Composer CMS: Content Make System

Gary B. Genett

v3.0 (2022-05-11)

Table of Contents

# 1 Composer CMS

| Composer Icon | “Creating Made Simple.” |
| --- | --- |
| [Composer CMS v3.0](https://github.com/garybgenett/composer/tree/v3.0) | [License: GPL](https://github.com/garybgenett/composer/blob/master/LICENSE.md) |
| [Gary B. Genett](http://www.garybgenett.net/projects/composer) | [composer@garybgenett.net](mailto:composer@garybgenett.net?subject=Composer%20CMS%20Submission&body=Thank%20you%20for%20sending%20a%20message%21) |

## 1.1 Overview

[**Composer**](https://github.com/garybgenett/composer) **is a simple but powerful CMS based on** [**Pandoc**](http://www.johnmacfarlane.net/pandoc)**,** [**Bootstrap**](https://getbootstrap.com) **and** [**GNU Make**](http://www.gnu.org/software/make)**.** It is a document and website build system that processes directories or individual files in [Markdown](http://daringfireball.net/projects/markdown) format.

Traditionally, CMS stands for Content Management System. [Composer](https://github.com/garybgenett/composer) is designed to be a Content **Make** System. Written content is vastly easier to manage as plain text, which can be crafted with simple editors and tracked with revision control. However, professional documentation, publications, and websites require formatting that is dynamic and feature-rich.

[Pandoc](http://www.johnmacfarlane.net/pandoc) is an extremely powerful document conversion tool, and is a widely used standard for processing [Markdown](http://daringfireball.net/projects/markdown) into other formats. While it has reasonable defaults, there are a large number of options, and additional tools are required for some formats and features.

[Composer](https://github.com/garybgenett/composer) consolidates all the necessary components, simplifies the options, and prettifies the output formats, all in one place. It also serves as a build system, so that large repositories can be managed as documentation archives or published as [Bootstrap Websites](#bootstrap-websites).

Composer Icon 

## 1.2 Quick Start

Use make help to get started:

make [-f .../Makefile] [variables] <filename>.<extension>  
make [-f .../Makefile] [variables] <target>

Create documents from source [Markdown](http://daringfireball.net/projects/markdown) files (see [Formatting Variables](#formatting-variables)):

make README.html  
make Composer-v3.0.Manual.html c\_list="README.md LICENSE.md"

Save a persistent configuration (see [Recommended Workflow](#recommended-workflow), [Configuration Settings](#configuration-settings) and [Special Targets](#special-targets)):

make template >.composer.mk  
$EDITOR .composer.mk  
 book-Composer-v3.0.Manual.html: README.md LICENSE.md  
make clean  
make all

Recursively install and build an entire directory tree (see [Recommended Workflow](#recommended-workflow)):

cd .../documents  
mv .../composer .Composer  
make -f .Composer/Makefile install-all  
make all-all

See help-all for full details and additional targets.

## 1.3 Principles

The guiding principles of [Composer](https://github.com/garybgenett/composer):

* All source files in readable plain text
* Professional output, suitable for publication
* Minimal dependencies, and entirely command-line driven
* Separate content and formatting; writing and publishing are independent
* Inheritance and dependencies; global, tree, directory and file overrides
* Fast; both to initiate commands and for processing to complete

Direct support for key document types (see [Document Formatting](#document-formatting)):

* [HTML](#html) & [Bootstrap Websites](#bootstrap-websites)
* [PDF](#pdf)
* [EPUB](#epub)
* [Reveal.js Presentations](#revealjs-presentations)
* [Microsoft Word & PowerPoint](#microsoft-word--powerpoint)

## 1.4 Requirements

[Composer](https://github.com/garybgenett/composer) has almost no external dependencies. All needed components are integrated directly into the repository, including [Pandoc](http://www.johnmacfarlane.net/pandoc). It does require a minimal command-line environment based on [GNU](http://www.gnu.org) tools, which is standard for all [GNU/Linux](https://gnu.org/gnu/linux-and-gnu.html) systems. The [Windows Subsystem for Linux](https://docs.microsoft.com/en-us/windows/wsl) for Windows and [MacPorts](https://www.macports.org) for macOS both provide suitable environments.

The one large external requirement is [TeX Live](https://tug.org/texlive), and it can be installed using the package managers of each of the above systems. It is only necessary for creating [PDF](#pdf) files.

Below are the versions of the components in the repository, and the tested versions of external tools for this iteration of [Composer](https://github.com/garybgenett/composer). Use [check](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) to validate your system.

| Repository | Commit | License |
| --- | --- | --- |
| [Pandoc](http://www.johnmacfarlane.net/pandoc) | 2.18 | GPL |
| [YQ](https://mikefarah.gitbook.io/yq) | v4.24.2 | MIT |
| [Bootstrap](https://getbootstrap.com) | v5.1.3 | MIT |
| [Markdown Viewer](https://github.com/Thiht/markdown-viewer) | 059f3192d4ebf5fa9776 | MIT |
| [Reveal.js](https://revealjs.com) | 4.3.1 | MIT |

| Project | Composer Version |
| --- | --- |
| GNU Bash | 5.0.18 |
| - GNU Coreutils | 8.31 |
| - GNU Findutils | 4.8.0 |
| - GNU Sed | 4.8 |
| [GNU Make](http://www.gnu.org/software/make) | 4.2.1 |
| - [Pandoc](http://www.johnmacfarlane.net/pandoc) | 2.18 |
| - [YQ](https://mikefarah.gitbook.io/yq) | 4.24.2 |
| - [TeX Live](https://tug.org/texlive) (pdf) | 2021 3.14159 2.6-1.40.22 |

[Markdown Viewer](https://github.com/Thiht/markdown-viewer) is included both for its CSS stylesheets, and for real-time rendering of [Markdown](http://daringfireball.net/projects/markdown) files as they are being written. To install, follow the instructions in the README.md, and select the appropriate manifest.\*.json file for your browser.

The versions of the integrated repositories can be changed, if desired (see [Repository Versions](#repository-versions)).

# 2 Composer Operation

## 2.1 Recommended Workflow

The ideal workflow is to put [Composer](https://github.com/garybgenett/composer) in a top-level .Composer for each directory tree you want to manage, creating a structure similar to this:

.../.Composer  
.../  
.../tld/  
.../tld/sub/

Then, it can be converted to a [Composer](https://github.com/garybgenett/composer) documentation archive ([Quick Start](#quick-start) example):

make -f .Composer/Makefile install-all  
make all-all

If specific settings need to be used, either globally or per-directory, .composer.mk files can be created (see [Configuration Settings](#configuration-settings), [Quick Start](#quick-start) example):

make template >.composer.mk  
$EDITOR .composer.mk

Custom targets can also be defined, using standard [GNU Make](http://www.gnu.org/software/make) syntax (see [Custom Targets](#custom-targets)).

[GNU Make](http://www.gnu.org/software/make) does not support file and directory names with spaces in them, and neither does [Composer](https://github.com/garybgenett/composer). Documentation archives which have such files or directories will produce unexpected results.

It is fully supported for input files to be symbolic links to files that reside outside the documentation archive:

cd .../tld  
ln -rs .../README.md ./  
make README.html

Finally, it is best practice to [install-force](#install--install-all--install-force) after every [Composer](https://github.com/garybgenett/composer) upgrade, in case there are any changes to the Makefile template (see [Primary Targets](#primary-targets)).

The archive is ready, and each directory is both a part of the collective and its own individual instance. Targets can be run per-file, per-directory, or recursively through an entire directory tree. The most commonly used targets are in [Primary Targets](#primary-targets).

**Welcome to** [**Composer**](https://github.com/garybgenett/composer)**. Happy Making!**

## 2.2 Document Formatting

As outlined in [Overview](#overview) and [Principles](#principles), a primary goal of [Composer](https://github.com/garybgenett/composer) is to produce beautiful and professional output. [Pandoc](http://www.johnmacfarlane.net/pandoc) does reasonably well at this, and yet its primary focus is document conversion, not document formatting. [Composer](https://github.com/garybgenett/composer) fills this gap by specifically tuning a select list of the most commonly used document formats.

Further options for each document type are in [Formatting Variables](#formatting-variables). All improvements not exposed as variables will apply to all documents created with a given instance of [Composer](https://github.com/garybgenett/composer).

Note that all the files referenced below are embedded in the ‘Embedded Files’ and ‘Heredoc’ sections of the Makefile. They are exported by the [\_release](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d) target, and will be overwritten whenever it is run.

### 2.2.1 HTML

In addition to being a helpful real-time rendering tool, [Markdown Viewer](https://github.com/Thiht/markdown-viewer) includes several CSS stylesheets that are much more visually appealing than the [Pandoc](http://www.johnmacfarlane.net/pandoc) default, and which behave like normal webpages, so [Composer](https://github.com/garybgenett/composer) uses them for all [HTML](#html)-based document types, including [EPUB](#epub).

Information on installing [Markdown Viewer](https://github.com/Thiht/markdown-viewer) for use as a [Markdown](http://daringfireball.net/projects/markdown) rendering tool is in [Requirements](#requirements).

### 2.2.2 Bootstrap Websites

[Bootstrap](https://getbootstrap.com) is a leading web development framework, capable of building static webpages that behave dynamically. Static sites are very easy and inexpensive to host, and are extremely responsive compared to truly dynamic webpages.

[Composer](https://github.com/garybgenett/composer) uses this framework to transform an archive of simple text files into a modern website, with the appearance and behavior of dynamically indexed pages.

*(This feature is reserved for a future release as the* [*site*](#site) *target, along with* [*page*](#page--post) *and* [*post*](#page--post) *in* [*Special Targets*](#special-targets)*.)*

### 2.2.3 PDF

The default formatting for [PDF](#pdf) is geared towards academic papers and the typesetting of printed books, instead of documents that are intended to be purely digital.

Internally, [Pandoc](http://www.johnmacfarlane.net/pandoc) first converts to LaTeX, and then uses [TeX Live](https://tug.org/texlive) to convert into the final [PDF](#pdf). [Composer](https://github.com/garybgenett/composer) inserts customized LaTeX to modify the final output:

.../artifacts/pdf.latex

### 2.2.4 EPUB

The [EPUB](#epub) format is essentially packaged [HTML](#html), so [Composer](https://github.com/garybgenett/composer) uses the same [Markdown Viewer](https://github.com/Thiht/markdown-viewer) CSS stylesheets for it.

### 2.2.5 Reveal.js Presentations

The CSS for [Reveal.js](https://revealjs.com) presentations has been modified to create a more traditional and readable end result. The customized version is at:

.../artifacts/revealjs.css

It links in a default theme from the .../revealjs/dist/theme directory. Edit the location in the file, or use [c\_css](#c_css) to select a different theme.

It is set up so that a logo can be placed in the upper right hand corner on each slide, for presentations that need to be branded. Simply copy an image file to the logo location:

.../artifacts/logo.img

To have different logos for different directories (using [Recommended Workflow](#recommended-workflow), [Configuration Settings](#configuration-settings) and [Precedence Rules](#precedence-rules)):

cd .../presentations  
cp .../logo.img ./  
ln -rs .../.Composer/artifacts/revealjs.css ./.composer.css  
echo 'override c\_type := revealjs' >>./.composer.mk  
make all

### 2.2.6 Microsoft Word & PowerPoint

The internal [Pandoc](http://www.johnmacfarlane.net/pandoc) templates for these are exported by [Composer](https://github.com/garybgenett/composer), so they are available for customization:

.../artifacts/reference.docx  
.../artifacts/reference.pptx

They are not currently modified by [Composer](https://github.com/garybgenett/composer).

## 2.3 Configuration Settings

[Composer](https://github.com/garybgenett/composer) uses .composer.mk files for persistent settings and definition of [Custom Targets](#custom-targets). By default, they only apply to the directory they are in (see [COMPOSER\_INCLUDE](#composer_include) in [Control Variables](#control-variables)). The values in the most local file override all others (see [Precedence Rules](#precedence-rules)).

The easiest way to create a new .composer.mk is with the [template](#template) target ([Quick Start](#quick-start) example):

make template >.composer.mk  
$EDITOR .composer.mk

All variable definitions must be in the override [variable] := [value] format from the [template](#template) target. Doing otherwise will result in unexpected behavior, and is not supported. The regular expression that is used to detect them:

override[[:space:]]+([^[:space:]]+)[[:space:]]+[:][=]

Variables can also be specified per-target, using [GNU Make](http://www.gnu.org/software/make) syntax (these are the settings used to process the [Composer](https://github.com/garybgenett/composer) README.\* files):

README.%: override c\_css := css\_alt  
README.%: override c\_toc := 0  
README.epub: override c\_css :=  
README.revealjs.html: override c\_css :=  
README.revealjs.html: override c\_toc :=

In this case, there are multiple definitions that could apply to README.revealjs.html, due to the % wildcard. Since the most specific target match is used, the final values for both [c\_css](#c_css) and [c\_toc](#c_toc) would be empty.

## 2.4 Precedence Rules

The order of precedence for .composer.mk files is global-to-local (see [COMPOSER\_INCLUDE](#composer_include) in [Control Variables](#control-variables)). This means that the values in the most local file override all others.

Variable aliases, such as COMPOSER\_DEBUGIT/c\_debug/V are prioritized in the order shown, with COMPOSER\_\* taking precedence over c\_\*, over the short alias.

Selection of the CSS file can be done using .composer.css or the [c\_css](#c_css) variable, with .composer.css taking precedence (unless [c\_css](#c_css) comes from .composer.mk). The process for .composer.css files is identical to .composer.mk (see [COMPOSER\_INCLUDE](#composer_include) in [Control Variables](#control-variables)).

All values in .composer.mk take precedence over everything else, including .composer.css and environment variables.

## 2.5 Specifying Dependencies

If there are files or directories that have dependencies on other files or directories being processed first, this can be done simply using [GNU Make](http://www.gnu.org/software/make) syntax in .composer.mk:

LICENSE.html: README.html  
all-subdirs-artifacts: all-subdirs-bootstrap

This would require README.html to be completed before LICENSE.html, and for bootstrap to be processed before artifacts. Directories need to be specified with the all-subdirs-\* syntax in order to avoid conflicts with target names (see [Custom Targets](#custom-targets)). Good examples of this are the internal [docs](#internal-targets) and [test](#internal-targets) targets, which are common directory names.

Chaining of dependencies can be as complex and layered as [GNU Make](http://www.gnu.org/software/make) will support. Note that if a file or directory is set to depend on a target, that target will be run whenever the file or directory is called.

## 2.6 Custom Targets

If needed, custom targets can be defined inside a .composer.mk file (see [Configuration Settings](#configuration-settings)), using standard [GNU Make](http://www.gnu.org/software/make) syntax. Naming them as [\*-clean](#clean--clean-all---clean) or [\*-all](#all--all-all---all) will include them in runs of the respective targets. Targets with any other names will need to be run manually, or included in [COMPOSER\_TARGETS](#composer_targets) (see [Control Variables](#control-variables)).

There are a few limitations when naming custom targets. Targets starting with the regular expression [\_.] are hidden, and are skipped by auto-detection. Additionally, there is a list of reserved targets in [Reserved](#reserved), along with a list of reserved variables.

Any included .composer.mk files are sourced early in the main [Composer](https://github.com/garybgenett/composer) Makefile, so matching targets and most variables will be overridden. In the case of conflicting targets, [GNU Make](http://www.gnu.org/software/make) will produce warning messages. Variables will have their values changed silently. Changing the values of internal [Composer](https://github.com/garybgenett/composer) variables is not recommended or supported.

A final note is that [\*-clean](#clean--clean-all---clean) and [\*-all](#all--all-all---all) targets are stripped from [COMPOSER\_TARGETS](#composer_targets). In cases where this results in an empty [COMPOSER\_TARGETS](#composer_targets), there will be a message and no actions will be taken.

## 2.7 Repository Versions

There are a few internal variables used by [\_update](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d) to select the repository and binary versions of integrated components (see [Requirements](#requirements)). These are exposed for configuration, but only within .composer.mk:

* PANDOC\_VER (must be a binary version number)
* PANDOC\_CMT (defaults to PANDOC\_VER)
* YQ\_VER (must be a binary version number)
* YQ\_CMT (defaults to YQ\_VER)
* BOOTSTRAP\_CMT
* MDVIEWER\_CMT
* REVEALJS\_CMT

Binaries for [Pandoc](http://www.johnmacfarlane.net/pandoc) and [YQ](https://mikefarah.gitbook.io/yq) are installed in their respective directories. By moving or removing them, or changing the version number and foregoing [\_update-all](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d) (see [Additional Targets](#additional-targets)), the system versions will be used instead. This will work as long as the commit versions match, so that supporting files are in alignment.

It is possible that changing the versions will introduce incompatibilities with [Composer](https://github.com/garybgenett/composer), which are usually impacts to the prettification of output files (see [Document Formatting](#document-formatting)).

# 3 Composer Variables

## 3.1 Formatting Variables

| Variable | Purpose | Value |
| --- | --- | --- |
| [c\_type](#c_type--c_base--c_list) ~ T | Desired output format | html |
| [c\_base](#c_type--c_base--c_list) ~ B | Base of output file | README |
| [c\_list](#c_type--c_base--c_list) ~ L | List of input files(s) | README.md |
| [c\_lang](#c_lang) ~ g | Language for document headers | en-US |
| [c\_css](#c_css) ~ s | Location of CSS file | (.composer.css) |
| [c\_toc](#c_toc) ~ c | Table of contents depth |  |
| [c\_level](#c_level) ~ l | Chapter/slide header level | 2 |
| [c\_margin](#c_margin) ~ m | Size of margins ([PDF](#pdf)) | 0.8in |
| [c\_options](#c_options) ~ o | Custom Pandoc options |  |

| Values: c\_type | Format | Extension |
| --- | --- | --- |
| html | HyperText Markup Language | \*.html |
| pdf | Portable Document Format | \*.pdf |
| epub | Electronic Publication | \*.epub |
| revealjs | Reveal.js Presentation | \*.revealjs.html |
| docx | Microsoft Word | \*.docx |
| pptx | Microsoft PowerPoint | \*.pptx |
| text | Plain Text (well-formatted) | \*.txt |
| markdown | Pandoc Markdown (for testing) | \*.md.txt |

* *Other* [*c\_type*](#c_type--c_base--c_list) *values will be passed directly to* [*Pandoc*](http://www.johnmacfarlane.net/pandoc)
* *Special values for* [*c\_css*](#c_css)*:*
  + *css\_alt ~ Use the alternate default stylesheet*
  + *0 ~ Revert to the* [*Pandoc*](http://www.johnmacfarlane.net/pandoc) *default*
* *Special value 0 for* [*c\_toc*](#c_toc) *~ List all headers, and number sections*
* *Special value 0 for* [*c\_level*](#c_level) *~ Varies by* [*c\_type*](#c_type--c_base--c_list) *(see* [*help-all*](#help--help-all)*)*
* *An empty* [*c\_margin*](#c_margin) *value enables individual margins:*
  + *c\_margin\_top ~ mt*
  + *c\_margin\_bottom ~ mb*
  + *c\_margin\_left ~ ml*
  + *c\_margin\_right ~ mr*

### 3.1.1 c\_type / c\_base / c\_list

The [compose](#compose) target uses these variables to decide what to build and how. The output file is [c\_base].<extension>, and is constructed from the [c\_list](#c_type--c_base--c_list) input files, in order. The <extension> is selected based on the [c\_type](#c_type--c_base--c_list) table above. Generally, it is not required to use the [compose](#compose) target directly for supported [c\_type](#c_type--c_base--c_list) files, since it is run automatically based on what output file <extension> is specified.

The automatic input file detection works by matching one of the following ([Quick Start](#quick-start) example):

make README.html ~ README (empty [COMPOSER\_EXT])  
make README.html ~ README.md  
make README.md.html ~ README.md  
make Composer-v3.0.Manual.html c\_list="README.md LICENSE.md"

Other values for [c\_type](#c_type--c_base--c_list), such as json or man, for example, can be passed through to [Pandoc](http://www.johnmacfarlane.net/pandoc) manually:

make compose c\_type="json" c\_base="README" c\_list="README.md"  
make compose c\_type="man" c\_base="Composer-v3.0.Manual" c\_list="README.md"

Any of the file types supported by [Pandoc](http://www.johnmacfarlane.net/pandoc) can be created this way. The only limitation is that the input files must be in [Markdown](http://daringfireball.net/projects/markdown) format.

### 3.1.2 c\_lang

* Primarily for [PDF](#pdf), this specifies the language that the table of contents ([c\_toc](#c_toc)) and chapter headings ([c\_level](#c_level)) will use.

### 3.1.3 c\_css

* By default, a CSS stylesheet from [Markdown Viewer](https://github.com/Thiht/markdown-viewer) is used for [HTML](#html) and [EPUB](#epub), and one of the [Reveal.js](https://revealjs.com) themes is used for [Reveal.js Presentations](#revealjs-presentations). This variable allows for selection of a different file in all cases.
* The special value css\_alt selects the alternate default stylesheet. Using 0 reverts to the [Pandoc](http://www.johnmacfarlane.net/pandoc) default.
* This value can be overridden by the presence of .composer.css files. See [Precedence Rules](#precedence-rules) for details.

### 3.1.4 c\_toc

* Setting this to a value of [1-6] creates a table of contents at the beginning of the document. The numerical value is how many header levels deep the table should go. A value of 6 lists all header levels.
* Using a value of 0 lists all header levels, and additionally numbers all the sections, for reference.

### 3.1.5 c\_level

* This value has different effects, depending on the [c\_type](#c_type--c_base--c_list) of the output document.
* For [HTML](#html), any value enables section-divs, which wraps headings and their section content in <section> tags and attaches identifiers to them instead of the headings themselves. This is for CSS styling, and is generally desired.
* For [PDF](#pdf), there are 3 top-level division types: part, chapter, and section. This sets the top-level header to the specified type, which changes the way the document is presented. Using part divides the document into “Parts”, each starting with a stand-alone title page. With this division type, each second-level heading starts a new “Chapter”. A chapter simply starts a new section on a new page, and lower-level headings continue as running portions within it. Finally, section creates one long running document with no blank pages or section breaks (like a [HTML](#html) page). To set the desired value:
  + part ~ 0
  + chapter ~ 2
  + section ~ Any other value
* For [EPUB](#epub), this creates chapter breaks at the specified level, starting the section on a new page. The special 0 simply sets it to the default value of 2.
* For [Reveal.js Presentations](#revealjs-presentations), the top-level headings can persist on the screen when moving through slides in their sections, or they can rotate out as their own individual slides. Setting to 0 enables persistent headings, and all other values use the default.
* An empty value defers to the [Pandoc](http://www.johnmacfarlane.net/pandoc) defaults in all cases.

### 3.1.6 c\_margin

* The default margins for [PDF](#pdf) are formatted for typesetting of printed books, where there is a large amount of open space around the edges and the text on each page is shifted away from where the binding would be. This is generally not what is desired in a purely digital [PDF](#pdf) document.
* This is one value for all the margins. Setting it to an empty value exposes variables for each of the individual margins: c\_margin\_top, c\_margin\_bottom, c\_margin\_left and c\_margin\_right.

### 3.1.7 c\_options

* In some cases, it may be desirable to add additional [Pandoc](http://www.johnmacfarlane.net/pandoc) options. Anything put in this variable will be passed directly to [Pandoc](http://www.johnmacfarlane.net/pandoc) as additional command-line arguments.

## 3.2 Control Variables

| Variable | Purpose | Value |
| --- | --- | --- |
| [MAKEJOBS](#makejobs) | Parallel processing threads | 1 (makejobs) |
| [COMPOSER\_DOCOLOR](#composer_docolor) | Enable title/color sequences | (boolean) |
| [COMPOSER\_DEBUGIT](#composer_debugit) | Use verbose output | (debugit) |
| [COMPOSER\_INCLUDE](#composer_include) | Include all: .composer.mk | (boolean) |
| [COMPOSER\_DEPENDS](#composer_depends) | Sub-directories first: [all](#all--all-all---all) | (boolean) |
| [COMPOSER\_LOG](#composer_log) | Timestamped command log | .composed |
| [COMPOSER\_EXT](#composer_ext) | Markdown file extension | .md |
| [COMPOSER\_TARGETS](#composer_targets) | See: [all](#all--all-all---all)/[clean](#clean--clean-all---clean) | [config](#X7cd51c86338f63207ee723d7b0c6010c67ea80b)/[targets](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) |
| [COMPOSER\_SUBDIRS](#composer_subdirs) | See: [all](#all--all-all---all)/[clean](#clean--clean-all---clean)/[install](#install--install-all--install-force) | [config](#X7cd51c86338f63207ee723d7b0c6010c67ea80b)/[targets](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) |
| [COMPOSER\_IGNORES](#composer_ignores) | See: [all](#all--all-all---all)/[clean](#clean--clean-all---clean)/[install](#install--install-all--install-force) | [config](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) |

* [*MAKEJOBS*](#makejobs) *~ c\_jobs ~ J*
* [*COMPOSER\_DOCOLOR*](#composer_docolor) *~ c\_color ~ C*
* [*COMPOSER\_DEBUGIT*](#composer_debugit) *~ c\_debug ~ V*
* *(makejobs) = empty is disabled / number of threads / 0 is no limit*
* *(debugit) = empty is disabled / any value enables / 0 is full tracing*
* *(boolean) = empty is disabled / any value enables*

### 3.2.1 MAKEJOBS

* By default, [Composer](https://github.com/garybgenett/composer) progresses linearly, doing one task at a time. If there are dependencies between items, this can be beneficial, since it ensures things will happen in a particular order. The downside, however, is that it is very slow.
* [Composer](https://github.com/garybgenett/composer) supports [GNU Make](http://www.gnu.org/software/make) parallel execution, where multiple threads can be working through tasks independently. Experiment with lower values first. When recursing through large directories, each make that instantiates into a sub-directory has it’s own jobs server, so the total number of threads running can proliferate rapidly.
* This can drastically speed up execution, processing thousands of files and directories in minutes. However, values that are too high can exhaust system resources. With great power comes great responsibility.
* A value of 0 does parallel execution with no thread limit.

### 3.2.2 COMPOSER\_DOCOLOR

* [Composer](https://github.com/garybgenett/composer) uses colors to make all output and [help](#help--help-all) text easier to read. The escape sequences used to accomplish this can create mixed results when reading in an output file or a $PAGER, or just make it harder to read for some.
* This is also used internally for targets like [debug-file](#debug--debug-file) and [template](#template), where plain text is required.

### 3.2.3 COMPOSER\_DEBUGIT

* Provides more explicit details about what is happening at each step. Produces a lot more output, and can be slower. It will also be hard to read unless [MAKEJOBS](#makejobs) is set to 1.
* Full tracing using 0 also displays [GNU Make](http://www.gnu.org/software/make) debugging output.
* *When doing* [*debug*](#debug--debug-file)*, this is used to pass a list of targets to test (see* [*Additional Targets*](#additional-targets)*).*

### 3.2.4 COMPOSER\_INCLUDE

* On every run, [Composer](https://github.com/garybgenett/composer) walks through the MAKEFILE\_LIST, all the way back to the main Makefile, looking for .composer.mk files in each directory. By default, it only reads the one in its main directory and the current directory, in that order. Enabling this causes all of them to be read.
* In the example directory tree below, normally the .composer.mk in .Composer is read first, and then tld/sub/.composer.mk. With this enabled, it will read all of them in order from top to bottom: .Composer/.composer.mk, .composer.mk, tld/.composer.mk, and finally tld/sub/.composer.mk.
* This is why it is best practice to have a .Composer directory at the top level for each documentation archive (see [Recommended Workflow](#recommended-workflow)). Not only does it allow for strict version control of [Composer](https://github.com/garybgenett/composer) per-archive, it also provides a mechanism for setting [Composer Variables](#composer-variables) globally.
* Care should be taken setting “Local” variables from [template](#template) (see [Templates](#templates)) when using this option. In that case, they will be propagated down the tree. This may be desired in some cases, but it will require that each directory set these manually, which could require a lot of maintenance.
* This setting also causes .composer.css files to be processed in an identical manner (see [Precedence Rules](#precedence-rules)).

Example directory tree (see [Recommended Workflow](#recommended-workflow)):

.../.Composer/Makefile  
.../.Composer/.composer.mk  
.../Makefile  
.../.composer.mk  
.../tld/Makefile  
.../tld/.composer.mk  
.../tld/sub/Makefile  
.../tld/sub/.composer.mk

### 3.2.5 COMPOSER\_DEPENDS

* When doing [all-all](#all--all-all---all), [Composer](https://github.com/garybgenett/composer) will process the current directory before recursing into sub-directories. This reverses that, and sub-directories will be processed first.
* In the example directory tree in [COMPOSER\_INCLUDE](#composer_include) above, the default would be: .../, .../tld, and then .../tld/sub. If the higher-level directories have dependencies on the sub-directories being run first, this will support that by doing them in reverse order, processing them from bottom to top.
* It should be noted that enabling this disables [MAKEJOBS](#makejobs), to ensure linear processing, and that it has no effect on [install](#install--install-all--install-force) or [clean](#clean--clean-all---clean).

### 3.2.6 COMPOSER\_LOG

* [Composer](https://github.com/garybgenett/composer) appends to a .composed log file in the current directory whenever it executes [Pandoc](http://www.johnmacfarlane.net/pandoc). This provides some accounting, and is used by [list](#list) to determine which \*.md files have been updated since the last time [Composer](https://github.com/garybgenett/composer) was run.
* This setting can change the name of the log file, or disable it completely (empty value).
* It is removed each time [clean](#clean--clean-all---clean) is run.

### 3.2.7 COMPOSER\_EXT

* The [Markdown](http://daringfireball.net/projects/markdown) file extension [Composer](https://github.com/garybgenett/composer) uses: \*.md. This is for auto-detection of files to add to [COMPOSER\_TARGETS](#composer_targets), files to output for [list](#list), and other tasks. This is a widely used standard, including [GitHub](https://github.com). Another commonly used extension is: \*.markdown.
* In some cases, they do not have any extension, such as README and LICENSE in source code directories. Setting this to an empty value causes them to be detected and output. It also causes all other files to be processed, because it becomes the wildcard \*, so use with care. It is likely best to use [COMPOSER\_TARGETS](#composer_targets) to explicitly set the targets list in these situations.

### 3.2.8 COMPOSER\_TARGETS

* The list of output files to create or delete with [clean](#clean--clean-all---clean) and [all](#all--all-all---all). [Composer](https://github.com/garybgenett/composer) does auto-detection using [c\_type](#c_type--c_base--c_list) and [COMPOSER\_EXT](#composer_ext), so this does not usually need to be set. Hidden files that start with . are skipped.
* Setting this manually disables auto-detection. It can also include non-file targets added into a .composer.mk file (see [Custom Targets](#custom-targets)).
* The .null target is special, and when used as a value for [COMPOSER\_TARGETS](#composer_targets) or [COMPOSER\_SUBDIRS](#composer_subdirs) it will display a message and do nothing. A side-effect of this target is that an actual file or directory named .null will never be created or removed by [Composer](https://github.com/garybgenett/composer).
* An empty value triggers auto-detection
* Use [config](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) or [targets](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) to check the current value.

### 3.2.9 COMPOSER\_SUBDIRS

* The list of sub-directories to recurse into with [install](#install--install-all--install-force), [clean](#clean--clean-all---clean), and [all](#all--all-all---all). The behavior and configuration is identical to [COMPOSER\_TARGETS](#composer_targets) above, including auto-detection and the .null target. Hidden directories that start with . are skipped.
* An empty value triggers auto-detection
* Use [config](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) or [targets](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) to check the current value.

### 3.2.10 COMPOSER\_IGNORES

* The list of [COMPOSER\_TARGETS](#composer_targets) and [COMPOSER\_SUBDIRS](#composer_subdirs) to skip with [install](#install--install-all--install-force), [clean](#clean--clean-all---clean), and [all](#all--all-all---all). This allows for selective auto-detection, when the list of items to process is larger than those to leave alone.
* Use [config](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) to check the current value.

# 4 Composer Targets

## 4.1 Primary Targets

| Target | Purpose |
| --- | --- |
| [help](#help--help-all) | Basic help overview (default) |
| [help-all](#help--help-all) | Console version of README.md (mostly identical) |
| [template](#template) | Print settings template: .composer.mk |
| [compose](#compose) | Document creation engine (see [Formatting Variables](#formatting-variables)) |
| [site](#site) | Recursively create [Bootstrap Websites](#bootstrap-websites) |
| [install](#install--install-all--install-force) | Current directory initialization: Makefile |
| [install-all](#install--install-all--install-force) | Do [install](#install--install-all--install-force) recursively (no overwrite) |
| [install-force](#install--install-all--install-force) | Recursively force overwrite of Makefile files |
| [clean](#clean--clean-all---clean) | Remove output files: [COMPOSER\_TARGETS](#composer_targets) :: [\*-clean](#clean--clean-all---clean) |
| [clean-all](#clean--clean-all---clean) | Do [clean](#clean--clean-all---clean) recursively: [COMPOSER\_SUBDIRS](#composer_subdirs) |
| [\*-clean](#clean--clean-all---clean) | Any targets named this way will also be run by [clean](#clean--clean-all---clean) |
| [all](#all--all-all---all) | Create output files: [COMPOSER\_TARGETS](#composer_targets) :: [\*-all](#all--all-all---all) |
| [all-all](#all--all-all---all) | Do [all](#all--all-all---all) recursively: [COMPOSER\_SUBDIRS](#composer_subdirs) |
| [\*-all](#all--all-all---all) | Any targets named this way will also be run by [all](#all--all-all---all) |
| [list](#list) | Print updated files: \*.md >> .composed |

### 4.1.1 help / help-all

* Outputs all of the documentation for [Composer](https://github.com/garybgenett/composer). The README.md has a few extra sections covering internal targets, along with reserved target and variable names, but is otherwise identical to the [help-all](#help--help-all) output.

### 4.1.2 template

* Prints a useful template for creating new .composer.mk files (see [Configuration Settings](#configuration-settings) and [Templates](#templates)).

### 4.1.3 compose

* This is the very core of [Composer](https://github.com/garybgenett/composer), and does the actual work of the [Pandoc](http://www.johnmacfarlane.net/pandoc) conversion. Details are in the [c\_type / c\_base / c\_list](#c_type--c_base--c_list) section of [Formatting Variables](#formatting-variables).

### 4.1.4 site

* *(This feature is reserved for a future release to create* [*Bootstrap Websites*](#bootstrap-websites)*. It will also include* [*page*](#page--post) *and* [*post*](#page--post) *from* [*Special Targets*](#special-targets)*.)*

### 4.1.5 install / install-all / install-force

* Creates the necessary Makefile files to set up a directory or a directory tree as a [Composer](https://github.com/garybgenett/composer) archive. By default, it will not overwrite any existing files.
* Doing a simple [install](#install--install-all--install-force) will only create a file in the current directory, whereas [install-all](#install--install-all--install-force) will recurse through the entire directory tree. A full [install-force](#install--install-all--install-force) is the same as [install-all](#install--install-all--install-force), with the exception that it will overwrite all Makefile files.
* The topmost directory will have the Makefile created for it modified to point to [Composer](https://github.com/garybgenett/composer).

### 4.1.6 clean / clean-all / \*-clean

* Deletes all [COMPOSER\_TARGETS](#composer_targets) output files in the current directory, after first running all [\*-clean](#clean--clean-all---clean) targets, including those for [Specials](#special-targets).
* Doing [clean-all](#clean--clean-all---clean) does the same thing recursively, through all the [COMPOSER\_SUBDIRS](#composer_subdirs).

### 4.1.7 all / all-all / \*-all

* Creates all [COMPOSER\_TARGETS](#composer_targets) output files in the current directory, after first running all [\*-all](#all--all-all---all) targets, including those for [Specials](#special-targets).
* Doing [all-all](#all--all-all---all) does the same thing recursively, through all the [COMPOSER\_SUBDIRS](#composer_subdirs).

### 4.1.8 list

* Outputs all the [COMPOSER\_EXT](#composer_ext) files that have been modified since [COMPOSER\_LOG](#composer_log) was last updated (see both in [Control Variables](#control-variables)). Acts as a quick reference to see if anything has changed.
* Since the [COMPOSER\_LOG](#composer_log) file is updated whenever [Pandoc](http://www.johnmacfarlane.net/pandoc) is executed, this target will primarily be useful when [all](#all--all-all---all) is the only target used to create files in the directory.

## 4.2 Special Targets

There are a few targets considered [Specials](#special-targets), that have unique properties:

| Base Name | Purpose |
| --- | --- |
| [book](#book) | Concatenate a source list into a single output file |
| [page](#page--post) | *(Reserved for the future* [*site*](#site) *feature)* |
| [post](#page--post) | *(Reserved for the future* [*site*](#site) *feature)* |

For each of these base names, there are a standard set of actual targets:

| Target | Purpose |
| --- | --- |
| %s-clean | Called by [clean](#clean--clean-all---clean), removes all %-\* files |
| %s-all | Called by [all](#all--all-all---all), creates all %-\* files |
| %s | Main target, which is a wrapper to %s-all |
| %-\* | Target files will be processed according to the base |

### 4.2.1 book

An example [book](#book) definition in a .composer.mk file ([Quick Start](#quick-start) example):

book-Composer-v3.0.Manual.html: README.md LICENSE.md

This configures it so that books will create Composer-v3.0.Manual.html from README.md and LICENSE.md, concatenated together in order. The primary purpose of this [Special](#special-targets) is to gather multiple source files in this manner, so that larger works can be comprised of multiple files, such as a book with each chapter in a separate file.

### 4.2.2 page / post

*(Both* [*page*](#page--post) *and* [*post*](#page--post) *are reserved for the future* [*site*](#site) *feature, which will build website pages using* [*Bootstrap*](https://getbootstrap.com)*.)*

## 4.3 Additional Targets

| Target | Purpose |
| --- | --- |
| [debug](#debug--debug-file) | Diagnostics, tests targets list in [COMPOSER\_DEBUGIT](#composer_debugit) |
| [debug-file](#debug--debug-file) | Export [debug](#debug--debug-file) results to a plain text file |
| [check](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) | List system packages and versions (see [Requirements](#requirements)) |
| [check-all](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) | Complete [check](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) package list, and system information |
| [config](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) | Show values of all [Composer Variables](#composer-variables) |
| [config-all](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) | Complete [config](#X7cd51c86338f63207ee723d7b0c6010c67ea80b), including environment variables |
| [targets](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) | List all available targets for the current directory |
| [\_commit](#Xb7fd9f4e9cd92dc66b184ee05783330e72d7768) | Timestamped [Git](https://git-scm.com) commit of the current directory tree |
| [\_commit-all](#Xb7fd9f4e9cd92dc66b184ee05783330e72d7768) | Automatic [\_commit](#Xb7fd9f4e9cd92dc66b184ee05783330e72d7768), without $EDITOR step |
| [\_release](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d) | Full upgrade to current release, repository preparation |
| [\_update](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d) | Update all included components (see [Requirements](#requirements)) |
| [\_update-all](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d) | Complete [\_update](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d), including binaries: [Pandoc](http://www.johnmacfarlane.net/pandoc), [YQ](https://mikefarah.gitbook.io/yq) |

### 4.3.1 debug / debug-file

* This is the tool to use for any support issues. Submit the output file to: [composer@garybgenett.net](mailto:composer@garybgenett.net?subject=Composer%20CMS%20Submission&body=Thank%20you%20for%20sending%20a%20message%21)
* Internally, it also runs:
  + [test](#internal-targets)
  + [check-all](#X7cd51c86338f63207ee723d7b0c6010c67ea80b)
  + [config-all](#X7cd51c86338f63207ee723d7b0c6010c67ea80b)
  + [targets](#X7cd51c86338f63207ee723d7b0c6010c67ea80b)
* If issues are occurring when running a particular set of targets, list them in [COMPOSER\_DEBUGIT](#composer_debugit).
* For general issues, run in the top-level directory (see [Recommended Workflow](#recommended-workflow)). For specific issues, run in the directory where the issue is occurring.

For example:

make COMPOSER\_DEBUGIT="books README.html" debug-file

### 4.3.2 check / check-all / config / config-all / targets

* Useful targets for validating tooling and configurations.
* Use [check](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) to see the list of components and their versions, in relation to the system installed versions. Doing [check-all](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) will show the complete list of tools that are used by [Composer](https://github.com/garybgenett/composer).
* The current values of all [Composer Variables](#composer-variables) is output by [config](#X7cd51c86338f63207ee723d7b0c6010c67ea80b), and [config-all](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) will additionally output all environment variables.
* A structured list of detected targets, available [Specials](#special-targets), [\*-clean](#clean--clean-all---clean) and [\*-all](#all--all-all---all) targets, [COMPOSER\_TARGETS](#composer_targets), and [COMPOSER\_SUBDIRS](#composer_subdirs) is printed by [targets](#X7cd51c86338f63207ee723d7b0c6010c67ea80b).
* Together, [config](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) and [targets](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) reveal the entire internal state of [Composer](https://github.com/garybgenett/composer).

### 4.3.3 \_commit / \_commit-all

* Using the directory structure in [Recommended Workflow](#recommended-workflow), .../ is considered the top-level directory. Meaning, it is the last directory before linking to [Composer](https://github.com/garybgenett/composer).
* If the top-level directory is a [Git](https://git-scm.com) repository (it has <directory>.git or <directory>/.git), this target creates a commit of the current directory tree with the title format below.
* For example, if it is run in the .../tld directory, that entire tree would be in the commit, including .../tld/sub. The purpose of this is to create quick and easy checkpoints when working on documentation that does not necessarily fit in a process where there are specific atomic steps being accomplished.
* When this target is run in a [Composer](https://github.com/garybgenett/composer) directory, it uses itself as the top-level directory.

Commit title format:

[Composer CMS v3.0 :: 2022-05-11T10:49:51-07:00]

### 4.3.4 \_release / \_update / \_update-all

* Using the repository configuration (see [Repository Versions](#repository-versions)), these fetch and install all external components.
* The [\_update-all](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d) target also fetches the [Pandoc](http://www.johnmacfarlane.net/pandoc) and [YQ](https://mikefarah.gitbook.io/yq) binaries, whereas [\_update](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d) only fetches the repositories.
* In addition to doing [\_update-all](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d), [\_release](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d) performs the steps necessary to turn the current directory into a complete clone of [Composer](https://github.com/garybgenett/composer).
* If rsync is installed, [\_release](#X9ce04701391ce92c6223e73c84c5d6ea8e0621d) can be used to rapidly replicate [Composer](https://github.com/garybgenett/composer), like below.
* One of the unique features of [Composer](https://github.com/garybgenett/composer) is that everything needed to compose itself is embedded in the Makefile.

Rapid cloning (requires rsync):

mkdir .../clone  
cd .../clone  
make -f .../.Composer/Makefile \_release

## 4.4 Internal Targets

| Target | Purpose |
| --- | --- |
| [help-force](#internal-targets) | Complete README.md content (similar to [help-all](#help--help-all)) |
| [.template-install](#internal-targets) | The Makefile used by [install](#install--install-all--install-force) (see [Templates](#templates)) |
| [.template](#internal-targets) | The .composer.mk used by [template](#template) (see [Templates](#templates)) |
| [docs](#internal-targets) | Extracts embedded files from Makefile, and does [all](#all--all-all---all) |
| [headers](#internal-targets) | Series of targets that handle all informational output |
| [headers-template](#internal-targets) | For testing default [headers](#internal-targets) output |
| [headers-template-all](#internal-targets) | For testing complete [headers](#internal-targets) output |
| [.make\_database](#internal-targets) | Complete contents of [GNU Make](http://www.gnu.org/software/make) internal state |
| [.all\_targets](#internal-targets) | Extracted list of all targets from [.make\_database](#internal-targets) |
| [.null](#internal-targets) | Placeholder to specify or detect empty values |
| [test](#internal-targets) | Test suite, validates all supported features |
| [test-file](#internal-targets) | Export [test](#internal-targets) results to a plain text file |
| [check-force](#internal-targets) | Minimized [check](#X7cd51c86338f63207ee723d7b0c6010c67ea80b) output (used for [Requirements](#requirements)) |
| [subdirs](#internal-targets) | Expands [COMPOSER\_SUBDIRS](#composer_subdirs) into \*-subdirs-\* targets |

*(None of these are intended to be run directly during normal use, and are only documented for completeness.)*

# 5 Reference

## 5.1 Templates

The [install](#install--install-all--install-force) target Makefile template (for reference only):

override COMPOSER\_MY\_PATH := $(abspath $(dir $(lastword $(MAKEFILE\_LIST))))  
override COMPOSER\_TEACHER := $(abspath $(dir $(COMPOSER\_MY\_PATH)))/Makefile  
include $(COMPOSER\_TEACHER)

Use the [template](#template) target to create .composer.mk files:

# >> Global  
# override MAKEJOBS := 1  
# override COMPOSER\_DOCOLOR :=  
# override COMPOSER\_DEBUGIT :=  
# override COMPOSER\_INCLUDE :=  
# override COMPOSER\_DEPENDS :=  
# override COMPOSER\_LOG := .composed  
# override COMPOSER\_EXT := .md  
# override c\_type := html  
# override c\_lang := en-US  
# override c\_css :=  
# override c\_toc :=  
# override c\_level := 2  
# override c\_margin := 0.8in  
# override c\_margin\_top :=  
# override c\_margin\_bottom :=  
# override c\_margin\_left :=  
# override c\_margin\_right :=  
# override c\_options :=  
# >> Local  
# override COMPOSER\_TARGETS := README.html README.pdf README.epub README.revealjs.html README.docx  
# override COMPOSER\_SUBDIRS :=  
# override COMPOSER\_IGNORES :=  
# override c\_base := README  
# override c\_list := README.md  
# >> Special  
# book-Composer.book.html: README.md LICENSE.md  
# page-Composer.page.html: README.md LICENSE.md  
# post-Composer.post.html: README.md LICENSE.md

## 5.2 Reserved

Reserved target names, including use as prefixes:

.all\_targets  
.make\_database  
.null  
all  
book  
books  
check  
clean  
compose  
config  
debug  
docs  
headers  
help  
install  
list  
page  
pages  
post  
posts  
site  
subdirs  
targets  
template  
test  
\_commit  
\_release  
\_update

Reserved variable names:

~  
7Z  
7Z\_VER  
BASE64  
BASH  
BASH\_VER  
BOOTSTRAP\_CMT  
BOOTSTRAP\_DIR  
BOOTSTRAP\_LIC  
BOOTSTRAP\_SRC  
CAT  
CHECKIT  
CHMOD  
CLEANER  
CODEBLOCK  
COLUMNS  
COLUMN\_2  
COMMENTED  
COMPOSER  
COMPOSER\_ART  
COMPOSER\_BASENAME  
COMPOSER\_COMPOSER  
COMPOSER\_CONTENTS  
COMPOSER\_CONTENTS\_DIRS  
COMPOSER\_CONTENTS\_FILES  
COMPOSER\_CSS  
COMPOSER\_DEBUGIT  
COMPOSER\_DEBUGIT\_ALL  
COMPOSER\_DEPENDS  
COMPOSER\_DIR  
COMPOSER\_DOCOLOR  
COMPOSER\_DOITALL\_check  
COMPOSER\_DOITALL\_clean  
COMPOSER\_DOITALL\_config  
COMPOSER\_DOITALL\_\_commit  
COMPOSER\_DOITALL\_debug  
COMPOSER\_DOITALL\_all  
COMPOSER\_DOITALL\_install  
COMPOSER\_DOITALL\_test  
COMPOSER\_DOITALL\_\_update  
COMPOSER\_EXPORTED  
COMPOSER\_EXPORTED\_NOT  
COMPOSER\_EXT  
COMPOSER\_EXT\_DEFAULT  
COMPOSER\_FILENAME  
COMPOSER\_FIND  
COMPOSER\_FULLNAME  
COMPOSER\_HEADLINE  
COMPOSER\_IGNORES  
COMPOSER\_INCLUDE  
COMPOSER\_INCLUDES  
COMPOSER\_INCLUDES\_LIST  
COMPOSER\_LICENSE  
COMPOSER\_LOG  
COMPOSER\_LOG\_DEFAULT  
COMPOSER\_MY\_PATH  
COMPOSER\_NOTHING  
COMPOSER\_OPTIONS  
COMPOSER\_PANDOC  
COMPOSER\_PKG  
COMPOSER\_REGEX  
COMPOSER\_REGEX\_DEFINE  
COMPOSER\_REGEX\_EVAL  
COMPOSER\_REGEX\_OVERRIDE  
COMPOSER\_REGEX\_PREFIX  
COMPOSER\_RELEASE  
COMPOSER\_RESERVED  
COMPOSER\_RESERVED\_SPECIAL  
COMPOSER\_RESERVED\_SPECIAL\_TARGETS  
COMPOSER\_ROOT  
COMPOSER\_SETTINGS  
COMPOSER\_SRC  
COMPOSER\_SUBDIRS  
COMPOSER\_TAGLINE  
COMPOSER\_TARGETS  
COMPOSER\_TEACHER  
COMPOSER\_TECHNAME  
COMPOSER\_TIMESTAMP  
COMPOSER\_TMP  
COMPOSER\_VERSION  
CONFIGS  
CONVICT  
COREUTILS\_VER  
CP  
CREATOR  
CSS\_ALT  
DATE  
DATEMARK  
DATENAME  
DATESTAMP  
DEBUGIT  
DEPTH\_DEFAULT  
DEPTH\_MAX  
DESC\_DOCX  
DESC\_EPUB  
DESC\_HTML  
DESC\_LINT  
DESC\_LPDF  
DESC\_PPTX  
DESC\_PRES  
DESC\_TEXT  
DIFF  
DIFFUTILS\_VER  
DISTRIB  
DIST\_ICON\_v1.0  
DIST\_SCREENSHOT\_v1.0  
DIST\_SCREENSHOT\_v3.0  
DIVIDE  
DOFORCE  
DOITALL  
DOMAKE  
DO\_BOOK  
DO\_HEREDOC  
DO\_PAGE  
DO\_POST  
ECHO  
ENDOLINE  
ENV  
EXAMPLE  
EXPR  
EXTENSION  
EXTN\_DEFAULT  
EXTN\_DOCX  
EXTN\_EPUB  
EXTN\_HTML  
EXTN\_LINT  
EXTN\_LPDF  
EXTN\_PPTX  
EXTN\_PRES  
EXTN\_TEXT  
FIND  
FINDUTILS\_VER  
GIT  
GIT\_OPTS\_CONVICT  
GIT\_REPO  
GIT\_REPO\_DO  
GIT\_RUN  
GIT\_RUN\_COMPOSER  
GIT\_VER  
GZIP\_BIN  
GZIP\_VER  
HEAD  
HEADERS  
HEADER\_L  
HEAD\_MAIN  
HELPOUT  
HEREDOC\_GITIGNORE  
HEREDOC\_LICENSE  
HEREDOC\_REVEALJS\_CSS  
HEREDOC\_TEX\_PDF\_TEMPLATE  
HEREDOC\_docs\_.composer.mk  
INPUT  
INSTALL  
LESS\_BIN  
LESS\_VER  
LINERULE  
LISTING  
LN  
LS  
MAKEFILE  
MAKEFILE\_LIST  
MAKEFLAGS  
MAKEJOBS  
MAKEJOBS\_DEFAULT  
MAKEJOBS\_OPTS  
MAKE\_DB  
MAKE\_OPTIONS  
MAKE\_VER  
MARKER  
MDVIEWER\_CMT  
MDVIEWER\_CSS  
MDVIEWER\_CSS\_ALT  
MDVIEWER\_DIR  
MDVIEWER\_LIC  
MDVIEWER\_SRC  
MKDIR  
MV  
NEWLINE  
NOTHING  
NOTHING\_IGNORES  
NULL  
OS\_TYPE  
OS\_UNAME  
OUTPUT  
OUTPUT\_FILENAME  
OUT\_LICENSE  
OUT\_MANUAL  
OUT\_README  
PANDOC  
PANDOC\_BIN  
PANDOC\_CMT  
PANDOC\_CMT\_DISPLAY  
PANDOC\_DIR  
PANDOC\_EXTENSIONS  
PANDOC\_LIC  
PANDOC\_LNX\_BIN  
PANDOC\_LNX\_DST  
PANDOC\_LNX\_SRC  
PANDOC\_MAC\_BIN  
PANDOC\_MAC\_DST  
PANDOC\_MAC\_SRC  
PANDOC\_OPTIONS  
PANDOC\_OPTIONS\_DATA  
PANDOC\_OPTIONS\_ERROR  
PANDOC\_SRC  
PANDOC\_TEX\_PDF  
PANDOC\_URL  
PANDOC\_VER  
PANDOC\_WIN\_BIN  
PANDOC\_WIN\_DST  
PANDOC\_WIN\_SRC  
PATH\_LIST  
PRINT  
PRINTER  
PRINTF  
PUBLISH  
READ\_ALIASES  
REALMAKE  
REALPATH  
REVEALJS\_CMT  
REVEALJS\_CSS  
REVEALJS\_CSS\_THEME  
REVEALJS\_DIR  
REVEALJS\_LIC  
REVEALJS\_LOGO  
REVEALJS\_SRC  
RM  
RSYNC  
RSYNC\_VER  
RUNMAKE  
SED  
SED\_VER  
SHELL  
SORT  
SOURCE\_INCLUDES  
SPECIAL\_VAL  
SUBDIRS  
TABLE\_C2  
TABLE\_M2  
TABLE\_M3  
TAIL  
TAR  
TARGETS  
TAR\_VER  
TEE  
TESTING  
TESTING\_COMPOSER\_DIR  
TESTING\_COMPOSER\_MAKEFILE  
TESTING\_DIR  
TESTING\_ENV  
TESTING\_LOGFILE  
TEX\_PDF  
TEX\_PDF\_TEMPLATE  
TEX\_PDF\_VER  
TITLE\_LN  
TMPL\_DOCX  
TMPL\_EPUB  
TMPL\_HTML  
TMPL\_LINT  
TMPL\_LPDF  
TMPL\_PPTX  
TMPL\_PRES  
TMPL\_TEXT  
TOKEN  
TR  
TRUE  
TYPE\_DEFAULT  
TYPE\_DOCX  
TYPE\_DO\_BOOK  
TYPE\_DO\_PAGE  
TYPE\_DO\_POST  
TYPE\_EPUB  
TYPE\_HTML  
TYPE\_LINT  
TYPE\_LPDF  
TYPE\_PPTX  
TYPE\_PRES  
TYPE\_TARGETS  
TYPE\_TEXT  
UNAME  
UPGRADE  
VIM\_FOLDING  
VIM\_OPTIONS  
WC  
WGET  
WGET\_PACKAGE  
WGET\_PACKAGE\_DO  
WGET\_VER  
XARGS  
YQ  
YQ\_BIN  
YQ\_CMT  
YQ\_CMT\_DISPLAY  
YQ\_DIR  
YQ\_LIC  
YQ\_LNX\_BIN  
YQ\_LNX\_DST  
YQ\_LNX\_SRC  
YQ\_MAC\_BIN  
YQ\_MAC\_DST  
YQ\_MAC\_SRC  
YQ\_SRC  
YQ\_URL  
YQ\_VER  
YQ\_WIN\_BIN  
YQ\_WIN\_DST  
YQ\_WIN\_SRC  
c\_base  
c\_css  
c\_css\_select  
c\_lang  
c\_level  
c\_list  
c\_margin  
c\_margin\_bottom  
c\_margin\_left  
c\_margin\_right  
c\_margin\_top  
c\_options  
c\_toc  
c\_type  
template-print  
template-var  
template-var-static  
headers  
headers-dir  
headers-file  
headers-list  
headers-note  
headers-release  
headers-rm  
headers-run  
headers-skip  
headers-vars  
headers-compose  
headers-subdirs  
help-force-targets-FORMAT  
help-force-targets-SECTIONS  
help-force-targets-TITLES  
help-all-CUSTOM  
help-all-DEPENDS  
help-all-FORMAT  
help-all-GOALS  
help-all-LINKS  
help-all-LINKS\_EXT  
help-all-ORDERS  
help-all-OVERVIEW  
help-all-REQUIRE  
help-all-REQUIRE\_POST  
help-all-SECTION  
help-all-SETTINGS  
help-all-TARGETS\_ADDITIONAL  
help-all-TARGETS\_INTERNAL  
help-all-TARGETS\_PRIMARY  
help-all-TARGETS\_SPECIALS  
help-all-TITLE  
help-all-VARIABLES\_CONTROL  
help-all-VARIABLES\_FORMAT  
help-all-VERSIONS  
help-all-WORKFLOW  
install-Makefile  
subdirs-template  
targets-list  
test-COMPOSER\_INCLUDE-done  
test-COMPOSER\_INCLUDE-init  
test-count  
test-done  
test-fail  
test-find  
test-hold  
test-init  
test-load  
test-log  
test-make  
test-mark  
test-pwd  
test-run  
test-speed-init  
test-speed-init-load  
test-headers  
\_C  
\_D  
\_E  
\_F  
\_H  
\_M  
\_N  
\_S

*Happy Making!*