Package 'readODS'

September 2, 2026

Type Package

Title Read and Write ODS Files

Version 2.3.6

Description Read ODS (OpenDocument Spreadsheet) into R as data frame. Also support writing data frame into ODS file.

URL https://docs.ropensci.org/readODS/, https://github.com/ropensci/readODS

BugReports https://github.com/ropensci/readODS/issues

Imports cellranger, minty (>= 0.0.5), stringi, tibble, vctrs (>= 0.4.2), zip, tools, withr

LinkingTo cpp11 (>= 0.5.0)

Suggests spelling, testthat, datasets, covr, knitr, rmarkdown, readr (>= 1.2.1), desc

License GPL-3

RoxygenNote 7.3.2

Encoding UTF-8

VignetteBuilder knitr

Depends R (>= 4.1)

Language en-GB

NeedsCompilation yes

Author Gerrit-Jan Schutten [aut],
Chung-hong Chan [aut, cre] (ORCID: <https://orcid.org/0000-0002-6232-7530>),
Peter Brohan [aut],
Detlef Steuer [aut] (ORCID: <https://orcid.org/0000-0003-2676-5290>),
Thomas J. Leeper [aut] (ORCID: <https://orcid.org/0000-0003-4097-6326>),
John Foster [ctb],
Sergio Oller [ctb],
Jim Hester [ctb] (ORCID: <https://orcid.org/0000-0002-2739-7082>),
Stephen Watts [ctb],

Arthur Katossky [ctb],
Stas Malavin [ctb],
Duncan Garmonsway [ctb],
Mehrad Mahmoudian [ctb],
Matt Kerlogue [ctb],
Michal Lauer [ctb],
Till Straube [ctb],
Mauricio Vargas Sepulveda [ctb],
Marcin Kalicinski [ctb, cph] (Author of included RapidXML code)

Maintainer Chung-hong Chan <chainsawtiney@gmail.com>

Repository CRAN

Date/Publication 2026-09-02 05:20:13 UTC

Contents

list_ods_sheets 2
readODS_progress 4
read_ods 4
write_ods 7

Index **10**

`list_ods_sheets` *Get information in an (F)ODS File*

Description

`list_ods_sheets` lists all sheets in an (f)ods file. The function can be used for listing sheets in both ods and flat ods files. (`list_fods_sheets`) is also available, which can only list sheets in flat ods files.

Usage

```
list_ods_sheets(
  path,
  include_external_data = FALSE,
  ods_format = c("auto", "ods", "fods"),
  guess = FALSE
)

list_fods_sheets(path, include_external_data = FALSE)

ods_sheets(path)
```

Arguments

path	Path to the (f)ods file
include_external_data	A boolean value to show or hide sheets containing archived linked data (default false)
ods_format	character, must be "auto", "ods" or "fods". The default "auto" is to determine the format automatically. By default, the format is determined by file extension, unless guess is FALSE.
guess	logical, If the file extension is absent or not recognized, this controls whether we attempt to guess format based on the file signature or "magic number".

Details

The default "include_external_data" for ods_sheets is TRUE to maintain compatibility with version 1 of readODS. It will change to TRUE in version 3.

Value

A character vector of sheet names

Author(s)

Peter Brohan peter.brohan+cran@gmail.com, Chung-hong Chan chainsawtiney@gmail.com, Gerrit-Jan Schutten phonixor@gmail.com

See Also

use read_ods to read the data

Examples

```
## Not run:
# Get the list of names of sheets
list_ods_sheets("starwars.ods")
list_ods_sheets("starwars.fods")
# Using list_fods_sheets, although you don't have to
list_fods_sheets("starwars.fods")

## End(Not run)
```

readODS_progress *Determine whether to show progress bar*

Description

By default, readODS displays a progress bar unless one of the following is TRUE:

- The progress is explicitly disabled by setting options(readODS.show_progress = FALSE).
- The code is run in a non-interactive session (interactive() is FALSE).
- The code is run by knitr / rmarkdown.

Usage

```
readODS_progress()
```

Value

logical indicating whether to show progress

read_ods *Read Data From (F)ODS File*

Description

read_ods is a function to read a single sheet from an (f)ods file and return a data frame. The function can be used for reading both ods and flat ods files. (`read_fods`) is also available, which can only read flat ods files.

Usage

```
read_ods(
  path,
  sheet = 1,
  col_names = TRUE,
  col_types = NULL,
  na = "",
  skip = 0,
  formula_as_formula = FALSE,
  range = NULL,
  row_names = FALSE,
  strings_as_factors = FALSE,
  verbose = FALSE,
  as_tibble = TRUE,
  .name_repair = "unique",
  ods_format = c("auto", "ods", "fods"),
```

```
  guess = FALSE,
  trim_ws = TRUE,
  n_max = Inf,
  guess_max = min(1000, n_max),
  progress = readODS_progress()
)

read_fods(
  path,
  sheet = 1,
  col_names = TRUE,
  col_types = NULL,
  na = "",
  skip = 0,
  formula_as_formula = FALSE,
  range = NULL,
  row_names = FALSE,
  strings_as_factors = FALSE,
  verbose = FALSE,
  as_tibble = TRUE,
  .name_repair = "unique",
  trim_ws = TRUE,
  n_max = Inf,
  guess_max = min(1000, n_max),
  progress = readODS_progress()
)
```

Arguments

path	path to the (f)ods file.
sheet	sheet to read. Either a string (the sheet name), or an integer sheet number. The default is 1.
col_names	logical, indicating whether the file contains the names of the variables as its first line. Default is TRUE.
col_types	Either NULL to guess from the spreadsheet or refer to `readr::type_convert()` to specify cols specification. It can also be a shorthand such as "ccf" ("character", "character", "factor"), a list, or an object created by `readr::cols()`. NA will return a data frame with all columns being "characters". Please note that it will not speed up the reading by a lot by specifying this parameter explicitly. It is more for accuracy.
na	Character vector of strings to use for missing values. By default read_ods converts blank cells to missing data. It can also be set to NULL, so that empty cells are treated as NA.
skip	the number of lines of the data file to skip before beginning to read data. If this parameter is larger than the total number of lines in the ods file, an empty data frame is returned.

formula_as_formula
: logical, a switch to display formulas as formulas "SUM(A1:A3)" or as the resulting value "3"... or "8".. . Default is FALSE.

range
: selection of rectangle using Excel-like cell range, such as `range = "D12:F15"` or `range = "R1C12:R6C15"`. Cell range processing is handled by the `cellranger::as.cell_limits()`. If sheet name is in the range, such as `range = "Sheet2!A2:B7"`, this sheet name is used instead of the provided `sheet`. If `sheet` is not the default value (1), a warning is given.

row_names
: logical, indicating whether the file contains the names of the rows as its first column. Default is FALSE.

strings_as_factors
: logical, if character columns to be converted to factors. Default is FALSE.

verbose
: logical, if messages should be displayed. Default is FALSE.

as_tibble
: logical, if the output should be a tibble (as opposed to a data.frame). Default is TRUE.

.name_repair
: A string or function passed on as `.name_repair` to `tibble::as_tibble()`

  - `"minimal"`: No name repair
  - `"unique"` : Make sure names are unique and not empty
  - `"check_unique"`: Check names are unique, but do not repair
  - `"universal"` : Checks names are unique and valid R variables names in scope
  - A function to apply custom name repair.

  Default is `"unique"`.

ods_format
: character, must be "auto", "ods" or "fods". The default "auto" is to determine the format automatically. By default, the format is determined by file extension, unless `guess` is `FALSE`.

guess
: logical, If the file extension is absent or not recognized, this controls whether we attempt to guess format based on the file signature or "magic number".

trim_ws
: logical, should leading and trailing whitespace be trimmed?

n_max
: numeric, Maximum number of data rows to read. Ignored if `range` is given.

guess_max
: numeric, Maximum number of data rows to use for guessing column types. Defaults to min(1000, n_max).

progress
: logical, Display a progress bar? By default, shows a progress bar in interactive sessions unless disabled. See `readODS_progress` for more details.

Value

A tibble (`tibble`) or data frame (`data.frame`) containing a representation of data in the (f)ods file.

Author(s)

Peter Brohan peter.brohan+cran@gmail.com, Chung-hong Chan chainsawtiney@gmail.com, Gerrit-Jan Schutten phonixor@gmail.com

Examples

```
## Not run:
# Read an ODS file
read_ods("starwars.ods")
# Read a specific sheet, e.g. the 2nd sheet
read_ods("starwars.ods", sheet = 2)
# Read a specific range, e.g. A1:C11
read_ods("starwars.ods", sheet = 2, range = "A1:C11")
# Read an FODS file
read_ods("starwars.fods")
# Read a specific sheet, e.g. the 2nd sheet
read_ods("starwars.fods", sheet = 2)
# Read a specific range, e.g. A1:C11
read_ods("starwars.fods", sheet = 2, range = "A1:C11")
# Give a warning and read from Sheet1 (not 2)
read_ods("starwars.fods", sheet = 2, range = "Sheet1!A1:C11")
# Specifying col_types as shorthand, the third column as factor; other by guessing
read_ods("starwars.ods", col_types = "??f")
# Specifying col_types as list
read_ods("starwars.ods", col_types = list(species = "f"))
# Using read_fods, although you don't have to
read_ods("starwars.fods")

## End(Not run)
```

`write_ods` *Write Data to (F)ODS File*

Description

Function to write a single data frame or a list of data frames to a (f)ods file.

Usage

```
write_ods(
  x,
  path = tempfile(fileext = ".ods"),
  sheet = "Sheet1",
  append = FALSE,
  update = FALSE,
  row_names = FALSE,
  col_names = TRUE,
  na_as_string = FALSE,
  padding = FALSE
)

write_fods(
  x,
```

```
  path = tempfile(fileext = ".fods"),
  sheet = "Sheet1",
  append = FALSE,
  update = FALSE,
  row_names = FALSE,
  col_names = TRUE,
  na_as_string = FALSE,
  padding = FALSE
)
```

Arguments

x	data frame or list of data frames that will be sheets in the (f)ods. If the list is named, the names are used as sheet names
path	Path to the (f)ods file to write
sheet	Name of the sheet; ignore if x is a list of data frames
append	logical, TRUE indicates that x should be appended to the existing file (path) as a new sheet. If a sheet with the same sheet_name exists, an exception is thrown. See update. Please also note that writing is slightly slower if TRUE. Default is FALSE. Ignore if x is a list of data frames
update	logical, TRUE indicates that the sheet with sheet_name in the existing file (path) should be updated with the content of x. If a sheet with sheet_name does not exist, an exception is thrown. Please also note that writing is slightly slower if TRUE. Default is FALSE. Ignore if x is a list of data frames
row_names	logical, TRUE indicates that row names of x are to be included in the sheet. Default is FALSE
col_names	logical, TRUE indicates that column names of x are to be included in the sheet. Default is TRUE
na_as_string	logical, TRUE indicates that NAs are written as string; FALSE indicates that NAs are written as empty cells
padding	logical, TRUE indicates that the sheet is padded with repeated empty cells to the maximum size, either 2^20 x 1024 (if the number of columns of x is less than or equal 1024) or 2^20 x 16,384 (otherwise). This is the default behaviour of Microsoft Excel. Default is FALSE

Details

This function emulates `writexl::write_xlsx()` except in the handling of list columns. The expected behaviour for this is undefined and the two functions behave differently. This function handles list columns by converting them to character vectors of R code (similar to the output of `dput()`), which is probably not ideal.

Value

A (F)ODS file written to the file path location specified by the user. The value of `path` is also returned invisibly

Author(s)

Detlef Steuer steuer@hsu-hh.de, Thomas J. Leeper thosjleeper@gmail.com, John Foster john.x.foster@nab.com.au, Chung-hong Chan chainsawtiney@gmail.com

Examples

```
## Not run:
# preserve the row names
write_ods(mtcars, "mtcars.ods", row_names = TRUE)
# append a sheet to an existing file
write_ods(PlantGrowth, "mtcars.ods", append = TRUE, sheet = "plant")
# This is however faster
write_ods(list("Sheet1" = mtcars, "plant" = PlantGrowth), "mtcars.ods", row_names = TRUE)
# write flat ODS file
write_fods(mtcars, "mtcars.fods", sheet = "mtcars")

## End(Not run)
```

Index

cellranger::as.cell_limits(), *6*

dput(), *8*

list_fods_sheets *(*list_ods_sheets*)*, 2
list_ods_sheets, 2

ods_sheets *(*list_ods_sheets*)*, 2

read_fods *(*read_ods*)*, 4
read_ods, *3*, 4
readODS_progress, 4, *6*
readr::cols(), *5*
readr::type_convert(), *5*

tibble::as_tibble(), *6*

write_fods *(*write_ods*)*, 7
write_ods, 7
writexl::write_xlsx(), *8*