

# Composer for System Administrators

## Composer

**Composer** is an opensource PHP packaging system to manage dependancies.

This makes it easy to adapt package installs and updates to your workflow, allowing you to test new/updated packages in a development environment, put the changes in your version control system (git, Subversion, Mercurial, etc.), pull in those changes on a staging environment and, when approved, put it in production.



### **composer.phar or composer ?**

The following examples use a composer install global command, as alternative use `php composer.phar <command>`.  
Read the answer in the FAQ : [What Composer command-line do you have to use ?](#)



See the [Composer documentation](#) for further information

## Technical pre-requisites

Composer requires PHP 5.3.2+ to run.

## Useful Composer commands for System Administrators



Note : as usual with CLI, you can type

```
$> php composer.phar help [--xml] [--format="..."] [--raw] [command_name]
```

to get the help of the command.

You will find on this page some useful commands and an extract of the Composer Documentation. The interesting options part is an extract of available options

## show

The show command displays detailed information about a package, or lists all packages available.

### Usage:

```
php composer.phar show [-i|--installed] [-p|--platform] [-a|--available] [-s|--self]
[-N|--name-only] [-P|--path] [package] [version]
```

## require

The require command adds required packages to your composer.json and installs them. If you do not want to install the new dependencies immediately you can call it with `--no-update`

### Usage:

```
php composer.phar require [--dev] [--prefer-source] [--prefer-dist] [--no-progress]
[--no-update] [--update-no-dev] [--update-with-dependencies] [packages1] ...
[packagesN]
```

## Interesting options

--prefer-source	Forces installation from package sources when possible, including VCS information.
--prefer-dist	Forces installation from package dist even for dev versions.
--no-progress	Do not output download progress.
--no-update	Disables the automatic update of the dependencies.
--update-with-dependencies	Allows inherited dependencies to be updated with explicit dependencies.

## search

The search command searches for packages by its name.

### Example :

```
$> php composer.phar search symfony composer
```

can returns to you a list like this :

› [Expand](#)

`symfony/assetic-bundle` Integrates Assetic into Symfony2 [source](#)  
`symfony/monolog-bundle` Symfony MonologBundle  
`ezsystems/ngsymfonytools-bundle` Bundle of the legacy netgen/ngsymfonytools extension  
`symfony-cmf/routing` Extends the Symfony2 routing component for dynamic routes and chaining several routers  
`doctrine/doctrine-bundle` Symfony DoctrineBundle  
`nelmio/cors-bundle` Adds CORS (Cross-Origin Resource Sharing) headers support in your Symfony2 application  
`tedivm/stash-bundle` Incorporates the Stash caching library into Symfony.  
`egulias/listeners-debug-command-bundle` Symfony 2 console command to debug listeners  
`hautelook/templated-uri-router` Symfony2 RFC-6570 compatible router and URL Generator  
`hautelook/templated-uri-bundle` Symfony2 Bundle that provides a RFC-6570 compatible router and URL Generator.  
`symfony/swiftmailer-bundle` Symfony SwiftmailerBundle  
`white-october/pagerfanta-bundle` Bundle to use Pagerfanta with Symfony2  
`symfony/icu` Contains an excerpt of the ICU data and classes to load it.  
`symfony/symfony` The Symfony PHP framework  
`sensio/distribution-bundle` The base bundle for the Symfony Distributions  
`symfony/symfony` The Symfony PHP framework  
`symfony/console` Symfony Console Component  
`symfony/filesystem` Symfony Filesystem Component  
`symfony/finder` Symfony Finder Component  
`symfony/process` Symfony Process Component  
`symfony/yaml` Symfony Yaml Component  
`symfony/translation` Symfony Translation Component  
`symfony/debug` Symfony Debug Component  
`symfony/routing` Symfony Routing Component  
`symfony/icu` Contains an excerpt of the ICU data to be used with `symfony/intl`.  
`symfony/config` Symfony Config Component  
`symfony/validator` Symfony Validator Component  
`symfony/stopwatch` Symfony Stopwatch Component  
`symfony-cmf/symfony-cmf` Symfony Content Management Framework

## validate

The validate command validates a given composer.json

## Usage

```
$> php composer.phar validate [--no-check-all] [file]
```

## Interesting options

option	description
<code>--no-check-all</code>	Do not make a complete validation
<code>--profile</code>	Display timing and memory usage information
<code>--working-dir (-d)</code>	If specified, use the given directory as working directory.

# Automate installation

Note that you can add some scripts to the Composer dependencies installation.

The available events are :

- **pre-install-cmd**
- **post-install-cmd**
- **pre-update-cmd**
- **post-update-cmd**
- **pre-status-cmd**
- **post-status-cmd**
- **pre-package-install**
- **post-package-install**
- **pre-package-update**
- **post-package-update**
- **pre-package-uninstall**
- **post-package-uninstall**
- **pre-autoload-dump**
- **post-autoload-dump**
- **post-root-package-install**
- **post-create-project-cmd**
- **pre-archive-cmd**
- **post-archive-cmd**



See [the Composer documentation](#) about scripts for more information