).property('openInNewTab')}
9   alignment = ${q(node).property('alignment')}
10  style = ${q(node).property('style')}
11  link = Neos.Neos:NodeUri {
12      node = Neos.Fusion:Case {
13          withDimension {
```

@@ -0,0 +1,8 @@

```
1 prototype(Neos.NeosIo:Icon) < prototype(Neos.Neos:Content) {
2   templatePath = 'resource://Neos.NeosIo/Private/Templates/NodeTypes/Icon.html'
3
4   alignment = ${q(node).property('alignment')}
5   icon = ${q(node).property('icon')}
6   color = ${q(node).property('color')}
7   size = ${q(node).property('size')}
8 }
```





```

13 13 use Neos\Flow\Http\Uri;
14 14 use Neos\Flow\Log\Utility\LogEnvironment;
15 15 use Psr\Log\LoggerInterface;

14 16

15 17 /**
16 18  * Abstract base class for api connectors.
    @@ -54,9 +56,9 @@ abstract class AbstractApiConnector

54 56     /**
55 57     * @Flow\Inject
56 58     *
57 59     * @var \Neos\Flow\Log\SystemLoggerInterface
58 60     * @var LoggerInterface
59 61     */
60 62     protected $systemLogger;
61 63     protected $logger;

62 64     /**
63 65     * Creates a valid cache identifier.
    @@ -120,8 +122,8 @@ abstract class AbstractApiConnector

20 122         $response = $browser->request($requestUri, 'GET');

21 123

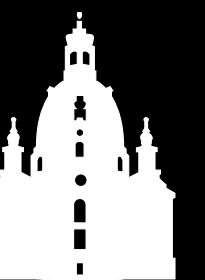
22 124         if ($response->getStatusCode() !== 200) {
23 125             $this->systemLogger->log(sprintf('Get request to Api failed with code "%s"!', $response->getStatusCode()),
24 126             LOG_ERR, 1453193835);
125 127             $this->logger->error(sprintf('Get request to Api failed with code "%s"!', $response->getStatusCode()),
126 128             LogEnvironment::fromMethodName(__METHOD__));
25 129         }
26 130

```

```

@@ -19,19 +19,31 @@ use Neos\FluidAdaptor\Core\ViewHelper\AbstractViewHelper;
19 19  */
20 20 class RelativeDateViewHelper extends AbstractViewHelper
21 21 {
22 22  /**
23 23   * Initialize arguments
24 24   *
25 25   * @return void
26 26   * @api
27 27   */
28 28 public function initializeArguments()
29 29 {
30 30     $this->registerArgument('date', \DateTimeInterface::class, 'the URI that will be put in the href attribute of the rendered
31 31     $this->registerArgument('compact', 'string', 'Open the linked document in new Tab', false, false);
32 32 }
33 33
22 34  /**
23 35   * Renders a DateTime formatted relative to the current date.
24 36   * Shows the time if the date is the current date.
25 37   * Shows the month and date if the date is the current year.
26 38   * Shows the year/month/date if the date is not the current year.
27 39   *
28 40   * @param \DateTime $date
29 41   * @param boolean $compact
30 42   * @return string an <img...> html tag
31 43   * @throws \InvalidArgumentException
32 44   */
33 45 public function render(\DateTime $date = null, $compact = false)
34 46 public function render()
35 47 {
36 48     $date = $this->arguments['date'];
37 49     $compact = $this->arguments['compact'];
38 50     if ($date === null) {
39 51         $date = $this->renderChildren();
40 52     }

```



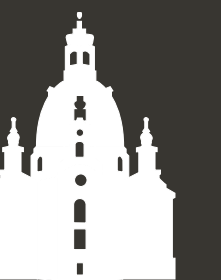
```
@@ -1,21 +1,14 @@
1 1 Neos:
2 2     Flow:
3 3     http:
4     chain:
5         'process':
6             chain:
7                 # move redirect component before routing, to make redirects work even if
8                 # there is still a visible node "behind the URL"
9                 redirect:
10                    position: 'before routing'
11                    'dynamicBaseUri':
12                        position: 'before routing'
13                        component: 'Acmecom\Site\Http\Component\DynamicBaseUriComponent'
14                        componentOptions:
15                            'bareHost': '%env:PROXY_BARE_HOST%'
16                            'pathPrefix':
17                                '*': '/magazin'
18                                'fr': '/magazine'
4     middlewares:
5         # move redirect component before routing, to make redirects work even if
6         # there is still a visible node "behind the URL"
7         'redirect':
8             position: 'before routing'
9             'dynamicBaseUri':
10                 position: 'before routing'
11                 middleware: 'Acmecom\Site\Http\Middleware\DynamicBaseUriMiddleware'
19 12 trustedProxies:
20 13     headers:
21 14         host: 'X-Forwarded-Server'
```



class DynamicBaseUriComponent implements  
ComponentInterface



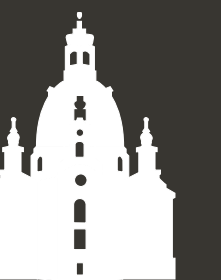
class DynamicBaseUriMiddleware implements  
MiddlewareInterface



```
public function handle(ComponentContext  
$componentContext)
```



```
public function process(ServerRequestInterface  
$request, RequestHandlerInterface $handler):  
ResponseInterface
```



# If you get stuck

Finally, if you get stuck with an update even though you were well-prepared

- Search for your issue, chances are someone else found a solution already
- Ask the community for help on Slack or [discuss.neos.io](https://discuss.neos.io)
- Ask your agency for help
- If you are an agency, ask me or my company for help 😎



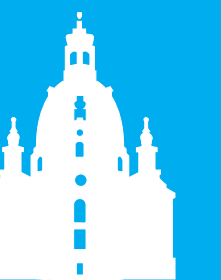


# Test and deploy

Now test your site. You may

- Run unit and functional tests if you have them
- Run a link checker on the site
- Check logs for error messages and warnings
- Click around and manually test important or prominent features

If all is well, deploy and be happy



# Problems after an update

If some new problems appear after an update – ideally found before going into production – there are two possible causes:

## **You made a mistake.**

Double-check your changes and try to verify against the previous version.

## **You found a bug.**

Check the issue tracker for a new report that looks like your problem



# Recap: Why, When & How

## Why?

Because you get new features and bug fixes.

## When?

Whenever a new version is released.

## How?

Read & follow instructions, test & deploy.

Done!





# Done. Questions!



# Thanks for listening!

Further reading:

<https://www.flownative.com/en/blog/neoscon-2022-updating-neos.html>

Contact me at:

- [@kdambekalns](https://twitter.com/kdambekalns)
- [karsten@flownative.com](mailto:karsten@flownative.com)



# Flownative



DRESDEN 2022