

# Package ‘reportfactory’

May 9, 2026

**Title** Lightweight Infrastructure for Handling Multiple R Markdown Documents

**Version** 0.4.0

**Description** Provides an infrastructure for handling multiple R Markdown reports, including automated curation and time-stamping of outputs, parameterisation and provision of helper functions to manage dependencies.

**License** MIT + file LICENSE

**URL** <https://github.com/reconverse/reportfactory>

**BugReports** <https://github.com/reconverse/reportfactory/issues>

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Suggests** testthat, covr

**Imports** rprojroot, fs, rmarkdown, utils, yaml, callr, rstudioapi, knitr

**SystemRequirements** pandoc (>= 1.14) - <http://pandoc.org>

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Thibaut Jombart [aut],  
Amy Gimma [ctb],  
Tim Taylor [aut, cre] (ORCID: <<https://orcid.org/0000-0002-8587-7113>>)

**Maintainer** Tim Taylor <[tim.taylor@hidden elephants.co.uk](mailto:tim.taylor@hidden elephants.co.uk)>

**Repository** CRAN

**Date/Publication** 2021-08-09 12:30:02 UTC

## Contents

compile_reports . . . . .	2
factory_overview . . . . .	3
install_deps . . . . .	3
list_deps . . . . .	4

list_outputs	5
list_reports	5
new_factory	6
validate_factory	8

## Index 9

---

compile_reports	<i>Compile one or several R Markdown reports</i>
-----------------	--

---

### Description

Compile one or several R Markdown reports

### Usage

```
compile_reports(
  reports = NULL,
  factory = ".",
  ignore.case = FALSE,
  params = NULL,
  quiet = TRUE,
  subfolder = NULL,
  timestamp = format(Sys.time(), "%Y-%m-%d_T%H-%M-%S"),
  ...
)
```

### Arguments

reports	Either a regular expression (passed directly to <code>grep()</code> ) that matches to the report paths you would like to compile or an integer/logical vector. If reports is an integer or logical vector then a call of <code>compile_reports(factory, reports = idx)</code> is equivalent to <code>compile_reports(factory, list_reports(factory)[idx])</code> .
factory	The path to the report factory or a folder within the desired factory. Defaults to the current directory.
ignore.case	if FALSE (default), the report pattern matching is case sensitive and if TRUE, case is ignored during matching.
params	A named list of parameters to be used for compiling reports, passed to <code>rmarkdown::render()</code> as the <code>params</code> argument. Values specified here will take precedence over default values specified in YAML headers of the reports. Note that the set of parameter is used for all compiled reports.
quiet	A logical indicating if messages from R Markdown compilation should be displayed; TRUE by default.
subfolder	Name of subfolder to store results. Not required but helps distinguish output if mapping over multiple parameters. If provided, "subfolder" will be placed before the timestamp when storing compilation outputs.

timestamp      A character indicating the date-time format to be used for timestamps. Timestamps are used in the folder structure of outputs. If NULL, the format format(Sys.time(), "%Y-%m-%d\_T%H-%M-%S") will be used. Note that the timestamp corresponds to the time of the call to compile\_reports(), so that multiple reports compiled using a single call to the function will have identical timestamps.

...              further arguments passed to rmarkdown::render()

**Value**

Invisible NULL (called for side effects only).

---

factory_overview	<i>Generate an overview of a report factory</i>
------------------	---

---

**Description**

Print contents of directories in a tree-like format.

**Usage**

```
factory_overview(path = ".", ...)
```

**Arguments**

path            The path to the report factory or a folder within the desired factory. Defaults to the current directory.

...            Additional arguments passed to fs::dir\_tree().

**Value**

Invisibly returns a character of the files and directories within the desired folder.

---

install_deps	<i>Install dependencies of reports within</i>
--------------	---

---

**Description**

This function can be used to install package dependencies based on the reports within the factory.

**Usage**

```
install_deps(factory = ".", update = FALSE, ...)
```

**Arguments**

factory	The path to the report factory or a folder within the desired factory. Defaults to the current directory.
update	A logical indicating if packages which are already installed should be re-installed (TRUE); otherwise, only missing packages are installed; defaults to FALSE.
...	Arguments to be passed to <code>install.packages()</code> .

**Value**

Invisible NULL (called for side effects only).

**See Also**

`list_deps()` to list dependencies of packages

---

list_deps	<i>List dependencies of reports within a factory</i>
-----------	--

---

**Description**

List package dependencies based on the reports and scripts within the `report_sources` and `scripts` directories respectively.

**Usage**

```
list_deps(
  factory = ".",
  missing = FALSE,
  check_r = TRUE,
  exclude_readme = TRUE,
  parse_first = FALSE
)
```

**Arguments**

factory	The path to the report factory or a folder within the desired factory. Defaults to the current directory.
missing	A logical indicating if only missing dependencies should be listed (TRUE); otherwise, all packages needed in the reports are listed; defaults to FALSE.
check_r	If true, R scripts contained within the factory will also be checked. Note that this will error if the script cannot be parsed.
exclude_readme	If TRUE (default) README files will not be checked for dependencies.
parse_first	If TRUE code will first be parsed for validity and unevaluated Rmd chunks will not be checked for dependencies. The default value is FALSE and, in this case, files will simply be checked line by line for calls to <code>library</code> , <code>require</code> or use of <code>double</code> , <code>::</code> , and <code>triple</code> , <code>:::</code> function calls.

**Value**

A character vector of package dependencies.

**Note**

This function requires that any R scripts present in the factory are valid syntax else the function will error.

---

list_outputs	<i>List outputs of the factory</i>
--------------	------------------------------------

---

**Description**

This function can be used to list available report outputs stored in the factory, in inside the outputs folder (or any subfolder within).

**Usage**

```
list_outputs(factory = ".", pattern = NULL, ...)
```

**Arguments**

factory	The path to the report factory or a folder within the desired factory. Defaults to the current directory.
pattern	An optional regular expression used to look for specific patterns in report names.
...	additional parameters to pass to grep()

**Value**

A character vector containing the names of the reports in the specified factory (empty if there were no files).

---

list_reports	<i>List reports in the factory</i>
--------------	------------------------------------

---

**Description**

This function can be used to list available reports stored in the factory, in inside the report\_sources folder (or any subfolder within).

**Usage**

```
list_reports(factory = ".", pattern = NULL, ...)
```

**Arguments**

factory	The path to the report factory or a folder within the desired factory. Defaults to the current directory.
pattern	An optional regular expression used to look for specific patterns in report names.
...	additional parameters to pass to grep()

**Value**

A character vector containing the names of the reports in the specified factory (empty if there were no files).

---

new_factory	<i>Create a new report factory</i>
-------------	------------------------------------

---

**Description**

This function can be used to create a new report factory. By default, the factory is created with a template of report, and the working environment is moved to the newly created factory.

**Usage**

```
new_factory(
  factory = "new_factory",
  path = ".",
  report_sources = "report_sources",
  outputs = "outputs",
  move_in = TRUE,
  create_README = TRUE,
  create_example_report = TRUE,
  create_data_folders = TRUE,
  create_scripts_folder = TRUE,
  use_here = TRUE,
  use_rproj = TRUE,
  create_gitignore = TRUE
)
```

**Arguments**

factory	The name of the report factory to be created.
path	The folder where the report factory should be created. This will default to the current directory.
report_sources	The name of the folder to be used for report templates; defaults to 'report_sources/'.
outputs	The name of the folder to be used for saving the built reports; defaults to 'outputs/'.

move_in	A logical indicating if the current session should move into the created factory; defaults to TRUE. If use_rproj is also TRUE and RStudio is being used then the corresponding project will be opened.
create_README	A logical indicating if a 'README' file should be created; defaults to TRUE.
create_example_report	A logical indicating if new_factory() should create an example report in the 'report_sources' folder along with some example data in the 'data/raw' folder; defaults to TRUE.
create_data_folders	a logical indicating if new_factory() should create folders to store data; defaults to TRUE.
create_scripts_folder	a logical indicating if new_factory() should create folders to store R scripts; defaults to TRUE.
use_here	a logical indicating if new_factory() should create a .here file that can be used with here::here(); defaults to TRUE.
use_rproj	a logical indicating if new_factory() should create a .Rproj file that can be used with RStudio; defaults to TRUE.
create_gitignore	a logical indicating if new_factory() should create a minimal '.gitignore' file; defaults to TRUE.

## Details

Assuming the default names are used then new\_factory will create a report factory folder (called "new\_factory") that includes:

- report\_sources: a folder for storing the .Rmd reports
- outputs: a folder storing the compiled reports
- factory\_config: a control file used to anchor a report factory

Depending on the values of the logical arguments, the factory may also include:

- README.md: Example README with instructions on how to use report factory.
- .gitignore: a file used to tell git to ignore certain files including the produced outputs in outputs().
- data/raw/: a folder for storing raw data
- data/raw/example\_data.csv: a set of data for use with the example report
- data/clean/: a folder for storing cleaned data
- scripts/: a folder to store additional code that may be called in reports
- report\_sources/example\_report.Rmd: an example .Rmd report template
- .here: a file to anchor calls to here::here()

## Value

the report factory folder location (invisibly)

## Examples

```
f1 <- new_factory("new_factory_1", move_in = FALSE)
f2 <- new_factory("new_factory_2", move_in = TRUE)
```

---

validate\_factory      *Inspect and validate the content of a factory*

---

## Description

validate\_factory() can be used to inspect the content of a factory and make everything looks fine. This includes various sanity checks listed in details that will error if a problem is found.

## Usage

```
validate_factory(factory = ".")
```

## Arguments

factory      The path to the report factory or a folder within the desired factory. Defaults to the current directory.

## Details

Checks run on the factory include:

- the factory directory exists;
- the factory\_config file exist;
- all mandatory folders exist - by default these are 'report\_sources/' and 'outputs/';

## Value

A list with 4 entries:

- root - the root folder path of the factory;
- factory\_name - the name of the report factory;
- report\_sources - the name of the report\_sources folder; and
- outputs - the name of the outputs folder.

# Index

`compile_reports`, 2  
`factory_overview`, 3  
`install_deps`, 3  
`list_deps`, 4  
`list_outputs`, 5  
`list_reports`, 5  
`new_factory`, 6  
`validate_factory`, 8