Package 'rcldf'

September 30, 2025

Type Package

Title Read Linguistic Data in the Cross Linguistic Data Format (CLDF)

Version 1.5.1

Maintainer Simon J. Greenhill <simon@simon.net.nz>

Description Cross-Linguistic Data Format (CLDF) is a framework for storing cross-linguistic data, ensuring compatibility and ease of data exchange between different linguistic datasets see Forkel et al. (2018) <doi:10.1038/sdata.2018.205>. The 'rcldf' package is designed to facilitate the manipulation and analysis of these datasets by simplifying the loading, querying, and visualisation of CLDF datasets making it easier to conduct comparative linguistic analyses, manage language data, and apply statistical methods directly within R.

License Apache License (>= 2.0)

Encoding UTF-8

Imports archive, bib2df (>= 1.1.1), csvwr, digest, dplyr, jsonlite, logger, magrittr, purrr, readr, remotes, rlang, tools, urltools, utils

Suggests ggplot2, patchwork, testthat, mockthat, spelling, covr, knitr, rmarkdown, qpdf

URL https://github.com/SimonGreenhill/rcldf

BugReports https://github.com/SimonGreenhill/rcldf/issues

Language en-US

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Author Simon J. Greenhill [aut, cre]

Repository CRAN

Date/Publication 2025-09-30 07:20:02 UTC

Contents

`add_dataframe` *Adds a dataframe.*

Description

Adds a dataframe.

Usage

```
add_dataframe(table, filename, group)
```

Arguments

table	a metadata section from the CLDF metadata.
filename	the filename.
group	a grouping from the metadata.

Value

A dataframe

---

as.cldf.wide — *Extracts a CLDF table as a 'wide' dataframe by resolving all foreign key links*

Description

Extracts a CLDF table as a 'wide' dataframe by resolving all foreign key links

Usage

```
as.cldf.wide(object, table)
```

Arguments

object	the CLDF dataset.
table	the name of the table to extract.

Value

A tibble dataframe

Examples

```
md <- system.file("extdata/huon", "cldf-metadata.json", package = "rcldf")
cldfobj <- cldf(md)
forms <- as.cldf.wide(cldfobj, 'FormTable')
```

cldf *Reads a Cross-Linguistic Data Format dataset into an object.*

Description

Reads a Cross-Linguistic Data Format dataset into an object.

included here to match people expecting e.g. readr::read_csv etc

Usage

```
cldf(
  mdpath,
  load_bib = FALSE,
  cache_dir = tools::R_user_dir("rcldf", which = "cache")
)

read_cldf(
  mdpath,
  load_bib = FALSE,
  cache_dir = tools::R_user_dir("rcldf", which = "cache")
)
```

Arguments

mdpath	the path to the directory or metadata JSON file.
load_bib	a boolean flag (TRUE/FALSE, default FALSE) to load the sources.bib BibTeX file. load_bib=FALSE can easily speed up loading of a CLDF dataset by an order of magnitude or two, so we do not load sources by default.
cache_dir	a directory to cache downloaded files to

Value

A cldf object

Examples

```
cldfobj <- cldf(system.file("extdata/huon", "cldf-metadata.json", package = "rcldf"))
```

coalesce_truth *Coalesce value to truthiness*

Description

Determine whether the input is true, with missing values being interpreted as false.

Usage

```
coalesce_truth(x)
```

Arguments

x	logical, NA or NULL

Value

FALSE if x is anything but TRUE

datatype_to_type *Map csvw datatypes to R types*

Description

Translate csvw datatypes to R types. This implementation currently targets readr::cols column specifications.

Usage

```
datatype_to_type(datatypes)
```

Arguments

datatypes	a list of csvw datatypes

Details

rcldf adds some overrides here to add e.g. anyURI etc.

Value

a `readr::cols` specification - a list of collectors

Examples

```
cspec <- datatype_to_type(list("double", list(base="date", format="yyyy-MM-dd")))
readr::read_csv(readr::readr_example("challenge.csv"), col_types=cspec)
```

default_dialect *CSVW default dialect*

Description

The CSVW Default Dialect specification described in CSV Dialect Description Format.

Usage

```
default_dialect
```

Format

An object of class `list` of length 13.

Value

a list specifying a default csv dialect

default_schema *Create a default table schema given a csv file and dialect*

Description

If neither the table nor the group have a `tableSchema` annotation, then this default schema will used.

Usage

```
default_schema(filename, dialect = default_dialect)
```

Arguments

`filename`	a csv file
`dialect`	specification of the csv's dialect (default: `default_dialect`)

Value

a table schema

get_cache_dir *Returns the cache dir.*

Description

Returns the cache dir.

Usage

```
get_cache_dir(cache_dir = NA)
```

Arguments

cache_dir	a directory to use

Value

A string of the cache dir

get_details *Returns a dataframe of with details on the CLDF dataset in* path.

Description

Returns a dataframe of with details on the CLDF dataset in path.

Usage

```
get_details(path, cache_dir = NA)
```

Arguments

path	the path to resolve
cache_dir	a directory to cache downloaded files to

Value

A dataframe.

get_dir_size *Returns the filesize in bytes of a directory.*

Description

Returns the filesize in bytes of a directory.

Usage

```
get_dir_size(path)
```

Arguments

path	a directory to size

Value

A numeric of the file size in bytes

get_filename *Get a filename from url value in metadata (handles .zip files)*

Description

Get a filename from url value in metadata (handles .zip files)

Usage

```
get_filename(base_dir, url)
```

Arguments

base_dir	the base_dir
url	the url statement

Value

A string

get_from_zenodo *Downloads and installs a CLDF dataset from a Zenodo endpoint*

Description

Downloads and installs a CLDF dataset from a Zenodo endpoint

Usage

```
get_from_zenodo(zid, load_bib = FALSE, cache_dir = NULL)
```

Arguments

zid	Zenodo endpoint conceptid
load_bib	load sources (TRUE/FALSE, default FALSE)
cache_dir	A cache_dir to use. If NULL it will use get_cache_dir

Value

A cldf object

get_separators *Identifies the separator characters specified by the CLDF metadata.*

Description

Identifies the separator characters specified by the CLDF metadata.

Usage

```
get_separators(metadata)
```

Arguments

metadata	• a CLDF metadata.

Value

A dataframe with three columns (name, separator, url).

get_tablename *Convert a CLDF URL tablename to a short tablename*

Description

Convert a CLDF URL tablename to a short tablename

Usage

```
get_tablename(conformsto, url = NA)
```

Arguments

conformsto	the dc:conforms to statement
url	the url statement

Value

A string

Examples

```
get_tablename("http://cldf.clld.org/v1.0/terms.rdf#ValueTable")
```

get_table_from *Extracts a single table from a CLDF dataset.*

Description

Extracts a single table from a CLDF dataset.

Usage

```
get_table_from(
  table,
  mdpath,
  cache_dir = tools::R_user_dir("rcldf", which = "cache")
)
```

Arguments

table	a CLDF table type
mdpath	a path to a CLDF file
cache_dir	a directory to cache downloaded files to

Value

a dataframe

Examples

```
md_json <- system.file("extdata/huon", "cldf-metadata.json", package = "rcldf")
df <- get_table_from("LanguageTable", md_json)
```

---

is_github *Returns TRUE if* url *looks like a github URL*

Description

Returns TRUE if url looks like a github URL

Usage

```
is_github(url)
```

Arguments

url	A string

Value

A boolean TRUE/FALSE

Examples

```
is_github('https://github.com/SimonGreenhill/rcldf/')
```

---

is_url *Returns TRUE if* url *looks like a URL*

Description

Returns TRUE if url looks like a URL

Usage

```
is_url(url)
```

Arguments

url	A string

Value

A boolean TRUE/FALSE

Examples

```
is_url('http://simon.net.nz')
```

`list_cache_files` *Returns a dataframe of directories in the cache dir*

Description

Returns a dataframe of directories in the cache dir

Usage

```
list_cache_files(cache_dir = NULL)
```

Arguments

`cache_dir`	the cache directory to use. If NULL then R_user_dir will be used.

Value

A dataframe of the directories

`load_clts` *Returns a CLDF dataset object of the latest CLTS version.*

Description

Returns a CLDF dataset object of the latest CLTS version.

Usage

```
load_clts(load_bib = FALSE, cache_dir = NULL)
```

Arguments

`load_bib`	load sources (TRUE/FALSE, default FALSE)
`cache_dir`	A cache_dir to use. If NULL it will use get_cache_dir

Value

A `cldf` object

load_concepticon *Returns a CLDF dataset object of the latest Concepticon version.*

Description

Returns a CLDF dataset object of the latest Concepticon version.

Usage

```
load_concepticon(load_bib = FALSE, cache_dir = NULL)
```

Arguments

load_bib	load sources (TRUE/FALSE, default FALSE)
cache_dir	A cache_dir to use. If NULL it will use get_cache_dir

Value

A cldf object

load_glottolog *Returns a CLDF dataset object of the latest glottolog version.*

Description

Returns a CLDF dataset object of the latest glottolog version.

Usage

```
load_glottolog(load_bib = FALSE, cache_dir = NULL)
```

Arguments

load_bib	load sources (TRUE/FALSE, default FALSE)
cache_dir	A cache_dir to use. If NULL it will use get_cache_dir

Value

A cldf object

make_cache_key *Returns the cachekey for the given path.*

Description

Returns the cachekey for the given path.

Usage

```
make_cache_key(path)
```

Arguments

path	a path to generate the cachekey for.

Value

A string.

nullify *Converts all values specified in the CLDF metadata as* null *to R's* NA.

Description

Note that this is run by default on loading a dataset with cldf()

Usage

```
nullify(cldfobj, nulls = NULL)
```

Arguments

cldfobj	a CLDF Object
nulls	a dataframe of null values to replace (default=NULL).

Value

A cldf object

Examples

```
cldfobj <- cldf(system.file("extdata/huon", "cldf-metadata.json", package = "rcldf"))
cldfobj <- nullify(cldfobj)
```

print.cldf *Summarises the CLDF file*

Description

Summarises the CLDF file

Usage

```
## S3 method for class 'cldf'
print(x, ...)
```

Arguments

x	the CLDF dataset
...	Arguments to be passed to or from other methods. Currently not used.

Value

No return value, called for side effects.

Examples

```
cldfobj <- cldf(system.file("extdata/huon", "cldf-metadata.json", package = "rcldf"))
print(cldfobj)
```

read_bib *Adds BibTeX source information into a CLDF dataset*

Description

Adds BibTeX source information into a CLDF dataset

Usage

```
read_bib(object)
```

Arguments

object	A CLDF object

Value

A tibble dataframe

relabel *Relabels a column in a dataset for merging.*

Description

Relabels a column in a dataset for merging.

Usage

```
relabel(column, table)
```

Arguments

column	the tablename.
table	the tablename.

Value

A string of "column.table"

resolve_path *Helper function to resolve the path (e.g. directory or md.json file)*

Description

Helper function to resolve the path (e.g. directory or md.json file)

Usage

```
resolve_path(path, cache_dir = NA)
```

Arguments

path	the path to resolve
cache_dir	a directory to cache downloaded files to

Value

A list of two items: `path` - string containing the path to the metadata.json file `metadata` - a csvwr metadata object

separate *Expands all values with separators.*

Description

Note that this is run by default on loading a dataset with cldf()

Usage

```
separate(cldfobj, separators = NULL)
```

Arguments

cldfobj	a CLDF Object
separators	a dataframe of separator values to replace (default=NULL).

Value

A cldf object

Examples

```
cldfobj <- cldf(system.file("extdata/huon", "cldf-metadata.json", package = "rcldf"))
cldfobj <- separate(cldfobj)
```

set_cache_dir *Sets the cache dir for the current session.*

Description

Sets the cache dir for the current session.

Usage

```
set_cache_dir(cache_dir = NA)
```

Arguments

cache_dir	a directory to use

Value

NULL. Sets an environment value.

summary.cldf *Summarises the CLDF file*

Description

Summarises the CLDF file

Usage

```
## S3 method for class 'cldf'
summary(object, ...)
```

Arguments

object	the CLDF dataset
...	Arguments to be passed to or from other methods. Currently not used.

Value

None

Examples

```
cldfobj <- cldf(system.file("extdata/huon", "cldf-metadata.json", package = "rcldf"))
summary(cldfobj)
```

Index