

# Package ‘opendataformat’

May 9, 2026

**Title** Reading and Writing Open Data Format Files

**Version** 2.2.2

**Maintainer** Tom Hartl <tom.hartl196@gmail.com>

**Description** The Open Data Format (ODF) is a new, non-proprietary, multilingual, metadata enriched, and zip-compressed data format with metadata structured in the Data Documentation Initiative (DDI) Codebook standard. This package allows reading and writing of data files in the Open Data Format (ODF) in R, and displaying metadata in different languages. For further information on the Open Data Format, see <<https://opendataformat.github.io/>>.

**URL** <https://github.com/opendataformat/r-package-opendataformat>

**BugReports** <https://github.com/opendataformat/r-package-opendataformat/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**Imports** cli, zip, magrittr, xml2, data.table, tibble (>= 3.2.1),  
jsonlite

**Depends** R (>= 3.6)

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, devtools, ISLR, dplyr,  
haven

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Tom Hartl [aut, cre] (ORCID: <<https://orcid.org/0009-0006-4226-9835>>),  
Claudia Saalbach [ctb]

**Repository** CRAN

**Date/Publication** 2025-11-26 08:20:08 UTC

## Contents

opendataformat-package . . . . .	2
as_odf_tbl . . . . .	3
data_odf . . . . .	4
docu_odf . . . . .	5
getmetadata_odf . . . . .	6
merge.odf_tbl . . . . .	7
read_odf . . . . .	9
setlanguage_odf . . . . .	10
write_odf . . . . .	11
<b>Index</b>	<b>13</b>

---

opendataformat-package

*Open Data Format*

---

## Description

The package is designed to support the use of the open data format. For this purpose, three main functions have been developed:

**read\_odf():**

Import data from the Open Data Format to an R data frame.

**write\_odf():**

Export data from an R data frame to the open data format.

**docu\_odf():**

Get access to information about the dataset and variables via the R-Studio Viewer or the web browser.

**setlanguage\_odf():**

Set the default language for displaying the metadata for docu\_odf and getmetadata\_odf

**getmetadata\_odf():**

Retrieve specific metadata like variable labels, or value labels.

**as\_odf\_tbl():**

Convert data frame (data.frame object or any subclass) to an ODF tibble (odf\_tbl class object).

## Author(s)

Tom Hartl (<thartl@diw.de>), Claudia Saalbach (<csaalbach@diw.de>)

Other Contributors: KonsortSWD/NFDI, DIW Berlin

**See Also**

More information about the Open Data Format specification and data examples are available here: <https://opendataformat.github.io/>

---

as_odf_tbl	<i>Converts a data frame to odf_tbl</i>
------------	---

---

**Description**

Converts a data.frame (or any subclass) object to an odf\_tbl (Open Data Format tibble).

**Usage**

```
as_odf_tbl(x, active_language = "en", language_of_metadata = NA)
```

**Arguments**

`x` a data.frame that should be converted to an odf\_tbl. (languages = "all").

`active_language` Select the language that should be the active metadata language. Default is 'en', or the first language occurring.

`language_of_metadata` Language of metadata, where language tag is missing, which is metadata in attributes label, description and labels. Default is NA.

**Value**

odf\_tbl with attributes including dataset and variable information.

**Examples**

```
# Create a dataframe with 4 variables id, name, age, and diagnosis
exampledata <- data.frame(id = 1:5,
                          name = c("Klaus", "Anna", "Rebecca",
                                    "Kevin", "Janina"),
                          age = c(55, 40, 19, 25, 60),
                          diagnosis = c(1,3,3,2,1))

# Add metadata for dataset
attr(exampledata, "name") <- "patientdata"
attr(exampledata, "label_en") <- "Patient Data"
attr(exampledata, "description_en") <- "Patient database of the practice Dr. Sommer"
attr(exampledata, "url") <- "www.example.url.en"

# Add metadata for diagnosis variable with label, description and value labels.
attr(exampledata$id, "name") <- "id"
attr(exampledata$id, "label_en") <- "Patiend ID"
attr(exampledata$id, "description_en") <- "Practice Patiend ID"
attr(exampledata$diagnosis, "name") <- "diagnose"
```

```

attr(exampladata$diagnosis, "label_en") <- "Diagnosis"
attr(exampladata$diagnosis, "description_en") <- "Diagnosis patient last visit"
valuelabels_diagnosis <- 1:4
names(valuelabels_diagnosis) <- c("Covid", "Influenza", "Common cold", "Tonsillitis")
attr(exampladata$diagnosis, "labels_en") <- valuelabels_diagnosis
# use as_odf_tbl() to transform dataframe to odf_tibble
example_odf <- as_odf_tbl(exampladata)

# Display metadata using docu_odf
docu_odf(example_odf, style = "print")

# Display metadata of diagnosis Variable
docu_odf(example_odf$diagnosis, style = "print")

```

---

data\_odf

*data\_odf*

---

## Description

example data with attributes specified for the Open Data Format.

## Usage

```
data_odf
```

## Format

A data frame with 20 rows and 7 variables:

**bap87** Current Health.

**bap9201** Hours of sleep, normal workday.

**bap9001** Pressed For Time Last 4 Weeks.

**bap9002** Run-down, Melancholy Last 4 Weeks.

**bap9003** Well-balanced Last 4 Weeks.

**bap96** Height.

**name** Firstname.

## Source

<https://github.com/opendataformat/Specification/tree/main/Example>

---

docu\_odf *Get documentation from R data frame.*

---

### Description

Get access to information about the dataset and variables via the R-Studio Viewer or the web browser.

### Usage

```
docu_odf(
  input,
  languages = "current",
  style = "viewer",
  replace_missing_language = FALSE,
  variables = "yes"
)
```

### Arguments

input	R data frame (df) or variable from an R data frame (df\$var).
languages	Select the language in which the descriptions and labels of the data will be displayed. <ul style="list-style-type: none"> <li>• By default the language that is set to current is displayed (languages = "current").</li> <li>• The default-option chooses either the default language(if labels and descriptions without a language tag exist)Otherwise the current language is displayed. (languages = "default").</li> <li>• You can choose to view all available language variants by selecting (languages = "all"),</li> <li>• or you can select the language by language code, e.g. languages = "en".</li> </ul>
style	Selects where the output should be displayed (console ore viewer).By default the metadata information is displayed in the viewer if the viewer is available. (style = "console") (style = "print") <ul style="list-style-type: none"> <li>• You can choose to display the code in both the console and the viewer (style = "both") (style = "all")</li> <li>• You can choose to display the code only in the viewer (style = "viewer") (style = "html")</li> </ul>
replace_missing_language	If only one language is specified in languages and replace_missing_language is set to TRUE. In case of a missing label or description, the default or english label/description is displayed additionally (if one of these is available).
variables	Indicate whether a list with all the variables should be displayed with the dataset metadata. If the input is a variable/column, the variables-argument will be ignored. Set (variables = "yes") to display the list of variables.

**Value**

Documentation.

**Examples**

```
# get example data from the opendataformat package
df <- get(data("data_odf"))

# view documentation about the dataset in the language that is currently set
docu_odf(df)

# view information from a selected variable in language "en"
docu_odf(df$bap87, languages = "en")

# view dataset information for all available languages
docu_odf(df, languages = "all")

# print information to the R console
docu_odf(df$bap87, style = "print")

# print information to the R viewer
docu_odf(df$bap87, style = "viewer")

# Since the label for language de is missing, in this case the
# english label will be displayed additionally.
attributes(df$bap87)["label_de"] <- ""
docu_odf(df$bap87, languages = "de", replace_missing_language = TRUE)
```

---

getmetadata_odf	<i>Get variable labels or other metadata from a data frame in open-dataformat.</i>
-----------------	--

---

**Description**

Get access to information about the dataset and variables via the R-Studio Viewer or the web browser.

**Usage**

```
getmetadata_odf(input, type, language = "active")
```

**Arguments**

input	R data frame (df) or variable from an R data frame (df\$var).
type	The metadata type you want to retrieve. Possible options are "label", "description", "url", "type", "valuelabels", or "languages".

- language      Select the language in which the labels of the variables will be displayed. If no language is selected, the current/active language of the data frame will be used.
- By default the language that is set to current is displayed (language = "current").
  - You can select the language by language code, e.g. language = "en".

## Value

Documentation.

## Examples

```
# get example data from the opendataformat package
df <- get(data("data_odf"))
# view the variable labels for all variables in English
getmetadata_odf(input = df, type = "label", language = "en")

# view the value labels for variable bap87 in English
getmetadata_odf(input = df$bap87, type = "value_label", language = "en")

# view the description for variable bap87 in English
getmetadata_odf(input = df$bap87, type = "description", language = "en")
```

---

merge.odf_tbl	<i>Merge method for odf tibbles.</i>
---------------	--------------------------------------

---

## Description

Merge two odf tibbles in R while keeping attributes with metadata.

## Usage

```
## S3 method for class 'odf_tbl'
merge(
  x,
  y,
  by = NULL,
  by.x = NULL,
  by.y = NULL,
  all = FALSE,
  all.x = all,
  all.y = all,
  sort = TRUE,
  suffixes = c(".x", ".y"),
  no.dups = TRUE,
  allow.cartesian = getOption("datatable.allow.cartesian"),
  incomparables = NULL,
```

```
    ...
  )
```

## Arguments

<code>x, y</code>	odf tibbles, or objects to be coerced to one
<code>by</code>	A vector of shared column names in <code>x</code> and <code>y</code> to merge on. This defaults to the shared key columns between the two tables. If <code>y</code> has no key columns, this defaults to the key of <code>x</code> .
<code>by.x, by.y</code>	Vectors of column names in <code>x</code> and <code>y</code> to merge on.
<code>all</code>	logical; <code>all = TRUE</code> is shorthand to save setting both <code>all.x = TRUE</code> and <code>all.y = TRUE</code> .
<code>all.x</code>	logical; if <code>TRUE</code> , rows from <code>x</code> which have no matching row in <code>y</code> are included. These rows will have 'NA's in the columns that are usually filled with values from <code>y</code> . The default is <code>FALSE</code> so that only rows with data from both <code>x</code> and <code>y</code> are included in the output.
<code>all.y</code>	logical; analogous to <code>all.x</code> above.
<code>sort</code>	logical. If <code>TRUE</code> (default), the rows of the merged <code>data.table</code> are sorted by setting the key to the <code>by / by.x</code> columns. If <code>FALSE</code> , unlike base R's <code>merge</code> for which row order is unspecified, the row order in <code>x</code> is retained (including retaining the position of missings when <code>all.x=TRUE</code> ), followed by <code>y</code> rows that don't match <code>x</code> (when <code>all.y=TRUE</code> ) retaining the order those appear in <code>y</code> .
<code>suffixes</code>	A character(2) specifying the suffixes to be used for making non- <code>by</code> column names unique. The suffix behaviour works in a similar fashion as the <code>merge.data.frame</code> method does.
<code>no.dups</code>	logical indicating that suffixes are also appended to non- <code>by.y</code> column names in <code>y</code> when they have the same column name as any <code>by.x</code> .
<code>allow.cartesian</code>	See <code>allow.cartesian</code> in <a href="#">data.table</a> .
<code>incomparables</code>	values which cannot be matched and therefore are excluded from <code>by</code> columns.
<code>...</code>	Not used at this time.

## Details

`merge` is a generic function in base R. It dispatches to either the `merge.data.frame` method, `merge.odf_tbl` or `merge.data.table` method depending on the class of its first argument. `merge.odf_tbl` uses the `merge.data.table` to join `data.frame` and adds the attributes containing metadata from the two original `odf.data.frames`. Note that, unlike SQL join, NA is matched against NA (and NaN against NaN) while merging. For a more `data.table`-centric way of merging two `data.tables`, see [data.table](#). See FAQ 1.11 for a detailed comparison of `merge`.

## Value

A new `odf tibble` build from the two input `data.frames` with the variable attributes from the original `data.frames`. Sorted by the columns set (or inferred for) the `by` argument if argument `sort` is set to `TRUE`. For variables/columns occurring in both `x` and `y`, attributes are taken from `x`.

## Examples

```
# get path to example data from the opendataformat package (data.zip)
path <- system.file("extdata", "data.odf.zip", package = "opendataformat")

# read four columns of example data specified as ODF from ZIP file
df <- read_odf(file = path, select = 1:4)

# read other columns of example data specified as ODF from ZIP file
df2 <- read_odf(file = path, select = 4:7)

# generate a variable for joining both datasets:
df$id<-1:20
df2$id<-1:20

# merge both datasets by id column
merged_df<-merge(df, df2)

#merge both datasets by shared key columns between the two tables
merged_df2<-merge(df, df2)
```

---

read\_odf

*Read data specified as Open Data Format.*

---

## Description

Import data from the Open Data Format to an R data frame.

## Usage

```
read_odf(
  file,
  languages = "all",
  nrows = Inf,
  skip = 0,
  select = NULL,
  na.strings = getOption("datatable.na.strings", "NA")
)
```

## Arguments

file	the name of the file which the data are to be read from. By default all available language variants are imported (languages = "all").
languages	integer: the maximum number of rows to read in. Negative and other invalid values are ignored.
nrows	Maximum number of lines to read.
skip	Select the number of rows to be skipped (without the column names).

select	A vector of column names or numbers to keep, drop the rest. In all forms of select, order that the columns are specified determines the order of the columns in the result.
na.strings	A character vector of strings which are to be interpreted as NA values. By default, ",," for columns of all types, including type character is read as NA for consistency. ",", is unambiguous and read as an empty string. To read ,NA, as NA, set na.strings="NA". To read ,, as blank string "", set na.strings=NULL. When they occur in the file, the strings in na.strings should not appear quoted since that is how the string literal ,"NA", is distinguished from NA, for example, when na.strings="NA".

**Value**

R dataframe with attributes including dataset and variable information.

**Examples**

```
# get path to example data from the opendataformat package (data.zip)
path <- system.file("extdata", "data.odf.zip", package = "opendataformat")
path

# read example data specified as Open Data Format from ZIP file
df <- read_odf(file = path)
attributes(df)
attributes(df$bap87)

# read example data with language selection
df <- read_odf(file = path, languages = "de")
attributes(df$bap87)
```

---

setlanguage\_odf

*Change language of dataframe metadata*


---

**Description**

Changes the active language of a dataframe with metadata for the docu\_odf function.

**Usage**

```
setlanguage_odf(dataframe, language)
```

**Arguments**

dataframe	R data frame (df) enriched with metadata in the odf-format.
language	Select the language to which you want to switch the metadata.

**Value**

Dataframe

**Examples**

```
# get example data from the opendataformat package
df <- get(data("data_odf"))

# Switch dataset df to language "en"
df <- setlanguage_odf(df, language = "en")

# Display dataset information for dataset df in language "en"
docu_odf(df)
```

---

`write_odf`*Write R data frame to the Open Data Format.*

---

**Description**

Export data from an R data frame to a ZIP file that stores the data as Open Data Format.

**Usage**

```
write_odf(
  x,
  file,
  languages = "all",
  export_data = TRUE,
  verbose = TRUE,
  compression_level = 5,
  odf_version = "1.1.0"
)
```

**Arguments**

- |                        |   |
|------------------------|---|
| <code>x</code>         | R data frame (df) to be writtem.  |
| <code>file</code>      | Path to ZIP file or name of zip file to save the odf-dataset in the working directory.  |
| <code>languages</code> | Select the language in which the descriptions and labels of the data will be exported <ul style="list-style-type: none"><li>• By default all available language variants are exported (<code>languages = "all"</code>).</li><li>• You can also choose to export only the default language (<code>languages = "default"</code>),</li><li>• Or only the current language (<code>languages = "current"</code>),</li><li>• or you can select the language by language code, e.g. <code>languages = "en"</code>.</li></ul> |

export_data	Choose, if you want to export the file that holds the data (data.csv).Default is TRUE.
verbose	Display more messages.
compression_level	A number between 1 and 9. 9 compresses best, but it also takes the longest.
odf_version	The ODF version of the output file. Default is the actual/most recent version. <ul style="list-style-type: none"><li>• By default the data and metadata are exported (export_data = TRUE).</li><li>• To export only metadata and no data, select export_data = FALSE</li></ul>

### Value

ZIP file and unzipped directory containing the data as CSV file and the metadata as XML file (DDI Codebook 2.5.).

### Examples

```
# get example data from the opendataformat package
df <- get(data("data_odf"))

# write R data frame with attributes to the file my_data.zip specified
# as Open Data Format.
write_odf(x = df, paste0(tempdir(), "/my_data.zip"))

# write R data frame with attributes to the file my_data.zip
# with selected language.
write_odf(x = df, paste0(tempdir(), "/my_data.zip"), languages = "en")

# write R data frame with attributes to the file my_data.zip but only
# metadata, no data.
write_odf(x = df, file = paste0(tempdir(), "/my_data.zip"), export_data = FALSE)
```

# Index

## \* datasets

data\_odf, 4

as\_odf\_tbl, 3

data.table, 8

data\_odf, 4

docu\_odf, 5

getmetadata\_odf, 6

merge, 8

merge.odf\_tbl, 7

opendataformat

(opendataformat-package), 2

opendataformat-package, 2

read\_odf, 9

setlanguage\_odf, 10

write\_odf, 11