Package 'ECOTOXr'

July 9, 2026

Type Package

Title Download and Extract Data from US EPA's ECOTOX Database

Version 1.2.5

Author Pepijn de Vries [aut, cre, dtc] (ORCID:
<https://orcid.org/0000-0002-7961-6646>)

Maintainer Pepijn de Vries <pepijn.devries@outlook.com>

Description The US EPA ECOTOX database is a freely available database with a treasure of aquatic and terrestrial ecotoxicological data. As the online search interface doesn't come with an API, this package provides the means to easily access and search the database in R. To this end, all raw tables are downloaded from the EPA website and stored in a local SQLite database <doi:10.1016/j.chemosphere.2024.143078>.

Depends R (>= 4.1.0), RSQLite (>= 2.3.4)

Imports bit64 (>= 4.6.0), cli (>= 3.6.4), dbplyr (>= 2.4.0), dplyr (>= 1.1.4), httr2 (>= 1.0.0), jsonlite (>= 1.8.8), lifecycle (>= 1.0.4), purrr (>= 1.0.2), rappdirs (>= 0.3.3), readr (>= 2.1.4), readxl (>= 1.4.3), rlang (>= 1.1.2), rvest (>= 1.0.3), stringr (>= 1.5.1), tibble (>= 3.2.1), tidyr (>= 1.3.0), tidyselect (>= 1.2.0), tools (>= 4.0.0), units (>= 0.8.5), utils

Suggests DBI, htmltools, kableExtra, knitr, rmarkdown, spelling, standartox, testthat (>= 3.0.0), webchem

URL https://github.com/pepijn-devries/ECOTOXr, https://pepijn-devries.github.io/ECOTOXr/, https://doi.org/10.1016/j.chemosphere.2024.143078

BugReports https://github.com/pepijn-devries/ECOTOXr/issues

License GPL (>= 3)

Encoding UTF-8

Language en-GB

Config/testthat/edition 3

VignetteBuilder knitr

Collate 'ECOTOXr-package.r' 'cas_handlers.r' 'database_access.r' 'helpers.r' 'imports.r' 'init.r' 'online.r' 'process_date.r' 'process_unit.r' 'process_numeric.r' 'wrappers.r'

Config/roxygen2/version 8.0.0

NeedsCompilation no

Repository CRAN

Date/Publication 2026-07-09 08:00:02 UTC

Contents

as_date_ecotox . . . 2
as_numeric_ecotox . . . 4
as_unit_ecotox . . . 6
build_ecotox_sqlite . . . 8
cas . . . 9
check_ecotox_availability . . . 13
check_ecotox_build . . . 14
check_ecotox_version . . . 15
cite_ecotox . . . 16
dbConnectEcotox . . . 17
download_ecotox_data . . . 18
get_ecotox_info . . . 19
get_ecotox_sqlite_file . . . 20
get_ecotox_url . . . 22
list_ecotox_fields . . . 23
migrate_ecotox_path . . . 24
mixed_to_single_unit . . . 25
process_ecotox_dates . . . 26
process_ecotox_numerics . . . 27
process_ecotox_units . . . 28
search_ecotox . . . 30
websearch_comptox . . . 33
websearch_ecotox . . . 35

Index **38**

`as_date_ecotox` *Values represented by ECOTOX* `character` *to dates*

Description

[Experimental] Similar to `as.Date()`, but it also performs some text sanitising before coercing text to dates.

Usage

```
as_date_ecotox(x, dd = 1L, mm = 1L, nr = 1L, ..., warn = TRUE)
```

Arguments

x	A vector of character strings. It expects fields as commonly returned from the ECOTOX database.
dd	Replacement values for unspecified days in a date. Defaults to 1L. If you want dates with unspecified days to result in NA, use dd = -1L.
mm	Replacement values for unspecified months in a date. Defaults to 1L. If you want dates with unspecified months to result in NA, use mm = -1L.
nr	Replacement values for generically unspecified values in a date. Defaults to 1L. If you want dates with unspecified values to result in NA, use nr = -1L.
...	Passed to as.Date().
warn	If set to FALSE warnings while converting text to dates are suppressed.

Details

The following steps are performed (in the order as listed) to sanitise text before coercing it to numerics:

- Trim whitespaces
- Replace hyphens with forward slashes
- Replace double forward slashes, forward slashes followed by a zero and spaces, with a single forward slash
- Replace "mm" or "dd" (case insensitive) with the value specified as argument. Add a forward slash to it when missing.
- Treat "na", "nr", "xx" and "00" (case insensitive) as unreported values when followed by a forward slash. Replace it with the nr argument
- Remove alphabetical characters when directly followed by a numerical character.
- Replace literal month names with its numerical calendar value (1-12).
- When the date consists of one value, assume it is a calender year and add dd and mm as day and month value.
- If a date consists of two numbers, assume it is month, followed by year. In that case insert the dd value for the day.

It is your own responsibility to check if the sanitising steps are appropriate for your analyses.

Value

A vector of Date class objects with the same length as x.

Author(s)

Pepijn de Vries

See Also

Other ecotox-sanitisers: `as_numeric_ecotox()`, `as_unit_ecotox()`, `mixed_to_single_unit()`, `process_ecotox_dates()`, `process_ecotox_numerics()`, `process_ecotox_units()`

Examples

```
## a vector of commonly used notations in the database to represent
## dates. Most frequent format is %m/%d/%Y
char_date <- c("5-19-1987   ", "5/dd/2021", "3/19/yyyy", "1985", "mm/19/1999",
               "October 2004", "nr/nr/2015")

as_date_ecotox(char_date)

## Set unspecified days to 15:
as_date_ecotox(char_date, dd = 15L)

## Unspecified days should result in NA:
as_date_ecotox(char_date, dd = -1L)

## Set unspecified months to 6:
as_date_ecotox(char_date, mm = 6L)

## Set generically unspecified value to 6:
as_date_ecotox(char_date, nr = 6L)
```

---

as_numeric_ecotox *Values represented by ECOTOX* character *to* numeric

---

Description

[Experimental] Similar to `as.numeric()`, but it also performs some text sanitising before coercing text to numerics.

Usage

```
as_numeric_ecotox(x, range_fun = NULL, ..., warn = TRUE)
```

Arguments

x	A vector of character strings. It expects fields as commonly returned from the ECOTOX database.
range_fun	Function to summarise range values. If NULL range values are returned as NA
...	Arguments passed to range_fun.
warn	If set to FALSE warnings while converting text to numerics are suppressed.

Details

The following steps are performed to sanitise text before coercing it to numerics:

- Notes labelled with "x" or "*" are removed.
- Operators (">", ">=", "<", "<=", "~", "=", "ca", "er") are removed.
- Text between brackets ("()") is removed (including the brackets)
- Comma's are considered to be a thousand separator when they are located at any fourth character (from the right) and removed. Comma's at any other location is assumed to be a decimal separator and is replaced by a period.
- If there is a hyphen present (not preceded by an ""e" or "E") it is probably representing a range of values. When range_fun is NULL it will result in a NA. Otherwise, the numbers are split at the hyphen and aggregated with range_fun

It is your own responsibility to check if the sanitising steps are appropriate for your analyses.

Value

A vector of numeric values with the same length as x.

Author(s)

Pepijn de Vries

See Also

Other ecotox-sanitisers: as_date_ecotox(), as_unit_ecotox(), mixed_to_single_unit(), process_ecotox_dates(), process_ecotox_numerics(), process_ecotox_units()

Examples

```
## a vector of commonly used notations in the database to represent
## numeric values
char_num <- c("10", " 2", "3 ", "~5", "9.2*", "2,33",
              "2,333", "2.1(1.0 - 3.2)", "1-5", "1e-3")

## Text fields reported as ranges are returned as `NA`:
as_numeric_ecotox(char_num, warn = FALSE)

## Text fields reported as ranges are processed with `range_fun`
as_numeric_ecotox(char_num, range_fun = median)
```

`as_unit_ecotox` *Text from the ECOTOX database to* `mixed_units`

Description

[Experimental] Convert text to units after sanitising.

Usage

```
as_unit_ecotox(
  x,
  type = c("concentration", "duration", "length", "media", "application", "size",
    "weight", "unknown"),
  ...,
  warn = TRUE
)
```

Arguments

`x`	A vector of `character` strings. It expects fields as commonly returned from the ECOTOX database.
`type`	The type of unit that can help the sanitation process. See the 'usage' section for available options. These options are linked to the different unit tables in the database (see `vignette("ecotox-schema")`). It can help to interpret ambiguous units correctly. For instance, 'dpm' can both mean 'disintegrations per minute' (`type = "concentration"`) and 'days post-moult' (`type = "duration"`).
`...`	Ignored.
`warn`	If set to `FALSE` warnings while converting text to units are suppressed.

Details

The following steps are performed (in the order as listed) to sanitise text before coercing it to units:

- The following is removed:
  - Leading/trailing white spaces
  - Square brackets and commas
  - A list of common prefixes
  - Double spaces are replaced by single spaces
  - Brackets around multiply symbol
- The following is corrected/adjusted:
  - 'for' is interpreted as multiplication
  - Scientific notation of numbers is standardised where possible.
  - A list of ambiguous patterns is replaced with more explicit strings. For instance, 'deg' is replaced with 'degree'.

- The following miscellaneous corrections are made:
  - A list of 'known' annotations are removed from the units
  - A list of elements known to represent counts are renamed 'counts'.
  - Percentages are renamed as explicit concentration in mass per volume or volume per volume units where possible
  - 'CI' is renamed 'Curies'.
  - 'M' is renamed 'mol/L'.
  - Units expressed as 'parts per ...' are explicitly renamed to mass over volume, or volume over volume where possible
- Type specific sanitation steps
  - Concentration units:
    - 'K' is renamed 'Karmen'
    - 'dpm' is renamed 'counts/min' (i.e., disintegrations per minute)
  - Media units:
    - 'K' is renamed 'Kelvin'
    - 'C' is renamed 'Celsius'
- Some final miscellaneous adjustments:
  - Scientific notation in numbers is not supported by the units package. Numbers are formatted in decimal notation where possible.
  - Spaces are removed if preceded by numeric and followed by alphabetical character
  - All equivalents of ambiguous synonyms for time units are explicitly renamed to their respective unit (e.g., 'dph' (days post hatching) -> 'day')
  - unreported/missing units are renamed 'unit'

It is your own responsibility to check if the sanitising steps are appropriate for your analyses.

Value

A vector of `?units::unit` class objects with the same length as `x`.

Author(s)

Pepijn de Vries

See Also

Other ecotox-sanitisers: `as_date_ecotox()`, `as_numeric_ecotox()`, `mixed_to_single_unit()`, `process_ecotox_dates()`, `process_ecotox_numerics()`, `process_ecotox_units()`

Examples

```
## Try parsing a random set of units from the database:
c("ppm-d", "ml/2.5 cm eu", "fl oz/10 gal/1k sqft", "kg/100 L",
  "mopm", "ng/kg", "ug", "AI ng/g", "PH", "pm", "uM/cm3", "1e-4 mM",
  "degree", "fs", "mg/TI", "RR", "ug/g org/d", "1e+4 IU/TI", "pg/mg TE",
  "pmol/mg", "1e-9/l", "no >15 cm", "umol/mg pro", "cc/org/wk", "PIg/L",
  "ug/100 ul/org", "ae mg/kg diet/d", "umol/mg/h", "cmol/kg d soil",
```

```
  "ug/L diet", "kg/100 kg sd", "1e+6 cells", "ul diet", "S", "mmol/h/g TI",
  "g/70 d", "vg", "ng/200 mg diet", "uS/cm2", "AI ml/ha", "AI pt/acre",
  "mg P/h/g TI", "no/m", "kg/ton sd", "ug/g wet wt", "AI mg/2 L diet",
  "nmol/TI", "umol/g wet wt", "PSU", "Wijs number") |>
  as_unit_ecotox(warn = FALSE)

## Adding the type of measurement can affect interpretation:
as_unit_ecotox(c("C", "K"), type = "concentration")
as_unit_ecotox(c("C", "K"), type = "media")
```

`build_ecotox_sqlite` *Build an SQLite database from zip archived tables downloaded from EPA website*

Description

[Stable] This function is called automatically after `download_ecotox_data()`. The database files can also be downloaded manually from the EPA website from which a local database can be build using this function.

Usage

```
build_ecotox_sqlite(source, destination = get_ecotox_path(), write_log = TRUE)
```

Arguments

`source`	A `character` string pointing to the directory path where the text files with the raw tables are located. These can be obtained by extracting the zip archive from `https://cfpub.epa.gov/ecotox/` and look for 'Download ASCII Data'.
`destination`	A `character` string representing the destination path for the SQLite file. By default this is `get_ecotox_path()`.
`write_log`	A `logical` value indicating whether a log file should be written in the destination path `TRUE`. The log contains information on the source and destination path, the version of this package, the creation date, and the operating system on which the database was created.

Details

Raw data downloaded from the EPA website is in itself not very efficient to work with in R. The files are large and would put a large strain on R when loading completely into the system's memory. Instead use this function to build an SQLite database from the tables. That way, the data can be queried without having to load it all into memory.

EPA provides the raw table from the ECOTOX database as text files with pipe-characters ('|') as table column separators. Although not documented, the tables appear not to contain comment or quotation characters. There are records containing the reserved pipe-character that will confuse the table parser. For these records, the pipe-character is replaced with a dash character ('-').

In addition, while reading the tables as text files, this package attempts to decode the text as UTF8. Unfortunately, this process appears to be platform-dependent, and may therefore result in different end-results on different platforms. This problem only seems to occur for characters that are listed as 'control characters' under UTF8. This will have consequences for reproducibility, but only if you build search queries that look for such special characters. It is therefore advised to stick to common (non-accented) alpha-numerical characters in your searches, for the sake of reproducibility.

Use '`suppressMessages()`' to suppress the progress report.

Value

Returns `NULL` invisibly.

Author(s)

Pepijn de Vries

See Also

Other database-build-functions: `check_ecotox_build()`, `check_ecotox_version()`, `download_ecotox_data()`, `get_ecotox_url()`

Examples

```
source_path <- tempfile()
dir.create(source_path)

## This is a small mockup file resembling the larger zip
## files that can be downloaded with `download_ecotox_data()`:

source_file <- system.file("ecotox-test.zip", package = "ECOTOXr")

unzip(source_file, exdir = source_path)

build_ecotox_sqlite(source_path, tempdir())
```

---

cas *Functions for handling chemical abstract service (CAS) registry numbers*

Description

[Stable] Functions for handling chemical abstract service (CAS) registry numbers

Usage

```
cas(length = 0L)

is.cas(x)

as.cas(x)

## S3 method for class 'cas'
x[[i]]

## S3 method for class 'cas'
x[i]

## S3 replacement method for class 'cas'
x[[i]] <- value

## S3 replacement method for class 'cas'
x[i] <- value

## S3 method for class 'cas'
format(x, hyphenate = TRUE, ...)

## S3 method for class 'cas'
as.character(x, ...)

show.cas(x, ...)

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

## S3 method for class 'cas'
as.list(x, ...)

## S3 method for class 'cas'
as.double(x, ...)

## S3 method for class 'cas'
as.integer(x, ...)

## S3 method for class 'cas'
c(...)

## S3 method for class 'cas'
as.data.frame(...)
```

Arguments

length	A non-negative integer specifying the desired length. Double values will be coerced to integer: supplying an argument of length other than one is an error.
x	Object from which data needs to be extracted or replaced, or needs to be coerced into a specific format. For nearly all of the functions documented here, this needs to be an object of the S3 class 'cas', which can be created with as.cas. For as.cas, x can be a character (CAS registry number with or without hyphenation) or a numeric value. Note that as.cas will only accept correctly formatted and valid CAS registry numbers.
i	Index specifying element(s) to extract or replace. See also base::Extract().
value	A replacement value, can be anything that can be converted into an S3 cas-class object with as.cas.
hyphenate	A logical value indicating whether the formatted CAS number needs to be hyphenated. Default is TRUE.
...	Arguments passed to other functions

Details

In the database CAS registry numbers are stored as text (type character). As CAS numbers can consist of a maximum of 10 digits (plus two hyphens) this means that each CAS number can consume up to 12 bytes of memory or disk space. By storing the data numerically, only 5 bytes are required. These functions provide the means to handle CAS registry numbers and coerce from and to different formats and types.

Value

Functions cas, c and as.cas return S3 class 'cas' objects. Coercion functions (starting with 'as') return the object as specified by their respective function names (i.e., integer, double, character, list and data.frame). The show.cas and print functions also return formatted charaters. The function is.cas will return a single logical value, indicating whether x is a valid S3 cas-class object. The square brackets return the selected index/indices, or the vector of cas objects where the selected elements are replaced by value.

Author(s)

Pepijn de Vries

Examples

```
## This will generate a vector of cas objects containing 10
## fictive (0-00-0), but valid registry numbers:
cas(10)

## This is a cas-object:
is.cas(cas(0L))

## This is not a cas-object:
is.cas(0L)
```

```
## Three different ways of creating a cas object from
## Benzene's CAS registry number (the result is the same)
as.cas("71-43-2")
as.cas("71432")
as.cas(71432L)

## This is one way of creating a vector with multiple CAS registry numbers:
cas_data <- as.cas(c("64175", "71432", "58082"))

## This is how you select a specific element(s) from the vector:
cas_data[2:3]
cas_data[[2]]

## You can also replace specific elements in the vector:
cas_data[1] <- "7440-23-5"
cas_data[[2]] <- "129-00-0"

## You can format CAS numbers with or without hyphens:
format(cas_data, TRUE)
format(cas_data, FALSE)

## The same can be achieved using as.character
as.character(cas_data, TRUE)
as.character(cas_data, FALSE)

## There are also show and print methods available:
show(cas_data)
print(cas_data)

## Numeric values can be obtained from CAS using as.numeric, as.double or as.integer
as.numeric(cas_data)

## Be careful, however. Some CAS numbers cannot be represented by R's 32 bit integers
## and will produce NA's. This will work OK:
huge_cas <- as.cas("9999999-99-5")

## Not run:
## This will not:
as.integer(huge_cas)

## End(Not run)

## The trick applied by this package is that the final
## validation digit is stored separately as attribute:
unclass(huge_cas)

## This is how cas objects can be concatenated:
cas_data <- c(huge_cas, cas_data)

## This will create a data.frame
as.data.frame(cas_data)
```

```
## This will create a list:
as.list(cas_data)
```

`check_ecotox_availability` *Check whether a ECOTOX database exists locally*

Description

[Stable] Tests whether a local copy of the US EPA ECOTOX database exists in `get_ecotox_path()`.

Usage

```
check_ecotox_availability(target = get_ecotox_path())
```

Arguments

`target`	A `character` string specifying the path where to look for the database file.

Details

When arguments are omitted, this function will look in the default directory (`get_ecotox_path()`). However, it is possible to build a database file elsewhere if necessary.

Value

Returns a `logical` value indicating whether a copy of the database exists. It also returns a `files` attribute that lists which copies of the database are found.

Author(s)

Pepijn de Vries

See Also

Other database-access-functions: `check_ecotox_build()`, `check_ecotox_version()`, `cite_ecotox()`, `dbConnectEcotox()`, `get_ecotox_info()`, `get_ecotox_sqlite_file()`, `list_ecotox_fields()`, `migrate_ecotox_path()`

Examples

```
check_ecotox_availability()
```

`check_ecotox_build` *Check the locally build database for validity*

Description

[Stable] Performs some simple tests to check whether the locally built database is not corrupted.

Usage

```
check_ecotox_build(path = get_ecotox_path(), version, ...)
```

Arguments

`path`	A `character` string with the path to the location of the local database (default is `get_ecotox_path()`).
`version`	A `character` string referring to the release version of the database you wish to locate. It should have the same format as the date in the EPA download link, which is month, day, year, separated by underscores ("%m_%d_%Y"). When missing, the most recent available copy is selected automatically.
`...`	Arguments that are passed to `dbConnect()` method or `dbDisconnect()` method.

Details

For now this function tests if all expected tables are present in the locally built database. Note that in later release of the database some tables were added. Therefore for older builds this function might return `FALSE` whereas it is actually just fine (just out-dated).

Furthermore, this function tests if all tables contain one or more records. Obviously, this is no guarantee that the database is valid, but it is a start.

More tests may be added in future releases.

Value

Returns an indicative logical value whether the database is not corrupted. `TRUE` indicates the database is most likely OK. `FALSE` indicates that something might be wrong. Additional messages (when `FALSE`) are included as attributes containing hints on the outcome of the tests. See also the 'details' section.

Author(s)

Pepijn de Vries

See Also

Other database-access-functions: `check_ecotox_availability()`, `check_ecotox_version()`, `cite_ecotox()`, `dbConnectEcotox()`, `get_ecotox_info()`, `get_ecotox_sqlite_file()`, `list_ecotox_fields()`, `migrate_ecotox_path()`

Other database-build-functions: `build_ecotox_sqlite()`, `check_ecotox_version()`, `download_ecotox_data()`, `get_ecotox_url()`

Examples

```
if (check_ecotox_availability()) {
  check_ecotox_build()
}
```

`check_ecotox_version` *Check if the locally build database is up to date*

Description

[Stable] Checks the version of the database available online from the EPA against the specified version (latest by default) of the database build locally. Returns `TRUE` when they are the same.

Usage

```
check_ecotox_version(path = get_ecotox_path(), version, verbose = TRUE, ...)
```

Arguments

`path`	When you have a copy of the database somewhere other than the default directory (`get_ecotox_path()`), you can provide the path here.
`version`	A `character` string referring to the release version of the database you wish to locate. It should have the same format as the date in the EPA download link, which is month, day, year, separated by underscores ("%m_%d_%Y"). When missing, the most recent available copy is selected automatically.
`verbose`	A `logical` value. If true messages are shown on the console reporting on the check.
`...`	Arguments passed to `get_ecotox_url()`

Value

Returns a `logical` value invisibly indicating whether the locally build is up to date with the latest release by the EPA.

Author(s)

Pepijn de Vries

See Also

Other database-access-functions: `check_ecotox_availability()`, `check_ecotox_build()`, `cite_ecotox()`, `dbConnectEcotox()`, `get_ecotox_info()`, `get_ecotox_sqlite_file()`, `list_ecotox_fields()`, `migrate_ecotox_path()`

Other database-build-functions: `build_ecotox_sqlite()`, `check_ecotox_build()`, `download_ecotox_data()`, `get_ecotox_url()`

Examples

```
if (interactive() && check_ecotox_availability()) {
  check_ecotox_version()
}
```

`cite_ecotox` *Cite the downloaded copy of the ECOTOX database*

Description

[Stable] Cite the downloaded copy of the ECOTOX database and this package (`citation("ECOTOXr")`) for reproducible results.

Usage

```
cite_ecotox(path = get_ecotox_path(), version)
```

Arguments

`path`	A `character` string with the path to the location of the local database (default is `get_ecotox_path()`).
`version`	A `character` string referring to the release version of the database you wish to locate. It should have the same format as the date in the EPA download link, which is month, day, year, separated by underscores ("%m_%d_%Y"). When missing, the most recent available copy is selected automatically.

Details

When you download a copy of the EPA ECOTOX database using `download_ecotox_data()`, a BibTex file is stored that registers the database release version and the access (= download) date. Use this function to obtain a citation to that specific download.

In order for others to reproduce your results, it is key to cite the data source as accurately as possible.

Value

Returns a `vector` of `bibentry()`'s, containing a reference to the downloaded database and this package.

Author(s)

Pepijn de Vries

See Also

Other database-access-functions: `check_ecotox_availability()`, `check_ecotox_build()`, `check_ecotox_version()`, `dbConnectEcotox()`, `get_ecotox_info()`, `get_ecotox_sqlite_file()`, `list_ecotox_fields()`, `migrate_ecotox_path()`

Examples

```
## In order to cite downloaded database and this package:
cite_ecotox() |> suppressWarnings()
```

---

dbConnectEcotox *Open or close a connection to the local ECOTOX database*

Description

[Stable] Wrappers for `dbConnect()` and `dbDisconnect()` methods.

Usage

```
dbConnectEcotox(path = get_ecotox_path(), version, ...)

dbDisconnectEcotox(conn, ...)
```

Arguments

path	A `character` string with the path to the location of the local database (default is `get_ecotox_path()`).
version	A `character` string referring to the release version of the database you wish to locate. It should have the same format as the date in the EPA download link, which is month, day, year, separated by underscores ("%m_%d_%Y"). When missing, the most recent available copy is selected automatically.
...	Arguments that are passed to `dbConnect()` method or `dbDisconnect()` method.
conn	An open connection to the ECOTOX database that needs to be closed.

Details

Open or close a connection to the local ECOTOX database. These functions are only required when you want to send custom queries to the database. For most searches the `search_ecotox()` function will be adequate.

Value

A database connection in the form of a `DBI::DBIConnection-class()` object. The object is tagged with: a time stamp; the package version used; and the file path of the SQLite database used in the connection. These tags are added as attributes to the object.

Author(s)

Pepijn de Vries

See Also

Other database-access-functions: `check_ecotox_availability()`, `check_ecotox_build()`, `check_ecotox_version()`, `cite_ecotox()`, `get_ecotox_info()`, `get_ecotox_sqlite_file()`, `list_ecotox_fields()`, `migrate_ecotox_path()`

Examples

```
## This will only work when a copy of the database exists:
if (check_ecotox_availability()) {
  con <- dbConnectEcotox()

  ## check if the connection works by listing the tables in the database:
  dbListTables(con)

  ## Let's be a good boy/girl and close the connection to the database when we're done:
  dbDisconnectEcotox(con)
}
```

`download_ecotox_data` *Download and extract ECOTOX database files and compose database*

Description

[Stable] In order for this package to fully function, a local copy of the ECOTOX database needs to be build. This function will download the required data and build the database.

Usage

```
download_ecotox_data(
  target = get_ecotox_path(),
  write_log = TRUE,
  ask = interactive(),
  verify_ssl = getOption("ECOTOXr_verify_ssl"),
  ...
)
```

Arguments

`target`	Target directory where the files will be downloaded and the database compiled. Default is `get_ecotox_path()`.
`write_log`	A `logical` value indicating whether a log file should be written to the target path `TRUE`.
`ask`	There are several steps in which files are (potentially) overwritten or deleted. In those cases the user is asked on the command line what to do in those cases. Set this parameter to `FALSE` in order to continue without warning and asking.
`verify_ssl`	When set to `FALSE` the SSL certificate of the host (EPA) is not verified. Can also be set as option: `options(ECOTOXr_verify_ssl = TRUE)`. Default is `TRUE`.
`...`	Arguments passed on to `httr2::req_options()`.

Details

This function will attempt to find the latest download url for the ECOTOX database from the EPA website (see `get_ecotox_url()`). When found it will attempt to download the zipped archive containing all required data. This data is then extracted and a local copy of the database is build.

Use '`suppressMessages()`' to suppress the progress report.

Value

Returns `NULL` invisibly.

Known issues

On some machines this function fails to connect to the database download URL from the EPA website due to missing SSL certificates. Unfortunately, there is no easy fix for this in this package. A work around is to download and unzip the file manually using a different machine or browser that is less strict with SSL certificates. You can then call `build_ecotox_sqlite()` and point the `source` location to the manually extracted zip archive. For this purpose `get_ecotox_url()` can be used. Alternatively, one could try to call `download_ecotox_data()` by setting `verify_ssl = FALSE`; but only do so when you trust the download URL from get_ecotox_URL().

Author(s)

Pepijn de Vries

See Also

Other database-build-functions: `build_ecotox_sqlite()`, `check_ecotox_build()`, `check_ecotox_version()`, `get_ecotox_url()`

Other online-functions: `get_ecotox_url()`, `websearch_ecotox()`

Examples

```
## Not run:
## This will download and build the database in your temp dir:
if (interactive()) {
  download_ecotox_data(tempdir())
}

## End(Not run)
```

`get_ecotox_info` *Get information on the local ECOTOX database when available*

Description

[Stable] Get information on how and when the local ECOTOX database was build.

Usage

```
get_ecotox_info(path = get_ecotox_path(), version)
```

Arguments

path	A character string with the path to the location of the local database (default is get_ecotox_path()).
version	A character string referring to the release version of the database you wish to locate. It should have the same format as the date in the EPA download link, which is month, day, year, separated by underscores ("%m_%d_%Y"). When missing, the most recent available copy is selected automatically.

Details

Get information on how and when the local ECOTOX database was build. This information is retrieved from the log-file that is (optionally) stored with the local database when calling download_ecotox_data() or build_ecotox_sqlite().

Value

Returns a vector of characters, containing a information on the selected local ECOTOX database.

Author(s)

Pepijn de Vries

See Also

Other database-access-functions: check_ecotox_availability(), check_ecotox_build(), check_ecotox_version(), cite_ecotox(), dbConnectEcotox(), get_ecotox_sqlite_file(), list_ecotox_fields(), migrate_ecotox_path()

Examples

```
if (check_ecotox_availability()) {
  ## Show info on the current database (only works when one is downloaded and build):
  get_ecotox_info()
}
```

---

get_ecotox_sqlite_file

The local path to the ECOTOX database (directory or sqlite file)

Description

[Stable] Obtain the local path to where the ECOTOX database is (or will be) placed.

Usage

```
get_ecotox_sqlite_file(path = get_ecotox_path(), version)

get_ecotox_path()
```

Arguments

path	When you have a copy of the database somewhere other than the default directory (get_ecotox_path()), you can provide the path here.
version	A character string referring to the release version of the database you wish to locate. It should have the same format as the date in the EPA download link, which is month, day, year, separated by underscores ("%m_%d_%Y"). When missing, the most recent available copy is selected automatically.

Details

It can be useful to know where the database is located on your disk. This function returns the location as provided by tools::R_user_dir(), or as specified by you using options(ECOTOXr_path = "mypath").

Value

Returns a character string of the path. get_ecotox_path will return the default directory of the database. get_ecotox_sqlite_file will return the path to the sqlite file when it exists.

Author(s)

Pepijn de Vries

See Also

Other database-access-functions: check_ecotox_availability(), check_ecotox_build(), check_ecotox_version(), cite_ecotox(), dbConnectEcotox(), get_ecotox_info(), list_ecotox_fields(), migrate_ecotox_path()

Examples

```
get_ecotox_path()

if (check_ecotox_availability()) {
  ## This will only work if a local database exists:
  get_ecotox_sqlite_file()
}
```

`get_ecotox_url` *Get ECOTOX download URL from EPA website*

Description

[Stable] This function downloads the webpage at https://cfpub.epa.gov/ecotox/index.cfm. It then searches for the download link for the complete ECOTOX database and extract its URL.

Usage

```
get_ecotox_url(verify_ssl = getOption("ECOTOXr_verify_ssl"), ...)
```

Arguments

`verify_ssl`	When set to `FALSE` the SSL certificate of the host (EPA) is not verified. Can also be set as option: `options(ECOTOXr_verify_ssl = TRUE)`. Default is `TRUE`.
`...`	arguments passed on to `httr2::req_options()`

Details

This function is called by `download_ecotox_data()` which tries to download the file from the resulting URL. On some machines this fails due to issues with the SSL certificate. The user can try to download the file by using this URL in a different browser (or on a different machine). Alternatively, the user could try to use `[download_ecotox_data](verify_ssl = FALE)` when the download URL is trusted.

Value

Returns a `character` string containing the download URL of the latest version of the EPA ECOTOX database.

Author(s)

Pepijn de Vries

See Also

Other database-build-functions: `build_ecotox_sqlite()`, `check_ecotox_build()`, `check_ecotox_version()`, `download_ecotox_data()`

Other online-functions: `download_ecotox_data()`, `websearch_ecotox()`

Examples

```
if (interactive()) {
  get_ecotox_url()
}
```

`list_ecotox_fields` *List the field names that are available from the ECOTOX database*

Description

[Stable] List the field names (table headers) that are available from the ECOTOX database

Usage

```
list_ecotox_fields(
  which = c("default", "extended", "full", "all"),
  include_table = TRUE
)
```

Arguments

`which`	A `character` string that specifies which fields to return. Can be any of: '`default`': returns default output field names; '`all`': returns all fields; '`extended`': returns all fields of the default tables; or '`full`': returns all fields except those from tables 'chemical_carriers', 'media_characteristics', 'doses', 'dose_responses', 'dose_response_details', 'dose_response_links' and 'dose_stat_method_codes'.
`include_table`	A `logical` value indicating whether the table name should be included as prefix. Default is `TRUE`.

Details

This can be useful when specifying a `search_ecotox()`, to identify which fields are available from the database, for searching and output.

Not that when requesting '`all`' fields, you will get all fields available from the latest EPA release of the ECOTOX database. This means that not necessarily all fields are available in your local build of the database.

Value

Returns a `vector` of type `character` containing the field names from the ECOTOX database.

Author(s)

Pepijn de Vries

See Also

Other database-access-functions: `check_ecotox_availability()`, `check_ecotox_build()`, `check_ecotox_version()`, `cite_ecotox()`, `dbConnectEcotox()`, `get_ecotox_info()`, `get_ecotox_sqlite_file()`, `migrate_ecotox_path()`

Examples

```
## Fields that are included in search results by default:
list_ecotox_fields("default")

## All fields that are available from the ECOTOX database:
list_ecotox_fields("all")

## All except fields from the tables 'chemical_carriers', 'media_characteristics',
## 'doses', 'dose_responses', 'dose_response_details', 'dose_response_links' and
## 'dose_stat_method_codes' that are available from the ECOTOX database:
list_ecotox_fields("full")
```

---

migrate_ecotox_path *Temporary function to switch to new ECOTOX path*

Description

Since ECOTOXr v1.2.5, the path is located with the base package `tools` instead of the previously used `rappdirs`. Use this function to migrate your old files to this new location.

Usage

```
migrate_ecotox_path(overwrite = FALSE)
```

Arguments

`overwrite`	`logical` value. Should files that already exist in the new location be overwritten?

Value

Returns `NULL` invisibly.

See Also

Other database-access-functions: `check_ecotox_availability()`, `check_ecotox_build()`, `check_ecotox_version()`, `cite_ecotox()`, `dbConnectEcotox()`, `get_ecotox_info()`, `get_ecotox_sqlite_file()`, `list_ecotox_fields()`

`mixed_to_single_unit` *Convert mixed units to a specific unit*

Description

[Experimental] Converts a list of mixed units to a specific unit, using the `units` package.

Usage

```
mixed_to_single_unit(x, target_unit)
```

Arguments

`x`	A mixed units object (`units::mixed_units()`) to be converted to the `target_unit`
`target_unit`	A `character` string representing the target unit

Value

Returns a units object (`?units::units`). Values with units that cannot be converted to the `target_unit` is returned as `NA`.

Author(s)

Pepijn de Vries

See Also

Other ecotox-sanitisers: `as_date_ecotox()`, `as_numeric_ecotox()`, `as_unit_ecotox()`, `process_ecotox_dates()`, `process_ecotox_numerics()`, `process_ecotox_units()`

Examples

```
mishmash <- as_unit_ecotox(c("mg/L", "ppt w/v", "% w/v", "mmol/L"))

## Note that 'mmol/L' cannot be converted to 'ug/L'
## without a molar mass. It is returned as `NA`
mixed_to_single_unit(mishmash, "ug/L")

mishmash <- as_unit_ecotox(c("h", "sec", "mi", "dph"))

mixed_to_single_unit(mishmash, "h")
```

process_ecotox_dates *Process ECOTOX search results by converting* character *to dates where relevant*

Description

[Experimental] The function search_ecotox() returns fields from the ECOTOX database as is. Fields that represent dates are usually formatted as "%m\%d\%Y". Unfortunately, this format is not consistently used throughout the database. process_ecotox_dates() takes a data.frame returned by search_ecotox(), locates date columns, represented by text, sanitises the text and converts them to Date objects. It will sanitise the date fields as much as possible. It will correct most dates. Dates without a specified calender year, a date range, illegal date format (even after sanitation) are returned as NA.

Usage

```
process_ecotox_dates(x, .fns = as_date_ecotox, ..., .names = NULL)
```

Arguments

x	A data.frame obtained with search_ecotox(), for which the dates need to be processed.
.fns	Function to convert character to Date. By default as_date_ecotox() is used which also sanitises the input. You can also use as.Date() if you don't want the sanitation step. You can also write a custom function.
...	Arguments passed to .fns.
.names	A 'glue' specification used to rename the date columns. By default it is "{.col}", which will overwrite existing text columns with date columns. You can for instance add a suffix with "{.col}_date" if you want to rename the resulting date columns.

Value

Returns a data.frame in which the columns containing date information is converted from the character format from the database to actual date objects ("POSIXlt" and "POSIXct").

Author(s)

Pepijn de Vries

See Also

Other ecotox-sanitisers: as_date_ecotox(), as_numeric_ecotox(), as_unit_ecotox(), mixed_to_single_unit(), process_ecotox_numerics(), process_ecotox_units()

Examples

```
if (check_ecotox_availability()) {
  df <- search_ecotox(
    list(
      latin_name    = list(
        terms          = c("Skeletonema", "Daphnia"),
        method         = "contains"
      ),
      chemical_name = list(
        terms          = "benzene",
        method         = "exact"
      )
    ), list_ecotox_fields("full"))

  df_dat <-
    process_ecotox_dates(df, warn = FALSE)
}
```

`process_ecotox_numerics`

Process ECOTOX search results by converting `character` *to* `numeric` *where relevant*

Description

[Experimental] The function `search_ecotox()` returns fields from the ECOTOX database as is. Many numeric values are stored in the database as text. It is not uncommon that these text fields cannot be converted directly and need some sanitising first. `process_ecotox_numerics()` takes a `data.frame` returned by `search_ecotox()`, locates numeric columns, represented by text, sanitises the text and converts them to numerics.

Usage

```
process_ecotox_numerics(
  x,
  .fns = as_numeric_ecotox,
  ...,
  add_units = FALSE,
  .names = NULL
)
```

Arguments

`x`	A `data.frame` obtained with `search_ecotox()`, for which the numerics need to be processed.
`.fns`	Function to convert `character` to `numeric`. By default `as_numeric_ecotox()` is used which also sanitises the input. You can also use `as.numeric()` if you don't want the sanitation step. You can also write a custom function.

...	Arguments passed to .fns.
add_units	A logical value. When set to TRUE corresponding units are parsed with as_unit_ecotox() (if available) and added to the numeric value.
.names	A 'glue' specification used to rename the numeric columns. By default it is "{.col}", which will overwrite existing text columns with numeric columns. You can for instance add a suffix with "{.col}_num" if you want to rename the resulting numeric columns.

Value

Returns a data.frame in which the columns containing numeric information is converted from the character format from the database to actual numerics.

Author(s)

Pepijn de Vries

See Also

Other ecotox-sanitisers: as_date_ecotox(), as_numeric_ecotox(), as_unit_ecotox(), mixed_to_single_unit(), process_ecotox_dates(), process_ecotox_units()

Examples

```
if (check_ecotox_availability()) {
  df <- search_ecotox(
    list(
      latin_name    = list(
        terms          = c("Skeletonema", "Daphnia"),
        method         = "contains"
      ),
      chemical_name = list(
        terms          = "benzene",
        method         = "exact"
      )
    ), list_ecotox_fields("full"))

  df_num <-
    process_ecotox_numerics(df, add_units = TRUE, warn = FALSE)
}
```

---

process_ecotox_units *Process ECOTOX search results by converting* character *to units where relevant*

Description

[Experimental] The function `search_ecotox()` returns fields from the ECOTOX database as is. Fields that represent units are not standardised in the database. Therefore, this format is not consistently used throughout the database. `process_ecotox_units()` takes a `data.frame` returned by `search_ecotox()`, locates unit columns, represented by text, sanitises the text and converts them to `units::mixed_units()` objects. It will sanitise the unit fields as much as possible. Units that could not be interpreted are returned as arbitrary `unit`.

Usage

```
process_ecotox_units(x, .fns = as_unit_ecotox, ..., .names = NULL)
```

Arguments

`x`	A `data.frame` obtained with `search_ecotox()`, for which the units need to be processed.
`.fns`	Function to convert `character` to unit. By default `as_unit_ecotox()` is used which also sanitises the input. You can also write a custom function.
`...`	Arguments passed to `.fns`.
`.names`	A 'glue' specification used to rename the unit columns. By default it is `"{.col}"`, which will overwrite existing text columns with unit columns. You can for instance add a suffix with `"{.col}_unit"` if you want to rename the resulting unit columns.

Value

Returns a `data.frame` in which the columns containing unit information is converted from the character format from the database to actual unit objects (`?units::units`).

Author(s)

Pepijn de Vries

See Also

Other ecotox-sanitisers: `as_date_ecotox()`, `as_numeric_ecotox()`, `as_unit_ecotox()`, `mixed_to_single_unit()`, `process_ecotox_dates()`, `process_ecotox_numerics()`

Examples

```
if (check_ecotox_availability()) {
  df <- search_ecotox(
    list(
      latin_name    = list(
        terms          = c("Skeletonema", "Daphnia"),
        method         = "contains"
      ),
      chemical_name = list(
        terms          = "benzene",
```

```
        method         = "exact"
      )
    ), list_ecotox_fields("full"))

  df_unit <-
    process_ecotox_units(df, warn = FALSE)
}
```

---

search_ecotox *Search and retrieve toxicity records from the database*

Description

[Stable] Create (and execute) an SQL search query based on basic search terms and options. This allows you to search the database, without having to understand SQL.

Usage

```
search_ecotox(
  search,
  output_fields = list_ecotox_fields("default"),
  group_by_results = TRUE,
  compute = FALSE,
  as_data_frame = TRUE,
  ...
)

search_ecotox_lazy(
  search,
  output_fields = list_ecotox_fields("default"),
  compute = FALSE,
  ...
)

search_query_ecotox(search, output_fields = list_ecotox_fields("default"), ...)
```

Arguments

search	A named `list` containing the search terms. The names of the elements should refer to the field (i.e. table header) in which the terms are searched. Use `list_ecotox_fields()` to obtain a list of available field names. Each element in that list should contain another list with at least one element named 'terms'. This should contain a `vector` of `character` strings with search terms. Optionally, a second element named 'method' can be provided which should be set to either '`contains`' (default, when missing) or '`exact`'. In the first case the query will match any record in the indicated field that contains the search term. In case of '`exact`' it will only return exact matches. Note that searches are not case sensitive, but are picky with special (accented) characters.

While building the local database (see build_ecotox_sqlite) such special characters may be treated differently on different operating systems. For the sake of reproducibility, the user is advised to stick with non-accented alpha-numeric characters.

Search terms for a specific field (table header) will be combined with 'or'. Meaning that any record that matches any of the terms are returned. For instance when 'latin_name' 'Daphnia magna' and 'Skeletonema costatum' are searched, results for both species are returned. Search terms across fields (table headers) are combined with 'and', which will narrow the search. For instance if 'chemical_name' 'benzene' is searched in combination with 'latin_name' 'Daphnia magna', only tests where Daphnia magna are exposed to benzene are returned.

When this search behaviour described above is not desirable, the user can either adjust the query manually, or use this function to perform several separate searches and combine the results afterwards.

Beware that some field names are ambiguous and occur in multiple tables (like `cas_number' and code'`). When searching such fields, the search result may not be as expected.

`output_fields` A `vector` of `character` strings indicating which field names (table headers) should be included in the output. By default `[list_ecotox_fields]("default")` is used. Use `[list_ecotox_fields]("all")` to list all available fields.

`group_by_results`

Ecological test results are generally the most informative element in the ECOTOX database. Therefore, this search function returns a table with unique results in each row.

However, some tables in the database (such as 'chemical_carriers' and 'dose_responses') have a one to many relationship with test results. This means that multiple chemical carriers can be linked to a single test result, similarly, multiple doses can also be linked to a single test result.

By default the search results are grouped by test results. As a result not all doses or chemical carriers may be displayed in the output. Set the `group_by_results` parameter to `FALSE` in order to force SQLite to output all data (e.g., all carriers). But beware that test results may be duplicated in those cases.

`compute` The ECOTOXr package tries to construct database queries as lazy as possible. Meaning that R moves as much of the heavy lifting as possible to the database. When your search becomes complicated (e.g., when including many output fields), you may run into trouble and hit the SQL parser limits. In those cases you can set this parameter to `TRUE`. Database queries are then computed in the process of joining tables. This is generally slower. Alternatively, you could try to include less output fields in order to simplify the query.

`as_data_frame` **[Experimental]** `logical` value indicating whether the result should be converted into a `data.frame` (default is `TRUE`). When set to `FALSE` the data will be returned as a `tbl_df()`.

`...` Arguments passed to `dbConnectEcotox()` and other functions. You can use this when the database is not located at the default path (`get_ecotox_path()`).

Details

The ECOTOX database is stored locally as an SQLite file, which can be queried with SQL. These functions allow you to automatically generate an SQL query and send it to the database, without having to understand SQL. The function `search_query_ecotox` generates and returns the SQL query (which can be edited by hand if desired). You can also directly call `search_ecotox`, this will first generate the query, send it to the database and retrieve the result.

Although the generated query is not optimized for speed, it should be able to process most common searches within an acceptable time. The time required for retrieving data from a search query depends on the complexity of the query, the size of the query and the speed of your machine. Most queries should be completed within seconds (or several minutes at most) on modern machines. If your search require optimisation for speed, you could try reordering the search fields. You can also edit the query generated with `search_query_ecotox` by hand and retrieve it with `DBI::dbGetQuery()`.

Note that this package is actively maintained and this function may be revised in future versions. In order to create reproducible results the user must: always work with an official release from CRAN and document the package and database version that are used to generate specific results (see also `cite_ecotox()`).

Value

In case of `search_query_ecotox`, a `character` string containing an SQL query is returned. This query is built based on the provided search terms and options.

In case of `search_ecotox` a `data.frame` is returned based on the search query built with `search_query_ecotox`. The `data.frame` is unmodified as returned by SQLite, meaning that all fields are returned as `characters` (even where the field types are 'date' or 'numeric'). Therefore, retrieved search results may need some post-processing with `process_ecotox_numerics()` `as_numeric_ecotox()`

The results are tagged with: a time stamp; the package version used; and the file path of the SQLite database used in the search (when applicable). These tags are added as attributes to the output table or query.

Author(s)

Pepijn de Vries

See Also

Other search-functions: `websearch_ecotox()`

Examples

```
## let's find the ids of all ecotox tests on species
## where Latin names contain either of 2 specific genus names and
## where they were exposed to the chemical benzene
if (check_ecotox_availability()) {
  search <-
    list(
      latin_name    = list(
        terms         = c("Skeletonema", "Daphnia"),
```

```
      method         = "contains"
    ),
    chemical_name = list(
      terms          = "benzene",
      method         = "exact"
    )
  )
  ## rows in result each represent a unique test id from the database
  result <- search_ecotox(search)
  query  <- search_query_ecotox(search)
  cat(query)
} else {
  print("Sorry, you need to use 'download_ecotox_data()' first in order for this to work.")
}
```

---

websearch_comptox *Search and retrieve substance information from* https://comptox.epa.gov/dashboard

---

Description

[Experimental] Search https://comptox.epa.gov/dashboard for substances and their chemico-physical properties and meta-information.

Usage

```
websearch_comptox(
  searchItems,
  identifierTypes = c("chemical_name", "CASRN", "INCHIKEY", "dtxsid"),
  inputType = c("IDENTIFIER", "DTXCID", "INCHIKEY_SKELETON", "MSREADY_FORMULA",
    "EXACT_FORMULA", "MASS"),
  downloadItems = c("DTXCID", "CASRN", "INCHIKEY", "IUPAC_NAME", "SMILES",
   "INCHI_STRING", "MS_READY_SMILES", "QSAR_READY_SMILES", "MOLECULAR_FORMULA",
  "AVERAGE_MASS", "MONOISOTOPIC_MASS", "TOXPRINTS_CHEMOTYPER", "TOXPRINTS_FINGERPRINT",
  "QC_LEVEL", "SAFETY_DATA", "EXPOCAST", "DATA_SOURCES", "TOXVAL_DATA", "IRIS_LINK",
  "PPRTV_LINK", "WIKIPEDIA_ARTICLE", "QC_NOTES", "BIOCONCENTRATION_FACTOR_TEST_PRED",
    "BOILING_POINT_DEGC_TEST_PRED", "48HR_DAPHNIA_LC50_MOL/L_TEST_PRED",
    "DENSITY_G/CM^3_TEST_PRED", "DEVTOX_TEST_PRED",

    "96HR_FATHEAD_MINNOW_MOL/L_TEST_PRED", "FLASH_POINT_DEGC_TEST_PRED",
    "MELTING_POINT_DEGC_TEST_PRED", "AMES_MUTAGENICITY_TEST_PRED",
    "ORAL_RAT_LD50_MOL/KG_TEST_PRED", "SURFACE_TENSION_DYN/CM_TEST_PRED",
    "THERMAL_CONDUCTIVITY_MW/(M*K)_TEST_PRED",
   "TETRAHYMENA_PYRIFORMIS_IGC50_MOL/L_TEST_PRED", "VISCOSITY_CP_CP_TEST_PRED",
    "VAPOR_PRESSURE_MMHG_TEST_PRED", "WATER_SOLUBILITY_MOL/L_TEST_PRED",
    "ATMOSPHERIC_HYDROXYLATION_RATE_(AOH)_CM3/MOLECULE*SEC_OPERA_PRED",
    "BIOCONCENTRATION_FACTOR_OPERA_PRED",
```

```
   "BIODEGRADATION_HALF_LIFE_DAYS_DAYS_OPERA_PRED", "BOILING_POINT_DEGC_OPERA_PRED",
    "HENRYS_LAW_ATM-M3/MOLE_OPERA_PRED", "OPERA_KM_DAYS_OPERA_PRED",
    "OCTANOL_AIR_PARTITION_COEFF_LOGKOA_OPERA_PRED",
    "SOIL_ADSORPTION_COEFFICIENT_KOC_L/KG_OPERA_PRED",
    "OCTANOL_WATER_PARTITION_LOGP_OPERA_PRED", "MELTING_POINT_DEGC_OPERA_PRED",
   "OPERA_PKAA_OPERA_PRED", "OPERA_PKAB_OPERA_PRED", "VAPOR_PRESSURE_MMHG_OPERA_PRED",
    "WATER_SOLUBILITY_MOL/L_OPERA_PRED", "LOGD5.5", "LOGD7.4", "READY_BIO_DEG",
    "EXPOCAST_MEDIAN_EXPOSURE_PREDICTION_MG/KG-BW/DAY",
     "NHANES",
    "TOXCAST_NUMBER_OF_ASSAYS/TOTAL", "TOXCAST_PERCENT_ACTIVE"),
  massError = 0,
  timeout = 300,
  verify_ssl = getOption("ECOTOXr_verify_ssl"),
  ...
)
```

Arguments

searchItems	A vector of characters where each element is a substance descriptor (any of the selected identifierTypes) you wish to query.
identifierTypes	Substance identifiers for searching CompTox. Only used when inputType is set to "IDENTIFIER".
inputType	Type of input used for searching CompTox. See usage section for valid entries.
downloadItems	Output fields of CompTox data for requested substances
massError	Error tolerance when searching for substances based on their monoisotopic mass. Only used for inputType = "MASS".
timeout	Time in seconds (default is 300 secs), that the routine will wait for the download link to get ready. It will throw an error if it takes longer than the specified timeout.
verify_ssl	When set to FALSE the SSL certificate of the host (EPA) is not verified. Can also be set as option: options(ECOTOXr_verify_ssl = TRUE). Default is TRUE.
...	Arguments passed on to httr2::req_options() requests.

Details

The CompTox Chemicals Dashboard is a freely accessible online U.S. EPA database. It contains information on physico-chemical properties, environmental fate and transport, exposure, usage, *in vivo* toxicity, and *in vitro* bioassay of a wide range of substances.

The function described here to search and retrieve records from the online database is experimental. This is because this feature is not formally supported by the EPA, and it may break in future incarnations of the online database. The function forms an interface between R and the CompTox website and is therefore limited by the restrictions documented there.

Value

Returns a named list of dplyr::tibbles containing the search results for the requested output tables and fields. Results are unpolished and 'as is' returned by EPA's web service.

Author(s)

Pepijn de Vries

References

Official US EPA CompTox website: https://comptox.epa.gov/dashboard/

Williams, A.J., Grulke, C.M., Edwards, J., McEachran, A.D., Mansouri, K, Baker, N.C., Patlewicz, G., Shah, I., Wambaugh, J.F., Judson, R.S. & Richard, A.M. (2017), The CompTox Chemistry Dashboard: a community data resource for environmental chemistry. *J Cheminform*, 9(61) doi:10.1186/s1332101702476

Examples

```
if (interactive()){
  ## search for substance name 'benzene' and CAS registration number 108-88-3
  ## on https://comptox.epa.gov/dashboard:
  comptox_results <- websearch_comptox(c("benzene", "108-88-3"))

  ## search for substances with mono-isotopic mass of 100+/-5:
  comptox_results2 <- websearch_comptox("100", inputType = "MASS", massError = 5)
}
```

---

websearch_ecotox *Search and retrieve toxicity records from the online database*

Description

[Experimental] Functions to search and retrieve records from the online database at https://cfpub.epa.gov/ecotox/search.cfm.

Usage

```
websearch_ecotox(
  fields = list_ecotox_web_fields(),
  habitat = c("aquire", "terrestrial"),
  verify_ssl = getOption("ECOTOXr_verify_ssl"),
  ...
)

list_ecotox_web_fields(...)
```

Arguments

fields	A named list of characters, used to build a search for for the online search query of https://cfpub.epa.gov/ecotox/search.cfm. Use list_ecotox_web_fields() to construct a valid list.

habitat	Use aquire (default) to retrieve aquatic data, terrestrial for, you've guessed it, terrestrial data.
verify_ssl	When set to FALSE the SSL certificate of the host (EPA) is not verified. Can also be set as option: options(ECOTOXr_verify_ssl = TRUE). Default is TRUE.
...	In case of list_ecotox_web_fields() the dots can be used as search field values used to update the returned list of fields. In case of websearch_ecotox() the dots can be used to pass custom options to the underlying httr2::req_options() call. For available field names, use names(list_ecotox_web_fields())

Details

The functions described here to search and retrieve records from the online database are experimental. This is because this feature is not formally supported by the EPA, and it may break in future iterations of the online database. The functions form an interface between R and the ECOTOX website and is therefore limited by its restrictions as described in the package documentation: ECOTOXr. The functions should therefore be used with caution.

Value

Returns named list of dplyr::tibbles with search results. Results are unpolished and 'as is' returned by EPA's web service.

list_ecotox_web_fields() returns a named list with fields that can be used in a web search of EPA's ECOTOX database, using websearch_ecotox().

Note

IMPORTANT: when you plan to perform multiple adjacent searches (for instance in a loop), please insert a call to Sys.sleep(). This to avoid overloading the server and getting your IP address banned from the server.

Author(s)

Pepijn de Vries

See Also

Other online-functions: download_ecotox_data(), get_ecotox_url()

Other search-functions: search_ecotox()

Examples

```
if (interactive()) {
  search_fields <-
    list_ecotox_web_fields(
      txAdvancedSpecEntries     = "daphnia magna",
      RBSPECSEARCHTYPE          = "EXACT",
      txAdvancedChemicalEntries = "benzene",
      RBCHEMSEARCHTYPE          = "EXACT")
```

```
  search_results <- websearch_ecotox(search_fields)
}
```

Index

∗ **cas-handlers**
  cas, 9
∗ **database-access-functions**
  check_ecotox_availability, 13
  check_ecotox_build, 14
  check_ecotox_version, 15
  cite_ecotox, 16
  dbConnectEcotox, 17
  get_ecotox_info, 19
  get_ecotox_sqlite_file, 20
  list_ecotox_fields, 23
  migrate_ecotox_path, 24
∗ **database-build-functions**
  build_ecotox_sqlite, 8
  check_ecotox_build, 14
  check_ecotox_version, 15
  download_ecotox_data, 18
  get_ecotox_url, 22
∗ **ecotox-sanitisers**
  as_date_ecotox, 2
  as_numeric_ecotox, 4
  as_unit_ecotox, 6
  mixed_to_single_unit, 25
  process_ecotox_dates, 26
  process_ecotox_numerics, 27
  process_ecotox_units, 28
∗ **online-functions**
  download_ecotox_data, 18
  get_ecotox_url, 22
  websearch_ecotox, 35
∗ **search-functions**
  search_ecotox, 30
  websearch_ecotox, 35
[.cas *(cas)*, 9
[<-.cas *(cas)*, 9
[[.cas *(cas)*, 9
[[<-.cas *(cas)*, 9

as.cas *(cas)*, 9
as.character.cas *(cas)*, 9
as.data.frame.cas *(cas)*, 9
as.double.cas *(cas)*, 9
as.integer.cas *(cas)*, 9
as.list.cas *(cas)*, 9
as_date_ecotox, 2
as_date_ecotox(), *5*, *7*, *25, 26*, *28, 29*
as_numeric_ecotox, 4
as_numeric_ecotox(), *4*, *7*, *25, 26*, *28, 29*
as_unit_ecotox, 6
as_unit_ecotox(), *4, 5*, *25, 26*, *28, 29*

base::Extract(), *11*
bibentry(), *16*
build_ecotox_sqlite, 8, *31*
build_ecotox_sqlite(), *14, 15*, *19, 20*, *22*

c.cas *(cas)*, 9
cas, 9
check_ecotox_availability, 13
check_ecotox_availability(), *14–17*, *20, 21*, *23, 24*
check_ecotox_build, 14
check_ecotox_build(), *9*, *13*, *15–17*, *19–24*
check_ecotox_version, 15
check_ecotox_version(), *9*, *13, 14*, *16, 17*, *19–24*
cite_ecotox, 16
cite_ecotox(), *13–15*, *17*, *20, 21*, *23, 24*, *32*

dbConnect(), *14*, *17*
dbConnectEcotox, 17
dbConnectEcotox(), *13–16*, *20, 21*, *23, 24*, *31*
dbDisconnect(), *14*, *17*
dbDisconnectEcotox *(dbConnectEcotox)*, 17
DBI::dbGetQuery(), *32*
download_ecotox_data, 18
download_ecotox_data(), *8, 9*, *14–16*, *19, 20*, *22*, *36*
dplyr::tibble, *34*, *36*

ECOTOXr, *36*

format.cas *(cas)*, 9

get_ecotox_info, 19
get_ecotox_info(), *13–17*, *21*, *23, 24*
get_ecotox_path
(get_ecotox_sqlite_file), 20
get_ecotox_path(), *8*, *13–18*, *20, 21*, *31*
get_ecotox_sqlite_file, 20
get_ecotox_sqlite_file(), *13–17*, *20*, *23, 24*
get_ecotox_url, 22
get_ecotox_URL(), *19*
get_ecotox_url(), *9*, *14, 15*, *19*, *36*

httr2::req_options(), *18*, *22*, *34*, *36*

is.cas *(cas)*, 9

list_ecotox_fields, 23
list_ecotox_fields(), *13–17*, *20, 21*, *24*, *30*
list_ecotox_web_fields
(websearch_ecotox), 35
list_ecotox_web_fields(), *35, 36*

migrate_ecotox_path, 24
migrate_ecotox_path(), *13–17*, *20, 21*, *23*
mixed_to_single_unit, 25
mixed_to_single_unit(), *4, 5*, *7*, *26*, *28, 29*

print.cas *(cas)*, 9
process_ecotox_dates, 26
process_ecotox_dates(), *4, 5*, *7*, *25*, *28, 29*
process_ecotox_numerics, 27
process_ecotox_numerics(), *4, 5*, *7*, *25, 26*, *29*
process_ecotox_units, 28
process_ecotox_units(), *4, 5*, *7*, *25, 26*, *28*

search_ecotox, 30
search_ecotox(), *17*, *23*, *36*
search_ecotox_lazy *(search_ecotox)*, 30
search_query_ecotox *(search_ecotox)*, 30
show.cas *(cas)*, 9
suppressMessages(), *9*, *19*
Sys.sleep(), *36*

tbl_df(), *31*
tools::R_user_dir(), *21*

websearch_comptox, 33
websearch_ecotox, 35
websearch_ecotox(), *19*, *22*, *32*, *36*