

Package ‘orgutils’

October 14, 2022

Type Package

Title Helper Functions for Org Files

Version 0.4-1

Date 2017-03-21

Author Enrico Schumann

Maintainer Enrico Schumann <es@enricoschumann.net>

Description Helper functions for Org files (<<http://orgmode.org/>>):
a generic function 'toOrg' for transforming R objects into Org
markup (most useful for data frames; there are also methods for
Dates/POSIXt) and a function to read Org tables into data frames.

License GPL (>= 2)

Imports textutils

Suggests RUnit

NeedsCompilation no

Repository CRAN

Date/Publication 2017-03-21 21:28:17 UTC

R topics documented:

orgutils-package	2
readOrg	2
toOrg	4
Index	7

orgutils-package *Org Utils*

Description

Helper functions to interact with Org files: read Org tables, convert R objects to Org markup.

Details

Org mode is a major mode for Emacs; see <http://orgmode.org/manual/Summary.html#Summary> for a summary of what it does.

The **orgutils** package provides helper functions for interacting with Org files (reading Org tables, convert R objects to Org markup) without Emacs. Since Org syntax is very human-readable, such conversions are useful also, for instance, in plain-text emails or reports.

There are several other packages that help you work with Org files as well, such as **orgR** or **ascii**.

Author(s)

Enrico Schumann <es@enricoschumann.net>

References

Org mode manual <http://orgmode.org/>

See Also

[toOrg](#), [readOrg](#)

readOrg *Read Org Tables*

Description

Read an Org table from a file.

Usage

```
readOrg(file, header = TRUE, dec = ".", comment.char = "",
        encoding = "", strip.white = TRUE,
        stringsAsFactors = FALSE,
        table.name = NULL, ...)
```

Arguments

file	character
header	logical
dec	character
comment.char	character
encoding	the encoding of the file
strip.white	logical
stringsAsFactors	logical; note that the default FALSE differs from read.csv
table.name	character: the name of the table to read (a regular expression)
...	further arguments

Details

When header is TRUE, readOrg will read the first 10 lines of the file (or the table, when table.name is specified) and try to find an org table header. Formatting instructions such as <5> are discarded. Then the function uses [read.csv](#) to read the remainder of the file/table.

When table.name is specified, the function looks for a line that starts with #+NAME: <table.name> and reads the table that follows that line.

For empty files, readOrg behaves like [read.csv](#): when completely empty, it fails; when headers are found, a zero-row [data.frame](#) is returned.

Value

A [data.frame](#).

Author(s)

Enrico Schumann

References

Org manual <http://orgmode.org/manual/index.html>

See Also

[read.csv](#)

Examples

```
## Not run:
## create an Org file with a table and read the table
tmp <-
"#+TITLE: A Table

Next comes a table.
```

```

#+name: test_table
| a | b |
|---+---|
| 1 | 2 |
| 3 | 4 |

That was a table.
"

fname <- tempfile("testfile", fileext = ".org")
writeLines(tmp, fname)

require("org")
readOrg(fname, table.name = "test_table")

## End(Not run)

```

toOrg

Generate Org-mode Markup

Description

Transform R objects into Org-mode objects.

Usage

```

toOrg(x, ...)

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

## S3 method for class 'data.frame'
toOrg(x, row.names = NULL, ...)

## S3 method for class 'Date'
toOrg(x, inactive = FALSE, ...)

## S3 method for class 'POSIXt'
toOrg(x, inactive = FALSE, ...)

```

Arguments

x	an object
row.names	NULL, logical or character. If TRUE, <code>row.names</code> of x are added as the first column, with column name "row.names". If a character string, the string is used as the column name. See Examples. If NULL, <code>row.names</code> are added when they are not 1, 2, ... (i.e. row numbers). If FALSE, <code>row.names</code> are not added.

inactive logical: use inactive timestamps? See <http://orgmode.org/manual/Creating-timestamps.html> .

... other arguments

Details

Transforms an object `x` into character vectors with Org markup. Most useful when `x` is a [data.frame](#). `toOrg` is meant for snippets of code, not for producing whole Org documents.

When you work with POSIXt, make sure that a potential timezone does not cause trouble: Org does not support timezones.

Value

A character vector, usually with class `org`. In some cases, class `character` is additionally attached. To save it to a file, use [writeLines](#).

Author(s)

Enrico Schumann

References

Org mode manual <http://orgmode.org/manual/index.html>

See Also

[toLatex](#), [as.orgtable](#)

Examples

```
toOrg(data.frame(a = 1:3, row.names = LETTERS[1:3]))
## => | row.names | a |
##   |-----+---|
##   | A           | 1 |
##   | B           | 2 |
##   | C           | 3 |
```

```
toOrg(data.frame(a = 1:3))
## => | a |
##   |---|
##   | 1 |
##   | 2 |
##   | 3 |
```

```
toOrg(data.frame(a = 1:3), row.names = TRUE)
## => | row.names | a |
##   |-----+---|
##   | 1           | 1 |
##   | 2           | 2 |
##   | 3           | 3 |
```

```

toOrg(data.frame(a = 1:5), row.names = "row numbers")
## => | row numbers | a |
##    |-----+---|
##    | 1           | 1 |
##    | 2           | 2 |
##    | 3           | 3 |
##    | 4           | 4 |
##    | 5           | 5 |

## Not run:
writeLines(toOrg(data.frame(a = 1:3)), "~/Desktop/my_table.org")
## End(Not run)

## Dates/Times
toOrg(as.Date("2015-01-01"))           ## <2015-01-01 Thu>
toOrg(as.Date("2015-01-01"), inactive = TRUE) ## [2015-01-01 Thu]
toOrg(Sys.time())                     ## <2017-03-20 Mon 13:23:18>

## Convert Org dates to Date

## see ?strptime: Each input string is processed as far as
##                 necessary for the format specified: any
##                 trailing characters are ignored.
d <- toOrg(as.Date("2015-01-01"))
as.Date(d, "<%Y-%m-%d")

```

Index

- * **package**
 - orgutils-package, 2
- as.orgtable, 5
- character, 5
- data.frame, 3, 5
- org (orgutils-package), 2
- orgutils-package, 2
- print.org (toOrg), 4
- read.csv, 3
- readOrg, 2, 2
- row.names, 4
- toLatex, 5
- toOrg, 2, 4
- writelines, 5