

Managing Drupal

with Composer

and Git

and Drush

Bill Seremetis (bserem) - Drupal Implementor

Vasilis Papoutsakis (Zekvyrin) - Back-End Developer

Ways to maintain a Drupal project

Bad Way

still usable today

1. Get the zip file
2. Extract it and upload with ftp
3. Install
4. Get a module/theme zip file
5. Go to step #1. Repeat.

I just hope it is SFTP

Ok Way

1. Use SSH instead of FTP.
2. Install Drush, get Drupal code with `drush dl`

Well... at least you use SSH.

Ok Improved

1. Add **everything** in a Git repo.

Hurray! You have an ugly git history, but you have a history!

Really Nice Way

Valid until D7

Use SSH+Drush and manage your **custom code** with Git:

1. Ignore all Drupal Core+Contrib code
2. Use a **drush makefile**
3. Do whatever you think of about third-party libraries... (ssh and unzip and track with git)

Congratulations! This is probably the best way to manage a D6/D7 site.

It is good, it uses all best practices and it works. Period.

Notes:

In D8 we substitute drush makefiles with composer.

Enter composer

Drupal is off the island!

composer

What everybody else was using for years

- Package manager for PHP
- For packages! Not extensions
- Can handle dependencies of third-party packages

Drupal acted smart and adopted composer in D8.

It was, mostly, a good thing...

Notes:

Composer is not PEAR.

How to use composer

1. Install composer (<https://getcomposer.org/>)
2. Create your Drupal codebase using [drupal-composer/drupal-project](#)
3. Add contrib modules/themes
4. Do your Drupal work

Note: the above method is not suitable for core development.

Notes:

For core development you need to git clone Drupal core and run composer install

there, instead of using drupal-project mentioned above.

Composer-Project

- Github: `https://github.com/drupal-composer/drupal-project`
- Recommended by drupal.org
- Downloads Drupal code + dependencies
- Also downloads Drupal Console & Drush locally
- Manages patches

```
composer create-project drupal-composer/drupal-project:8.x-dev <directory> --stability dev --no-interaction
```

Starting a new project

Steps to be done by the first developer

- Download and install Drupal (`drush site:install`)
- Export configuration (`drush config:export`)
- Commit and push to Git repository

Other developers can now join (clone and get db)

Install from an existing config (D8.6+)

```
$ drush site:install --existing-config
```

- Requires Drupal 8.6+ and Drush 9.4+
- Works for specific profiles like minimal (not standard yet). Work in progress!
- Alternatives: config_installer module, dump & share initial database

Adding a module

```
composer require drupal/<modulename>
```

```
composer require drupal/<modulename>:<version>
```

Notes:

Command is in the modules release page:

Adding a dev-only module

```
composer require --dev drupal/devel
```

Use this with `config split` or CMI2 once it is released

Identify outdated code

```
composer outdated drupal/*
```

What used to be `drush ups`

Update Core

```
composer update drupal/core webflo/drupal-core-require-  
dev --with-dependencies
```

Don't ask why... just do it :)

Update a module

```
composer update --with-dependencies drupal/<modulename>
```


Remove a module

```
composer remove drupal/<modulename>
```

If the module was added with `--dev` flag, you might get a prompt to confirm the removal from require-dev

Patches

Composer project comes with composer-patches

```
"extra": {  
  "patches": {  
    "drupal/pdf": {  
      "library adjust weight": "https://www.drupal.org/fi"  
    },  
  },  
}
```

Lock hash warnings

well.. sometimes hash on composer.lock isn't correct and will throw warnings.

```
composer update --lock
```

(will just update hash and nothing else)

Git

Use the default `.gitignore` provided by drupal-project.

What should be committed

- composer.json and composer.lock
- configuration folder (def: `./config`)
- any custom code:

`./web/modules/custom`

`./web/themes/custom`

What shouldn't be committed

- Vendor folder `./vendor`
- Drupal Core `./web/core`
- Contrib modules (`./web/modules/contrib`) or themes (`./web/themes/contrib`)

Common pitfalls

- Always `import` first and then work and export
- Avoid installing dev-dependencies on production
- Do not run `composer update`

Deploy to Production

Don't run Composer require or update on the production server!

- Ideally rsync/copy files from somewhere else.
- If you can't do otherwiser, run with warm caches.
- Lock the DB!

Notes:

A patched core will remove ALL CORE files and re-add them.
EVERY TIME.

Discuss about rsyncing the "app-package" to production.
We can simply use the committed composer.lock and get predictable results using `composer install`

Worth knowing

- Drupal Composer Project comes with drush 9 and drupal console by default.
- Drush 9 is installed on a per-project level. Not a system level.
- You need to add `drush-launcher` in your binaries so as to locate and use the proper drush.
- Most third-party php libraries will be downloaded automatically
- For non-php assets you can use `https://asset-packagist.org/`

Notes:

Drush 9 does not work on Aegir/BOA yet.

Dev only config

Common approach:

`config_split` allows you to split some configuration and enable/disable them per environment

For example having devel active only on dev, or reroute_email activated anywhere but on production

```
$config['config_split.config_split.dev']['status'] = TRUE;
```

Alternative:

```
drush cex --skip-modules=devel,reroute_email  
drush cim --skip-modules=devel,reroute_email
```

Configuration Sync

- Export config with `drush cex`
- Add all files or selectively (ex: the view you just worked)
- Commit
- Import on other instance with `drush cim`

Structure Sync

Allows to export/import taxonomies, blocks, menu links as configuration!

```
drush eb/ib (blocks)
drush et/it (taxonomies)
drush em/im (menu links)
```

.env files

```
MYSQL_DATABASE='db_name'  
MYSQL_HOSTNAME='localhost'  
MYSQL_PASSWORD='secret'  
MYSQL_PORT='3306'  
MYSQL_USER='db_user'
```

Change settings.php:

```
$databases['default']['default'] = [  
    'database' => getenv('MYSQL_DATABASE'),  
    ...  
    ...  
    ...  
];
```

Thanks!

Bill Seremetis (bserem)

Vasilis Papoutsakis (Zekvyrin)

zehnplus.ch