

Composer, Updates and Patches

TABLE OF CONTENTS

- [Creating a local patch to a contrib module](#)
- [composer.json patches in separate file](#)
- [Stop files being overwritten during composer operations](#)
- [Updating Drupal Core](#)
- [Test composer \(dry run\)](#)
- [Version constraints](#)
- [Troubleshooting](#)
 - [Composer won't update Drupal core](#)
 - [The big reset button](#)
- [Reference](#)

views 129

Creating a local patch to a contrib module

See Making a patch at <https://www.drupal.org/node/707484>

In this case, I had the file_entity module installed and wanted to hide the tab "[files.]{underline}" The tab item is provided by a task (read "menu tab") in the web/modules/contrib/file_entity/file_entity.links.task.yml

```
entity.file.collection:  
  route_name: entity.file.collection  
  base_route: system.admin_content  
  title: 'Files'  
  description: 'Manage files for your site.'
```

For my patch, I want to remove this section of the file_entity.links.task.yml file.

First get the repo/git version of the module

```
$ composer update drupal/file_entity --prefer-source
```

Change the file in the text editor

Run git diff to see the changes:

```
$ git diff
```

The output shows:

```
diff --git a/file_entity.links.task.yml b/file_entity.links.task.yml
index 3ea93fc..039f7f9 100644
--- a/file_entity.links.task.yml
+++ b/file_entity.links.task.yml
@@ -15,12 +15,6 @@ entity.file.edit_form:
   base_route: entity.file.canonical
   weight: 0

-entity.file.collection:
-  - route_name: entity.file.collection
-  - base_route: system.admin_content
-  - title: 'Files'
-  - description: 'Manage files for your site.'
-

entity.file.add_form:
  route_name: entity.file.add_form
  base_route: entity.file.add_form
```

Create the patch

```
git diff >file_entity_disable_file_menu_tab.patch
```

Add the patch to the patches section of composer.json. Notice below the line starting with "drupal/file_entity":

```
"patches": {
  "drupal/commerce": {
    "Allow order types to have no carts": "https://www.drupal.org/files/issues/2018-03-16/commerce-direct-checkout-50.patch"
  },
  "drupal/views_load_more": {
    "Template change to keep up with core": "https://www.drupal.org/files/issues/views-load-more-pager-class-2543714-02.patch" ,
    "Problems with exposed filters": "https://www.drupal.org/files/issues/views_load_more-problems-with-exposed-filters-2630306-4.patch"
  },
  "drupal/easy_breadcrumb": {
    "Titles in breadcrumbs are double-escaped": "https://www.drupal.org/files/issues/2018-06-21/2979389-7-easy-breadcrumb--double-escaped-titles.patch"
  },
  "drupal/file_entity": {
    "Temporarily disable the files menu tab": "./patches/file_entity_disable_file_menu_tab.patch"
  }
}
```

Revert the file in git and then try to apply the patch.

Here is the patch command way to un-apply or revert a patch (-R means revert)

```
patch -p1 -R < ./patches/fix_scary_module.patch
```

To apply the patch:

```
patch -p1 < ./patches/fix_scary_module.patch
```

composer.json patches in separate file

To separate patches into a different file other than composer json add "patches-file" section under "extra". See example below:

```

"extra": {
  "installer-paths": {
    "web/core": ["type:drupal-core"],
    "web/libraries/{${name}}": ["type:drupal-library"],
    "web/modules/contrib/{${name}}": ["type:drupal-module"],
    "web/profiles/contrib/{${name}}": ["type:drupal-profile"],
    "web/themes/contrib/{${name}}": ["type:drupal-theme"],
    "drush/Commands/contrib/{${name}}": ["type:drupal-drush"],
    "web/modules/custom/{${name}}": ["type:drupal-custom-module"],
    "web/themes/custom/{${name}}": ["type:drupal-custom-theme"]
  },
  "drupal-scaffold": {
    "locations": {
      "web-root": "web/"
    },
    "excludes": [
      "robots.txt",
      ".htaccess"
    ]
  },
  "patches-file": "patches/composer_patches.json"
}

```

If composer install fails, try `composer -vvv` for verbose output

If the issue is that it can't find the file for example if it displays the following:

```

- Applying patches for drupal/addtocalendar
./patches/add_to_calendar_smart_date_handling.patch (Add support for smart_date fields)
patch 'p1' --no-backup-if-mismatch -d 'web/modules/contrib/addtocalendar' < '/Users/selwyn/Sites/tglobal/patches/add_to_calendar_smart_date_handling.patch'
Executing command (CWD): patch 'p1' --no-backup-if-mismatch -d 'web/modules/contrib/addtocalendar' < '/Users/selwyn/Sites/tglobal/patches/add_to_calendar_smart_date_h
can't find file to patch at input line 5
Perhaps you used the wrong -p or --strip option?

```

This means the patch is trying to run the patch in the directory `web/modules/contrib/addtocalendar` (notice the `-d web/modules/contrib/addtocalendar` above)

In this case, recreate the patch with the `--no-prefix` option i.e.

```
git diff --no-prefix >./patches/patch2.patch
```

Then composer install will apply the patch correctly

More at <https://github.com/cweagans/composer-patches/issues/146>

Stop files being overwritten during composer operations

Depending on your `composer.json`, files like `development.services.yml` may be overwritten from during scaffolding. To prevent certain scaffold files from being overwritten every time you run a Composer command you can specify them in the “extra” section of your project's `composer.json`. See the docs on Excluding scaffold files.

The following snippet prevents the `development.services.yml` from being regularly overwritten:

```
"drupal-scaffold": {
  "locations": {
    "web-root": "web/"
  },
  "file-mapping": {
    "[web-root]/sites/development.services.yml": false
  }
},
```

The code above is from <https://www.drupal.org/docs/develop/development-tools/disable-caching#s-beware-of-scaffolding>

and from https://www.drupal.org/docs/develop/using-composer/using-drupals-composer-scaffold#toc_6: Sometimes, a project might prefer to entirely replace a scaffold file provided by a dependency, and receive no further updates for it. This can be done by setting the value for the scaffold file to exclude to false. In the example below, three files are excluded from being overwritten:

```
"name": "my/project",
...
"extra": {
  "drupal-scaffold": {
    "locations": {
      "web-root": "web/"
    },
    "file-mapping": {
      "[web-root]/robots.txt": false
      "[web-root]/.htaccess": false,
      "[web-root]/sites/development.services.yml": false
    },
    ...
  }
}
```

More at <https://drupal.stackexchange.com/questions/290989/composer-keeps-overwriting-htaccess-and-other-files-every-time-i-do-anything>

Updating Drupal Core

if there is drupal/core-recommended in your composer.json use:

```
$ composer update drupal/core-recommended -W
```

if there is no drupal/core-recommended in your composer.json use:

```
$ composer update drupal/core -W
```

Note composer update -W is the same as composer update --with-dependencies

Test composer (dry run)

If you want to run through an installation without actually installing a package, you can use `--dry-run`. This will simulate the installation and show you what would happen.

```
composer update --dry-run "drupal/*"
```

produces something like:

```
Package operations: 0 installs, 4 updates, 0 removals
- Updating drupal/core (8.8.2) to drupal/core (8.8.4)
- Updating drupal/config_direct_save (1.0.0) to drupal/config_direct_save (1.1.0)
- Updating drupal/core-recommended (8.8.2) to drupal/core-recommended (8.8.4)
- Updating drupal/crop (1.5.0) to drupal/crop (2.0.0)
```

Version constraints

1. The caret constraint (^): this will allow any new versions except BREAKING ones—in other words, the first number in the version cannot increase, but the others can. `drupal/foo:^1.0` would allow anything greater than or equal to 1.0 but less than 2.0.x . If you need to specify a version, this is the recommended method.

2. The tilde constraint (~): this is a bit more restrictive than the caret constraint. It means composer can download a higher version of the last digit specified only. For example, `drupal/foo:~1.2` will allow anything greater than or equal to version 1.2 (i.e., 1.2.0, 1.3.0, 1.4.0, ..., 1.999.999), but it won't allow that first 1 to increment to a 2.x release. Likewise, `drupal/foo:~1.2.3` will allow anything from 1.2.3 to 1.2.999, but not 1.3.0.

3. The other constraints are a little more self-explanatory. You can specify a version range with operators, a specific stability level (e.g., -stable or -dev), or even specify wildcards with *.

Version range: By using comparison operators you can specify ranges of valid versions. Valid operators are >, >=, <, <=, !=.

You can define multiple ranges. Ranges separated by a space () or comma (,) will be treated as a logical AND. A double pipe (||) will be treated as a logical OR. AND has higher precedence than OR.

Note: Be careful when using unbounded ranges as you might end up unexpectedly installing versions that break backwards compatibility. Consider using the caret operator instead for safety.

Examples:

- `>=1.0`
- `>=1.0 <2.0`
- `>=1.0 <1.1 || >=1.2`

More at <https://getcomposer.org/doc/articles/versions.md>

Troubleshooting

Composer won't update Drupal core

The `prohibits` command tells you which packages are blocking a given package from being installed. Specify a version constraint to verify whether upgrades can be performed in your project, and if not why not.

Why won't composer install Drupal version 8.9.1.

`composer why-not drupal/core:8.9.1`

The big reset button

If composer barfs with a bunch of errors, try removing `vendor/`, `/core`, `modules/contrib` (and optionally `composer.lock` using:

```
$ rm -fr core/ modules/contrib/ vendor/
```

Then try run composer install again to see how it does:

```
$ composer install --ignore-platform-reqs
```

Note `--ignore-platform-reqs` is only necessary if your php on your host computer is different to the version in your DDEV containers.

You could always use this for DDEV:



```
$ ddev composer install
```

Reference

- Drupal 8 composer best practices (Jan 2018) <https://www.lullabot.com/articles/drupal-8-composer-best-practices>
- Making a patch (Dec 2022) <https://www.drupal.org/node/707484>
- Composer Documentation <https://getcomposer.org/doc/>
- Composer documentation article on versions and constraints <https://getcomposer.org/doc/articles/versions.md>

- Using Drupal's Composer Scaffold updated Dec 2922 https://www.drupal.org/docs/develop/using-composer/using-drupals-composer-scaffold#toc_6

[Back to top](#)

[Drupal at your fingertips](#) by [Selwyn Polit](#) is licensed under [CC BY 4.0](#)  

Page last modified: Apr 28 2023.

[Edit this page on GitHub](#)

This site uses [Just the Docs](#), a documentation theme for Jekyll.