

# Package ‘dockerfiler’

July 6, 2022

**Title** Easy Dockerfile Creation from R

**Version** 0.2.0

**Description** Build a Dockerfile straight from your R session.  
'dockerfiler' allows you to create step by step a Dockerfile, and provide convenient tools to wrap R code inside this Dockerfile.

**License** MIT + file LICENSE

**URL** <https://github.com/ThinkR-open/dockerfiler>

**BugReports** <https://github.com/ThinkR-open/dockerfiler/issues>

**Imports** attempt (>= 0.3.1), cli (>= 2.3.0), desc (>= 1.2.0), fs (>= 1.5.0), glue (>= 1.4.2), jsonlite (>= 1.7.2), pak (>= 0.2.0), pkgbuild (>= 1.2.0), R6 (>= 2.5.0), remotes (>= 2.2.0), renv (>= 0.12.0), usethis (>= 2.0.1), utils

**Suggests** knitr (>= 1.31), rmarkdown (>= 2.6), testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Colin Fay [cre, aut] (<<https://orcid.org/0000-0001-7343-1846>>),  
Vincent Guyader [aut] (<<https://orcid.org/0000-0003-0671-9270>>),  
Josiah Parry [aut] (<<https://orcid.org/0000-0001-9910-865X>>),  
Sébastien Rochette [aut] (<<https://orcid.org/0000-0002-1565-9313>>)

**Maintainer** Colin Fay <contact@colinfay.me>

**Repository** CRAN

**Date/Publication** 2022-07-06 20:30:10 UTC

## R topics documented:

Dockerfile . . . . .	2
docker_ignore_add . . . . .	6

dock_from_desc . . . . .	6
dock_from_renv . . . . .	7
get_sysreqs . . . . .	8
r . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

Dockerfile	<i>A Dockerfile template</i>
------------	------------------------------

---

### Description

A Dockerfile template

A Dockerfile template

### Value

A dockerfile template

### Methods

RUN add a RUN command

ADD add a ADD command

COPY add a COPY command

WORKDIR add a WORKDIR command

EXPOSE add an EXPOSE command

VOLUME add a VOLUME command

CMD add a CMD command

LABEL add a LABEL command

ENV add a ENV command

ENTRYPOINT add a ENTRYPOINT command

VOLUME add a VOLUME command

USER add a USER command

ARG add an ARG command

ONBUILD add a ONBUILD command

STOPSIGNAL add a STOPSIGNAL command

HEALTHCHECK add a HEALTHCHECK command

STOPSIGNAL add a STOPSIGNAL command

SHELL add a SHELL command

MAINTAINER add a MAINTAINER command

custom add a custom command

write save the Dockerfile

switch\_cmd switch two command

remove\_cmd remove\_cmd one or more command(s)

**Methods****Public methods:**

- `Dockerfile$new()`
- `Dockerfile$RUN()`
- `Dockerfile$ADD()`
- `Dockerfile$COPY()`
- `Dockerfile$WORKDIR()`
- `Dockerfile$EXPOSE()`
- `Dockerfile$VOLUME()`
- `Dockerfile$CMD()`
- `Dockerfile$LABEL()`
- `Dockerfile$ENV()`
- `Dockerfile$ENTRYPOINT()`
- `Dockerfile$USER()`
- `Dockerfile$ARG()`
- `Dockerfile$ONBUILD()`
- `Dockerfile$STOPSIGNAL()`
- `Dockerfile$HEALTHCHECK()`
- `Dockerfile$SHELL()`
- `Dockerfile$MAINTAINER()`
- `Dockerfile$custom()`
- `Dockerfile$print()`
- `Dockerfile$write()`
- `Dockerfile$switch_cmd()`
- `Dockerfile$remove_cmd()`
- `Dockerfile$add_after()`
- `Dockerfile$clone()`

**Method new():**

*Usage:*

`Dockerfile$new(FROM = "rocker/r-base", AS = NULL)`

**Method RUN():**

*Usage:*

`Dockerfile$RUN(cmd)`

**Method ADD():**

*Usage:*

`Dockerfile$ADD(from, to, force = TRUE)`

**Method COPY():**

*Usage:*

`Dockerfile$COPY(from, to, force = TRUE)`

**Method WORKDIR():***Usage:*

Dockerfile\$WORKDIR(when)

**Method EXPOSE():***Usage:*

Dockerfile\$EXPOSE(port)

**Method VOLUME():***Usage:*

Dockerfile\$VOLUME(volume)

**Method CMD():***Usage:*

Dockerfile\$CMD(cmd)

**Method LABEL():***Usage:*

Dockerfile\$LABEL(key, value)

**Method ENV():***Usage:*

Dockerfile\$ENV(key, value)

**Method ENTRYPOINT():***Usage:*

Dockerfile\$ENTRYPOINT(cmd)

**Method USER():***Usage:*

Dockerfile\$USER(user)

**Method ARG():***Usage:*

Dockerfile\$ARG(arg, ahead = FALSE)

**Method ONBUILD():***Usage:*

Dockerfile\$ONBUILD(cmd)

**Method STOPSIGNAL():***Usage:*

Dockerfile\$STOPSIGNAL(signal)

**Method HEALTHCHECK():**

*Usage:*

Dockerfile\$HEALTHCHECK(check)

**Method** SHELL():

*Usage:*

Dockerfile\$SHELL(shell)

**Method** MAINTAINER():

*Usage:*

Dockerfile\$MAINTAINER(name, email)

**Method** custom():

*Usage:*

Dockerfile\$custom(base, cmd)

**Method** print():

*Usage:*

Dockerfile\$print()

**Method** write():

*Usage:*

Dockerfile\$write(as = "Dockerfile")

**Method** switch\_cmd():

*Usage:*

Dockerfile\$switch\_cmd(a, b)

**Method** remove\_cmd():

*Usage:*

Dockerfile\$remove\_cmd(wher)

**Method** add\_after():

*Usage:*

Dockerfile\$add\_after(cmd, after)

**Method** clone(): The objects of this class are cloneable with this method.

*Usage:*

Dockerfile\$clone(deep = FALSE)

*Arguments:*

deep Whether to make a deep clone.

## Examples

```
my_dock <- Dockerfile$new()
```

`docker_ignore_add`      *Create a dockerignore file*

---

### **Description**

Create a dockerignore file

### **Usage**

```
docker_ignore_add(path)
```

### **Arguments**

`path`                      Where to write the file

### **Value**

The path to the `‘.dockerignore‘` file, invisibly.

### **Examples**

```
if (interactive()) {  
  docker_ignore_add()  
}
```

---

`dock_from_desc`              *Create a Dockerfile from a DESCRIPTION*

---

### **Description**

Create a Dockerfile from a DESCRIPTION

### **Usage**

```
dock_from_desc(  
  path = "DESCRIPTION",  
  FROM = paste0("rocker/r-ver:", R.Version()$major, ".", R.Version()$minor),  
  AS = NULL,  
  sysreqs = TRUE,  
  repos = c(CRAN = "https://cran.rstudio.com/"),  
  expand = FALSE,  
  update_tar_gz = TRUE,  
  build_from_source = TRUE,  
  extra_sysreqs = NULL  
)
```

**Arguments**

path	path to the DESCRIPTION file to use as an input.
FROM	The FROM of the Dockerfile. Default is FROM rocker/r-ver: 'R.Version()\$major'. 'R.Version()\$minor'.
AS	The AS of the Dockerfile. Default it NULL.
sysreqs	boolean. If TRUE, the Dockerfile will contain sysreq installation.
repos	character. The URL(s) of the repositories to use for 'options("repos")'.
expand	boolean. If 'TRUE' each system requirement will have its own 'RUN' line.
update_tar_gz	boolean. If 'TRUE' and 'build_from_source' is also 'TRUE', an updated tar.gz is created.
build_from_source	boolean. If 'TRUE' no tar.gz is created and the Dockerfile directly mount the source folder.
extra_sysreqs	character vector. Extra debian system requirements. Will be installed with apt-get install.

---

dock_from_renv	<i>Create a Dockerfile from an 'renv.lock' file</i>
----------------	---

---

**Description**

Create a Dockerfile from an 'renv.lock' file

**Usage**

```
dock_from_renv(
  lockfile = "renv.lock",
  distro = "focal",
  FROM = "rocker/r-base",
  AS = NULL,
  sysreqs = TRUE,
  repos = c(CRAN = "https://cran.rstudio.com/"),
  expand = FALSE,
  extra_sysreqs = NULL
)
```

**Arguments**

lockfile	Path to an 'renv.lock' file to use as an input..
distro	One of "focal", "bionic", "xenial", "centos7", or "centos8". See available distributions at <a href="https://hub.docker.com/r/rstudio/r-base/">https://hub.docker.com/r/rstudio/r-base/</a> .
FROM	Docker image to start FROM Default is FROM rocker/r-base
AS	The AS of the Dockerfile. Default it NULL.
sysreqs	boolean. If 'TRUE', the Dockerfile will contain sysreq installation.

repos	character. The URL(s) of the repositories to use for 'options("repos")'.
expand	boolean. If 'TRUE' each system requirement will have its own 'RUN' line.
extra_sysreqs	character vector. Extra debian system requirements. Will be installed with apt-get install.

### Details

System requirements for packages are provided through RStudio Package Manager via the pak package. The install commands provided from pak are added as 'RUN' directives within the 'Dockerfile'.

The R version is taken from the 'renv.lock' file. Packages are installed using 'renv::restore()' which ensures that the proper package version and source is used when installed.

### Value

A R6 object of class 'Dockerfile'.

### Examples

```
## Not run:
dock <- dock_from_renv("renv.lock", distro = "xenial")
dock$write("Dockerfile")

## End(Not run)
```

---

get_sysreqs	<i>Get system requirements</i>
-------------	--------------------------------

---

### Description

This function retrieves information about the system requirements using the <<https://sysreqs.r-hub.io>> API.

### Usage

```
get_sysreqs/packages, quiet = TRUE, batch_n = 30)
```

### Arguments

packages	character vector. Packages names.
quiet	Boolean. If 'TRUE' the function is quiet.
batch_n	numeric. Number of simultaneous packages to ask.

### Value

A vector of system requirements.



---

r *Turn an R call into an Unix call*

---

**Description**

Turn an R call into an Unix call

**Usage**

```
r(code)
```

**Arguments**

code            the function to call

**Value**

an unix R call

**Examples**

```
r(print("yeay"))  
r(install.packages("plumber", repo = "http://cran.irsn.fr/"))
```

# Index

dock\_from\_desc, [6](#)  
dock\_from\_renv, [7](#)  
docker\_ignore\_add, [6](#)  
Dockerfile, [2](#)  
  
get\_sysreqs, [8](#)  
  
r, [9](#)