# Autoloads and class maps

How Exponential finds PHP classes - the generated class maps for the kernel, the extensions, kernel overrides and tests, every option of bin/php/ezpgenerateautoloads.php, .autoloadignore, Composer's autoloader beside...

Source file: `doc/eng-us/help/php-developers/autoloads-and-class-maps.md` Last changed 2026-11-07
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/php-developers/autoloads-and-class-maps.md

This page is for PHP developers who add, rename or move classes. Exponential does not use PSR-4. It finds classes through arrays that map each class name to its file, generated by scanning the source. A class missing from the arrays does not exist for the application, however correct its file is.

## Contents

1. The class maps
2. The generator and its options
3. What the generator scans
4. Kernel overrides
5. Composer's autoloader
6. When to regenerate
7. When a class is not found

## The class maps

| Map | Covers | Generated with | Loaded |
|---|---|---|---|
| `autoload/ezp_kernel.php` | kernel and library classes | `-k` (kernel developers; committed) | always |
| `var/autoload/ezp_extension.php` | classes of the active extension directories | `-e` | when it exists |
| `var/autoload/ezp_override.php` | kernel classes replaced by extensions | `-o` | only with `EZP_AUTOLOAD_ALLOW_KERNEL_OVERRIDE` |
| `var/autoload/ezp_tests.php` | test classes | `-s` | only in tests ( `ezpAutoloader::enableTests()` |

The autoloader in `autoload.php` merges them; when a class appears in the extension map and the kernel map, the extension's entry comes first in the merge. Runtime maps leave every `tests/` directory out.

## The generator and its options

**Shell**

```
php bin/php/ezpgenerateautoloads.php -e          # extensions (the one you need most)
php bin/php/ezpgenerateautoloads.php -k          # kernel (kernel developers)
php bin/php/ezpgenerateautoloads.php -o          # kernel overrides
php bin/php/ezpgenerateautoloads.php -s          # tests
php bin/php/ezpgenerateautoloads.php -e -n -v    # dry run, verbose: what would change
php bin/php/console exp:ezpgenerateautoloads -e  # the same through the console
```

| Option | Meaning |
|---|---|
| `-e`, `--extension` | the extension map |
| `-k`, `--kernel` | the kernel map |
| `-o`, `--kernel-override` | the kernel override map |
| `-s`, `--tests` | the test map |
| `-n`, `--dry-run` | write nothing |
| `-v`, `--verbose`, `-p`, `--progress` | more output |
| `-t`, `--target` | another directory for the generated file |
| `--exclude` | folders to leave out of the search |
| `<extension>` | generate for one extension |

`composer install` and `composer update` run the generator afterwards (`"legacy-scripts"` in `composer.json`).

## What the generator scans

- `-e` walks the extension directories (`extension/` and `[ExtensionSettings] AdditionalExtensionDirectories[]`), PHP file by PHP file, and lists every class, interface and trait it finds with its path.
- `.autoloadignore` at the root names directories the generator must never walk. `vendor/` is one: Composer resolves its classes, and listing them in the kernel map once pulled in hundreds of classes that nothing kept up to date.
- A duplicate class name is reported; rename one of the two classes (Class naming and the exp prefix).

## Kernel overrides

An extension can replace a kernel class with its own copy of the same name, listed in `var/autoload/ezp_override.php` (`-o`). It is active only when `config.php` defines `EZP_AUTOLOAD_ALLOW_KERNEL_OVERRIDE` as `true` (`config.php-RECOMMENDED` shows the line, commented out).

This is the heaviest mechanism and the last resort: nothing registers the replacement by name, two extensions replacing the same class conflict without knowing it, and the copy must follow every kernel update by hand. Prefer a registered extension point, a handler named by a setting, an event listener, or a `[RunnableSettings] Implementation[]` subclass (Kernel overrides, the last resort, Extension points from PHP).

## Composer's autoloader

Libraries in `vendor/` (the expcomponents with their `ezc*` classes, PHPUnit, `mongodb/mongodb` and the rest) are loaded by Composer's own autoloader, which `vendor/autoload.php` registers. An extension that needs a library declares it in its own `composer.json`; it never copies `vendor/` into the extension (Composer for PHP developers).

## When to regenerate

| Change | Run |
|---|---|
| a new, renamed or moved class in an extension | `-e` |
| an extension activated or added | `-e` |
| a new kernel class | `-k`, and commit `autoload/ezp_kernel.php` |
| a new kernel override | `-o` |
| a new test class | `-s` |
| under Velocity | `exp:velocity deploy` (`--kernel` for a kernel class): regenerates, then restarts the workers, which keep the map they started with |

## When a class is not found

| Symptom | Cause | Fix |
|---|---|---|
| `Class "acmeReportService" not found` | map not regenerated | `ezpgenerateautoloads.php -e` |
| works on the command line, fails on the web | the web server's workers started before the class existed | reload PHP-FPM or `exp:velocity deploy` |
| a setting names a class and pages break | the setting went live before the class was deployed | deploy code first; `class_exists()` where a missing class must not break a page |
| the generator reports a duplicate | two files declare the class | rename one |
| a class inside `tests/` is not found at runtime | runtime maps leave `tests/` out | move it out of `tests/` |

## Related pages

- Autoloads and class naming (application creators)
- Autoloads and class naming (extension developers)
- Additional extension directories
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

### Become an Exponential Partner

Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.

Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.