

# Package: io (via r-universe)

September 1, 2024

**Type** Package

**Title** A Unified Framework for Input-Output Operations in R

**Version** 0.3.4

**Author** David J. H. Shih

**Maintainer** David J. H. Shih <djh.shih@gmail.com>

**Description** One function to read files. One function to write files.  
One function to direct plots to screen or file. Automatic file  
format inference and directory structure creation.

**Imports** stringr, methods

**Depends** filenamer

**Suggests** XML (>= 3.98-1.1), rhdf5 (>= 2.26.1), yaml (>= 2.1.13),  
rjson, arrow, SummarizedExperiment, testthat

**URL** <https://bitbucket.org/djhshih/io>

**BugReports** <https://bitbucket.org/djhshih/io/issues>

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Repository** <https://djhshih.r-universe.dev>

**RemoteUrl** <https://bitbucket.org/djhshih/io>

**RemoteRef** HEAD

**RemoteSha** 17b054673c08bb65c94224b0f5322746626836ca

## Contents

io_supported . . . . .	2
list_files . . . . .	2
qcache . . . . .	3
qdraw . . . . .	4
qread . . . . .	5
qwrite . . . . .	6
qwrite.rds . . . . .	7

**Index** **8**


---

<code>io_supported</code>	<i>Determine input-output support for data or file type</i>
---------------------------	---

---

**Description**

This function returns whether a type is supported by `qread` or `qwrite`.

**Usage**

```
io_supported(type)
```

**Arguments**

<code>type</code>	data or file type
-------------------	-------------------

**Value**

a `data.frame` with logical entries; TRUE if type is supported, FALSE otherwise

**Examples**

```
io_supported("rds")
```

---

<code>list_files</code>	<i>List the files in a directory.</i>
-------------------------	---------------------------------------

---

**Description**

This function extends `list.files` by excluding the listing of directories.

**Usage**

```
list_files(path = ".", full.names = FALSE, ...)
```

**Arguments**

<code>path</code>	a character vector of path names
<code>full.names</code>	whether to return absolute paths
<code>...</code>	other arguments passed to <code>list.files</code>

**Value**

a character vector of only names of files

**Examples**

```
list.files(R.home())  
list_files(R.home())
```

---

qcache

*Cached data*

---

**Description**

If the data does not already exist as a file, evaluate an expression to generate the data and save it to file.

**Usage**

```
qcache(expr, file, cache = getOption("cache"), mkpath = TRUE, verbose = TRUE)
```

**Arguments**

expr	expression for generating data
file	filename
cache	whether to cache the result
mkpath	whether to create parent directories (if they do not already exists)
verbose	whether to output message

**Value**

a data object (object type depends on the expression)

**Examples**

```
## Not run:  
  
qcache(  
  {  
    data(cars)  
    with(cars, dist / speed)  
  },  
  file = "times.vtr"  
)  
  
## End(Not run)
```

---

qdraw

*Draw plot*


---

### Description

This function draws a plot to screen, a file, or both.

### Usage

```
qdraw(
  expr,
  file = NULL,
  device = getOption("plot.device"),
  width = NULL,
  height = NULL,
  aspect.ratio = NULL,
  units = NULL,
  res = NULL,
  mkpath = TRUE,
  symlink = TRUE,
  ...
)
```

### Arguments

expr	expression for plotting
file	filename
device	plot device
width	plot width [default: 5]
height	plot height [default: 5]
aspect.ratio	ratio of width to height
units	unit of plot dimension [default: "in"]
res	bitmap resolution, used only by bitmap formats [default: 300]
mkpath	whether to create parent directories (if they do not already exist)
symlink	whether to create a symlink to file with a simplified filename (ignored if file is not a filename object); an existing file will not be overwritten but an existing symlink will be
...	other arguments passed to the plot device function

**Details**

To send the plot to screen, set device to NA (default). Optionally, to print the plot on screen to a file, specify file.

If device is NULL, the plot will be sent directly to the the specified file using a printing device inferred from the file extension (no graphical window will open).

Set the global option `plot.device` to affect multiple plots. Graphical parameters including width, height, res, units are obtained from the global option `getOption("plot")`.

**Examples**

```
## Not run:
# Set device to jpeg (remember to update file extensions for printed plots)
options(plot.device=jpeg)
qdraw(plot(1:10), "plot.jpeg")

# Enable automatic plot format inference
options(plot.device=NULL)

# Plot directly to file (format is inferred from filename extension)
qdraw(plot(1:10), "plot.pdf")

# Plot to screen, then print to file (display will not be closed)
qdraw(plot(1:10), "plot.png", device=NA)

# If an error occurs, be sure to clear the current plot
dev.off()
# or clear all plots
graphics.off()

## End(Not run)
```

---

qread

*Data input*


---

**Description**

This function reads a file in a specified format.

**Usage**

```
qread(file, type = NULL, ...)
```

**Arguments**

file	file name (character or <code>filenamer::filename</code> ), a readable text-mode connection (for some types), or path to existing directory
type	data or file type
...	other arguments passed to the underlying function

**Details**

If type is NULL, the file type is inferred from the file extension. Use [io\\_supported](#) to check support for a file or data type.

**Value**

a data object (type depends on the underlying function)

**Examples**

```
## Not run:
data(cars)

# write data to an RDS file
qwrite(cars, "cars.rds")
# infer output type based on the class of the cars object
qwrite(cars, "cars.dfm", type=NA)

# read data back in
x1 <- qread("cars.rds")
# specify the type explicitly
x3 <- qread("cars.dfm", type="data.frame")

# read all files (with extension) in current directory
xs <- qread(".", pattern="cars")

## End(Not run)
```

---

qwrite

*Data output*


---

**Description**

This function writes an object to file in a specified format.

**Usage**

```
qwrite(x, file, type = NULL, mkpath = TRUE, symlink = TRUE, ...)
```

**Arguments**

x	data object to write
file	filename (character or <code>filenamer::filename</code> ), a readable text-mode connection (for some types), or path to existing directory
type	data or file type
mkpath	whether to create parent directories (if they do not already exists)

symlink	whether to create a symlink to file with a simplified file name (ignored if file is not a filename object); an existing file will not be overwritten but an existing symlink will be
...	other arguments passed to the underlying function

### Details

If type is NULL, the file type is inferred from the file extension. If type is NA or if the file extension is unavailable or unknown, type is inferred from class(x). Use [io\\_supported](#) to check support for a file or data type.

### Value

a data object (object type depends on the underlying function)

### Examples

```
## Not run:
data(cars)

# write data to a TSV file
qwrite(cars, "cars.tsv")
# infer output type based on the class of the cars object
qwrite(as.matrix(cars), "cars.mtx", type=NA)

## End(Not run)
```

---

qwrite.rds

*Write to RDS file*


---

### Description

Write to RDS file

### Usage

```
## S3 method for class 'rds'
qwrite(x, file, type, deref = TRUE, ...)
```

### Arguments

x	data object to write
file	filename (character or filename::filename), a readable text-mode connection (for some types), or path to existing directory
type	data or file type
deref	whether to dereference data links in DelayedMatrix
...	other arguments passed to saveRDS

# Index

`io_supported`, [2](#), [6](#), [7](#)

`list.files`, [2](#)

`list_files`, [2](#)

`qcache`, [3](#)

`qdraw`, [4](#)

`qread`, [2](#), [5](#)

`qwrite`, [2](#), [6](#)

`qwrite.rds`, [7](#)