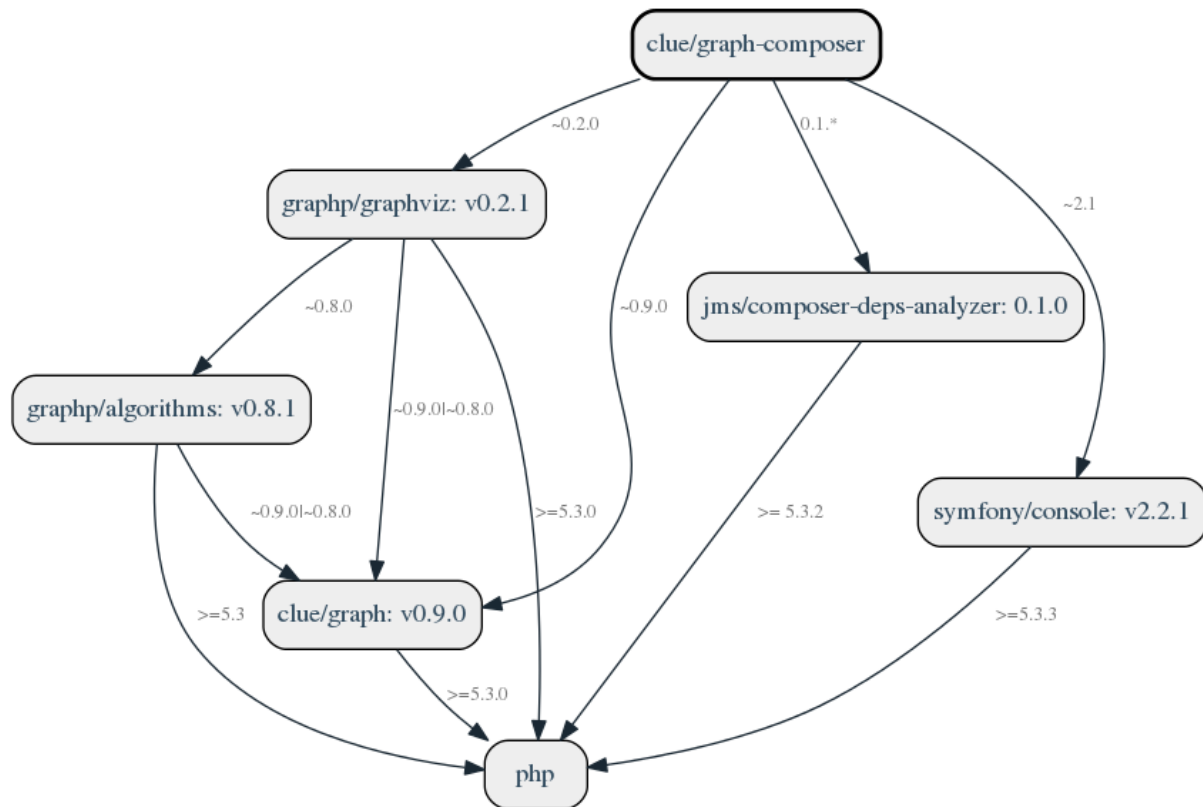


---

## clue/graph-composer

 CI  no status  downloads on GitHub 4k  installs on Packagist 88k

Graph visualization for your project's `composer.json` and its dependencies:



### Table of contents

- Usage
  - `graph-composer show`
  - `graph-composer export`
- Install
  - As a phar (recommended)
  - Installation using Composer
- Development
- Tests
- License

---

## Usage

Once clue/graph-composer is installed, you can use it via command line like this.

### graph-composer show

The `show` command creates a dependency graph for the given project path and opens the default desktop image viewer for you:

```
1 $ php graph-composer.phar show ~/path/to/your/project
```

- It accepts an optional argument which is the path to your project directory or composer.json file (defaults to checking the current directory for a composer.json file).
- You may optionally pass an `--format=[svg/svgz/png/jpeg/...]` option to set the image type (defaults to `svg`).

### graph-composer export

The `export` command works very much like the `show` command, but instead of opening your default image viewer, it will write the resulting graph to STDOUT or into an image file:

```
1 $ php graph-composer.phar export ~/path/to/your/project
```

- It accepts an optional argument which is the path to your project directory or composer.json file (defaults to checking the current directory for a composer.json file).
- It accepts an additional optional argument which is the path to write the resulting image to. Its file extension also sets the image format (unless you also explicitly pass the `--format` option).  
Example call:

```
1 $ php graph-composer.phar export ~/path/to/your/project export.png
```

If this argument is not given, it defaults to writing to STDOUT, which may be useful for scripting purposes:

```
1 $ php graph-composer.phar export ~/path/to/your/project | base64
```

- You may optionally pass an `--format=[svg/svgz/png/jpeg/...]` option to set the image type (defaults to `svg`).

---

## Install

You can grab a copy of clue/graph-composer in either of the following ways.

This project aims to run on any platform and thus does not require any PHP extensions and supports running on legacy PHP 5.3 through current PHP 8+. It's *highly recommended to use the latest supported PHP version* for this project.

The graph drawing feature is powered by the excellent GraphViz software. This means you'll have to install GraphViz ([dot](#) executable). The Graphviz homepage includes complete installation instructions for most common platforms, users of Debian/Ubuntu-based distributions may simply invoke:

```
1 $ sudo apt install graphviz
```

### As a phar (recommended)

Once you have PHP and GraphViz installed, you can simply download a pre-packaged and ready-to-use version of this project as a Phar to any directory. You can simply download the latest [graph-composer.phar](#) file from our releases page. The latest release can always be downloaded like this:

```
1 $ curl -JOL https://clue.engineering/graph-composer-latest.phar
```

That's it already. Once downloaded, you can verify everything works by running this:

```
1 $ cd ~/Downloads
2 $ php graph-composer.phar --version
```

If you prefer a global (system-wide) installation without having to type the [.phar](#) extension each time, you may simply invoke:

```
1 $ chmod +x graph-composer.phar
2 $ sudo mv graph-composer.phar /usr/local/bin/graph-composer
```

You can verify everything works by running:

```
1 $ graph-composer --version
```

There's no separate [update](#) procedure, simply download the latest release again and overwrite the existing phar.

---

## Installation using Composer

Alternatively, you can also install clue/graph-composer as part of your development dependencies. You will likely want to use the `require-dev` section to exclude clue/graph-composer in your production environment.

This method also requires PHP 5.3+, GraphViz and, of course, Composer.

You can either modify your `composer.json` manually or run the following command to include the latest tagged release:

```
1 $ composer require --dev clue/graph-composer
```

Now you should be able to invoke the following command in your project root:

```
1 $ ./vendor/bin/graph-composer show
```

Alternatively, you can install this globally for your user by running:

```
1 $ composer global require clue/graph-composer
```

Now, assuming you have `~/.composer/vendor/bin` in your path, you can invoke the following command:

```
1 $ graph-composer show ~/path/to/your/project
```

Note: You should only invoke and rely on the main graph-composer bin file. Installing this project as a non-dev dependency in order to use its source code as a library is *not supported*.

To update to the latest release, just run `composer update clue/graph-composer`. If you installed it globally via composer you can run `composer global update clue/graph-composer` instead.

## Development

clue/graph-composer is an open-source project and encourages everybody to participate in its development. You're interested in checking out how clue/graph-composer works under the hood and/or want to contribute to the development of clue/graph-composer? Then this section is for you!

The recommended way to install clue/graph-composer is to clone (or download) this repository and use Composer to download its dependencies. Therefore you'll need PHP, Composer, GraphViz, git and curl installed. For example, on a recent Ubuntu/debian system, simply run:

```
1 $ sudo apt install php7.2-cli git curl graphviz
```

---

```
2
3 $ git clone https://github.com/clue/graph-composer.git
4 $ cd graph-composer
5
6 $ curl -s https://getcomposer.org/installer | php
7 $ sudo mv composer.phar /usr/local/bin/composer
8
9 $ composer install
```

You can now verify everything works by running `clue/graph-composer` like this:

```
1 $ php bin/graph-composer show
```

If you want to distribute `clue/graph-composer` as a single standalone release file, you may compile the project into a single `graph-composer.phar` file like this:

```
1 $ composer build
```

Note that compiling will temporarily install a copy of this project to the local `build/` directory and install all non-development dependencies for distribution. This should only take a second or two if you've previously installed its dependencies already. The build script optionally accepts the version number (`VERSION` env) and an output file name or will otherwise try to look up the last release tag, such as `graph-composer-1.0.0.phar`.

You can now verify the resulting `graph-composer.phar` file works by running it like this:

```
1 $ ./graph-composer.phar --version
```

To update your development version to the latest version, just run this:

```
1 $ git pull
2 $ php composer.phar install
```

Made some changes to your local development version?

Make sure to let the world know! :shipit: We welcome PRs and would love to hear from you!

Happy hacking!

## Tests

To run the test suite, you first need to clone this repo and then install all dependencies through Composer:

```
1 $ composer install
```

---

To run the test suite, go to the project root and run:

```
1 $ php vendor/bin/phpunit
```

## License

This project is released under the permissive MIT license.

Did you know that I offer custom development services and issuing invoices for sponsorships of releases and for contributions? Contact me (@clue) for details.