

Package ‘desc’

May 1, 2018

Title Manipulate DESCRIPTION Files

Version 1.2.0

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

LazyData true

URL <https://github.com/r-lib/desc#readme>

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

Depends R (>= 3.1.0)

Suggests covr, testthat, whoami, withr

Imports assertthat, utils, R6, crayon, rprojroot

Encoding UTF-8

RoxygenNote 6.0.1.9000

Collate 'assertions.R' 'authors-at-r.R' 'built.R' 'classes.R'
'collate.R' 'constants.R' 'deps.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]

Repository CRAN

Date/Publication 2018-05-01 20:48:05 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_me	13
desc_add_remotes	14
desc_add_role	14
desc_add_to_collate	15
desc_add_urls	15
desc_bump_version	16
desc_change_maintainer	16
desc_clear_remotes	17
desc_clear_urls	18
desc_del	18
desc_del_author	19
desc_del_collate	19
desc_del_dep	20
desc_del_deps	21
desc_del_from_collate	21
desc_del_remotes	22
desc_del_role	22
desc_del_urls	23
desc_fields	23
desc_get	24
desc_get_author	24
desc_get_authors	25
desc_get_built	25
desc_get_collate	26
desc_get_deps	26
desc_get_field	27
desc_get_maintainer	28
desc_get_remotes	28
desc_get_urls	29
desc_get_version	29
desc_has_dep	30
desc_has_fields	30
desc_normalize	31
desc_print	31
desc_reformat_fields	32
desc_reorder_fields	32
desc_set	33
desc_set_authors	33
desc_set_collate	34

check_field 3

desc_set_dep	34
desc_set_deps	35
desc_set_remotes	36
desc_set_urls	36
desc_set_version	37
desc_to_latex	37
desc_validate	38

Index 39

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 <code>"!new"</code> is implemented. If it does not start with an exclamation mark, it will be interpreted as the <code>'file'</code> argument.
file	Name of the DESCRIPTION file to load. If all of <code>'cmd'</code> , <code>'file'</code> and <code>'text'</code> 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

Usage

```
description
```

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)
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.

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 `normalize` argument controls whether the fields are reformatted according to a standard style. By default they are not.

The API:

```
description$write(file = NULL, normalize = 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.

normalize: Whether to reformat the fields in a standard way.

Version numbers

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

version: A string or a [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 [package_version](#) object. It throws an error if the package does not have a 'Version' field.

`$set_version()` takes a string or a [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()
```

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 [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.

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

Add a new author. The arguments correspond to the arguments of the [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.

```
description$del_author(given = NULL, family = NULL, email = NULL,
  role = NULL, comment = 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$add_role(role, given = NULL, family = NULL, email = NULL,
  comment = NULL)
description$del_role(role, given = NULL, family = NULL, email = NULL,
  comment = NULL)
description$change_maintainer(given = NULL, family = NULL,
  email = NULL, comment = 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, file = ".", normalize = FALSE)
```

Arguments

given	Given name.
family	Family name.
email	Email address.
role	Role.
comment	Comment.
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 Authors@R: [desc_add_me](#), [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_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, file = ".", normalize = FALSE)
```

Arguments

role	Role to set for the user, defaults to contributor.
comment	Comment, 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 desc_normalize .

See Also

Other Authors@R: [desc_add_author](#), [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_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 desc_normalize .

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 and comment 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, 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.
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 Authors@R: [desc_add_author](#), [desc_add_me](#), [desc_change_maintainer](#), [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 desc_normalize .

See Also

Other Collate field: [desc_del_collate](#), [desc_del_from_collate](#), [desc_get_collate](#), [desc_set_collate](#)

desc_add_urls *Add URLs to the URL field in DESCRIPTION*

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 desc_normalize .

desc_bump_version *Increase the version number in DESCRIPTION*

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 desc_normalize .

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

Arguments

given	Given name.
family	Family name.
email	Email address.
comment	Comment.
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 .

Details

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

See Also

Other Authors@R: [desc_add_author](#), [desc_add_me](#), [desc_add_role](#), [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 desc_normalize .

desc_clear_urls *Remove all URLs from the URL field of DESCRIPTION*

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 desc_normalize .

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 desc_normalize .

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, 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.
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 Authors@R: [desc_add_author](#), [desc_add_me](#), [desc_add_role](#), [desc_change_maintainer](#), [desc_del_role](#), [desc_get_authors](#), [desc_get_author](#), [desc_get_maintainer](#), [desc_set_authors](#)

desc_del_collate	<i>Delete the Collate field from DESCRIPTION</i>
------------------	--

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	<i>Remove a package dependency from DESCRIPTION</i>
--------------	---

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. <code>LinkingTo</code> and <code>Imports</code> . 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 desc_normalize .

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 desc_normalize .

See Also

Other dependencies: [desc_del_dep](#), [desc_get_deps](#), [desc_has_dep](#), [desc_set_deps](#), [desc_set_dep](#)

desc_del_from_collate	<i>Remove files from the Collate field.</i>
-----------------------	---

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 desc_normalize .

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 desc_normalize .

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 and comment 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, 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.
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 Authors@R: [desc_add_author](#), [desc_add_me](#), [desc_add_role](#), [desc_change_maintainer](#), [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 desc_normalize .

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 Also

Other Authors@R: [desc_add_author](#), [desc_add_me](#), [desc_add_role](#), [desc_change_maintainer](#), [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 Also

Other Authors@R: [desc_add_author](#), [desc_add_me](#), [desc_add_role](#), [desc_change_maintainer](#), [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

Get a single field from a DESCRIPTION file, fail if not found

Usage

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

```
desc_get_or_fail(keys, 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.
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.

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](#), [desc_add_me](#), [desc_add_role](#), [desc_change_maintainer](#), [desc_del_author](#), [desc_del_role](#), [desc_get_authors](#), [desc_get_author](#), [desc_set_authors](#)

desc_get_remotes *List the locations in the Remotes field in DESCRIPTION*

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	<i>Query the package version in DESCRIPTION</i>
------------------	---

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 [package_version](#) object.

See Also

Other version numbers: [desc_bump_version](#), [desc_set_version](#)

desc_has_dep	<i>Check for a dependency</i>
--------------	-------------------------------

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.

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 *Reformat fields in a DESCRIPTION file*

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)
```

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 .

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.

See Also

Other simple queries: [desc_del](#), [desc_fields](#), [desc_get_field](#), [desc_get](#), [desc_has_fields](#)

desc_set_authors *Set authors in Authors@R, in DESCRIPTION*

Description

Set authors in Authors@R, in DESCRIPTION

Usage

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

Arguments

authors	Authors, to set, a person 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 desc_normalize .

See Also

Other Authors@R: [desc_add_author](#), [desc_add_me](#), [desc_add_role](#), [desc_change_maintainer](#), [desc_del_author](#), [desc_del_role](#), [desc_get_authors](#), [desc_get_author](#), [desc_get_maintainer](#)

desc_set_collate *Set the Collate field in DESCRIPTION*

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 desc_normalize .

See Also

Other Collate field: [desc_add_to_collate](#), [desc_del_collate](#), [desc_del_from_collate](#), [desc_get_collate](#)

desc_set_dep *Add a package dependency to a DESCRIPTION file*

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 desc_normalize .

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 desc_get_deps .
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 dependencies: [desc_del_deps](#), [desc_del_dep](#), [desc_get_deps](#), [desc_has_dep](#), [desc_set_dep](#)

desc_set_remotes	<i>Set the Remotes field in DESCRIPTION</i>
------------------	---

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 desc_normalize .

desc_set_urls	<i>Set the URL field in DESCRIPTION</i>
---------------	---

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 desc_normalize .

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 package_version 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 desc_normalize .

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

*Topic **datasets**

- cran_ascii_fields, 3
 - cran_valid_fields, 4
 - dep_types, 4
 - description, 5
- check_field, 3
- cran_ascii_fields, 3, 4
- cran_valid_fields, 4, 4
- dep_types, 4, 4, 9
- desc, 5
- desc_add_author, 12, 13, 15, 17, 19, 23, 25, 28, 34
- desc_add_me, 13, 13, 15, 17, 19, 23, 25, 28, 34
- desc_add_remotes, 14
- desc_add_role, 13, 14, 17, 19, 23, 25, 28, 34
- desc_add_to_collate, 15, 20, 21, 26, 34
- desc_add_urls, 15
- desc_bump_version, 16, 29, 37
- desc_change_maintainer, 13, 15, 16, 19, 23, 25, 28, 34
- desc_clear_remotes, 17
- desc_clear_urls, 18
- desc_del, 18, 23, 24, 27, 31, 33
- desc_del_author, 13, 15, 17, 19, 23, 25, 28, 34
- desc_del_collate, 15, 19, 21, 26, 34
- desc_del_dep, 20, 21, 27, 30, 35
- desc_del_deps, 20, 21, 27, 30, 35
- desc_del_from_collate, 15, 20, 21, 26, 34
- desc_del_remotes, 22
- desc_del_role, 13, 15, 17, 19, 22, 25, 28, 34
- desc_del_urls, 23
- desc_fields, 18, 23, 24, 27, 31, 33
- desc_get, 18, 23, 24, 27, 31, 33
- desc_get_author, 13, 15, 17, 19, 23, 24, 25, 28, 34
- desc_get_authors, 13, 15, 17, 19, 23, 25, 25, 28, 34
- desc_get_built, 25
- desc_get_collate, 15, 20, 21, 26, 34
- desc_get_deps, 20, 21, 26, 30, 35
- desc_get_field, 18, 23, 24, 27, 31, 33
- desc_get_maintainer, 13, 15, 17, 19, 23, 25, 28, 34
- desc_get_or_fail (desc_get_field), 27
- desc_get_remotes, 28
- desc_get_urls, 29
- desc_get_version, 16, 29, 37
- desc_has_dep, 20, 21, 27, 30, 35
- desc_has_fields, 18, 23, 24, 27, 30, 33
- desc_normalize, 13–23, 25, 31, 32–37
- desc_print, 31
- desc_reformat_fields, 31, 32, 32
- desc_reorder_fields, 31, 32, 32
- desc_set, 18, 23, 24, 27, 31, 33
- desc_set_authors, 13, 15, 17, 19, 23, 25, 28, 33
- desc_set_collate, 15, 20, 21, 26, 34
- desc_set_dep, 20, 21, 27, 30, 34, 35
- desc_set_deps, 20, 21, 27, 30, 35, 35
- desc_set_remotes, 36
- desc_set_urls, 36
- desc_set_version, 16, 29, 37
- desc_to_latex, 37
- desc_validate, 38
- description, 5
- package_version, 8, 29, 37
- person, 10, 24, 25, 33