Package ‘Rparadox’

September 29, 2025

Title Read Paradox Database Files into R

Version 0.1.5

Description Provides a simple and efficient way to read data from Paradox database files (.db) directly into R as modern 'tibble' data frames. It uses the underlying 'pxlib' C library, to handle the low-level file format details and provides a clean, user-friendly R interface.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.3

Imports blob, hms, tibble, stringi

Suggests rmarkdown, devtools, knitr, testthat (>= 3.0.0), usethis

Config/testthat/edition 3

URL `https://github.com/celebithil/Rparadox`, `https://github.com/steinm/pxlib`

BugReports `https://github.com/celebithil/Rparadox/issues`

VignetteBuilder knitr, rmarkdown

NeedsCompilation yes

Author Daniil Popov [aut, cre]

Maintainer Daniil Popov <`popov.daniil@gmail.com`>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2025-09-29 09:10:02 UTC

Contents

`pxlib_close_file` *Close a Paradox database file*

Description

This function explicitly closes a Paradox database file associated with a `pxdoc_t` external pointer and releases its resources.

Usage

```
pxlib_close_file(pxdoc)
```

Arguments

`pxdoc`	An external pointer of class 'pxdoc_t' obtained from `pxlib_open_file()`.

Value

Invisible `NULL`.

`pxlib_get_data` *Read Data from a Paradox File*

Description

Retrieves all records from an open Paradox database file and returns them as a tibble, ready for use in R.

Usage

```
pxlib_get_data(pxdoc)
```

Arguments

`pxdoc`	An object of class `pxdoc_t`, representing an open Paradox file connection. This object is obtained from `pxlib_open_file()`.

Details

This function provides a high-level interface for reading Paradox data. The heavy lifting is done by a C-level function (`R_pxlib_get_data`) which efficiently reads the raw data into memory. This R function then acts as a wrapper to perform crucial post-processing steps:

- It identifies columns containing binary (BLOB) data and correctly converts them into `blob` objects.
- It ensures that date/time columns are properly classed for seamless integration with other R functions.

The result is a clean, modern `tibble` that is fully compatible with the tidyverse ecosystem.

Value

A `tibble` containing the data from the Paradox file. Each row represents a record and each column represents a field. If the file contains no records, an empty tibble is returned.

Examples

```
# Define the path to the demo database included with the package
db_path <- system.file("extdata", "biolife.db", package = "Rparadox")

# Open the file handle
pxdoc <- pxlib_open_file(db_path)

if (!is.null(pxdoc)) {
  # Read all data into a tibble
  biolife_data <- pxlib_get_data(pxdoc)

  # Always close the file handle when finished
  pxlib_close_file(pxdoc)

  # Work with the data
  print(biolife_data)
}
```

`pxlib_metadata` *Get Metadata from a Paradox Database File*

Description

Retrieves metadata from an open Paradox file handle without reading the entire dataset.

Usage

```
pxlib_metadata(pxdoc)
```

Arguments

`pxdoc`	An object of class `pxdoc_t`, representing an open Paradox file connection, obtained from `pxlib_open_file()`.

Value

A list containing:

`num_records`	The total number of records in the database.
`num_fields`	The total number of fields (columns).
`encoding`	The character encoding specified in the file header (e.g., "CP1251").
`fields`	A data frame with details for each field, with names recoded to UTF-8.

Examples

```
db_path <- system.file("extdata", "country.db", package = "Rparadox")
pxdoc <- pxlib_open_file(db_path)
if (!is.null(pxdoc)) {
  metadata <- pxlib_metadata(pxdoc)
  print(metadata)
  pxlib_close_file(pxdoc)
}
```

`pxlib_open_file` *Open a Paradox Database File*

Description

Opens a Paradox database (.db) file and prepares it for reading. This function serves as the entry point for interacting with a Paradox database.

Usage

```
pxlib_open_file(path, encoding = NULL)
```

Arguments

`path`	A character string specifying the path to the Paradox (.db) file.
`encoding`	An optional character string specifying the input encoding of the data (e.g., "cp866", "cp1252"). If `NULL` (the default), the encoding is determined from the file header.

Details

This function initializes a connection to a Paradox file via the underlying C library. It automatically performs two key setup tasks:

1. **Encoding Override:** It allows the user to specify the character encoding of the source file via the `encoding` parameter. This is crucial for legacy files where the encoding stored in the header may be incorrect. If `encoding` is `NULL`, the function will attempt to use the codepage from the file header.
2. **BLOB File Attachment:** It automatically searches for an associated BLOB file (with a `.mb` extension, case-insensitively) in the same directory and, if found, attaches it to the database handle.

Value

An external pointer of class 'pxdoc_t' if the file is successfully opened, or `NULL` if an error occurs (e.g., file not found).

Examples

```
# Example 1: Open a bundled demo file (biolife.db)
db_path <- system.file("extdata", "biolife.db", package = "Rparadox")
pxdoc <- pxlib_open_file(db_path)
if (!is.null(pxdoc)) {
  # normally you'd read data here
  pxlib_close_file(pxdoc)
}

# Example 2: Open a file with overridden encoding (of_cp866.db)
db_path2 <- system.file("extdata", "of_cp866.db", package = "Rparadox")
pxdoc2 <- pxlib_open_file(db_path2, encoding = "cp866")
if (!is.null(pxdoc2)) {
  # read some data ...
  pxlib_close_file(pxdoc2)
}
```

`read_paradox` *Read a Paradox Database File into a Tibble*

Description

A high-level, user-friendly wrapper function that reads an entire Paradox database file (.db) and returns its contents as a tibble.

Usage

```
read_paradox(path, encoding = NULL)
```

Arguments

`path`	A character string specifying the path to the Paradox (.db) file.
`encoding`	An optional character string specifying the input encoding of the data (e.g., "cp866", "cp1252"). If `NULL` (the default), the encoding is determined from the file header.

Details

This function simplifies the process of reading Paradox files by handling the complete workflow in a single call:

1. It validates the input path and encoding.
2. It safely opens a handle to the file using `pxlib_open_file()`.
3. It ensures the file handle is always closed using `on.exit()`, even if errors occur during data reading.
4. It reads the data using `pxlib_get_data()`.
5. It returns a clean `tibble`.

If the specified file does not exist, the function will issue a warning and return an empty tibble.

Value

A `tibble` containing the data from the Paradox file.

Examples

```
# Read the demo database in one step
db_path <- system.file("extdata", "biolife.db", package = "Rparadox")
if (file.exists(db_path)) {
  biolife_data <- read_paradox(db_path)
  print(biolife_data)
}
```

Index