

De-Mystifying Composer

Presented by Dan Ficker

Twin Cities DrupalCamp, September 15, 2023

About The Presenter

Dan Ficker

- Staff Customer Success Engineer, Pantheon
- Been Using Drupal For 18 Years
- Site Builder, Developer & Debugger
- Lives in St. Paul, Minnesota



De-Mystifying Composer

Agenda

What Is Composer?

How To Use Composer

Composer And Drupal

Q&A / Tips & Tricks

Composer

What Is It? What Does It Do?

Do I Need To Use It?



Composer

Dependency Manager for PHP

- Official Site: getcomposer.org
- Command-line-based tool
- Similar to Node's "npm" and Ruby's "bundler"



A Dependency Manager for PHP

Latest: **2.6.2** ([changelog](#))

[Getting Started](#)

[Download](#)

[Documentation](#)

[Browse Packages](#)

[Issues](#)

[GitHub](#)

Authors: [Nils Adermann](#), [Jordi Boggiano](#) and many [community contributions](#)

What Is A “Dependency Manager”?

- In modern development, many “libraries” are used.
 - Library: A package of code that does one or more specific functions.
 - Libraries help to use maintained code instead of building from scratch.
- Drupal and other projects use many libraries to accomplish small tasks.
- A Dependency Manager helps with installing libraries, determining which versions can be installed, updating libraries, and other customizations.

What Is Composer?

The composer.json File

- In the root of a project, there is a composer.json file.
- This is a text file in JSON format.
- At its core is a set of requirements — which libraries are needed.
- Also included are various other configurations and customizations.

```
"name": "pantheon-upstreams/drupal-composer-managed",
"description": "Install Drupal 9 with Composer on Pantheon.",
"type": "project",
"license": "MIT",
"repositories": [
  {
    "type": "composer",
    "url": "https://packages.drupal.org/8"
  },
  {
    "type": "path",
    "url": "upstream-configuration"
  }
],
"require": {
  "pantheon-upstreams/upstream-configuration": "dev-main",
  "composer/installers": "^1.9",
  "drupal/core-composer-scaffold": "^10",
  "drupal/core-recommended": "^10",
  "pantheon-systems/drupal-integrations": "^10",
  "cweagans/composer-patches": "^1.7",
  "drush/drush": "^11 || ^12",
  "drupal/pathauto": "^1.9",
  "drupal/redirect": "^1.9"
},
"require-dev": {
  "drupal/core-dev": "^10"
},
"conflict": {
  "drupal/drupal": "*"
},
"minimum-stability": "stable",
"prefer-stable": true,
"extra": {
  "drupal-scaffold": {
    "locations": {
      "web-root": "./web"
    },
    "allowed-packages": ["pantheon-systems/drupal-integrations"],
    "file-mapping": {
      "[project-root]/.editorconfig": false,
```

What Is Composer?

The composer.lock File

- The composer.json only lists broad requirements — which libraries are needed and which versions are compatible.
- The composer.lock file is a list of exactly which versions of each library are currently installed.
- composer.json may require drupal/core version ^10. But composer.lock has drupal/core set to 10.1.2, for example.

```
{
  "_readme": [
    "This file locks the dependencies of your project to a known state",
    "Read more about it at https://getcomposer.org/doc/01-basic-usage.md#installing-dependencies",
    "This file is @generated automatically"
  ],
  "content-hash": "19a1930ca5fe868b388813c3a152bd51",
  "packages": [
    {
      "name": "asm89/stack-cors",
      "version": "v2.1.1",
      "source": {
        "type": "git",
        "url": "https://github.com/asm89/stack-cors.git",
        "reference": "73e5b88775c64ccc0b84fb60836b30dc9d92ac4a"
      },
      "dist": {
        "type": "zip",
        "url": "https://api.github.com/repos/asm89/stack-cors/zipball/73e5b88775c64ccc0b84fb60836b30dc9d92ac4a",
        "reference": "73e5b88775c64ccc0b84fb60836b30dc9d92ac4a",
        "shasum": ""
      },
      "require": {
        "php": "^7.2|^8.0",
        "symfony/http-foundation": "^4|^5|^6",
        "symfony/http-kernel": "^4|^5|^6"
      },
      "require-dev": {
        "phpunit/phpunit": "^7|^9",
        "squizlabs/php_codesniffer": "^3.5"
      },
      "type": "library",
      "extra": {
        "branch-alias": {
          "dev-master": "2.1-dev"
        }
      },
      "autoload": {
        "psr-4": {
          "Asm89\\Stack\\": "src/"
        }
      }
    },
  ],
}
```


What Is Composer?

How Composer Works

- Many libraries have other libraries they depend on. Drupal, for example, requires many Symfony libraries, and many Symfony libraries rely on other Symfony libraries, but modules/themes can require libraries as well.
- Composer figures out which version of each library will work with all the different requirements. It installs only that one version and loads it into Drupal when needed.
- Composer not only keeps track of what libraries and which specific version is needed, but what libraries can be updated to newer versions.

How To Use Composer



Composer Commands

Starting/Creating A Project

- Composer's data is stored in JSON and PHP files, but it's really best not to modify those directly. Use Composer's command-line tools to add/remove libraries and reconfigure Composer.
- To create a new site/project with the latest version of Drupal core:
 - `composer create-project drupal/recommended-project my_site_name_dir`
- To create a new library like a module or theme, run `composer init`. It will help create a `composer.json` file.

Composer Commands

composer require

- To add a module/theme/profile or other library to your site, run:
 - `composer require drupal/pathauto`
- Composer will add the library (if there's a compatible version) to the `composer.json`'s "require" section, install that version and update `composer.lock`.
- If it's a library only used in development, include `--dev` flag.

```
danielficker@Daniels-MacBook-Pro ficker-10-test % composer require drupal/pathauto
./composer.json has been updated
Running composer update drupal/pathauto
No patches supplied.
> DrupalComposerManaged\ComposerScripts::preUpdate
Loading composer repositories with package information
Updating dependencies
Lock file operations: 3 installs, 0 updates, 0 removals
  - Locking drupal/ctools (4.0.4)
  - Locking drupal/pathauto (1.11.0)
  - Locking drupal/token (1.12.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 3 installs, 0 updates, 0 removals
No patches supplied.
Gathering patches for dependencies. This might take a minute.
  - Installing drupal/token (1.12.0): Extracting archive
  - Installing drupal/ctools (4.0.4): Extracting archive
  - Installing drupal/pathauto (1.11.0): Extracting archive
Generating autoload files
87 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
phpstan/extension-installer: Extensions installed
> DrupalComposerManaged\ComposerScripts::postUpdate
No security vulnerability advisories found
Using version ^1.11 for drupal/pathauto
danielficker@Daniels-MacBook-Pro ficker-10-test % clear
danielficker@Daniels-MacBook-Pro ficker-10-test %
```

Composer Commands

composer install

- If no composer.lock file is present, Composer will install the latest versions of all libraries required in composer.json, and write a composer.lock file.
- If a composer.lock file is present, Composer installs whatever is listed there, even if it's not exactly what composer.json says to install (probably due to manual editing of files).

```
danielficker@Daniels-MacBook-Pro ficker-10-test % composer install
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.
Package operations: 154 installs, 0 updates, 0 removals
- Installing composer/installers (v1.12.0): Extracting archive
- Installing cweagans/composer-patches (1.7.3): Extracting archive
No patches supplied.
Gathering patches for dependencies. This might take a minute.
- Installing drupal/core-composer-scaffold (10.1.3): Extracting archive
- Installing phpstan/phpstan (1.10.33): Extracting archive
- Installing phpstan/extension-installer (1.3.1): Extracting archive
- Installing squizlabs/php_codesniffer (3.7.2): Extracting archive
- Installing dealerdirect/phpcodesniffer-composer-installer (v1.0.0): Extracting archive
- Installing composer/ca-bundle (1.3.7): Extracting archive
- Installing symfony/finder (v6.3.3): Extracting archive
- Installing composer/pcre (3.1.0): Extracting archive
- Installing composer/class-map-generator (1.1.0): Extracting archive
- Installing composer/metadata-minifier (1.0.0): Extracting archive
- Installing composer/spdx-licenses (1.5.7): Extracting archive
- Installing psr/log (3.0.0): Extracting archive
- Installing composer/xdebug-handler (3.0.3): Extracting archive
- Installing symfony/polyfill-mbstring (v1.27.0): Extracting archive
- Installing symfony/polyfill-intl-normalizer (v1.27.0): Extracting archive
- Installing symfony/polyfill-intl-grapheme (v1.27.0): Extracting archive
- Installing symfony/polyfill-ctype (v1.27.0): Extracting archive
- Installing symfony/string (v6.3.2): Extracting archive
- Installing psr/container (2.0.2): Extracting archive
- Installing symfony/service-contracts (v3.3.0): Extracting archive
- Installing symfony/deprecation-contracts (v3.3.0): Extracting archive
- Installing symfony/console (v6.3.4): Extracting archive
- Installing consolidation/log (3.0.0): Extracting archive
- Installing symfony/filesystem (v6.3.1): Extracting archive
- Installing composer/semver (3.3.2): Extracting archive
- Installing consolidation/self-update (2.2.0): Extracting archive
- Installing symfony/var-dumper (v6.3.4): Extracting archive
- Installing symfony/phpunit-bridge (v6.3.2): Extracting archive
- Installing symfony/lock (v6.3.2): Extracting archive
- Installing symfony/error-handler (v6.3.2): Extracting archive
- Installing masterminds/html5 (2.8.1): Extracting archive
- Installing symfony/dom-crawler (v6.3.4): Extracting archive
- Installing symfony/css-selector (v6.3.2): Extracting archive
```

Composer Commands

composer install (Continued)

- By default, `composer install` installs all `require-dev` libraries. If you're running `composer install` on/for a Production environment, add the `--no-dev` flag.
- This operation should not update/modify the `composer.json` or `composer.lock` file contents.
- Libraries are put in the `vendor/` folder by default and an auto-loader is built to load classes for the PHP application.
 - The `composer.json` for Drupal sites will set different locations for Drupal core and contributed modules/themes/profiles/libraries—where Drupal expects them to be in the directory structure.

Composer Commands

composer update

- This command checks if there are any updates and applies them if there are.
- Updates the composer.lock to reflect the update versions.
- It will not update to a newer version of a library if the newer version is listed as not compatible with one of the other dependencies.

```
danielficker@Daniels-MacBook-Pro ficker-10-test % composer update
No patches supplied.
> DrupalComposerManaged\ComposerScripts::preUpdate
Loading composer repositories with package information
Info from https://repo.packagist.org: #StandWithUkraine
Updating dependencies
Lock file operations: 6 installs, 72 updates, 1 removal
- Removing enlightn/security-checker (v1.10.0)
- Upgrading chi-teck/drupal-code-generator (2.6.2 => 3.2.0)
- Locking colinodell/psr-testlogger (v1.2.0)
- Upgrading composer/ca-bundle (1.3.5 => 1.3.7)
- Upgrading composer/class-map-generator (1.0.0 => 1.1.0)
- Upgrading composer/composer (2.5.5 => 2.6.2)
- Upgrading consolidation/annotated-command (4.8.2 => 4.9.1)
- Upgrading consolidation/output-formatters (4.2.4 => 4.3.2)
- Upgrading consolidation/self-update (2.1.0 => 2.2.0)
- Upgrading doctrine/annotations (1.13.3 => 1.14.3)
- Upgrading doctrine/deprecations (v1.0.0 => v1.1.1)
- Upgrading doctrine/lexer (1.2.3 => 2.1.0)
- Upgrading drupal/coder (8.3.18 => 8.3.21)
- Upgrading drupal/core (10.0.9 => 10.1.3)
- Upgrading drupal/core-composer-scaffold (10.0.9 => 10.1.3)
- Upgrading drupal/core-dev (10.0.9 => 10.1.3)
- Upgrading drupal/core-recommended (10.0.9 => 10.1.3)
- Upgrading drush/drush (11.5.1 => 12.2.0)
- Upgrading egulias/email-validator (3.2.5 => 4.0.1)
- Locking grasmash/yaml-cli (3.1.0)
- Upgrading guzzlehttp/guzzle (7.5.2 => 7.7.1)
- Upgrading guzzlehttp/promises (1.5.2 => 2.0.1)
- Upgrading guzzlehttp/psr7 (2.4.5 => 2.5.1)
- Upgrading masterminds/html5 (2.7.6 => 2.8.1)
- Locking mck89/peast (v1.15.4)
- Upgrading mglaman/phpstan-drupal (1.1.32 => 1.2.0)
- Upgrading nikip/php-parser (v4.15.4 => v4.17.1)
- Upgrading pantheon-systems/drupal-integrations (10.0.0 => 10.0.1)
- Upgrading pantheon-upstreams/upstream-configuration (dev-main 9542d5a => dev-main 8f0ed84)
- Upgrading phootwork/collection (v3.2.1 => v3.2.2)
- Upgrading phootwork/lang (v3.2.1 => v3.2.2)
- Upgrading phpdocumentor/type-resolver (1.7.1 => 1.7.3)
- Upgrading phpstan/extension-installer (1.3.0 => 1.3.1)
- Upgrading phpstan/phpdoc-parser (1.20.4 => 1.24.0)
```

Composer & Drupal

Use of Composer is basically required in modern Drupal.




Composer & Drupal

How To Use Composer

- Drupal's Documentation has detailed information: <https://www.drupal.org/docs/develop/using-composer>
- Drupal module pages include composer require commands right on pages for easy copy & paste.
- Modules/themes and core are updated by composer update.

Releases

8.x-1.9  released 30 August 2023

Works with Drupal: ^9.2 || ^10

✓ Recommended by the project's maintainer.

PHP 8.2 compatibility, 404 report ignore improvements, minor cleanup and bugfixes


Install: `composer require 'drupal/redirect:^1.9'` 

Development version: [8.x-1.x-dev](#) updated 30 Aug 2023 at 10:48 UTC

redirect 8.x-1.7

Install

Works with Drupal: ^8.8 || ^9

`composer require 'drupal/redirect:^1.7'` 

[Using Composer to manage Drupal site dependencies](#)

Alternative installation files

[Download tar.gz](#) 75.64 KB

[Download zip](#) 138.76 KB

Downloads are for manual installation, which is not recommended when using Drupal 8 or later.

View file hashes: [MD5](#), [SHA-1](#), [SHA-256](#)

Release notes

Compatibility with PHP 8.1, various bugfixes and improvements

Created by: [Berdir](#)
Created on: 13 Dec 2021 at 10:48 UTC
Last updated: 13 Dec 2021 at 10:48 UTC

 [8.x-1.7](#)

[Bug fixes](#)

[New features](#)

Other releases:

[8.x-1.9](#) - 30 August 2023

[8.x-1.8](#) - 29 August 2022

[8.x-1.7](#) - 13 December 2021

[8.x-1.6](#) - 12 May 2020

[8.x-1.5](#) - 18 December 2019

[8.x-1.4](#) - 28 June 2019

[8.x-1.3](#) - 16 October 2018

[8.x-1.2](#) - 16 October 2018

[8.x-1.1](#) - 16 October 2018

Composer & Drupal

Major Version Upgrades

- Docs again does this well: <https://www.drupal.org/docs/updating-drupal/updating-drupal-core-via-composer>
- The main command:
`composer update "drupal/core-*" --with-all-dependencies`
- Composer will not do the update if not all dependencies work with the latest versions.
 - When upgrading from Drupal 9 to 10, update as many modules/themes as you can to their latest version before trying to upgrade to Drupal 10.

Composer & Drupal On Pantheon

- Creating a new Drupal 10 site on the platform starts with a Composer-based site.
- The main Git repository and branches have no core/modules/themes code—it is added by a composer install after Git push to Pantheon.
- One-click ability to run composer update on dev/multidev in Pantheon Dashboard.

Development Mode SFTP Git Clone with Git ▾

31 updates are available Update Options ▾ Apply Updates

Last checked: a few seconds ago [Check Now](#)


Composer Updates

Update drupal/core (10.1.1) to (10.1.3)
Update drupal/core-recommended (10.1.1) to (10.1.3)
Update drupal/core-composer-scaffold (10.1.1) to (10.1.3)


0 Additions, 0 Removals, 31 Updates View Detail

a few seconds ago

Commit Log

 Removing a test file. 6853e2f446

From Daniel Ficker, 2 months ago | [See Diff](#) | [View Log](#)

 Updated drupal/core to 10.1.1 and 49 others 74a4a06f7f

Additions (5)
symfony/polyfill-php83 (v1.27.0)

Composer & Drupal

On Pantheon (Continued)

- To develop locally, just clone the Git repository and run `composer install`.
- Add remove/update the modules/themes/profiles/libraries as needed.
- Commit the changes to `composer.json` and `composer.lock` and push to Pantheon to deploy changes.
- Documentation:
 - <https://docs.pantheon.io/guides/integrated-composer>
 - <https://docs.pantheon.io/drupal-migration>

Q&A / Tips & Tricks

Thank You



TWIN CITIES
DRUPAL CAMP