

# Package ‘desc’

March 6, 2022

**Title** Manipulate DESCRIPTION Files

**Version** 1.4.1

**Maintainer** Gábor Csárdi <csardi.gabor@gmail.com>

**Description** Tools to read, write, create, and manipulate DESCRIPTION files. It is intended for packages that create or manipulate other packages.

**License** MIT + file LICENSE

**URL** <https://github.com/r-lib/desc#readme>, <http://r-lib.github.io/desc/>

**BugReports** <https://github.com/r-lib/desc/issues>

**Depends** R (>= 3.4)

**Imports** cli, R6, rprojroot, utils

**Suggests** callr, covr, gh, spelling, testthat, whoami, withr

**Config/Needs/website** tidyverse/tidytemplate

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-US

**RoxygenNote** 7.1.2

**Collate** 'assertions.R' 'authors-at-r.R' 'built.R' 'classes.R'  
'collate.R' 'constants.R' 'deps.R' 'desc-package.R'  
'description.R' 'encoding.R' 'latex.R' 'non-oo-api.R'  
'package-archives.R' 'read.R' 'remotes.R' 'str.R'  
'syntax\_checks.R' 'urls.R' 'utils.R' 'validate.R' 'version.R'

**NeedsCompilation** no

**Author** Gábor Csárdi [aut, cre],  
Kirill Müller [aut],  
Jim Hester [aut],  
Maëlle Salmon [ctb] (<<https://orcid.org/0000-0002-2815-0399>>),  
RStudio [cph, fnd]

**Repository** CRAN

**Date/Publication** 2022-03-06 13:20:02 UTC

**R topics documented:**

check_field . . . . .	3
cran_ascii_fields . . . . .	3
cran_valid_fields . . . . .	4
dep_types . . . . .	4
desc . . . . .	5
description . . . . .	5
desc_add_author . . . . .	12
desc_add_author_gh . . . . .	13
desc_add_me . . . . .	14
desc_add_orcid . . . . .	15
desc_add_remotes . . . . .	16
desc_add_role . . . . .	16
desc_add_to_collate . . . . .	17
desc_add_urls . . . . .	18
desc_bump_version . . . . .	18
desc_change_maintainer . . . . .	19
desc_clear_remotes . . . . .	20
desc_clear_urls . . . . .	20
desc_coerce_authors_at_r . . . . .	21
desc_del . . . . .	21
desc_del_author . . . . .	22
desc_del_collate . . . . .	23
desc_del_dep . . . . .	23
desc_del_deps . . . . .	24
desc_del_from_collate . . . . .	25
desc_del_remotes . . . . .	25
desc_del_role . . . . .	26
desc_del_urls . . . . .	27
desc_fields . . . . .	27
desc_get . . . . .	28
desc_get_author . . . . .	28
desc_get_authors . . . . .	29
desc_get_built . . . . .	29
desc_get_collate . . . . .	30
desc_get_deps . . . . .	30
desc_get_field . . . . .	31
desc_get_maintainer . . . . .	32
desc_get_remotes . . . . .	33
desc_get_urls . . . . .	33
desc_get_version . . . . .	34
desc_has_dep . . . . .	34
desc_has_fields . . . . .	35
desc_normalize . . . . .	35
desc_print . . . . .	36
desc_reformat_fields . . . . .	36
desc_reorder_fields . . . . .	37

*check\_field* 3

desc_set . . . . .	37
desc_set_authors . . . . .	38
desc_set_collate . . . . .	39
desc_set_dep . . . . .	39
desc_set_deps . . . . .	40
desc_set_remotes . . . . .	41
desc_set_urls . . . . .	41
desc_set_version . . . . .	42
desc_to_latex . . . . .	42
desc_validate . . . . .	43

**Index** 44

---

*check\_field*                      *Syntactical check of a DESCRIPTION field*

---

**Description**

Syntactical check of a DESCRIPTION field

**Usage**

`check_field(x, warn = FALSE, ...)`

**Arguments**

- `x`                      The field.
- `warn`                    Whether to generate a warning if the syntax check fails.
- `...`                    Additional arguments, they might be used in the future.

**Value**

TRUE if the field is syntactically correct, otherwise a character vector, containing one or multiple error messages.

---

*cran\_ascii\_fields*                      *The DESCRIPTION fields that are supposed to be in plain ASCII encoding*

---

**Description**

The DESCRIPTION fields that are supposed to be in plain ASCII encoding

**Usage**

`cran_ascii_fields`

**Format**

An object of class character of length 16.

**See Also**

Other field types: [cran\\_valid\\_fields](#), [dep\\_types](#)

---

cran_valid_fields	<i>A list of DESCRIPTION fields that are valid according to the CRAN checks</i>
-------------------	---

---

**Description**

A list of DESCRIPTION fields that are valid according to the CRAN checks

**Usage**

```
cran_valid_fields
```

**Format**

An object of class character of length 125.

**See Also**

Other field types: [cran\\_ascii\\_fields](#), [dep\\_types](#)

---

dep_types	<i>DESCRIPTION fields that denote package dependencies</i>
-----------	--

---

**Description**

Currently it has the following ones: Imports, Depends, Suggests, Enhances and LinkingTo. See the *Writing R Extensions* manual for when to use which.

**Usage**

```
dep_types
```

**Format**

An object of class character of length 5.

**See Also**

Other field types: [cran\\_ascii\\_fields](#), [cran\\_valid\\_fields](#)

---

desc *Read a DESCRIPTION file*

---

### Description

This is a convenience wrapper for `description$new()`. Very often you want to read an existing DESCRIPTION file, and to do this you can just supply the path to the file or its directory to `desc()`.

### Usage

```
desc(cmd = NULL, file = NULL, text = NULL, package = NULL)
```

### Arguments

cmd	A command to create a description from scratch. Currently only "!"new" is implemented. If it does not start with an exclamation mark, it will be interpreted as the file argument.
file	Name of the DESCRIPTION file to load. If all of cmd, file and text are NULL (the default), then the DESCRIPTION file in the current working directory is used. The file can also be an R package (source, or binary), in which case the DESCRIPTION file is extracted from it, but note that in this case <code>\$write()</code> cannot write the file back in the package archive.
text	A character scalar containing the full DESCRIPTION. Character vectors are collapsed into a character scalar, with newline as the separator.
package	If not NULL, then the name of an installed package and the DESCRIPTION file of this package will be loaded.

### Examples

```
desc(package = "desc")
DESCRIPTION <- system.file("DESCRIPTION", package = "desc")
desc(DESCRIPTION)
```

---

description *Read, write, update, validate DESCRIPTION files*

---

### Description

Read, write, update, validate DESCRIPTION files

### Format

An R6 class.

## Constructors

There are two ways of creating a description object. The first is reading an already existing DESCRIPTION file; simply give the name of the file as an argument. The default is DESCRIPTION:

```
x <- description$new()
x2 <- description$new("path/to/DESCRIPTION")
```

The second way is creating a description object from scratch, supply "!new" as an argument to do this.

```
x3 <- description$new("!new")
```

The complete API reference:

```
description$new(cmd = NULL, file = NULL, text = NULL,
  package = NULL)
```

- `cmd`: a command to create a description from scratch. Currently only "!new" is implemented. If it does not start with an exclamation mark, it will be interpreted as a file argument.
- `file`: name of the DESCRIPTION file to load. If it is a directory, then we assume that it is inside an R package and conduct a search for the package root directory, i.e. the first directory up the tree that contains a DESCRIPTION file. If `cmd`, `file`, `text` and `package` are all NULL (the default), then the search is started from the working directory. The file can also be an R package (source, or binary), in which case the DESCRIPTION file is extracted from it, but note that in this case `$write()` cannot write the file back in the package archive.
- `text`: a character scalar containing the full DESCRIPTION. Character vectors are collapsed into a character scalar, with newline as the separator.
- `package`: if not NULL, then the name of an installed package and the DESCRIPTION file of this package will be loaded.

## Setting and Querying fields

Set a field with `$set` and query it with `$get`:

```
x <- description$new("!new")
x$get("Package")
x$set("Package", "foobar")
x$set(Title = "Example Package for 'description'")
x$get("Package")
```

Note that `$set` has two forms. You can either give the field name and new value as two arguments; or you can use a single named argument, the argument name is the field name, the argument value is the field value.

The `$fields` method simply lists the fields in the object:

```
x$fields()
```

The `$has_fields` method checks if one or multiple fields are present in a description object:

```
x$has_fields("Package")
x$has_fields(c("Title", "foobar"))
```

The `$del` method removes the specified fields:

```
x$set(foo = "bar")
x$del("foo")
```

`$get_field` is similar to `$get`, but it queries a single field, it returns an unnamed vector if found, and returns the specified default value if not. By default it throws an error if the field is not found.

The complete API reference:

```
description$get(keys)
description$get_field(key, default, trim_ws = TRUE, squish_ws = trim_ws)
description$set(...)
description$fields()
description$has_fields(keys)
description$del(keys)
```

- `key`: a character string (length one), the key to query.
- `default`: If specified and `key` is missing, this value is returned. If not specified, an error is thrown.
- `trim_ws`: whether to trim leading and trailing whitespace from the returned value.
- `squish_ws`: whether to reduce repeated whitespace in the returned value.
- `keys`: a character vector of keys to query, check or delete.
- `...`: this must be either two unnamed arguments, the key and the value to set; or an arbitrary number of named arguments, names are used as keys, values as values to set.

## Normalizing

Format DESCRIPTION in a standard way. `$str` formats each field in a standard way and returns them (it does not change the object itself), `$print` is used to print it to the screen. The `$normalize` function normalizes each field (i.e. it changes the object). Normalization means reformatting the fields, via `{reformat_fields()}` and also reordering them via `{reorder_fields()}`. The format of the various fields is opinionated and you might like it or not. Note that `desc` only re-formats fields that it updates, and only on demand, so if your formatting preferences differ, you can still manually edit DESCRIPTION and `desc` will respect your edits.

```
description$str(by_field = FALSE, normalize = TRUE,
  mode = c("file", "screen"))
description$normalize()
description$reformat_fields()
description$reorder_fields()
description$print()
```

- `by_field`: whether to return the normalized format by field, or collapsed into a character scalar.
- `normalize`: whether to reorder and reformat the fields.
- `mode`: `file` mode formats the fields as they are written to a file with the `write` method. `screen` mode adds extra markup to some fields, e.g. formats the `Authors@R` field in a readable way.

### Writing it to file

The `$write` method writes the description to a file. By default it writes it to the file it was created from, if it was created from a file. Otherwise giving a file name is compulsory:

```
x$write(file = "DESCRIPTION")
```

The API:

```
description$write(file = NULL)
```

- `file`: path to write the description to. If it was created from a file in the first place, then it is written to the same file. Otherwise this argument must be specified.

### Version numbers

```
description$get_version()
description$set_version(version)
description$bump_version(which = c("patch", "minor", "major", "dev"))
```

- `version`: a string or a [base::package\\_version](#) object.
- `which`: which component of the version number to increase. See details just below.

These functions are simple helpers to make it easier to query, set and increase the version number of a package.

`$get_version()` returns the version number as a [base::package\\_version](#) object. It throws an error if the package does not have a `Version` field.

`$set_version()` takes a string or a [base::package\\_version](#) object and sets the `Version` field to it.

`$bump_version()` increases the version number. The `which` parameter specifies which component to increase. It can be a string referring to a component: `major`, `minor`, `patch` or `dev`, or an integer scalar, for the latter components are counted from one, and the beginning. I.e. component one is equivalent to `major`.

If a component is bumped, then the ones after it are zeroed out. Trailing zero components are omitted from the new version number, but if the old version number had at least two or three components, then the one will also have two or three.

The bumping of the `dev` version (the fourth component) is special: if the original version number had less than four components, and the `dev` version is bumped, then it is set to `9000` instead of `1`. This is a convention often used by R developers, it was originally invented by Winston Chang.

Both `$set_version()` and `$bump_version()` use dots to separate the version number components.

### Dependencies

These functions handle the fields that define how the R package uses another R packages. See [dep\\_types](#) for the list of fields in this group.

The `$get_deps` method returns all declared dependencies, in a data frame with columns: `type`, `package` and `version`. `type` is the name of the dependency field, `package` is the name of the R package, and `version` is the required version. If no specific versions are required, then this is a `"*"`.



The `$set_deps` method is the opposite of `$get_deps` and it sets all dependencies. The input is a data frame, with the same structure as the return value of `$get_deps`.

The `$has_dep` method checks if a package is included in the dependencies. It returns a logical scalar. If `type` is not `any`, then it has to match as well.

The `$del_deps` method removes all declared dependencies.

The `$set_dep` method adds or updates a single dependency. By default it adds the package to the `Imports` field.

The API:

```
description$set_dep(package, type = dep_types, version = "*")
description$set_deps(deps)
description$get_deps()
description$has_dep(package, type = c("any", dep_types))
description$del_dep(package, type = c("all", dep_types))
description$del_deps()
```

- `package`: name of the package to add to or remove from the dependencies.
- `type`: dependency type, see [dep\\_types](#). For `$del_dep` it may also be `"all"`, and then the package will be deleted from all dependency types.
- `version`: required version. Defaults to `"*"`, which means no explicit version requirements.
- `deps`: a data frame with columns `type`, `package` and `version`. `$get_deps` returns the same format.

### Collate fields

Collate fields contain lists of file names with R source code, and the package has a separate API for them. In brief, you can use `$add_to_collate` to add one or more files to the main or other collate field. You can use `$del_from_collate` to remove it from there.

The API:

```
description$set_collate(files, which = c("main", "windows", "unix"))
description$get_collate(which = c("main", "windows", "unix"))
description$del_collate(which = c("all", "main", "windows", "unix"))
description$add_to_collate(files, which = c("default", "all", "main",
  "windows", "unix"))
description$del_from_collate(files, which = c("all", "main",
  "windows", "unix"))
```

- `files`: the files to add or remove, in a character vector.
- `which`: which collate field to manipulate. `"default"` for `$add_to_collate` means all existing collate fields, or the main one if none exist.

### Authors

There is a specialized API for the `Authors@R` field, to add and remove authors, update their roles, change the maintainer, etc.

The API:

```
description$get_authors()
description$set_authors(authors)
description$get_author(role)
description$get_maintainer()
description$coerce_authors_at_r()
```

- authors: a person object, a list of authors.
- role: The role to query. See person for details.

\$get\_authors returns a person object, the parsed authors. See [utils::person\(\)](#) for details.

\$get\_author returns a person object, all authors with the specified role.

\$get\_maintainer returns the maintainer of the package. It works with Authors@R fields and with traditional Maintainer fields as well.

\$coerce\_authors\_at\_r converts an Author field to one with a person object. This coercion may be necessary for other functions such as \$get\_authors.

```
description$add_author(given = NULL, family = NULL, email = NULL,
  role = NULL, comment = NULL, orcid = NULL)
description$add_me(role = "ctb", comment = NULL, orcid = NULL)
description$add_author_gh(username, role = "ctb", comment = NULL, orcid = NULL)
```

Add a new author. The arguments correspond to the arguments of the [utils::person\(\)](#) function. `add_me` is a convenience function, it adds the current user as an author, and it needs the `whoami` package to be installed. It'll add your ORCID ID if you provide it as argument or save it as `ORCID_ID` environment variable in `.Renviro`. The full name is parsed by `add_me` and `add_author_gh` using `as.person` and collapsing the given name and the family name in order to e.g. have the first and middle names together as given name. This approach might be limited to some full name structures.

```
description$del_author(given = NULL, family = NULL, email = NULL,
  role = NULL, comment = NULL, orcid = NULL)
```

Remove an author, or multiple authors. The author(s) to be removed can be specified via any field(s). All authors matching all specifications will be removed. E.g. if only `given = "Joe"` is supplied, then all authors whose given name matches Joe will be removed. The specifications can be (PCRE) regular expressions.

```
description$add_role(role, given = NULL, family = NULL, email = NULL,
  comment = NULL, orcid = NULL)
description$add_orcid(orcid, given = NULL, family = NULL, email = NULL,
  comment = NULL, role = NULL)
description$del_role(role, given = NULL, family = NULL, email = NULL,
  comment = NULL, orcid = NULL)
description$change_maintainer(given = NULL, family = NULL,
  email = NULL, comment = NULL, orcid = NULL)
```

role is the role to add or delete. The other arguments are used to select a subset of the authors, on which the operation is performed, similarly to \$del\_author.

## URLs

We provide helper functions for manipulating URLs in the URL field:

```
description$get_urls()
description$set_urls(urls)
description$add_urls(urls)
description$del_urls(pattern)
description$clear_urls()
```

- `urls`: character vector of URLs to set or add.
- `pattern`: Perl compatible regular expression to specify the URLs to be removed.

`$get_urls()` returns all urls in a character vector. If no URL fields are present, a zero length vector is returned.

`$set_urls()` sets the URL field to the URLs specified in the character vector argument.

`$add_urls()` appends the specified URLs to the URL field. It creates the field if it does not exist. Duplicate URLs are removed.

`$del_urls()` deletes the URLs that match the specified pattern.

`$clear_urls()` deletes all URLs.

## Remotes

`devtools`, `remotes` and some other packages support the non-standard `Remotes` field in `DESCRIPTION`. This field can be used to specify locations of dependent packages: GitHub or BitBucket repositories, generic git repositories, etc. Please see the `Package remotes vignette` in the `devtools` package.

`desc` has helper functions for manipulating the `Remotes` field:

```
description$get_remotes()
description$get_remotes()
description$set_remotes(remotes)
description$add_remotes(remotes)
description$del_remotes(pattern)
description$clear_remotes()
```

- `remotes`: character vector of remote dependency locations to set or add.
- `pattern`: Perl compatible regular expression to specify the remote dependency locations to remove.

`$get_remotes()` returns all remotes in a character vector. If no URL fields are present, a zero length vector is returned.

`$set_remotes()` sets the URL field to the Remotes specified in the character vector argument.

`$add_remotes()` appends the specified remotes to the `Remotes` field. It creates the field if it does not exist. Duplicate remotes are removed.

`$del_remotes()` deletes the remotes that match the specified pattern.

`$clear_remotes()` deletes all remotes.

## Built

The `Built` field is used in binary packages to store information about when and how a binary package was built.

`$get_built()` returns the built information as a list with fields `R`, `Platform`, `Date`, `OStype`. It throws an error if the package does not have a `Built` field.

## Encodings

When creating a description object, `desc` observes the `Encoding` field, if present, and uses the specified encoding to parse the file. Internally, it converts all fields to UTF-8.

When writing a description object to a file, `desc` uses the `Encoding` field (if present), and converts all fields to the specified encoding.

We suggest that whenever you need to use non-ASCII characters in your package, you use the UTF-8 encoding, for maximum portability.

## Examples

```
## Create a template
desc <- description$new("!new")
desc

## Read a file
desc2 <- description$new(file = system.file("DESCRIPTION",
      package = "desc"))
desc2

## Remove a field
desc2$del("LazyData")

## Add another one
desc2$set(VignetteBuilder = "knitr")
desc2$get("VignetteBuilder")
desc2
```

---

desc\_add\_author

*Add an author to Authors@R in DESCRIPTION*

---

## Description

Add an author to Authors@R in DESCRIPTION

## Usage

```
desc_add_author(
  given = NULL,
  family = NULL,
  email = NULL,
  role = NULL,
```

```

    comment = NULL,
    orcid = NULL,
    file = ".",
    normalize = FALSE
  )

```

### Arguments

given	Given name.
family	Family name.
email	Email address.
role	Role.
comment	Comment.
orcid	ORCID.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

### See Also

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_author\(\)](#), [desc\\_get\\_maintainer\(\)](#), [desc\\_set\\_authors\(\)](#)

---

desc\_add\_author\_gh     *Add a GitHub user as an author to DESCRIPTION*

---

### Description

Uses the Authors@R field.

### Usage

```

desc_add_author_gh(
  username,
  role = "ctb",
  comment = NULL,
  orcid = NULL,
  file = ".",
  normalize = FALSE
)

```

**Arguments**

username	GitHub username of the GitHub user
role	Role to set for the user, defaults to contributor.
comment	Comment, empty by default.
orcid	ORCID, empty by default.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**Details**

desc\_add\_author\_gh is a convenience function, it adds the GitHub user as an author, and it needs the gh package to be installed. The full name is parsed using `as.person` and collapsing the given name and the family name in order to e.g. have the first and middle names together as given name. This approach might be limited to some full name structures.

**See Also**

Other Authors@R: [desc\\_add\\_author\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_author\(\)](#), [desc\\_get\\_maintainer\(\)](#), [desc\\_set\\_authors\(\)](#)

---

desc\_add\_me

*Add the current user as an author to DESCRIPTION*


---

**Description**

Uses the Authors@R field.

**Usage**

```
desc_add_me(
  role = "ctb",
  comment = NULL,
  orcid = NULL,
  file = ".",
  normalize = FALSE
)
```

**Arguments**

role	Role to set for the user, defaults to contributor.
comment	Comment, empty by default.
orcid	ORCID, empty by default.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**Details**

desc\_add\_me is a convenience function, it adds the current user as an author, and it needs the whoami package to be installed. It'll add your ORCID ID if you provide it as argument or save it as ORCID\_ID environment variable in .Renviron. The full name is parsed using as.person and collapsing the given name and the family name in order to e.g. have the first and middle names together as given name. This approach might be limited to some full name structures.

**See Also**

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_author\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_author\(\)](#), [desc\\_get\\_maintainer\(\)](#), [desc\\_set\\_authors\(\)](#)

---

desc_add_orcid	<i>Add an ORCID to one or more authors in Authors@R, in DESCRIPTION</i>
----------------	---

---

**Description**

The author(s) can be specified by a combination of the given, family, email, comment and role fields. If multiple filters are specified, then all must match to identify the author(s).

**Usage**

```
desc_add_orcid(
  orcid,
  given = NULL,
  family = NULL,
  email = NULL,
  comment = NULL,
  role = NULL,
  file = ".",
  normalize = FALSE
)
```

**Arguments**

orcid	orcid to add.
given	Given name to filter on. Regular expression.
family	Family name to filter on. Regular expression.
email	Email address to filter on. Regular expression.
comment	Comment field to filter on. Regular expression.
role	Role field to filter on.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_author\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_author\(\)](#), [desc\\_get\\_maintainer\(\)](#), [desc\\_set\\_authors\(\)](#)

---

desc\_add\_remotes      *Add locations in the Remotes field in DESCRIPTION*

---

**Description**

Add locations in the Remotes field in DESCRIPTION

**Usage**

```
desc_add_remotes(remotes, file = ".", normalize = FALSE)
```

**Arguments**

remotes	Character vector of remote locations to add. Duplicate locations are eliminated. Note that existing locations are not updated, so if you want to <i>change</i> a remote location of a package, you need to delete the old location first and then add the new one.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

---

desc\_add\_role      *Add a role to one or more authors in Authors@R, in DESCRIPTION*

---

**Description**

The author(s) can be specified by a combination of the given, family, email, comment and orcid fields. If multiple filters are specified, then all must match to identify the author(s).

**Usage**

```
desc_add_role(
  role,
  given = NULL,
  family = NULL,
  email = NULL,
  comment = NULL,
  orcid = NULL,
  file = ".",
  normalize = FALSE
)
```



**Arguments**

role	Role to add.
given	Given name to filter on. Regular expression.
family	Family name to filter on. Regular expression.
email	Email address to filter on. Regular expression.
comment	Comment field to filter on. Regular expression.
orcid	ORCID field to filter on.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_author\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_author\(\)](#), [desc\\_get\\_maintainer\(\)](#), [desc\\_set\\_authors\(\)](#)

---

desc\_add\_to\_collate    *Add one or more files to the Collate field, in DESCRIPTION*

---

**Description**

Add one or more files to the Collate field, in DESCRIPTION

**Usage**

```
desc_add_to_collate(
  files,
  which = c("default", "all", "main", "windows", "unix"),
  file = ".",
  normalize = FALSE
)
```

**Arguments**

files	Character vector, files to add.
which	Which collate field to use. Collate fields can be operating system type specific.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other Collate field: [desc\\_del\\_collate\(\)](#), [desc\\_del\\_from\\_collate\(\)](#), [desc\\_get\\_collate\(\)](#), [desc\\_set\\_collate\(\)](#)

---

desc_add_urls	<i>Add URLs to the URL field in DESCRIPTION</i>
---------------	---

---

**Description**

Add URLs to the URL field in DESCRIPTION

**Usage**

```
desc_add_urls(urls, file = ".", normalize = FALSE)
```

**Arguments**

urls	Character vector of URLs to add. Duplicate URLs are eliminated.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

---

desc_bump_version	<i>Increase the version number in DESCRIPTION</i>
-------------------	---

---

**Description**

The which parameter specifies which component to increase. It can be a string referring to a component: major, minor, patch or dev, or an integer scalar, for the latter components are counted from one, and the beginning. I.e. component one is equivalent to major.

**Usage**

```
desc_bump_version(which, file = ".", normalize = FALSE)
```

**Arguments**

which	Which component to increase. See details below.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**Details**

If a component is bumped, then the ones after it are zeroed out. Trailing zero components are omitted from the new version number, but if the old version number had at least two or three components, then the one will also have two or three.

The bumping of the dev version (the fourth component) is special: if the original version number had less than four components, and the dev version is bumped, then it is set to 9000 instead of 1. This is a convention often used by R developers, it was originally invented by Winston Chang.

Both `$set_version()` and `$bump_version()` use dots to separate the version number components.

**See Also**

Other version numbers: [desc\\_get\\_version\(\)](#), [desc\\_set\\_version\(\)](#)

---

desc\_change\_maintainer

*Change maintainer of the package, in DESCRIPTION*

---

**Description**

Only works with the Authors@R field.

**Usage**

```
desc_change_maintainer(  
  given = NULL,  
  family = NULL,  
  email = NULL,  
  comment = NULL,  
  orcid = NULL,  
  file = ".",  
  normalize = FALSE  
)
```

**Arguments**

given	Given name.
family	Family name.
email	Email address.
comment	Comment.
orcid	ORCID.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**Details**

The current maintainer is kept if they have at least another role.

**See Also**

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_author\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_author\(\)](#), [desc\\_get\\_maintainer\(\)](#), [desc\\_set\\_authors\(\)](#)

---

desc_clear_remotes	<i>Remove all locations from the Remotes field of DESCRIPTION</i>
--------------------	---

---

**Description**

This simply means that the field is deleted.

**Usage**

```
desc_clear_remotes(file = ".", normalize = FALSE)
```

**Arguments**

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

---

desc_clear_urls	<i>Remove all URLs from the URL field of DESCRIPTION</i>
-----------------	--

---

**Description**

Remove all URLs from the URL field of DESCRIPTION

**Usage**

```
desc_clear_urls(file = ".", normalize = FALSE)
```

**Arguments**

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

---

desc\_coerce\_authors\_at\_r

*Coerce Author and Maintainer Fields to Authors@R*

---

### Description

Convert the Author and Maintainer fields to Authors@R, which is necessary for other functions such as `desc_get_authors()`.

### Usage

```
desc_coerce_authors_at_r(file = ".", normalize = FALSE)
```

### Arguments

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

### Details

If the Authors@R field does not exist, `desc_coerce_authors_at_r` tries to parse the Author and Maintainer fields with `utils::as.person()` and writes them to the Authors@R field. Note that Author and Maintainer are free-form fields, so parsing them may fail.

### See Also

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_author\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_author\(\)](#), [desc\\_get\\_maintainer\(\)](#), [desc\\_set\\_authors\(\)](#)

---

desc\_del

*Remove fields from a DESCRIPTION file*

---

### Description

Remove fields from a DESCRIPTION file

### Usage

```
desc_del(keys, file = ".", normalize = FALSE)
```

**Arguments**

keys	Character vector of keys to remove.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other simple queries: [desc\\_fields\(\)](#), [desc\\_get\\_field\(\)](#), [desc\\_get\(\)](#), [desc\\_has\\_fields\(\)](#), [desc\\_set\(\)](#)

---

desc_del_author	<i>Remove one or more authors from DESCRIPTION.</i>
-----------------	---

---

**Description**

It uses the Authors@R field. The author(s) to be removed can be specified via any field(s). All authors matching all specifications will be removed. E.g. if only given = "Joe" is supplied, then all authors whose given name matches Joe will be removed. The specifications can be (PCRE) regular expressions.

**Usage**

```
desc_del_author(
  given = NULL,
  family = NULL,
  email = NULL,
  role = NULL,
  comment = NULL,
  orcid = NULL,
  file = ".",
  normalize = FALSE
)
```

**Arguments**

given	Given name to filter on. Regular expression.
family	Family name to filter on. Regular expression.
email	Email address to filter on. Regular expression.
role	Role to filter on. Regular expression.
comment	Comment field to filter on. Regular expression.
orcid	ORCID field to filter on.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_author\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_author\(\)](#), [desc\\_get\\_maintainer\(\)](#), [desc\\_set\\_authors\(\)](#)

---

desc\_del\_collate      *Delete the Collate field from DESCRIPTION*

---

**Description**

Delete the Collate field from DESCRIPTION

**Usage**

```
desc_del_collate(
  which = c("all", "main", "windows", "unix"),
  file = ".",
  normalize = FALSE
)
```

**Arguments**

**which**            Which collate field to use. Collate fields can be operating system type specific.

**file**             DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

**normalize**        Whether to normalize the write when writing back the result. See [desc\\_normalize\(\)](#).

**See Also**

Other Collate field: [desc\\_add\\_to\\_collate\(\)](#), [desc\\_del\\_from\\_collate\(\)](#), [desc\\_get\\_collate\(\)](#), [desc\\_set\\_collate\(\)](#)

---

desc\_del\_dep            *Remove a package dependency from DESCRIPTION*

---

**Description**

Remove a package dependency from DESCRIPTION

**Usage**

```
desc_del_dep(
  package,
  type = c("all", desc::dep_types),
  file = ".",
  normalize = FALSE
)
```

**Arguments**

package	Package dependency to remove.
type	Dependency type to remove. Sometimes a package is depended on via multiple dependency types, e.g. LinkingTo and Imports. Defaults to all types.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other dependencies: [desc\\_del\\_deps\(\)](#), [desc\\_get\\_deps\(\)](#), [desc\\_has\\_dep\(\)](#), [desc\\_set\\_deps\(\)](#), [desc\\_set\\_dep\(\)](#)

---

desc_del_deps	<i>Remove all dependencies from DESCRIPTION</i>
---------------	---

---

**Description**

Remove all dependencies from DESCRIPTION

**Usage**

```
desc_del_deps(file = ".", normalize = FALSE)
```

**Arguments**

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other dependencies: [desc\\_del\\_dep\(\)](#), [desc\\_get\\_deps\(\)](#), [desc\\_has\\_dep\(\)](#), [desc\\_set\\_deps\(\)](#), [desc\\_set\\_dep\(\)](#)



---

desc\_del\_from\_collate *Remove files from the Collate field.*

---

**Description**

Remove files from the Collate field.

**Usage**

```
desc_del_from_collate(
    files,
    which = c("all", "main", "windows", "unix"),
    file = ".",
    normalize = FALSE
)
```

**Arguments**

files	Files to remove from the Collate field.
which	Which collate field to use. Collate fields can be operating system type specific.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other Collate field: [desc\\_add\\_to\\_collate\(\)](#), [desc\\_del\\_collate\(\)](#), [desc\\_get\\_collate\(\)](#), [desc\\_set\\_collate\(\)](#)

---

desc\_del\_remotes *Delete locations from the Remotes field in DESCRIPTION*

---

**Description**

All locations matching the specified pattern are deleted.

**Usage**

```
desc_del_remotes(pattern, file = ".", normalize = FALSE)
```

**Arguments**

pattern	Perl-compatible regular expression, all locations matching this expression will be deleted.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

---

desc\_del\_role                      *Delete a role of an author, in DESCRIPTION*

---

### Description

The author(s) can be specified by a combination of the given, family, email, comment and orcid fields. If multiple filters are specified, then all must match to identify the author(s).

### Usage

```
desc_del_role(
  role,
  given = NULL,
  family = NULL,
  email = NULL,
  comment = NULL,
  orcid = NULL,
  file = ".",
  normalize = FALSE
)
```

### Arguments

role	Role to remove.
given	Given name to filter on. Regular expression.
family	Family name to filter on. Regular expression.
email	Email address to filter on. Regular expression.
comment	Comment field to filter on. Regular expression.
orcid	ORCID field to filter on.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

### See Also

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_author\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_author\(\)](#), [desc\\_get\\_maintainer\(\)](#), [desc\\_set\\_authors\(\)](#)

---

desc_del_urls	<i>Delete URLs from the URL field in DESCRIPTION</i>
---------------	--

---

**Description**

All URLs matching the specified pattern are deleted.

**Usage**

```
desc_del_urls(pattern, file = ".", normalize = FALSE)
```

**Arguments**

pattern	Perl-compatible regular expression, all URLs matching this expression will be deleted.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

---

desc_fields	<i>List all fields in a DESCRIPTION file</i>
-------------	--

---

**Description**

List all fields in a DESCRIPTION file

**Usage**

```
desc_fields(file = ".")
```

**Arguments**

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
------	--

**Value**

Character vector of fields.

**See Also**

Other simple queries: [desc\\_del\(\)](#), [desc\\_get\\_field\(\)](#), [desc\\_get\(\)](#), [desc\\_has\\_fields\(\)](#), [desc\\_set\(\)](#)

---

desc\_get                      *Get a field from a DESCRIPTION file*

---

**Description**

Get a field from a DESCRIPTION file

**Usage**

```
desc_get(keys, file = ".")
```

**Arguments**

keys	Character vector of fields to get.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

**Value**

Character vector, values of the specified keys. Non-existing keys return NA.

**See Also**

Other simple queries: [desc\\_del\(\)](#), [desc\\_fields\(\)](#), [desc\\_get\\_field\(\)](#), [desc\\_has\\_fields\(\)](#), [desc\\_set\(\)](#)

---

desc\_get\_author                *Query authors by role in Authors@R, in DESCRIPTION*

---

**Description**

Query authors by role in Authors@R, in DESCRIPTION

**Usage**

```
desc_get_author(role = "cre", file = ".")
```

**Arguments**

role	Role to query. Defaults to the package maintainer.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

**Value**

A person object, see [utils::person\(\)](#).

**See Also**

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_author\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_maintainer\(\)](#), [desc\\_set\\_authors\(\)](#)

---

desc\_get\_authors      *Query all authors in Authors@R, in DESCRIPTION*

---

**Description**

Query all authors in Authors@R, in DESCRIPTION

**Usage**

```
desc_get_authors(file = ".")
```

**Arguments**

`file`                DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

**Value**

A person object, see [utils::person\(\)](#).

**See Also**

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_author\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_maintainer\(\)](#), [desc\\_set\\_authors\(\)](#)

---

desc\_get\_built      *Query the built field in DESCRIPTION*

---

**Description**

If the file has no Built field then it throws an error.

**Usage**

```
desc_get_built(file = ".", normalize = FALSE)
```

**Arguments**

`file`                DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

`normalize`          Whether to normalize the write when writing back the result. See [desc\\_normalize\(\)](#).

**Value**

A list with fields R, Platform, Date, OStype.

---

desc_get_collate	<i>Query the Collate field in DESCRIPTION</i>
------------------	---

---

**Description**

Query the Collate field in DESCRIPTION

**Usage**

```
desc_get_collate(which = c("main", "windows", "unix"), file = ".")
```

**Arguments**

which	Which collate field to use. Collate fields can be operating system type specific.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

**Value**

Character vector of file names.

**See Also**

Other Collate field: [desc\\_add\\_to\\_collate\(\)](#), [desc\\_del\\_collate\(\)](#), [desc\\_del\\_from\\_collate\(\)](#), [desc\\_set\\_collate\(\)](#)

---

desc_get_deps	<i>List all package dependencies from a DESCRIPTION file</i>
---------------	--

---

**Description**

List all package dependencies from a DESCRIPTION file

**Usage**

```
desc_get_deps(file = ".")
```

**Arguments**

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
------	--

**Value**

Data frame with columns: type (dependency type), package, and version. For non-versioned dependencies version is \*.

**See Also**

Other dependencies: [desc\\_del\\_deps\(\)](#), [desc\\_del\\_dep\(\)](#), [desc\\_has\\_dep\(\)](#), [desc\\_set\\_deps\(\)](#), [desc\\_set\\_dep\(\)](#)

---

desc_get_field	<i>Get a single field from a DESCRIPTION file, fail if not found</i>
----------------	--

---

**Description**

`desc_get_list()` parses a comma separated list into a character vector.

**Usage**

```
desc_get_field(
  key,
  default = stop("Field '", key, "' not found"),
  trim_ws = TRUE,
  squish_ws = trim_ws,
  file = ".")
)
```

```
desc_get_or_fail(keys, file = ".")
```

```
desc_get_list(
  key,
  default = stop("Field '", key, "' not found"),
  sep = ",",
  trim_ws = TRUE,
  squish_ws = trim_ws,
  file = ".")
)
```

**Arguments**

key	The field to query.
default	Value to return if key is not found. By default it throws an error.
trim_ws	Whether to trim leading and trailing whitespace from the value. Defaults to TRUE.
squish_ws	Whether to reduce repeated whitespace in the value. Defaults to trim_ws.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

keys            Character vector of fields to get.  
 sep            Separator string for [desc\\_get\\_list\(\)](#).

**Value**

Character string, the value of key, or default if key is not found and default is specified.

**See Also**

Other simple queries: [desc\\_del\(\)](#), [desc\\_fields\(\)](#), [desc\\_get\(\)](#), [desc\\_has\\_fields\(\)](#), [desc\\_set\(\)](#)

---

desc\_get\_maintainer    *Query the package maintainer in DESCRIPTION*

---

**Description**

Either from the Maintainer or the Authors@R field.

**Usage**

```
desc_get_maintainer(file = ".")
```

**Arguments**

file            DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

**Value**

A character scalar.

**See Also**

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_author\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_author\(\)](#), [desc\\_set\\_authors\(\)](#)



---

desc_get_remotes	<i>List the locations in the Remotes field in DESCRIPTION</i>
------------------	---

---

**Description**

List the locations in the Remotes field in DESCRIPTION

**Usage**

```
desc_get_remotes(file = ".")
```

**Arguments**

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
------	--

**Value**

A character vectors or remote locations. A length zero vector is returned if there is no Remotes field in the package.

---

desc_get_urls	<i>Query the URL field in DESCRIPTION</i>
---------------	---

---

**Description**

Query the URL field in DESCRIPTION

**Usage**

```
desc_get_urls(file = ".")
```

**Arguments**

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
------	--

**Value**

A character vectors or URLs. A length zero vector is returned if there is no URL field in the package.

---

desc\_get\_version      *Query the package version in DESCRIPTION*

---

### Description

If the file has no Version field, or it is an invalid version string, then it throws an error.

### Usage

```
desc_get_version(file = ".")
```

### Arguments

file                    DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

### Value

A `base::package_version` object.

### See Also

Other version numbers: [desc\\_bump\\_version\(\)](#), [desc\\_set\\_version\(\)](#)

---

desc\_has\_dep            *Check for a dependency*

---

### Description

Check for a dependency

### Usage

```
desc_has_dep(package, type = c("any", desc::dep_types), file = ".")
```

### Arguments

package                The package name.  
 type                    A dependency type or “any”.  
 file                    DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

### Value

A logical scalar.

**See Also**

Other dependencies: [desc\\_del\\_deps\(\)](#), [desc\\_del\\_dep\(\)](#), [desc\\_get\\_deps\(\)](#), [desc\\_set\\_deps\(\)](#), [desc\\_set\\_dep\(\)](#)

---

desc_has_fields	<i>Check if some fields are present in a DESCRIPTION file</i>
-----------------	---

---

**Description**

Check if some fields are present in a DESCRIPTION file

**Usage**

```
desc_has_fields(keys, file = ".")
```

**Arguments**

keys	Character vector of keys to check.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

**Value**

Logical vector.

**See Also**

Other simple queries: [desc\\_del\(\)](#), [desc\\_fields\(\)](#), [desc\\_get\\_field\(\)](#), [desc\\_get\(\)](#), [desc\\_set\(\)](#)

---

desc_normalize	<i>Normalize a DESCRIPTION file</i>
----------------	-------------------------------------

---

**Description**

Re-formats and re-orders fields in DESCRIPTION in a standard way. Reorders the packages alphabetically.

**Usage**

```
desc_normalize(file = ".")
```

**Arguments**

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
------	--

**See Also**

Other repair functions: [desc\\_reformat\\_fields\(\)](#), [desc\\_reorder\\_fields\(\)](#)

---

desc_print	<i>Print the contents of a DESCRIPTION file to the screen</i>
------------	---

---

**Description**

Print the contents of a DESCRIPTION file to the screen

**Usage**

```
desc_print(file = ".")
```

**Arguments**

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
------	--

---

desc_reformat_fields	<i>Reformat fields in a DESCRIPTION file</i>
----------------------	--

---

**Description**

Reformat the fields in DESCRIPTION in a standard way.

**Usage**

```
desc_reformat_fields(file = ".")
```

**Arguments**

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
------	--

**See Also**

Other repair functions: [desc\\_normalize\(\)](#), [desc\\_reorder\\_fields\(\)](#)

---

desc\_reorder\_fields     *Reorder fields in a DESCRIPTION file*

---

### Description

Reorder the fields in DESCRIPTION in a standard way.

### Usage

```
desc_reorder_fields(file = ".")
```

### Arguments

file                    DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

### See Also

Other repair functions: [desc\\_normalize\(\)](#), [desc\\_reformat\\_fields\(\)](#)

---

desc\_set                    *Set one or more fields in a DESCRIPTION file*

---

### Description

Set one or more fields in a DESCRIPTION file

### Usage

```
desc_set(..., file = ".", normalize = FALSE)
```

```
desc_set_list(key, list_value, sep = ", ", file = ".", normalize = FALSE)
```

### Arguments

...                    Values to set, see details below.

file                    DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.

normalize                Whether to normalize the write when writing back the result. See [desc\\_normalize\(\)](#).

key                     Key to set in desc\_set\_list().

list\_value              Character vector, to collapse in desc\_set\_list().

sep                     Separator string for desc\_set\_list() list fields.

**Details**

desc\_set() supports two forms, the first is two unnamed arguments: the key and its value to set.

The second form requires named arguments: names are used as keys and values as values to set.

desc\_set\_list() collapses a character vector into string, separating the elements by commas.

**See Also**

Other simple queries: [desc\\_del\(\)](#), [desc\\_fields\(\)](#), [desc\\_get\\_field\(\)](#), [desc\\_get\(\)](#), [desc\\_has\\_fields\(\)](#)

---

desc_set_authors	<i>Set authors in Authors@R, in DESCRIPTION</i>
------------------	---

---

**Description**

Set authors in Authors@R, in DESCRIPTION

**Usage**

```
desc_set_authors(authors, file = ".", normalize = FALSE)
```

**Arguments**

authors	Authors, to set, a person object, see <a href="#">utils::person()</a> .
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other Authors@R: [desc\\_add\\_author\\_gh\(\)](#), [desc\\_add\\_author\(\)](#), [desc\\_add\\_me\(\)](#), [desc\\_add\\_orcid\(\)](#), [desc\\_add\\_role\(\)](#), [desc\\_change\\_maintainer\(\)](#), [desc\\_coerce\\_authors\\_at\\_r\(\)](#), [desc\\_del\\_author\(\)](#), [desc\\_del\\_role\(\)](#), [desc\\_get\\_authors\(\)](#), [desc\\_get\\_author\(\)](#), [desc\\_get\\_maintainer\(\)](#)

---

desc_set_collate	<i>Set the Collate field in DESCRIPTION</i>
------------------	---

---

**Description**

Set the Collate field in DESCRIPTION

**Usage**

```
desc_set_collate(  
  files,  
  which = c("main", "windows", "unix"),  
  file = ".",  
  normalize = FALSE  
)
```

**Arguments**

files	Collate field to set, as a character vector.
which	Which collate field to use. Collate fields can be operating system type specific.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other Collate field: [desc\\_add\\_to\\_collate\(\)](#), [desc\\_del\\_collate\(\)](#), [desc\\_del\\_from\\_collate\(\)](#), [desc\\_get\\_collate\(\)](#)

---

desc_set_dep	<i>Add a package dependency to a DESCRIPTION file</i>
--------------	---

---

**Description**

Add a package dependency to a DESCRIPTION file

**Usage**

```
desc_set_dep(  
  package,  
  type = desc::dep_types,  
  version = "*",  
  file = ".",  
  normalize = FALSE  
)
```

**Arguments**

package	Package to depend on.
type	Dependency type.
version	Version to depend on, for versioned dependencies.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other dependencies: [desc\\_del\\_deps\(\)](#), [desc\\_del\\_dep\(\)](#), [desc\\_get\\_deps\(\)](#), [desc\\_has\\_dep\(\)](#), [desc\\_set\\_deps\(\)](#)

---

desc_set_deps	<i>Set all package dependencies in DESCRIPTION</i>
---------------	--

---

**Description**

Set all package dependencies in DESCRIPTION

**Usage**

```
desc_set_deps(deps, file = ".", normalize = FALSE)
```

**Arguments**

deps	Package dependency data frame, in the same format returned by <a href="#">desc_get_deps()</a> .
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

**See Also**

Other dependencies: [desc\\_del\\_deps\(\)](#), [desc\\_del\\_dep\(\)](#), [desc\\_get\\_deps\(\)](#), [desc\\_has\\_dep\(\)](#), [desc\\_set\\_dep\(\)](#)



---

desc\_set\_remotes      *Set the Remotes field in DESCRIPTION*

---

**Description**

The specified locations replace the current ones. The Remotes field is created if it does not exist currently.

**Usage**

```
desc_set_remotes(remotes, file = ".", normalize = FALSE)
```

**Arguments**

remotes	A character vector of remote locations to set.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

---

desc\_set\_urls      *Set the URL field in DESCRIPTION*

---

**Description**

The specified urls replace the current ones. The URL field is created if it does not exist currently.

**Usage**

```
desc_set_urls(urls, file = ".", normalize = FALSE)
```

**Arguments**

urls	A character vector of urls to set.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

---

desc\_set\_version      *Set the package version in DESCRIPTION*

---

### Description

Both `$set_version()` and `$bump_version()` use dots to separate the version number components.

### Usage

```
desc_set_version(version, file = ".", normalize = FALSE)
```

### Arguments

version	A string or a <a href="#">base::package_version</a> object.
file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
normalize	Whether to normalize the write when writing back the result. See <a href="#">desc_normalize()</a> .

### See Also

Other version numbers: [desc\\_bump\\_version\(\)](#), [desc\\_get\\_version\(\)](#)

---

desc\_to\_latex      *Converts a DESCRIPTION file to LaTeX*

---

### Description

Returns the contents of the DESCRIPTION in LaTeX format.

### Usage

```
desc_to_latex(file = ".")
```

### Arguments

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
------	--

---

desc_validate	<i>Validate a DESCRIPTION file</i>
---------------	------------------------------------

---

**Description**

This function is not implemented yet.

**Usage**

```
desc_validate(file = ".")
```

**Arguments**

file	DESCRIPTION file to use. By default the DESCRIPTION file of the current package (i.e. the package the working directory is part of) is used.
------	--

# Index

- \* **Authors@R**
  - desc\_add\_author, 12
  - desc\_add\_author\_gh, 13
  - desc\_add\_me, 14
  - desc\_add\_orcid, 15
  - desc\_add\_role, 16
  - desc\_change\_maintainer, 19
  - desc\_coerce\_authors\_at\_r, 21
  - desc\_del\_author, 22
  - desc\_del\_role, 26
  - desc\_get\_author, 28
  - desc\_get\_authors, 29
  - desc\_get\_maintainer, 32
  - desc\_set\_authors, 38
- \* **Collate field**
  - desc\_add\_to\_collate, 17
  - desc\_del\_collate, 23
  - desc\_del\_from\_collate, 25
  - desc\_get\_collate, 30
  - desc\_set\_collate, 39
- \* **built**
  - desc\_get\_built, 29
- \* **datasets**
  - cran\_ascii\_fields, 3
  - cran\_valid\_fields, 4
  - dep\_types, 4
- \* **dependencies**
  - desc\_del\_dep, 23
  - desc\_del\_deps, 24
  - desc\_get\_deps, 30
  - desc\_has\_dep, 34
  - desc\_set\_dep, 39
  - desc\_set\_deps, 40
- \* **field types**
  - cran\_ascii\_fields, 3
  - cran\_valid\_fields, 4
  - dep\_types, 4
- \* **repair functions**
  - desc\_normalize, 35
  - desc\_reformat\_fields, 36
  - desc\_reorder\_fields, 37
- \* **simple queries**
  - desc\_del, 21
  - desc\_fields, 27
  - desc\_get, 28
  - desc\_get\_field, 31
  - desc\_has\_fields, 35
  - desc\_set, 37
- \* **version numbers**
  - desc\_bump\_version, 18
  - desc\_get\_version, 34
  - desc\_set\_version, 42
- base::package\_version, 8, 34, 42
- check\_field, 3
- cran\_ascii\_fields, 3, 4
- cran\_valid\_fields, 4, 4
- dep\_types, 4, 4, 8, 9
- desc, 5
- desc\_add\_author, 12, 14–17, 19, 21, 23, 26, 29, 32, 38
- desc\_add\_author\_gh, 13, 13, 15–17, 19, 21, 23, 26, 29, 32, 38
- desc\_add\_me, 13, 14, 14, 16, 17, 19, 21, 23, 26, 29, 32, 38
- desc\_add\_orcid, 13–15, 15, 17, 19, 21, 23, 26, 29, 32, 38
- desc\_add\_remotes, 16
- desc\_add\_role, 13–16, 16, 19, 21, 23, 26, 29, 32, 38
- desc\_add\_to\_collate, 17, 23, 25, 30, 39
- desc\_add\_urls, 18
- desc\_bump\_version, 18, 34, 42
- desc\_change\_maintainer, 13–17, 19, 21, 23, 26, 29, 32, 38
- desc\_clear\_remotes, 20
- desc\_clear\_urls, 20

`desc_coerce_authors_at_r`, 13–17, 19, 21, 23, 26, 29, 32, 38  
`desc_del`, 21, 27, 28, 32, 35, 38  
`desc_del_author`, 13–17, 19, 21, 22, 26, 29, 32, 38  
`desc_del_collate`, 17, 23, 25, 30, 39  
`desc_del_dep`, 23, 24, 31, 35, 40  
`desc_del_deps`, 24, 24, 31, 35, 40  
`desc_del_from_collate`, 17, 23, 25, 30, 39  
`desc_del_remotes`, 25  
`desc_del_role`, 13–17, 19, 21, 23, 26, 29, 32, 38  
`desc_del_urls`, 27  
`desc_fields`, 22, 27, 28, 32, 35, 38  
`desc_get`, 22, 27, 28, 32, 35, 38  
`desc_get_author`, 13–17, 19, 21, 23, 26, 28, 29, 32, 38  
`desc_get_authors`, 13–17, 19, 21, 23, 26, 29, 29, 32, 38  
`desc_get_built`, 29  
`desc_get_collate`, 17, 23, 25, 30, 39  
`desc_get_deps`, 24, 30, 35, 40  
`desc_get_deps()`, 40  
`desc_get_field`, 22, 27, 28, 31, 35, 38  
`desc_get_list(desc_get_field)`, 31  
`desc_get_list()`, 32  
`desc_get_maintainer`, 13–17, 19, 21, 23, 26, 29, 32, 38  
`desc_get_or_fail(desc_get_field)`, 31  
`desc_get_remotes`, 33  
`desc_get_urls`, 33  
`desc_get_version`, 19, 34, 42  
`desc_has_dep`, 24, 31, 34, 40  
`desc_has_fields`, 22, 27, 28, 32, 35, 38  
`desc_normalize`, 35, 36, 37  
`desc_normalize()`, 13–27, 29, 37–42  
`desc_print`, 36  
`desc_reformat_fields`, 36, 36, 37  
`desc_reorder_fields`, 36, 37  
`desc_set`, 22, 27, 28, 32, 35, 37  
`desc_set_authors`, 13–17, 19, 21, 23, 26, 29, 32, 38  
`desc_set_collate`, 17, 23, 25, 30, 39  
`desc_set_dep`, 24, 31, 35, 39, 40  
`desc_set_deps`, 24, 31, 35, 40, 40  
`desc_set_list(desc_set)`, 37  
`desc_set_remotes`, 41  
`desc_set_urls`, 41  
`desc_set_version`, 19, 34, 42  
`desc_to_latex`, 42  
`desc_validate`, 43  
`description`, 5  
`utils::as.person()`, 21  
`utils::person()`, 10, 28, 29, 38