

# Package ‘MeSHDbi’

March 31, 2025

**Title** DBI to construct MeSH-related package from sqlite file

**Description** The package is unified implementation of MeSH.db, MeSH.AOR.db, and MeSH.PCR.db and also is interface to construct GeneMeSH package (MeSH.XXX.eg.db). loadMeSHDbiPkg import sqlite file and generate MeSH.XXX.eg.db.

**Version** 1.42.0

**Author** Koki Tsuyuzaki

**Maintainer** Koki Tsuyuzaki <k.t.the-answer@hotmail.co.jp>

**Depends** R (>= 3.0.1)

**Imports** methods, AnnotationDbi (>= 1.31.19), RSQLite, Biobase

**Suggests** testthat

**License** Artistic-2.0

**biocViews** Annotation, AnnotationData, Infrastructure

**git\_url** <https://git.bioconductor.org/packages/MeSHDbi>

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** 6355ff4

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2025-03-31

## Contents

MeSHDb-class . . . . .	1
<b>Index</b>	<b>5</b>

---

MeSHDb-class	<i>MeSHDb objects</i>
--------------	-----------------------

---

## Description

MeSHDb is the simple class for providing the relationship between Entrez gene IDs and MeSH IDs. It provides the database connection and easily accessible with `columns`, `keytypes`, `keys` and `select`. Some users may use additional functions such as `dbconn`, `dbfile`, `dbschema`, `dbInfo`, `species`, `nomenclature`, `listDatabases`, and `meshVersion` for much complex data acquisition.

`columns` shows which kinds of data can be returned for the MeSHDb object.

`keytypes` allows the user to discover which keytypes can be passed in to `select` or `keys` and the keytype argument.

`keys` returns keys for the database contained in the MeSHDb object . This method is already documented in the keys manual page but is mentioned again here because it's usage with `select` is so intimate. By default it will return the primary keys for the database, but if used with the keytype argument, it will return the keys from that keytype.

`select` will retrieve the data as a `data.frame` based on parameters for selected keys, columns, and keytype arguments.

`dbconn` returns the connection with database in the package.

`dbfile` returns the absolute path sqlite file is saved.

`dbschema` returns the database schema.

`dbInfo` returns the many meta information about the package.

`species` returns the species name.

`nomenclature` returns the scientific name.

`listDatabases` returns the list of databases to correspond Gene ID and MeSH ID.

`meshVersion` returns the version of MeSH.

## Usage

```
columns(x)
keytypes(x)
keys(x, keytype, ...)
select(x, keys, columns, keytype, ...)
dbconn(x)
dbfile(x)
dbschema(x, file = "", show.indices = FALSE)
dbInfo(x)
species(object)
nomenclature(x)
listDatabases(x)
meshVersion(x)
```

## Arguments

<code>x</code>	the MeSHDb object converted by <code>MeSHDbi::MeSHDb</code> .
<code>object</code>	same as <code>x</code>
<code>keys</code>	the keys to select records for from the database. All possible keys are returned by using the <code>keys</code> method.
<code>columns</code>	the columns or kinds of things that can be retrieved from the database. As with <code>keys</code> , all possible columns are returned by using the <code>columns</code> method.

keytype	the keytype that matches the keys used. For the <code>select</code> methods, this is used to indicate the kind of ID being used with the <code>keys</code> argument. For the <code>keys</code> method this is used to indicate which kind of keys are desired from keys
...	other arguments.
file	The <code>file</code> argument must be a connection, or a character string naming the file to print to (see the <code>file</code> argument of the <code>cat</code> function for the details).
<code>show.indices</code>	The <code>CREATE INDEX</code> statements are not shown by default. Use <code>show.indices=TRUE</code> to get them.

### Value

`keys`, `columns`, `keytypes`, `dbfile`, `dbInfo`, `species`, and `nomenclature` each return a character vector or possible values. `select`, `dbschema`, `listDatabases`, and `meshVersion` each return a `data.frame`. `dbconn` returns database connection.

### Author(s)

Koki Tsuyuzaki

### See Also

[dbConnect](#)

### Examples

```
# library("MeSHDbi")
# library("AnnotationHub")

## Data retrieval from AnnotationHub
# ah <- AnnotationHub()
# dbfile <- query(ah, c("MeSHDb", "Bombyx mori", "v001"))[[1]]

## Constructor
# MeSH.Sil.eg.db <- MeSHDbi::MeSHDb(dbfile)

## show
# MeSH.Sil.eg.db

## dbconn
# dbconn(MeSH.Sil.eg.db)

## dbfile
# dbfile(MeSH.Sil.eg.db)

## dbschema
# dbschema(MeSH.Sil.eg.db)

## dbInfo
# dbInfo(MeSH.Sil.eg.db)

## species
# species(MeSH.Sil.eg.db)

## nomenclature
# nomenclature(MeSH.Sil.eg.db)
```

```
# # listDatabases
# listDatabases(MeSH.Sil.eg.db)

# # meshVersion
# meshVersion(MeSH.Sil.eg.db)

# # columns
# cols <- columns(MeSH.Sil.eg.db)

# # keytypes
# kys <- keytypes(MeSH.Sil.eg.db)

# # keys
# ks <- keys(MeSH.Sil.eg.db, keytype="GENEID")[seq(10)]

# # select
# out <- select(MeSH.Sil.eg.db,
#   columns=cols,
#   keys=ks,
#   keytype="GENEID")
```

# Index

cat, [3](#)  
class:MeSHDb (MeSHDb-class), [1](#)  
columns (MeSHDb-class), [1](#)  
columns, MeSHDb-method (MeSHDb-class), [1](#)  
  
dbconn (MeSHDb-class), [1](#)  
dbconn, MeSHDb-method (MeSHDb-class), [1](#)  
dbConnect, [3](#)  
dbfile (MeSHDb-class), [1](#)  
dbfile, MeSHDb-method (MeSHDb-class), [1](#)  
dbInfo (MeSHDb-class), [1](#)  
dbInfo, MeSHDb-method (MeSHDb-class), [1](#)  
dbschema (MeSHDb-class), [1](#)  
dbschema, MeSHDb-method (MeSHDb-class), [1](#)  
  
keys (MeSHDb-class), [1](#)  
keys, MeSHDb-method (MeSHDb-class), [1](#)  
keytypes (MeSHDb-class), [1](#)  
keytypes, MeSHDb-method (MeSHDb-class), [1](#)  
  
listDatabases (MeSHDb-class), [1](#)  
listDatabases, MeSHDb-method  
(MeSHDb-class), [1](#)  
  
MeSHDb (MeSHDb-class), [1](#)  
MeSHDb-class, [1](#)  
meshVersion (MeSHDb-class), [1](#)  
meshVersion, MeSHDb-method  
(MeSHDb-class), [1](#)  
  
nomenclature (MeSHDb-class), [1](#)  
nomenclature, MeSHDb-method  
(MeSHDb-class), [1](#)  
  
select (MeSHDb-class), [1](#)  
select, MeSHDb-method (MeSHDb-class), [1](#)  
species (MeSHDb-class), [1](#)  
species, MeSHDb-method (MeSHDb-class), [1](#)