Package ‘mdbr’

May 23, 2026

Title Work with Microsoft Access Files

Version 0.3.0

Description Work with Microsoft Access '.mdb' and '.accdb' files using the open source 'MDB Tools' library <https://github.com/mdbtools/mdbtools/>. The library is compiled and bundled with the package, so no external installation is required. Provides high-level helpers for reading tables, exporting to CSV or JSON, inspecting table definitions, and running SQL queries. Also exposes a full read-only 'DBI' interface for use with standard database workflows.

License GPL-3 | LGPL-2

URL https://k5cents.github.io/mdbr/, https://github.com/k5cents/mdbr, https://github.com/mdbtools/mdbtools/

BugReports https://github.com/k5cents/mdbr/issues

Imports DBI, lifecycle, methods, tibble, utils, jsonlite

Suggests readr, testthat (>= 3.0.0)

Encoding UTF-8

NeedsCompilation yes

SystemRequirements GNU make

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Author Kiernan Nicholls [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-9229-7897>), Bruno Tremblay [ctb] (DBI interface and bundled mdbtools source)

Maintainer Kiernan Nicholls <k5cents@gmail.com>

Repository CRAN

Date/Publication 2026-05-23 11:00:07 UTC

Contents

`export_mdb` *Export an Access database table as a text file*

Description

Convert the data of a table into a delimited text string. Save the string as a character vector or write it to a text file. This direct conversion makes it easy to read tables into R or a spreadsheet.

Usage

```
export_mdb(
  file,
  table,
  output = TRUE,
  delim = ",",
  quote = "\"",
  quote_escape = "double",
  col_names = TRUE,
  eol = "\n",
  date_format = "%Y-%m-%d %H:%M:%S"
)
```

Arguments

file	Path to the Microsoft Access file.
table	Name of the table, list with mdb_tables().
output	Controls where output is sent. TRUE (the default) returns the output as a character vector. "" prints to the R console and returns invisibly. NULL or FALSE discards the output. A character string is treated as a file path to write to, returning the path invisibly.
delim	Delimiter used to separate values.
quote	Single character used to quote strings. Defaults to ".
quote_escape	The type of escaping to use for quoted values, one of "double", "backslash" or "none". You can also use FALSE, which is equivalent to "none". The default is "double", which is expected format for Excel.
col_names	If FALSE, column names will not be included at the top of the file. If TRUE, column names will be included.
eol	The end of line character to use. Most commonly either "\n" for Unix style newlines, or "\r\n" for Windows style newlines.
date_format	The format in which date columns are converted. MDB Tools uses the strftime(3) format, similar to readr::parse_date(). No need to specify whole string. Defaults to ISO8601.

Value

Character string, invisible if path to file.

Examples

```
## Not run:
export_mdb(mdb_example(), "Airlines", output = TRUE)

## End(Not run)
```

---

mdb *Create an mdbr DBI Driver*

Description

mdb() is the canonical DBI-style constructor for connecting to Microsoft Access '.mdb' and '.accdb' files.

Usage

```
mdb()
```

Value

A DBI driver for '.mdb' and '.accdb' files.

Examples

```
db <- mdb_example()
conn <- DBI::dbConnect(mdb(), dbname = db)
DBI::dbListTables(conn)
DBI::dbDisconnect(conn)
```

---

mdb_array *Export Table As List Columns (mdb-array mimic)*

Description

`mdb-array(1)` emits C source; this wrapper returns a named R list of columns while keeping equivalent database/table inputs.

Usage

```
mdb_array(path, table, columns = NULL, n = -1L)
```

Arguments

`path`	Path to `.mdb`/`.accdb` file.
`table`	Table name.
`columns`	Optional character vector of columns.
`n`	Optional row limit (`LIMIT n`).

Value

Named list, one entry per selected column.

Examples

```
db <- mdbr:::.mdb_example_nwind_path()
if (nzchar(db)) {
  arr <- mdb_array(db, "Products", columns = c("ProductID", "ProductName"), n = 2)
  str(arr)
}
```

mdb_count *Count Rows In Table*

Description

mdb-count is a utility program distributed with MDB Tools. It outputs the number of rows in a table.

Usage

```
mdb_count(path, table = NULL, where = NULL, version = FALSE)
```

Arguments

path	Path to .mdb/.accdb file.
table	Table name.
where	Optional SQL predicate appended to WHERE. This is an R-side extension and is not part of the CLI.
version	Logical; when TRUE, return mdbtools version (--version).

Value

Integer row count or version string.

Examples

```
db <- mdbr:::.mdb_example_nwind_path()
if (nzchar(db)) {
  mdb_count(db, "Orders")
}
```

mdb_example *Get path to mdbr example*

Description

mdbr comes bundled with a sample file from the nycflights13 package in its inst/extdata directory. This function make it easy to access.

Usage

```
mdb_example(path = "nycflights13.mdb")
```

Arguments

path	path to the Microsoft Access file.

Value

A character string with the full path to the bundled example file.

Examples

```
mdb_example()
```

`mdb_export` *Export Data In An MDB Table To CSV Or INSERT SQL*

Description

`mdb-export` is a utility program distributed with MDB Tools. It produces CSV output for the given table. Such output is suitable for importation into databases or spreadsheets.

Usage

```
mdb_export(
  path,
  table,
  no_header = FALSE,
  delimiter = ",",
  row_delimiter = "\n",
  no_quote = FALSE,
  quote = "\"",
  escape = NULL,
  escape_invisible = FALSE,
  date_format = "%Y-%m-%d",
  datetime_format = "%Y-%m-%d %H:%M:%S",
  null = "",
  bin = c("strip", "raw", "octal", "hex"),
  boolean_words = FALSE,
  insert = NULL,
  namespace = NULL,
  batch_size = 1L,
  n = -1L
)
```

Arguments

`path`	Path to `.mdb`/`.accdb` file.
`table`	Table name.
`no_header`	Logical, equivalent to `-H`/`--no-header`.
`delimiter`	Equivalent to `-d`/`--delimiter`.
`row_delimiter`	Equivalent to `-R`/`--row-delimiter`.

no_quote	Equivalent to -Q/--no-quote.
quote	Equivalent to -q/--quote.
escape	Equivalent to -X/--escape.
escape_invisible	Equivalent to -e/--escape-invisible.
date_format	Equivalent to -D/--date-format.
datetime_format	Equivalent to -T/--datetime-format.
null	Equivalent to -0/--null.
bin	Binary mode (strip, raw, octal, hex) for parity.
boolean_words	Equivalent to -B/--boolean-words.
insert	Backend for -I/--insert mode.
namespace	Equivalent to -N/--namespace.
batch_size	Equivalent to -S/--batch-size.
n	Optional row limit (LIMIT n).

Details

Used with insert, it outputs backend-specific SQL INSERT statements. Most formatting options also apply in insert mode.

Value

Character scalar containing CSV or SQL INSERT text.

Examples

```
db <- mdbr:::.mdb_example_nwind_path()
if (nzchar(db)) {
  cat(mdb_export(db, "Products", n = 2))
}
```

---

mdb_header *MDB Header Summary (mdb-header mimic)*

Description

mdb-header(1) writes C files; this wrapper returns a structured summary.

Usage

```
mdb_header(path)
```

Arguments

path	Path to .mdb/.accdb file.

Value

Named list with version, table names and query names.

Examples

```
db <- mdbr:::.mdb_example_nwind_path()
if (nzchar(db)) {
  mdb_header(db)
}
```

---

mdb_hexdump *Hexdump MDB File (mdb-hexdump mimic)*

Description

Hexdump MDB File (mdb-hexdump mimic)

Usage

```
mdb_hexdump(path, pagenumber = NULL, page_size = 4096L, n = 256L)
```

Arguments

path	Path to file.
pagenumber	Optional page index (0-based) like mdb-hexdump file [pagenumber].
page_size	Page size in bytes (default 4096 for modern Jet/ACE).
n	Number of bytes to emit.

Value

Single hexadecimal string.

Examples

```
db <- mdbr:::.mdb_example_nwind_path()
if (nzchar(db)) {
  mdb_hexdump(db, n = 16)
}
```

mdb_json *Export Data In An MDB Table To JSON*

Description

mdb-json is a utility program distributed with MDB Tools. It produces JSON output for the given table. Such output is suitable for parsing in a variety of languages.

Usage

```
mdb_json(
  path,
  table,
  date_format = "%Y-%m-%d",
  time_format = "%Y-%m-%d %H:%M:%S",
  no_unprintable = FALSE,
  n = -1L
)
```

Arguments

path	Path to .mdb/.accdb file.
table	Table name.
date_format	Equivalent to -D/--date-format.
time_format	Equivalent to -T/--time-format.
no_unprintable	Equivalent to -U/--no-unprintable.
n	Optional row limit.

Value

JSON string.

Examples

```
db <- mdbr:::.mdb_example_nwind_path()
if (nzchar(db) && requireNamespace("jsonlite", quietly = TRUE)) {
  mdb_json(db, "Products", n = 2)
}
```

mdb_prop *Get Properties List From MDB Database*

Description

mdb-prop retrieves properties for one or more objects in an MDB database. name is the name of the table, query, or other object. propcol is the name of the MSysObjects column containing properties and defaults to LvProp.

Usage

```
mdb_prop(
  path,
  name = NULL,
  propcol = "LvProp",
  version = FALSE,
  as_list = TRUE
)
```

Arguments

path	Path to .mdb/.accdb file.
name	Object name (table, query, etc.); may be a character vector.
propcol	Property column name. Defaults to LvProp.
version	Logical; when TRUE, return mdbtools version.
as_list	Logical; defaults to TRUE. When TRUE, wraps the result in an mdblist object so the dedicated print method is used.

Value

A named list with one entry per element of name. Each entry is itself a named list of named character vectors, one per property block (e.g. "(none)" for table-level properties, or a column/field name). Access individual values with p[["Orders"]][["(none)"]]["Description"].

Examples

```
db <- mdbr:::.mdb_example_nwind_path()
if (nzchar(db)) {
  p <- mdb_prop(db, "Orders")
  p[["Orders"]][["(none)"]]["Description"]
  p2 <- mdb_prop(db, c("Orders", "Orders Qry"))
}
```

mdb_queries *List Or Retrieve Queries In An MDB Database*

Description

mdb-queries is a utility program distributed with MDB Tools. Without query, it lists the names of all saved queries in the database. With query, it returns the SQL text of the named query or queries.

Usage

```
mdb_queries(
  path,
  query = NULL,
  list = TRUE,
  newline = FALSE,
  delimiter = " ",
  as_text = FALSE,
  as_list = TRUE
)
```

Arguments

path	Path to .mdb/.accdb file.
query	Character vector of query names. When NULL (the default), the function lists available query names instead of fetching SQL.
list	Logical; when TRUE (the default) and query is NULL, return the list of query names. Set to FALSE to suppress listing and return character(0).
newline	Logical; when TRUE and as_text = TRUE, collapse names with "\n" instead of delimiter.
delimiter	Character scalar used to collapse query names when as_text = TRUE. Default " ".
as_text	Logical; when TRUE, collapses the query name list into a single string using delimiter (or "\n" if newline = TRUE).
as_list	Logical; defaults to TRUE. When TRUE and query is supplied, wraps the result in a named mdblist object.

Value

When query is NULL: a character vector of query names (or a collapsed string when as_text = TRUE). When query is supplied and as_list = TRUE: a named mdblist of SQL text strings, one per query. With a single query and as_list = FALSE: a character scalar.

Examples

```
db <- mdbr:::.mdb_example_nwind_path()
if (nzchar(db)) {
  mdb_queries(db)
  mdb_queries(db, "Orders Qry")
}
```

`mdb_schema` *Generate Schema DDL or Column Types*

Description

`mdb-schema` is a utility program distributed with MDB Tools. It produces DDL (data definition language) output for the given database, which can be passed to another database to recreate the Access table structure. With `mode = "legacy"` (the default), it returns a readr col spec for the table instead.

Usage

```
mdb_schema(
  path,
  table = NULL,
  mode = c("legacy", "ddl"),
  condense = FALSE,
  namespace = NULL,
  backend = c("access", "sybase", "oracle", "postgres", "mysql", "sqlite"),
  drop_table = FALSE,
  not_null = TRUE,
  default_values = FALSE,
  not_empty = FALSE,
  comments = TRUE,
  indexes = TRUE,
  relations = TRUE,
  as_list = TRUE
)
```

Arguments

`path`	Path to `.mdb`/`.accdb` file.
`table`	Table name(s). For `mode = "ddl"`, defaults to all user tables. For `mode = "legacy"`, exactly one table name must be given.
`mode`	`"legacy"` (default) returns a `readr::cols()` specification, matching the k5cents/mdbr canonical API. Requires the **readr** package. `"ddl"` returns DDL text.
`condense`	Logical; only used when `mode = "legacy"`. When `TRUE`, passes the col spec through `readr::cols_condense()`. Default `FALSE`.

namespace	Prefix identifiers with namespace, equivalent to -N/--namespace. Only used when mode = "ddl".
backend	Target DDL dialect. Supported values are access, sybase, oracle, postgres, mysql, and sqlite. Only used when mode = "ddl".
drop_table	Issue DROP TABLE statements.
not_null	Include NOT NULL constraints.
default_values	Include DEFAULT values.
not_empty	Include CHECK <> '' constraints.
comments	Include COMMENT ON statements.
indexes	Export indexes.
relations	Request foreign key constraints. Current library-mode implementation emits a placeholder comment; full FK export is not yet implemented.
as_list	Logical; defaults to TRUE. When TRUE, returns a named mdblist of DDL text entries keyed by table name. Only used when mode = "ddl".

Value

When mode = "legacy", a readr::cols() specification (optionally condensed via readr::cols_condense()). Requires the **readr** package. When mode = "ddl" and as_list = TRUE, a named mdblist of table-level DDL text.

Examples

```
db <- mdbr:::.mdb_example_nwind_path()
if (nzchar(db)) {
  mdb_schema(db, table = "Products")
  mdb_schema(db, table = "Products", mode = "ddl")
}
```

---

mdb_sql *SQL Interface To MDB Tools*

Description

mdb-sql is a utility program distributed with MDB Tools. It allows querying of an MDB database using a limited SQL subset language. The supported SQL is intentionally small: single-table queries, no aggregates, and limited WHERE support.

Usage

```
mdb_sql(
  path,
  statement = NULL,
  no_header = FALSE,
  no_footer = FALSE,
```

```
  no_pretty_print = FALSE,
  delimiter = "\t",
  input = NULL,
  output = NULL,
  as_text = FALSE
)
```

Arguments

path	Path to .mdb/.accdb file.
statement	SQL statement text.
no_header	Logical, equivalent to -H/--no-header (for text mode).
no_footer	Logical, equivalent to -F/--no-footer (for text mode).
no_pretty_print	Logical, equivalent to -p/--no-pretty-print.
delimiter	Delimiter equivalent to -d/--delimiter in plain text mode.
input	Input file equivalent to -i/--input.
output	Output file equivalent to -o/--output.
as_text	Logical; when TRUE, returns CLI-like text output.

Details

In addition to single statements, this wrapper accepts input files similar to mdb-sql -i file, strips go batch terminators, and executes the script one statement at a time.

Value

data.frame by default, or character scalar in text mode.

Examples

```
db <- mdbr:::.mdb_example_nwind_path()
if (nzchar(db)) {
  mdb_sql(db, "SELECT [ProductID], [ProductName] FROM [Products] LIMIT 3;")
}
```

---

mdb_tables *List tables in a Microsoft Access database*

Description

mdb-tables is a utility program distributed with MDB Tools. It outputs the names of all user tables (or other object types) in an MDB database file.

Usage

```
mdb_tables(
  file,
  system = FALSE,
  type = c("table", "query", "systable", "any", "all", "form", "macro", "report",
    "linkedtable", "module", "relationship", "dbprop"),
  show_type = FALSE
)
```

Arguments

file	Path to the Microsoft Access file.
system	Logical; include system (MSys*) tables. Equivalent to -S.
type	Object type to list: "table" (default), "query", "systable", "any", "all", "form", "macro", "report", "linkedtable", "module", "relationship", or "dbprop". Equivalent to -t.
show_type	Logical; prefix each entry with its type. Equivalent to -T.

Value

A character vector of object names.

Examples

```
db <- mdb_example()
mdb_tables(db)
mdb_tables(db, type = "query")
```

---

mdb_ver *Return MDB File Format Or MDB Tools Version*

Description

mdb-ver will return a single line of output corresponding to the program that produced the file: JET3 (for files produced by Access 97), JET4 (Access 2000, XP and 2003), ACE12 (Access 2007), ACE14 (Access 2010), ACE15 (Access 2013), or ACE16 (Access 2016).

Usage

```
mdb_ver(path = NULL, version = FALSE)
```

Arguments

path	Optional database path. When omitted, the wrapper returns the mdbtools package version for backward compatibility.
version	Logical, equivalent to -M/--version.

Value

Single character string with file format or mdbtools version.

Examples

```
mdb_ver()
db <- mdbr:::.mdb_example_nwind_path()
if (nzchar(db)) {
  mdb_ver(db)
}
```

`print.mdblist` *Print Method For* `mdblist`

Description

Pretty printer for multi-object text outputs returned by selected `mdb_*` helpers when `as_list = TRUE` (default).

Usage

```
## S3 method for class 'mdblist'
print(x, ...)
```

Arguments

`x`	A `mdblist` object.
`...`	Unused.

Value

The input object, invisibly.

`read_mdb` *Read a table as data frame*

Description

Reads a table directly from a Microsoft Access database using the bundled mdbtools C library. Column types are inferred from the MDB schema: integer, double, logical, POSIXct for DateTime columns, and character otherwise.

Usage

```
read_mdb(file, table, col_names = TRUE, col_types = NULL, ...)
```

Arguments

`file`	Path to the Microsoft Access file.
`table`	Name of the table, list with `mdb_tables()`.
`col_names`	Logical; when `FALSE` columns are named `V1`, `V2`, etc.
`col_types`	**[Deprecated]** Ignored. Type coercion is now handled automatically by the native reader.
`...`	**[Deprecated]** Ignored. Extra arguments were previously forwarded to `readr::read_delim()`, which is no longer used.

Value

A tibble.

Examples

```
## Not run:
read_mdb(mdb_example(), "Airlines")

## End(Not run)
```

Index