Package 'adbi'

September 25, 2026

Title 'DBI' Compliant Database Access Using 'ADBC'

Version 0.1.3

Description In order to make Arrow Database Connectivity ('ADBC' <https://arrow.apache.org/adbc/>) accessible from R, an interface compliant with the 'DBI' package is provided, using driver back-ends that are implemented in the 'adbcdrivermanager' framework. This enables interacting with database systems using the Arrow data format, thereby offering an efficient alternative to 'ODBC' for analytical applications.

License LGPL (>= 2.1)

URL https://adbi.r-dbi.org, https://github.com/r-dbi/adbi, https://arrow.apache.org/adbc/

BugReports https://github.com/r-dbi/adbi/issues

Depends R (>= 4.1.0)

Imports DBI (>= 1.2.0), methods, adbcdrivermanager (>= 0.8.0), nanoarrow (>= 0.3.0)

Suggests testthat, covr, DBItest (>= 1.8.3), adbcsqlite (>= 0.8.0), withr, bit64, utils, arrow

Config/Needs/website r-dbi/dbitemplate

Encoding UTF-8

Collate 'adbi-package.R' 'AdbiDriver.R' 'AdbiConnection.R' 'AdbiResult.R' 'AdbiResultArrow.R' 'dbAppendTable_AdbiConnection_character_data.frame.R' 'dbBegin_AdbiConnection.R' 'dbBindArrow_AdbiResult.R' 'dbBindArrow_AdbiResultArrow.R' 'dbBind_AdbiResult.R' 'dbBind_AdbiResultArrow.R' 'dbClearResult_AdbiResult.R' 'dbClearResult_AdbiResultArrow.R' 'dbColumnInfo_AdbiResult.R' 'dbColumnInfo_AdbiResultArrow.R' 'dbCommit_AdbiConnection.R' 'dbConnect_AdbiDriver.R' 'dbDataType_AdbiConnection.R' 'dbDataType_AdbiDriver.R' 'dbDisconnect_AdbiConnection.R' 'dbExistsTable_AdbiConnection_Id.R' 'dbExistsTable_AdbiConnection_SQL.R' 'dbExistsTable_AdbiConnection_character.R' 'dbFetchArrowChunk_AdbiResultArrow.R'

'dbFetchArrow_AdbiResultArrow.R' 'dbFetch_AdbiResult.R'
'dbGetInfo_AdbiConnection.R' 'dbGetInfo_AdbiDriver.R'
'dbGetRowCount_AdbiResult.R' 'dbGetRowCount_AdbiResultArrow.R'
'dbGetRowsAffected_AdbiResult.R'
'dbGetRowsAffected_AdbiResultArrow.R'
'dbGetStatement_AdbiResult.R'
'dbGetStatement_AdbiResultArrow.R'
'dbHasCompleted_AdbiResult.R'
'dbHasCompleted_AdbiResultArrow.R' 'dbIsValid_AdbiConnection.R'
'dbIsValid_AdbiDriver.R' 'dbIsValid_AdbiResult.R'
'dbIsValid_AdbiResultArrow.R'
'dbListFields_AdbiConnection_Id.R'
'dbListFields_AdbiConnection_SQL.R'
'dbListFields_AdbiConnection_character.R'
'dbListTables_AdbiConnection.R'
'dbQuoteIdentifier_AdbiConnection_character.R'
'dbQuoteLiteral_AdbiConnection_character.R'
'dbQuoteString_AdbiConnection_character.R'
'dbRemoveTable_AdbiConnection_character.R'
'dbRemoveTable_AdbiConnection_Id.R'
'dbRollback_AdbiConnection.R'
'dbSendQueryArrow_AdbiConnection.R'
'dbSendQuery_AdbiConnection_character.R'
'dbSendStatement_AdbiConnection_character.R'
'dbUnquoteIdentifier_AdbiConnection.R'
'dbWriteTable_AdbiConnection_Id_data.frame.R'
'dbWriteTable_AdbiConnection_SQL_data.frame.R'
'dbWriteTable_AdbiConnection_character_data.frame.R' 'export.R'
'show_AdbiConnection.R' 'show_AdbiDriver.R' 'show_AdbiResult.R'
'show_AdbiResultArrow.R' 'utils.R'

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Nicolas Bennett [aut, cre, cph],
Voltron Data [fnd]

Maintainer Nicolas Bennett <nicolas@cynkra.com>

Repository CRAN

Date/Publication 2026-09-25 13:00:09 UTC

Contents

adbi 3
dbFetch_AdbiResult 4
dbSendQueryArrow_AdbiConnection 5

Index **7**

adbi *Adbi driver*

Description

In order to open a database connection, DBI::dbConnect() dispatches on a driver object, which can be instantiated by calling adbi().

Usage

```
adbi(driver = NA_character_)

## S4 method for signature 'AdbiDriver'
dbConnect(drv, ..., bigint = NULL)

## S4 method for signature 'AdbiConnection'
dbDisconnect(conn, force = getOption("adbi.force_close_results", FALSE), ...)
```

Arguments

driver	A driver specification that can be evaluated (with no arguments) to give an adbcdrivermanager::adbc_driver(). See Details for more information.
drv	An object that inherits from DBI::DBIDriver, or an existing DBI::DBIConnection object (in order to clone an existing connection).
...	Extra arguments passed to DBI::dbConnect() are forwarded to adbcdrivermanager::adbc_database_
bigint	The R type that 64-bit integer types should be mapped to, default is bit64::integer64, if bit64 is installed and character otherwise
conn	A DBI::DBIConnection object, as returned by DBI::dbConnect().
force	Close open results when disconnecting

Details

To specify the type of adbc driver, adbi accepts as driver argument

- an object inheriting from adbc_driver,
- a function that can be evaluated with no arguments and returns an object inheriting from adbc_driver,
- a string of the form pkg::fun (where pkg:: is optional and defaults to fun), which can be used to look up such a function.

As default, an adbcdrivermanager::adbc_driver_monkey() object is created.

Value

A connection object (S4 class AdbiConnection, inheriting from DBI::DBIConnection) is returned by DBI::dbConnect(), while DBI::dbDisconnect() returns TRUE invisibly.

Examples

```
adbi()
if (requireNamespace("adbcsqlite")) {
  adbi("adbcsqlite")
}
library(DBI)
con <- dbConnect(adbi())
dbIsValid(con)
dbDisconnect(con)
dbIsValid(con)
```

`dbFetch_AdbiResult` *Fetch result sets*

Description

When fetching results using `DBI::dbFetch()`, the argument `n` can be specified to control chunk size per fetching operation. The default value of `-1` corresponds to retrieving the entire result set at once, while a positive integer will try returning as many rows (as long as `n` does not exceed the available number of rows), in line with standard DBI expectations. As data transfer is mediated by Arrow data structures, which are retrieved as array chunks, the underlying chunk size can be used by passing an `n` value `NA`.

Usage

```
## S4 method for signature 'AdbiResult'
dbFetch(res, n = -1, ...)
```

Arguments

`res`	An object inheriting from DBI::DBIResult, created by `DBI::dbSendQuery()`.
`n`	maximum number of records to retrieve per fetch. Use `n = -1` or `n = Inf` to retrieve all pending records. Some implementations may recognize other special values.
`...`	Other arguments passed on to methods.

Value

A `data.frame` with the requested number of rows (or zero rows if `DBI::dbFetch()` is called on a result set with no more remaining rows).

Examples

```
if (requireNamespace("adbcsqlite")) {
  library(DBI)
  con <- dbConnect(adbi::adbi("adbcsqlite"), uri = ":memory:")
  dbWriteTable(con, "swiss", swiss)
  res <- dbSendQuery(con, "SELECT * from swiss WHERE Agriculture < 30")
```

```
  dbFetch(res)
  dbClearResult(res)
  dbDisconnect(con)
}
```

---

dbSendQueryArrow_AdbiConnection

Create result sets

Description

Creating result sets using DBI::dbSendQuery() (and by extension using DBI::dbGetQuery()) mostly follows DBI specification. One way where adbi deviates from DBI mechanisms is how the bigint setting is not only per connection, but the per-connection setting can be overridden on a result set basis. As default, the connection setting is applied, but passing one of the accepted values as bigint when creating a result set will subsequently use that setting for all fetches using this result set.

Usage

```
## S4 method for signature 'AdbiConnection'
dbSendQueryArrow(
  conn,
  statement,
  ...,
  params = NULL,
  immediate = NULL,
  bigint = NULL
)

## S4 method for signature 'AdbiConnection,character'
dbSendQuery(
  conn,
  statement,
  ...,
  params = NULL,
  immediate = NULL,
  bigint = NULL
)

## S4 method for signature 'AdbiConnection,character'
dbSendStatement(
  conn,
  statement,
  ...,
  params = NULL,
  immediate = NULL,
  bigint = NULL
)
```

Arguments

conn	A DBI::DBIConnection object, as returned by DBI::dbConnect().
statement	a character string containing SQL.
...	Other parameters passed on to methods.
params	Optional query parameters (forwarded to DBI::dbBind())
immediate	Passing a value TRUE is intended for statements containing no placeholders and FALSE otherwise. The default value NULL will inspect the statement for presence of placeholders (will PREPARE the statement)
bigint	The R type that 64-bit integer types should be mapped to, default is chosen according to the connection setting

Details

Multiple open result sets per connection are supported and support can be disabled by setting options(adbi.allow_multiple_results = FALSE). If not enabled, creating a new result will finalize potential other results and throw a warning.

Value

An S4 class AdbiResult (inheriting from DBI::DBIResult).

See Also

adbi-driver

Examples

```
if (requireNamespace("adbcsqlite")) {
  library(DBI)
  con <- dbConnect(adbi::adbi("adbcsqlite"), uri = ":memory:")
  dbWriteTable(con, "swiss", swiss)
  str(
    dbGetQuery(con, "SELECT Examination from swiss WHERE Agriculture < 30")
  )
  str(
    dbGetQuery(con, "SELECT Examination from swiss WHERE Agriculture < 30",
      bigint = "integer")
  )
  dbDisconnect(con)
}
```

Index

adbcdrivermanager::adbc_database_init(), *3*
adbcdrivermanager::adbc_driver(), *3*
adbcdrivermanager::adbc_driver_monkey(), *3*
adbi, 3

bit64::integer64, *3*

dbConnect,AdbiDriver-method *(*adbi*)*, 3
dbConnect_AdbiDriver *(*adbi*)*, 3
dbDisconnect,AdbiConnection-method *(*adbi*)*, 3
dbDisconnect_AdbiConnection *(*adbi*)*, 3
dbFetch,AdbiResult-method *(*dbFetch_AdbiResult*)*, 4
dbFetch_AdbiResult, 4
DBI::dbBind(), *6*
DBI::dbConnect(), *3*, *6*
DBI::dbDisconnect(), *3*
DBI::dbFetch(), *4*
DBI::dbGetQuery(), *5*
DBI::DBIConnection, *3*, *6*
DBI::DBIDriver, *3*
DBI::DBIResult, *4*, *6*
DBI::dbSendQuery(), *4*, *5*
dbSendQuery,AdbiConnection,character-method *(*dbSendQueryArrow_AdbiConnection*)*, *5*
dbSendQuery_AdbiConnection_character *(*dbSendQueryArrow_AdbiConnection*)*, *5*
dbSendQueryArrow,AdbiConnection-method *(*dbSendQueryArrow_AdbiConnection*)*, *5*
dbSendQueryArrow_AdbiConnection, 5
dbSendStatement,AdbiConnection,character-method *(*dbSendQueryArrow_AdbiConnection*)*, *5*
dbSendStatement_AdbiConnection_character *(*dbSendQueryArrow_AdbiConnection*)*, *5*