

# Package ‘blockr.io’

January 7, 2026

**Title** Interactive File Import and Export Blocks

**Version** 0.1.0

**Description** Extends ‘blockr.core’ with interactive blocks for reading and writing data files. Supports CSV, Excel, Parquet, RDS, and other formats through a graphical interface without writing code directly. Includes file browser integration and configurable import/export options.

**URL** <https://bristolmyerssquibb.github.io/blockr.io/>

**BugReports** <https://github.com/BristolMyersSquibb/blockr.io/issues>

**License** GPL (>= 3)

**Depends** R (>= 4.1.0)

**Encoding** UTF-8

**Language** en-US

**RoxygenNote** 7.3.3

**VignetteBuilder** knitr

**Imports** blockr.core (>= 0.1.1), readxl, shiny, readr, shinyFiles, rio, arrow, bslib, rappdirs, shinyjs, writexl, zip

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Config/testthat/parallel** true

**NeedsCompilation** no

**Author** Christoph Sax [aut, cre] (ORCID: <<https://orcid.org/0000-0002-7192-7044>>), Nicolas Bennett [aut], David Granjon [aut], Mike Page [aut], Bristol Myers Squibb [fnd]

**Maintainer** Christoph Sax <[christoph@cynkra.com](mailto:christoph@cynkra.com)>

**Repository** CRAN

**Date/Publication** 2026-01-07 08:00:08 UTC

## Contents

new_read_block . . . . .	2
new_write_block . . . . .	4

## Index

7

---

new_read_block	<i>Unified file reading block</i>
----------------	-----------------------------------

---

### Description

A single block for reading files in various formats with smart UI that adapts based on detected file type. Supports "From Browser" (upload) and "From Server" (browse) modes with persistent storage for uploaded files.

### Usage

```
new_read_block(
  path = character(),
  source = "upload",
  combine = "auto",
  args = list(),
  ...
)
```

### Arguments

path	Character vector of file paths to pre-load. When provided, automatically switches to "path" mode regardless of the source parameter.
source	Either "upload" for file upload widget, "path" for file browser, or "url" for URL download. Default: "upload". Automatically set based on path parameter.
combine	Strategy for combining multiple files: "auto", "rbind", "cbind", "first"
args	Named list of format-specific reading parameters. Only specify values that differ from defaults. Available parameters: <ul style="list-style-type: none"> <li>• <b>For CSV files:</b> sep (default: ","), quote (default: "'"), encoding (default: "UTF-8"), skip (default: 0), n_max (default: Inf), col_names (default: TRUE)</li> <li>• <b>For Excel files:</b> sheet (default: NULL), range (default: NULL), skip (default: 0), n_max (default: Inf), col_names (default: TRUE)</li> </ul>
...	Forwarded to <a href="#">blockr.core::new_data_block()</a>

## Details

### File Handling Modes:

The block supports three modes:

#### From Browser mode (upload):

- User uploads files from their computer via the browser
- Files are copied to persistent storage directory (upload\_path)
- State stores permanent file paths
- Works across R sessions with state restoration

#### From Server mode (path):

- User picks files that already exist on the server
- No file copying, reads directly from original location
- State stores selected file paths
- When running locally, this is your computer's file system

#### URL mode:

- User provides a URL to a data file
- File is downloaded to temporary location each time
- Always fetches fresh data from URL
- State stores the URL (not file path)

### Smart Adaptive UI:

After file selection, the UI detects file type and shows relevant options:

- **CSV/TSV:** Delimiter, quote character, encoding options
- **Excel:** Sheet selection, cell range
- **Other formats:** Minimal or no options (handled automatically)

### Multi-file Support:

When multiple files are selected:

- **"auto":** Attempts rbind, falls back to first file if incompatible
- **"rbind":** Row-binds files (requires same columns)
- **"cbind":** Column-binds files (requires same row count)
- **"first":** Uses only the first file

## Value

A blockr data block that reads file(s) and returns a data.frame.

## Configuration

The following settings are retrieved from options and not stored in block state:

- **volumes:** File browser mount points. Set via `options(blockr.volumes = c(name = "path"))` or environment variable `BLOCKR_VOLUMES`. Default: `c(temp = tempdir())`
- **upload\_path:** Directory for persistent file storage. Set via `options(blockr.upload_path = "/path")` or environment variable `BLOCKR_UPLOAD_PATH`. Default: `rapdirs::user_data_dir("blockr")`

## Examples

```
# Create a read block for a CSV file
csv_file <- tempfile(fileext = ".csv")
write.csv(mtcars[1:5, ], csv_file, row.names = FALSE)
block <- new_read_block(path = csv_file)
block

# With custom CSV parameters
block <- new_read_block(
  path = csv_file,
  args = list(n_max = 3)
)

if (interactive()) {
  # Launch interactive app
  serve(new_read_block())
}
```

---

new\_write\_block      *Unified file writing block*

---

## Description

A variadic block for writing dataframes to files in various formats. Accepts multiple input dataframes and handles single files, multi-sheet Excel, or ZIP archives depending on format and number of inputs.

## Usage

```
new_write_block(
  directory = "",
  filename = "",
  format = "csv",
  mode = "download",
  auto_write = FALSE,
  args = list(),
  ...
)
```

## Arguments

directory	Character. Default directory for file output (browse mode only). Can be configured via options(blockr.write_dir = "/path") or environment variable BLOCKR_WRITE_DIR. Default: tempdir().
filename	Character. Optional fixed filename (without extension). <ul style="list-style-type: none"> <li>• <b>If provided:</b> Writes to same file path on every upstream change (auto-overwrite)</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>If empty</b> (default): Generates timestamped filename (e.g., data_20250127_143022.csv)</li> </ul>
format	Character. Output format: "csv", "excel", "parquet", or "feather". Default: "csv"
mode	Character. Either "download" for "To Browser" (triggers browser download), or "browse" for "To Server" (writes to server filesystem). Default: "download"
auto_write	Logical. When TRUE, automatically writes files when data changes (browse mode only). When FALSE (default), user must click "Submit" button to save. Has no effect in download mode.
args	Named list of format-specific writing parameters. Only specify values that differ from defaults. Available parameters: <ul style="list-style-type: none"> <li>• <b>For CSV files:</b> sep (default: ","), quote (default: TRUE), na (default: "")</li> <li>• <b>For Excel/Arrow:</b> Minimal options needed (handled by underlying packages)</li> </ul>
...	Forwarded to <a href="#">blockr.core::new_transform_block()</a>

## Details

### Variadic Inputs:

This block accepts multiple dataframe inputs (1 or more) similar to bind\_rows\_block. Inputs can be numbered ("1", "2", "3") or named ("sales\_data", "inventory"). Input names are used for sheet names (Excel) or filenames (multi-file ZIP).

### File Output Behavior:

#### Single input:

- Writes single file in specified format
- Filename: {filename}.{ext} or data\_{timestamp}.{ext}

#### Multiple inputs + Excel:

- Single Excel file with multiple sheets
- Sheet names derived from input names

#### Multiple inputs + CSV/Arrow:

- Single ZIP file containing individual files
- Each file named from input names

### Filename Behavior:

#### Fixed filename (filename = "output"):

- Reproducible path: always writes to {directory}/output.{ext}
- Overwrites file on every upstream data change
- Ideal for automated pipelines

#### Auto-timestamped (filename = ""):

- Unique files: {directory}/data\_YYYYMMDD\_HHMMSS.{ext}
- Preserves history, prevents accidental overwrites
- Safe default behavior

### Mode: To Browser vs To Server:

#### To Browser mode (download):

- Exports files to your computer
- Triggers a download to your browser's download folder
- Useful for exporting results

**To Server mode** (browse):

- Saves files directly on the server
- User selects directory with file browser
- Files persist on server
- When running locally, this is your computer's file system

**Value**

A blockr transform block that writes dataframes to files

**Examples**

```
# Create a write block for CSV output
block <- new_write_block(
  directory = tempdir(),
  filename = "output",
  format = "csv"
)
block

# Write block for Excel with auto-timestamp
block <- new_write_block(
  directory = tempdir(),
  filename = "",
  format = "excel"
)

if (interactive()) {
  # Launch interactive app
  serve(new_write_block())
}
```

# Index

blockr.core::new\_data\_block(), [2](#)  
blockr.core::new\_transform\_block(), [5](#)  
new\_read\_block, [2](#)  
new\_write\_block, [4](#)