

Package: fastsasconvert (via r-universe)

October 11, 2026

Title Convert SAS7BDAT Files to Parquet Without Loading Them

Version 0.1.0

Description Converts .sas7bdat files to Parquet, CSV or TSV, one file or a directory tree at a time, mirroring the input tree into an output root. Conversion happens entirely in Rust: the data never enters the R session, so files far larger than available memory can be converted. Companion to the 'fastsas' package, which reads SAS files into R.

License MIT + file LICENSE

URL <https://github.com/tkragholm/sas7bdat-parser-rs>

BugReports <https://github.com/tkragholm/sas7bdat-parser-rs/issues>

Encoding UTF-8

Copyright See inst/AUTHORS for the copyright holders of the Rust crates bundled in src/rust/vendor.tar.xz.

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

SystemRequirements Cargo (Rust's package manager), rustc >= 1.89

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/pak/sysreqs libclang-dev

Repository <https://tkragholm.r-universe.dev>

Date/Publication 2026-09-11 16:38:05 UTC

RemoteUrl <https://github.com/tkragholm/sas7bdat-parser-rs>

RemoteRef HEAD

RemoteSha 985de44e25887e8627fdbf7c60571288d74afcb7

RemoteSubdir crates/r-convert-plugin

Contents

convert_sas	2
Index	4

convert_sas	<i>Convert SAS7BDAT files to Parquet, CSV or TSV</i>
-------------	--

Description

Converts one file, several files, or every .sas7bdat under a directory, mirroring the input tree into output. The data never enters R: conversion happens entirely in Rust and only the per-file summary comes back, so trees far larger than the session’s memory convert fine.

Usage

```
convert_sas(  
  input,  
  output = NULL,  
  recursive = TRUE,  
  flatten = FALSE,  
  overwrite = FALSE,  
  sink = c("parquet", "csv", "tsv"),  
  compression = c("zstd", "lz4", "snappy", "none"),  
  io_backend = c("auto", "mmap", "buffered"),  
  threads = NULL,  
  tmp_dir = NULL  
)
```

Arguments

input	Paths to .sas7bdat files, directories, or glob patterns. Directories are searched for .sas7bdat files.
output	Output root. The tree below each input directory is recreated here. NULL writes each output beside its input.
recursive	Search directories recursively. Default TRUE.
flatten	Ignore the input tree and write every output directly into output. Default FALSE.
overwrite	Replace existing outputs instead of reporting them as errors.
sink	Output format: "parquet" (default), "csv" or "tsv".
compression	Parquet codec: "zstd" (default), "lz4", "snappy" or "none". Ignored for CSV and TSV.
io_backend	How inputs are read: "auto" (default) memory-maps local files and reads network shares sequentially, "mmap" always memory-maps, "buffered" always reads sequentially. "auto" can only detect a share on Windows, so pass "buffered" for a mounted share elsewhere.

threads	Decode threads per file. NULL (default) uses every logical core.
tmp_dir	Stage outputs here before moving them into place. NULL (default) stages beside the destination, where the move is a rename. Point this at a local disk to keep the write off a network link until there is a finished file to send — at the cost of copying it afterwards.

Value

A data frame with one row per converted input — zero rows if nothing matched, which is not an error: input, output, rows, columns, input_bytes, output_bytes, seconds, status and error. Carries an interrupted attribute, and discovered — the number of files found, which exceeds nrow() when a run was stopped early.

Failures are rows, not errors

A file that cannot be read does not stop the run. It comes back with status = "error" and the reason in error, and the remaining files still convert. Only argument mistakes throw. Retry a failed subset by filtering the returned frame and passing result\$input back in.

Interrupting a long run

Ctrl-C is checked between files, so a large tree can be stopped without losing the work already done — the frame returned covers the files that finished, and its interrupted attribute is TRUE. Interrupting cannot corrupt the output: each file is written to a temporary and moved into place only when complete, so a partially converted file never appears at its destination.

Resuming

With overwrite = FALSE (the default) an existing output is an error rather than a silent skip, which keeps a genuine name collision visible. To resume an interrupted run, filter out what already converted:

```
done <- file.exists(sub("\.sas7bdat$", ".parquet", files))
convert_sas(files[!done], "out")
```

Examples

```
## Not run:
# Mirror a tree of SAS files into a tree of Parquet files.
result <- convert_sas("//server/share/sas", "D:/parquet")
sum(result$status == "error")

# On a share where "auto" cannot tell the path is remote:
convert_sas("/mnt/share/sas", "out", io_backend = "buffered")

# Retry only what failed.
convert_sas(result$input[result$status == "error"], "D:/parquet")

## End(Not run)
```

Index

convert_sas, [2](#)