toxvaldbmain

May 27, 2025

Type Package

Title Builds the ToxValDB v9.6.2 Database

Version 9.6.2

Author Taylor Wall

Maintainer Taylor Wall <wall.taylor@epa.gov>

Description

ToxValDB is a database containing quantitative records from in vivo toxicology studies from many sources. The database has 2 main parts - toxval_source containing
source data in separate tables, and the main toxval schema which combines data from multiple sources
into a single format. This package moves data from toxval_source to toxval.
Data is read from files or other databases into toxval_source and then pulled
into toxval where terms are converted to standard values. This version is setup to build ToxValDB v9.6.2.

Imports DBI,
RMySQL,
openxlsx,
dplyr,
tidyr,
stringr,
tibble,
janitor,
XML,
miniUI,
RCurl,
gsubfn,
textclean,
magrittr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests knitr,
rmarkdown

VignetteBuilder knitr

Contents

`cas_checkSum` *cas_checkSum*

Description

Check CAS RN validity via checksum method

For a suspected CAS RN, determine validity by calculating final digit checksum

Usage

```
cas_checkSum(x)
```

Arguments

`x`	chr. Input vector of values to check. Standard CAS notation using hyphens is fine, as #' all non-digit characters are stripped for checksum calculation. Each element of *x* should contain #' only one suspected CAS RN to check.

Details

This function performs a very specific type of check for CAS validity, namely whether the final digit checksum follows the CAS standard. By default, it also ensures that the digit length is compatible with CAS standards.

Value

A `logical` vector of length *x* denoting whether each *x* is a valid CAS by the checksum method. `NA` input values will be set 0 (FAIL).

See Also

`str_detect`, `str_pad`

Examples

```
cas_good <- c("71-43-2", "18323-44-9", "7732-18-5") # benzene, clindamycin, water
cas_bad  <- c("61-43-2", "18323-40-9", "7732-18-4") # single digit change from good
cas_checkSum(c(cas_good, cas_bad))
```

`check.toxval_type.route.units` *Output distinct combinations of toxval_type, exposure_route, and toxval_units to check*

Description

Output distinct combinations of toxval_type, exposure_route, and toxval_units to check

Usage

```
check.toxval_type.route.units(
  toxval.db,
  source = NULL,
  subsource = NULL,
  load.dict = NULL
)
```

Arguments

`toxval.db`	The version of toxvaldb to use.
`source`	Source to check
`subsource`	Subsource to check (NULL default)
`load.dict`	Name of dictionary containing expected combinations, or NULL if dictionary should not be read

---

`chem.check.v2` *Check the chemicals from a file Names with special characters are cleaned and trimmed CASRN are fixed (dashes put in, trimmed) and check sums are calculated The output is sent to a file called chemcheck.xlsx in the source data file One option for using this is to edit the source file until no errors are found*

Description

Check the chemicals from a file Names with special characters are cleaned and trimmed CASRN are fixed (dashes put in, trimmed) and check sums are calculated The output is sent to a file called chemcheck.xlsx in the source data file One option for using this is to edit the source file until no errors are found

Usage

```
chem.check.v2(res0, in_source = NULL, verbose = FALSE)
```

Arguments

res0	The data frame in which chemicals names and CASRN will be replaced
in_source	The source to be processed. If source=NULL, process all sources
verbose	If TRUE, print diagnostic messages

Value

Return a list with fixed CASRN and name and flags indicating if fixes were made: res0=res0,name.OK=name.OK,casrn.OK=c

---

clean.last.character *clean.last.character*

Description

Clean unneeded characters from the end of a string

Usage

```
clean.last.character(x)
```

Arguments

x	String to be cleaned

Details

DETAILS

Value

The cleaned string

See Also

str_trim

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

clean.toxval.by.source *Delete a portion of the contents of the toxval database*

Description

Delete a portion of the contents of the toxval database

Usage

```
clean.toxval.by.source(toxval.db, source)
```

Arguments

toxval.db	The version of toxval from which the data is deleted.
source	The data source name

Value

The database will be altered

compare.versions *Compare versions of toxval*

Description

Compare versions of toxval

Usage

```
compare.versions(db1, db2)
```

Arguments

db1	The old version of the database
db2	The new version of the database

duplicate.hunter *Diagnose duplicates*

Description

Diagnose duplicates

Usage

```
duplicate.hunter(toxval.db, source = NULL, source_table = NULL)
```

Arguments

toxval.db	Database version
source	The source to be updated

Value

Write a file with the results

ecotox.select.study.duration *ECOTOX Select study_duration*

Description

Function to select the appropriate study_duration value from conc1_* fields

Usage

```
ecotox.select.study.duration(in_data, dur_col = NULL)
```

Arguments

in_data	Input ECOTOX dataframe

Value

Processed dataframe with new study_duration_values, units, and qualitifer fields

ecotox.select.toxval.numeric

ECOTOX Select toxval_numeric

Description

Function to select the appropriate toxval_numeric value from conc1_* fields

Usage

```
ecotox.select.toxval.numeric(in_data)
```

Arguments

in_data	Input ECOTOX dataframe

Value

Processed dataframe with new toxval_numeric, units, and qualitifer fields

ecotox.species.dictionary

Extract the ECOTOX species dictionary from the ECOTOX data

Description

Extract the ECOTOX species dictionary from the ECOTOX data

Usage

```
ecotox.species.dictionary(toxval.db, do.load = F, sys.date = "2023-05-03")
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
do.load	If TRUE, load the data from the input file and put into a global variable
source.db	The version of toxval source - used to manage chemicals
verbose	Whether the loaded rows should be printed to the console.
log	If TRUE, send output to a log file

`export.all.by.source` *Build a data frame of the data from toxval and export by source as a series of xlsx files*

Description

Build a data frame of the data from toxval and export by source as a series of xlsx files

Usage

```
export.all.by.source(
  toxval.db,
  source = NULL,
  subsource = NULL,
  include.qc.status = TRUE
)
```

Arguments

`toxval.db`	Database version
`source`	The source to be updated
`subsource`	The subsource to be updated
`include.qc.status`	Boolean whether to include teh qc_status field, or filter out "fail" records. Default is TRUE. #' @return for each source writes an Excel file with the name ../export/export_by_source_data/toxval_all_toxval.db_source.xlsx

`export.delete.qc_status.fail.by.source` *export.delete.qc_status.fail.by.source*

Description

Function to export log of qc_status "fail" records and delete fail records from ToxValDB.

Usage

```
export.delete.qc_status.fail.by.source(toxval.db, source, subsource)
```

Arguments

`toxval.db`	The database version to use
`source`	The source name
`subsource`	The specific subsource to process, if desired (Default: NULL)

Value

None. XLSX log genrated and SQL UPDATE statements are pushed to the database.

`export.for.missing.species` *Export the records with missing species and point of departure values*

Description

Export the records with missing species and point of departure values

Usage

```
export.for.missing.species(toxval.db, source = NULL)
```

Arguments

`toxval.db`	Database version
`source`	The source to be updated #' @return for each source writes an Excel file with the name ../export/export_by_source_data/toxval_all_toxval.db_source.xlsx

---

`export.for.oppt` *Build a data frame of the PODs and exports as xlsx*

Description

Build a data frame of the PODs and exports as xlsx

Usage

```
export.for.oppt(toxval.db, file.name = "TSCA PICS")
```

Arguments

`toxval.db`	Database version
`file.name`	If not NA, this is a file containing chemicals, and only those chemicals will be exported

Value

writes an Excel file with the name ../export/toxval_pod_summary_[human_eco]_Sys.Date().xlsx

export.for.toxval.qc *Build a data frame of the PODs and exports as xlsx*

Description

Build a data frame of the PODs and exports as xlsx

Usage

```
export.for.toxval.qc(toxval.db, source = NULL)
```

Arguments

toxval.db	Database version
human_eco	Either 'human health' or 'eco'
file.name	If not NA, this is a file containing chemicals, and only those chemicals will be exported

Value

writes an Excel file with the name ../export/toxval_pod_summary_[human_eco]_Sys.Date().xlsx

export.for.toxvaldb.manuscript *Build a data frame of the data for the toxval manuscript*

Description

Build a data frame of the data for the toxval manuscript

Usage

```
export.for.toxvaldb.manuscript(toxval.db)
```

Arguments

toxval.db	Database version
source	The source to be updated

Value

Write a file with the results

`export.input.dict.file.list` *export.input.dict.file.list*

Description

Pull list of directories, files, and dictionaries used throughout the ToxValDB workflow.

Usage

```
export.input.dict.file.list()
```

Value

Dataframe with "folder_name" field of files and folders in the toxval.config()$datapath directory.

See Also

`read_excel` `write_xlsx` `filter`, `pull`, `bind_rows`, `distinct`

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`export.missing.dictionary.entries` *Find "original" values that have not been included in the dictionaries and export them*

Description

Find "original" values that have not been included in the dictionaries and export them

Usage

```
export.missing.dictionary.entries(
  toxval.db,
  source = NULL,
  subsource = NULL,
  report.only = FALSE
)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
source	The source to be fixed. If source=NULL, fix all sources
subsource	The subsource to be fixed (NULL default)
report.only	Whether to report or write/export data. Default is FALSE (write/export data)

Value

An excel file in dictionaries with the missing entries (if report.only=TRUE, return tibble) "missing dictionary entries Sys.Date.xlsx"

---

export.missing.strains

Set the strain information in toxval

Description

Set the strain information in toxval

Usage

```
export.missing.strains(toxval.db, date_string = "2024-02-27")
```

Arguments

toxval.db	The version of the database to use
date_string	The date of the latest dictionary version

---

export.missing.toxval_type

Export any toxval_types that are not in the toxval_type dictionary

---

Description

Export any toxval_types that are not in the toxval_type dictionary

Usage

```
export.missing.toxval_type(toxval.db, report.only = FALSE)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
report.only	Whether to report or write/export data. Default is FALSE (write/export data)

Value

An excel file in dictionaries with the missing entries (if report.only=TRUE, return tibble) "dictionary/missing/missing_toxval_type Sys.Date.xlsx"

---

export.toxvaldb.subset

Build a data frame of the data from toxval for a subset of chemicals

---

Description

Build a data frame of the data from toxval for a subset of chemicals

Usage

```
export.toxvaldb.subset(toxval.db, filename)
```

Arguments

toxval.db	Database version
filename	The name of the file the be imported - should be a short name that will be used in the output filename. This is an xlsx file and needs a column labeled dtxsid

Value

Write a file with the results

---

export.update.source.info

export.update.source.info

Description

Pull updated field information for source info dictionary

Usage

```
export.update.source.info(toxval.db, source.db, dict_date = "2024-08-28")
```

Arguments

toxval.db	Database version
source.db	The source to be updated
dict_date	Date stamp of the dictionary file to update

Value

Overwrite existing source info dictionary with updated fields

fill.chemical.by.source *Fill the chemical table*

Description

Fill the chemical table

Usage

```
fill.chemical.by.source(toxval.db, source, verbose = TRUE)
```

Arguments

toxval.db	The version of toxvaldb to use.
source	The source to be used
verbose	If TRUE, print out extra diagnostic messages

fill.toxval.defaults *Set Toxval Defaults*

Description

Set Toxval Defaults

Usage

```
fill.toxval.defaults(toxval.db, mat)
```

Arguments

toxval.db	The version of toxval from which to set defaults.
mat	An input matrix of data

Value

The data matrix afer fixing

`fill.toxval.defaults.global.by.source`
Set Toxval Defaults globally, replacing blanks with -

Description

Set Toxval Defaults globally, replacing blanks with -

Usage

```
fill.toxval.defaults.global.by.source(
  toxval.db,
  source = NULL,
  subsource = NULL
)
```

Arguments

`toxval.db`	The version of toxval from which to set defaults.
`source`	The source to be fixed
`subsource`	The subsource to be fixed (NULL default)

`fix.all.param.by.source`
Alter the contents of toxval according to an excel dictionary file with fields - exposure_method, exposure_route, sex,strain, study_duration_class, study_duration_units, study_type, toxval_type, exposure_form, toxval_subtype

---

Description

Alter the contents of toxval according to an excel dictionary file with fields - exposure_method, exposure_route, sex,strain, study_duration_class, study_duration_units, study_type, toxval_type, exposure_form, toxval_subtype

Usage

```
fix.all.param.by.source(
  toxval.db,
  source = NULL,
  subsource = NULL,
  fill.toxval_fix = TRUE
)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
source	The source to be fixed. If source=NULL, fix all sources
subsource	The subsource to be fixed (NULL default)
fill.toxval_fix	If TRUE (default) read the dictionaries into the toxval_fix table

Value

The database will be altered

---

fix.associated.pod.critical_effect

Appends associated POD information to critical_effect for derived toxval_types

---

Description

Appends associated POD information to critical_effect for derived toxval_types

Usage

```
fix.associated.pod.critical_effect(res, map_fields)
```

Arguments

res	The data to be altered
map_fields	The fields used to map entries to associated PODs

---

fix.casrn

fix.casrn

Description

Fix a CASRN that has one of several problems

Usage

```
fix.casrn(casrn, cname = "", verbose = FALSE)
```

Arguments

casrn	Input CASRN to be fixed
cname	An optional chemical name
verbose	if TRUE, print the input values

Details

DETAILS

Value

the fixed CASRN

See Also

reexports

Examples

```
## Not run:
if(interactive()){
 fix.casrn("107028")
 # Expected output "107-02-8"
 }

## End(Not run)
```

`fix.critical_effect.icf.by.source`
standardize critical_effect in toxval table based on icf dictionary and toxval critical effects dictionary

---

Description

standardize critical_effect in toxval table based on icf dictionary and toxval critical effects dictionary

Usage

```
fix.critical_effect.icf.by.source(toxval.db, source, subsource = NULL)
```

Arguments

`toxval.db`	The version of toxvaldb to use.
`source`	THe source to be fixed
`subsource`	The subsource to be fixed (NULL default)

fix.dedup.hierarchy.by.source

Set qc_status as "fail" for lesser priority duplicates

Description

Set qc_status as "fail" for lesser priority duplicates

Usage

```
fix.dedup.hierarchy.by.source(
  toxval.db,
  source = NULL,
  subsource = NULL,
  priority_list = NULL,
  subsource_priority_list = NULL,
  criteria = c("dtxsid"),
  report.only = FALSE
)
```

Arguments

toxval.db	The version of toxvaldb to use.
source	Source to be fixed
subsource	Subsource to be fixed (NULL default)
priority_list	Named list describing source priority, with low priority index and high priority value
subsource_priority_list	Named list of subsources to deprecate per source, source index/subsource value
criteria	List of parameters used to make deduping decisions
report.only	Whether to report or write/export data. Default is FALSE (write/export data)

fix.derived.toxval_type.by.source

Set select normalized toxval fields to '-' if the record is a select toxval_type

Description

Set select normalized toxval fields to '-' if the record is a select toxval_type

Usage

```
fix.derived.toxval_type.by.source(
  toxval.db,
  source = NULL,
  subsource = NULL,
  report.only = FALSE
)
```

Arguments

toxval.db	The version of the database to use
source	The source to be fixed
subsource	The subsource to be fixed (NULL default)
report.only	Whether to update database or only report the results. Default FALSE.

---

fix.empty.by.source *Set all empty cells in toxval to '-'*

Description

Set all empty cells in toxval to '-'

Usage

```
fix.empty.by.source(toxval.db, source = NULL, subsource = NULL)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
source	The source to be fixed
subsource	The subsource to be fixed (NULL default)

Value

The database will be altered

`fix.empty.record_source.by.source` *Set all empty cells in record_source to '-'*

Description

Set all empty cells in record_source to '-'

Usage

```
fix.empty.record_source.by.source(toxval.db, source = NULL)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`source`	The source to be fixed

Value

The database will be altered

`fix.exposure.params` *Fix the exposure fields: exposure_method, exposure_route, exposure_form based on a 3 column dictionary ~/dictionary/exposure_route_method_form.xlsx*

Description

Fix the exposure fields: exposure_method, exposure_route, exposure_form based on a 3 column dictionary ~/dictionary/exposure_route_method_form.xlsx

Usage

```
fix.exposure.params(
  toxval.db,
  source = NULL,
  subsource = NULL,
  report.only = FALSE
)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
source	The source to be fixed. If source=NULL, fix all sources
subsource	The subsource to be fixed (NULL default)
report.only	Whether to report or write/export data. Default is FALSE (write/export data)
fill.toxval_fix	If TRUE (default) read the dictionaries into the toxval_fix table

Value

The database will be altered (if report.only=TRUE, return missing entries)

---

fix.exposure_route.not_specified.by.source

fix.exposure_route.not_specified.by.source

Description

Function to use a dictionary file to update cases where exposure_route reported as "not specified".

Usage

```
fix.exposure_route.not_specified.by.source(toxval.db, source, subsource)
```

Arguments

toxval.db	The database version to use
source	The source name
subsource	The specific subsource to process, if desired (Default: NULL)

Details

DETAILS

Value

None. SQL UPDATE statements are pushed to the database.

See Also

pull, mutate-joins, select, rename, mutate, filter, distinct read_excel write_xlsx

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`fix.generation.by.source` *Alter the contents of toxval according to an excel dictionary file with field generation*

Description

Alter the contents of toxval according to an excel dictionary file with field generation

Usage

```
fix.generation.by.source(toxval.db, source, subsource = NULL)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`source`	The source to be processed
`subsource`	The subsource to be processed (NULL default)

Value

The database will be altered

`fix.human_eco.by.source` *Fix the human_eco flag*

Description

Fix the human_eco flag

Usage

```
fix.human_eco.by.source(toxval.db, source = NULL, subsource = NULL)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
source	The source to be fixed. If NULL, fix all sources
subsource	The subsource to be fixed (NULL default)

Value

The database will be altered

---

fix.non_ascii.v2 *fix.non_ascii.v2*

Description

Flag and fix non-ascii characters in the database

Usage

```
fix.non_ascii.v2(df, source)
```

Arguments

df	The dataframe to be processed
source	Current ToxVal source
The	source to be fixed

Details

DETAILS

Value

The dataframe with non ascii characters replaced with cleaned versions

See Also

read.xlsx, write.xlsx str_trim stri_escape_unicode

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

fix.priority_id.by.source

Fix the priority_id in the toxval table based on source

Description

Fix the priority_id in the toxval table based on source

Usage

```
fix.priority_id.by.source(toxval.db, source = NULL, subsource = NULL)
```

Arguments

toxval.db	The version of toxvaldb to use.
source	The source to be fixed, If NULL, set for all sources
subsource	The subsource to be fixed (NULL default)

fix.qc_status.by.source

Fix the qa_status flag

Description

Fix the qa_status flag

Usage

```
fix.qc_status.by.source(
  toxval.db,
  source.db,
  source = NULL,
  subsource = NULL,
  reset = FALSE
)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
source	The source to be fixed
subsource	The subsource to be fixed (NULL default)
reset	If TRUE, reset all values to 'pass' before setting
sourcedb	The source database name

Value

The database will be altered

`fix.replace.unicode` *fix.replace.unicode*

Description

A function to check all character fields and handle unicode symbols, either by removing them or replacing them with alphabetic equivalents.

Usage

```
fix.replace.unicode(df)
```

Arguments

`df`	Character vector to check/replace unicode symbols.

Details

DETAILS

Value

Returns a modified version of the input vector with unicode replacements.

See Also

[stri_escape_unicode][stringi::stri_escape_unicode] [str_extract][stringr::str_extract]

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

fix.risk_assessment_class.by.source

Set the risk assessment class of toxval according to an excel dictionary. Values may beset multiple times, so the excel sheet should be ordered so that the last ones to be set are last

Description

Set the risk assessment class of toxval according to an excel dictionary. Values may beset multiple times, so the excel sheet should be ordered so that the last ones to be set are last

Usage

```
fix.risk_assessment_class.by.source(
  toxval.db,
  source = NULL,
  subsource = NULL,
  restart = TRUE,
  report.only = FALSE
)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
source	The source to be updated
subsource	The subsource to be updated (NULL default)
restart	If TRUE, delete all values and start from scratch
report.only	Whether to report or write/export data. Default is FALSE (write/export data)

fix.single.param.by.source

Alter the contents of toxval according to an excel dictionary

Description

Alter the contents of toxval according to an excel dictionary

Usage

```
fix.single.param.by.source(
  toxval.db,
  param,
  source,
  subsource = NULL,
```

```
  ignore = FALSE,
  report.only = FALSE,
  units.data = NULL
)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
param	The parameter value to be fixed
source	The source to be fixed
subsource	The subsource to be fixed (NULL default)
ignore	If TRUE allow missing values to be ignored
report.only	Whether to report or write/export data. Default is FALSE (write/export data)
units.data	A dataframe containing current units data if units are to be reported (NULL default)

Value

The database will be altered

---

fix.species.common_name

Fix issues with species common names

Description

Fix issues with species common names

Usage

```
fix.species.common_name(toxval.db)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.

fix.species.duplicates

Check for species duplicates - same common name but multiple species_ids

Description

Check for species duplicates - same common name but multiple species_ids

Usage

```
fix.species.duplicates(toxval.db, source = NULL, subsource = NULL)
```

Arguments

toxval.db	The version of the database to use
source	The source to be checked. If source=NULL, check all sources
subsource	The subsource to be checked (NULL default)

fix.species.v2

Set the species_id column in toxval

Description

This function replaces fix.species This function precedes toxvaldb.load.species

Usage

```
fix.species.v2(
  toxval.db,
  source = NULL,
  subsource = NULL,
  date_string = "2023-05-18"
)
```

Arguments

toxval.db	The version of the database to use
source	The source to be fixed
subsource	The subsource to be fixed (NULL default)
date_string	The date version of the dictionary

fix.strain.v2 *Set the strain information in toxval*

Description

Set the strain information in toxval

Usage

```
fix.strain.v2(
  toxval.db,
  source = NULL,
  subsource = NULL,
  date_string = "2024-04-08",
  reset = FALSE
)
```

Arguments

toxval.db	The version of the database to use
source	The source to be fixed. If NULL, fix for all sources
subsource	The subsource to be fixed (NULL default)
date_string	The date of the latest dictionary version

fix.study_duration.params *Fix the study duration fields: study_duration_value, study_duration_units, study_duration_class based on a 3 column dictionary ~/dictionary/exposure_route_method_form.xlsx*

Description

Fix the study duration fields: study_duration_value, study_duration_units, study_duration_class based on a 3 column dictionary ~/dictionary/exposure_route_method_form.xlsx

Usage

```
fix.study_duration.params(
  toxval.db,
  source = NULL,
  subsource = NULL,
  report.only = FALSE
)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
source	The source to be fixed. If source=NULL, fix all sources
subsource	The subsource to be fixed (NULL default)
report.only	Whether to report or write/export data. Default is FALSE (write/export data)
fill.toxval_fix	If TRUE (default) read the dictionaries into the toxval_fix table

Value

The database will be altered

---

fix.study_group *Set the study_group field*

Description

Set the study_group field

Usage

```
fix.study_group(
  toxval.db,
  source = NULL,
  subsource = NULL,
  report.only = FALSE
)
```

Arguments

toxval.db	Database version
source	The source to be updated
subsource	The subsource to be updated (NULL default)
report.only	Whether to apply study_group fix or just report what fixes would be applied.

Value

for each source writes an Excel file with the name ../export/export_by_source_data/toxval_all_toxval.db_source.xlsx

`fix.study_type.by.source` *Fix the study_type using manual curation on a source-by-source basis*

Description

This function replaces the original export.for_study_type and fix.study_type.manual, with the intention of making it easier to fix the study types on a source-by-source basis All of the work will happen in the directory ~/Repo/dictionary/study_type_by_source. Each source will have its own file and will not have a date attached to make maintenance easier. To start the process, run this with mode="export". This will write a source-specific file to the export_temp directory. Open either the xlsx or csv (if the xslx is corrupted) and place this file into the main directory (study_type_by_source) and edit it there as documented in the main documentation. Next run this function with mode="import". This will load your changes into the database. It is suggested that before working on a new source that the old version in the study_type_by_source get pushed to the old_versions directory

Usage

```
fix.study_type.by.source(
  toxval.db,
  mode = "export",
  source = NULL,
  subsource = NULL,
  custom.query.filter = NULL,
  report.only = FALSE
)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`mode`	Either export or import
`source`	The source you want to work on. If NULL, this will run all sources
`subsource`	The subsource to be fixed
`custom.query.filter`	Additional filters for the query. Example: custom.query.filter = paste0(" and b.human_eco='human health' and ", "e.toxval_type_supercategory in ('Point of Departure','Lethality Effect Level','Toxicity Value')")
`report.only`	Whether to report or write/export data. Default is FALSE (write/export data)

Value

The database will be altered

fix.trim_spaces *Trim leading and trailing blanks from all character columns*

Description

Trim leading and trailing blanks from all character columns

Usage

```
fix.trim_spaces(res)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
source	The source to be fixed. If source=NULL, fix all sources
fill.toxval_fix	If TRUE (default) read the dictionaries into the toxval_fix table

Value

The database will be altered

fix.units.by.source *Do all of the fixes to units*

Description

1. All of these steps operate on the toxval_units column.
2. Replace variant unit names with standard ones, running fix.single.param.new.by.source.R This fixes issues like variant names for mg/kg-day and uses the dictionary file dictionary/toxval_units_5.xlsx
3. Fix special characters in toxval_units
4. Fix issues with units containing extra characters for some ECOTOX records
5. Convert units that are multiples of standard ones (e.g. ppb to ppm). This uses the dictionary file dictionary/toxval_units conversions 2018-09-12.xlsx
6. Run conversions from molar to mg units, using MW. This uses the dictionary file dictionary/MW conversions.xlsx
7. Convert ppm to mg/m3 for inhalation studies. This uses the conversion Concentration (mg/m3) = 0.0409 x concentration (ppm) x molecular weight. See https://cfpub.epa.gov/ncer_abstracts/index.cfm/fuseaction/disp This function requires htat the DSSTox external chemical_id be set
8. Convert ppm to mg/kg-day in toxval according to a species-specific conversion factor for oral exposures. This uses the dictionary file dictionary/ppm to mgkgday by animal.xlsx See: www10.plala.or.jp/biostatistics/1-3.doc This probbaly assumes feed rather than water
9. Make sure that eco studies are in mg/L and human health in mg/m3

Usage

```
fix.units.by.source(
  toxval.db,
  source = NULL,
  subsource = NULL,
  do.convert.units = FALSE,
  report.only = FALSE,
  report.extra = FALSE
)
```

Arguments

toxval.db	The version of toxvaldb to use.
source	Source to be fixed
subsource	Subsource to be fixed (NULL default)
do.convert.units	If TRUE, so unit conversions, as opposed to just cleaning
report.only	Whether to report or write/export data. Default is FALSE (write/export data)
report.extra	If reporting, then choose whether to record extra conversion information (e.g. toxval_type, mw, species_id, etc.)

---

generate.originals *Duplicate any columns with '_original' Set Toxval Defaults*

Description

Duplicate any columns with '_original' Set Toxval Defaults

Usage

```
generate.originals(toxval.db, mat)
```

Arguments

toxval.db	The version of toxval from which to set defaults.
mat	THe matrix of data to be altered

Value

The altered input matrix

getDBConn *getDBConn*

Description

Get the names the database server, user, and pass or returns error message

Usage

```
getDBConn()
```

Details

DETAILS

Value

print the database connection information

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

import.dictionary *import the toxval and toxval_type dictionaries*

Description

import the toxval and toxval_type dictionaries

Usage

```
import.dictionary(toxval.db)
```

Arguments

toxval.db	The name of the database

`import.source.info.by.source` *Load Source Info for each source into toxval The information is in the file ~/dictionary/source_info 2023-11-30.xlsx*

Description

Load Source Info for each source into toxval The information is in the file ~/dictionary/source_info 2023-11-30.xlsx

Usage

```
import.source.info.by.source(
  toxval.db,
  source = NULL,
  dict_date = "2025-05-16"
)
```

Arguments

`toxval.db`	The version of toxval into which the source info is loaded.
`source`	The specific source to be loaded, If NULL, load for all sources
`dict_date`	Date stamp of the dictionary file to update

`load.dsstox` *Load DSSTox if needed from a file into a global variables (DSSTOX)*

Description

Load DSSTox if needed from a file into a global variables (DSSTOX)

Usage

```
load.dsstox(reprocess = FALSE)
```

log_message *Function to combine output log with output message*

Description

Function to combine output log with output message

Function to combine output log with output message

Usage

```
log_message(log_df, message_df_col)

log_message(log_df, message_df_col)
```

Arguments

log_df	Dataframe to which the log information will be appended
message_df_col	New message to add

mv_orchestrate *Orchestrate Materialized View(s) Creation*

Description

Function to create a new materialized view database table by querying an input database version based on an input field_dictionary.xlsx file.

Usage

```
mv_orchestrate(toxval.db, include.qc.status = FALSE)
```

Arguments

toxval.db	Version of the database to store the new view.
include.qc.status	Boolean whether to include the qc_status field, or filter out "fail" records. Default is FALSE.

Details

DETAILS

Value

None. SQL statements are executed to generate a database table.

See Also

read_excel unite rename, mutate, case_when, select, mutate-joins, filter, bind_rows character(0), MySQLDriver-class

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

---

printCurrentFunction *printCurrentFunction*

Description

Print the name of the current function

Usage

```
printCurrentFunction(comment.string = NA)
```

Arguments

comment.string	An optional string to be printed

Details

DETAILS

Value

None

See Also

flush.console

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

pull_jira_info *pull_jira_info*

Description

Script to process CSV export of Jira into a status log

Usage

```
pull_jira_info(
  jira_project = "TOXVAL",
  in_file = NULL,
  auth_token = NULL,
  ticket_filter_list = NULL
)
```

Arguments

jira_project	Jira project code (e.g. CVTDB)
in_file	File path to Jira ticket summary CSV.
auth_token	Authorization token for Jira
ticket_filter_list	List of Jira tickets to filter to

Details

DETAILS

Value

Summary DataFrame of Jira tickets by Epic, Label, and Status

See Also

[download.file][utils::download.file], [unzip][utils::unzip] [read_csv][readr::read_csv], [cols][readr::cols] [select][dplyr::select], [contains][dplyr::contains], [mutate][dplyr::mutate], [everything][dplyr::everything], [filter][dplyr::filter], [distinct][dplyr::distinct], [left_join][dplyr::left_join], [group_by][dplyr::group_by], [summarise][dplyr::summarise], [n][dplyr::n] [unite][tidyr::unite] [str_squish][stringr::str_squish]

Examples

```
## Not run:
if(interactive()){
 out = pull_jira_info(jira_project="project_name")
 }

## End(Not run)
```

qa_toxval_url_validation *qa_toxval_url_validation*

Description

Function to pull URLs and log their HTTP statuses

Usage

```
qa_toxval_url_validation(
  tbl_list = c("toxval", "record_source", "source_info"),
  db,
  log_suffix
)
```

Arguments

tbl_list	List of ToxVal tables to pull URLs from, Default: c("toxval", "record_source", "source_info")
db	ToxVal database name to pull URLs from
log_suffix	SUffix to add to end of log file to uniquely identify file, Default: Date stamp from Sys.Date()

Details

DETAILS

Value

None. Log file is generated

See Also

read_excel filter, mutate, reexports, bind pivot_longer keep GET, status_code write_xlsx

Examples

```
## Not run:
if(interactive()){
 qa_toxval_url_validation(tbl_list = c("toxval", "record_source", "source_info"), db="res_toxval_v94", log_suffi
 }

## End(Not run)
```

runInsert *runInsert*

Description

Insert a record into a database. if auto.increment=TRUE, return the auto incremented primary key of the record. otherwise, return -1

Usage

```
runInsert(query, db, do.halt = F, verbose = F, auto.increment.id = F)
```

Arguments

query	a properly formatted SQL query as a string
db	the name of the database
do.halt	if TRUE, halt on errors or warnings
verbose	if TRUE, print diagnostic information
auto.increment.id	PARAM_DESCRIPTION, Default: F
auto.increment	if TRUE, add the auto increment primary key even if not part of the query

Details

DETAILS

Value

Returns the database table auto incremented primary key ID

See Also

character(0), MySQLDriver-class

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

runInsertTable *runInsertTable*

Description

Inserts multiple rows into a database table

Usage

```
runInsertTable(mat, table, db, do.halt = TRUE, verbose = FALSE, get.id = TRUE)
```

Arguments

mat	data frame containing the data, with the column names corresponding
table	name of the database table to which data will be inserted
db	the name of the database
do.halt	if TRUE, halt on errors or warnings
verbose	if TRUE, print diagnostic information
get.id	Whether to return ID or not, Default: T

Details

DETAILS

Value

ID or None

See Also

character(0), MySQLDriver-class

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

runQuery *runQuery*

Description

Runs a database query and returns a result set

Usage

```
runQuery(query = NULL, db, do.halt = TRUE, verbose = FALSE)
```

Arguments

query	a properly formatted SQL query as a string
db	the name of the database
do.halt	if TRUE, halt on errors or warnings
verbose	if TRUE, print diagnostic information

Details

DETAILS

Value

Query results

See Also

character(0), MySQLDriver-class flush.console

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

runStatement *runStatement*

Description

Run a SQL statement, such as an ALTER or UPDATE

Usage

```
runStatement(query, db, do.halt = FALSE, verbose = FALSE)
```

Arguments

query	a properly formatted SQL query as a string
db	the name of the database
do.halt	if TRUE, halt on errors or warnings
verbose	if TRUE, print diagnostic information

Details

DETAILS

Value

None. SQL statement is run.

See Also

character(0), MySQLDriver-class

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

runUpdate *runUpdate*

Description

Runs a database query and returns a result set

Usage

```
runUpdate(
  table,
  updateQuery = NULL,
  updated_df = NULL,
  db,
  do.halt = TRUE,
  verbose = FALSE,
  trigger_check = TRUE
)
```

Arguments

table	table to update
updateQuery	a properly formatted SQL query as a string in the form of an UPDATE INNER JOIN
updated_df	a dataframe of updated data to temporarily write to database for INNER JOIN
db	the name of the database
do.halt	if TRUE, halt on errors or warnings
verbose	if TRUE, print diagnostic information
trigger_check	if FALSE, audit triggers are ignored/bypassed

Details

DETAILS

Value

None

See Also

character(0), MySQLDriver-class dbSendStatement

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`set.critical_effect_category`
Sets the final category for each term/study_type pair in the critical_effect_terms table

---

Description

Sets the final category for each term/study_type pair in the critical_effect_terms table

Usage

```
set.critical_effect_category(toxval.db)
```

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.

Value

None. SQL update statement is performed

`set.experimental_record.by.source`
Sets experimental_record flag by source for records in toxval

---

Description

Sets experimental_record flag by source for records in toxval

Usage

```
set.experimental_record.by.source(toxval.db, source = NULL)
```

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`source`	Name of source to set. Default NULL means set experimental record for all sources

Value

None. SQL update statement is performed

`set.initial.qc_status` *Set toxval qc_status to source table qc_status*

Description

Set toxval qc_status to source table qc_status

Usage

```
set.initial.qc_status(toxval.db, source.db, source, subsource = NULL)
```

Arguments

`toxval.db`	The database version to use
`source.db`	The source database name
`source`	The source name
`subsource`	The subsource to update, if desired (Default: NULL)

`set.qc.category.by.source`
Function for assigning QC Categories to sources in toxval via the qc_category field

---

Description

Function for assigning QC Categories to sources in toxval via the qc_category field

Usage

```
set.qc.category.by.source(
  toxval.db,
  source.db,
  source = NULL,
  confluence_url = "https://confluence.epa.gov/x/VuCkFg",
  confluence_access_token,
  jira_access_token,
  jira_rdata = "qc_category"
)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The source database to use.
source	The source to set a qc_category for
confluence_url	URL to QC tracking Confluence page
confluence_access_token	A personal access token for authentication in Confluence
jira_access_token	A personal access token for authentication in Jira
jira_rdata	Local RData file of Jira data to use for jira_tickets object

---

set.study_type.by.study_group

Normalize study_type within study_group

Description

Normalize study_type within study_group

Usage

```
set.study_type.by.study_group(
  toxval.db,
  source = NULL,
  subsource = NULL,
  report.only = TRUE,
  filter.matching = FALSE
)
```

Arguments

toxval.db	The version of toxvaldb to use.
source	Source to check
subsource	Subsource to check (NULL default)
report.only	Whether to alter ToxVal (FALSE) or simply record suggestions (TRUE), default TRUE
filter.matching	Whether to ignore entries whose current study_type matches suggestion, default FALSE

set.supersource.export.names

Populate export_source_name and supersource fields in ToxVal

Description

Populate export_source_name and supersource fields in ToxVal

Usage

```
set.supersource.export.names(
  toxval.db,
  source = NULL,
  version_date = "2025-05-16"
)
```

Arguments

toxval.db	The version of toxval from which to set defaults.
source	The source to be fixed (if NULL then edit all sources)
version_date	The date of the source_info dictionary to be used

set_extraction_doc_clowder_id

Inserts clowder document information into record_source table

Description

Inserts clowder document information into record_source table

Usage

```
set_extraction_doc_clowder_id(toxval.db, source.db, source = NULL)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The source database to use.
source	Name of source to set. Default NULL means set Clowder ID for all sources

Value

None. SQL insert statement is performed

set_toxval_relationship_by_toxval_type

Generic function for setting record relationships based on standardized rules

Description

Generic function for setting record relationships based on standardized rules

Usage

```
set_toxval_relationship_by_toxval_type(res, toxval.db)
```

Arguments

res	The data that has relationships to be represented
toxval.db	The version of toxval into which the tables are loaded.

source_chemical.chemidplus

special process to deal with source chemicals for ChemIDPlus

Description

special process to deal with source chemicals for ChemIDPlus

Usage

```
source_chemical.chemidplus(
  toxval.db,
  source.db,
  res,
  source,
  chem.check.halt = FALSE,
  casrn.col = "casrn",
  name.col = "name",
  verbose = F
)
```

Arguments

toxval.db	The version of toxval into which the source info is loaded.
source.db	The source database version
source	The xource to be processed (ECOTOX)
chem.check.halt	If TRUE, halt if there are errors in the chemical checking
casrn.col	Name of the column containing the CASRN
name.col	Name of the column containing chemical names
verbose	If TRUE, output extra diagnostics information

---

source_chemical.ecotox

special process to deal with source chemicals for ECOTOX

Description

special process to deal with source chemicals for ECOTOX

Usage

```
source_chemical.ecotox(
  toxval.db,
  source.db,
  res,
  source,
  chem.check.halt = FALSE,
  casrn.col = "casrn",
  name.col = "name",
  verbose = F
)
```

Arguments

toxval.db	The version of toxval into which the source info is loaded.
source.db	The source database version
source	The xource to be processed (ECOTOX)
chem.check.halt	If TRUE, halt if there are errors in the chemical checking
casrn.col	Name of the column containing the CASRN
name.col	Name of the column containing chemical names
verbose	If TRUE, output extra diagnostics information

source_chemical.extra *special process to deal with source chemicals for extra source (cancer, genetox, skin_eye, etc)*

Description

special process to deal with source chemicals for extra source (cancer, genetox, skin_eye, etc)

Usage

```
source_chemical.extra(
  toxval.db,
  source.db,
  res,
  source,
  chem.check.halt = FALSE,
  casrn.col = "casrn",
  name.col = "name",
  verbose = FALSE
)
```

Arguments

toxval.db	The version of toxval into which the source info is loaded.
source.db	The source database version
source	The source to be processed #' @param chem.check.halt If TRUE, halt if there are errors in the chemical checking
casrn.col	Name of the column containing the CASRN
name.col	Name of the column containing chemical names
verbose	If TRUE, output extra diagnostics information

source_chemical.toxrefdb *Special process to deal with source chemicals for ToxRefDB. This will put the chemicals into the source database source_chemical table*

Description

Special process to deal with source chemicals for ToxRefDB. This will put the chemicals into the source database source_chemical table

Usage

```
source_chemical.toxrefdb(
  toxval.db,
  source.db,
  res,
  source = "ToxRefDB",
  chem.check.halt = FALSE,
  casrn.col = "casrn",
  name.col = "name",
  verbose = FALSE
)
```

Arguments

toxval.db	The version of toxval into which the source info is loaded.
source.db	The source database version
res	The dataframe to which the chemical_id sill be added
source	The name of the source
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping
casrn.col	The name of the column containing the CASRN
name.col	The name ofhte column containing hte chemical name
verbose	If TRUE, write out diagnostic messages

Value

Returns the input dataframe with the chemical_id added

---

source_hash_vectorized

source_hash_vectorized

Description

Generate the hash key for a source table based on hashing columns

Usage

```
source_hash_vectorized(res, hashing_cols)
```

Arguments

res	The data frame to be processed
hashing_cols	Optional list of columns to use for generating source_hash

Details

DETAILS

Value

Input dataframe with new source_hash field

See Also

digest distinct

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

---

species.qc *Run some checks on the species information*

Description

Run some checks on the species information

Usage

```
species.qc(toxval.db)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.

---

species.strain.mismap *Find species and strain mismaps*

Description

Find species and strain mismaps

Usage

```
species.strain.mismap(toxval.db)
```

Arguments

toxval.db	Database version

Value

Write a file with the results

---

toxval.config *toxval.config*

Description

Define a set of global variables. These include the source path (datapath) and the source databases (e.g. dev_toxval_version and dev_toxval_source_version).

Usage

```
toxval.config()
```

Details

DETAILS

Value

Returns a set of parameters to be used throughout the package

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`toxval.init.db` *Initialize the database. THis sill load the species, info and dictionary tables*

Description

Initialize the database. THis sill load the species, info and dictionary tables

Usage

```
toxval.init.db(toxval.db, version_info, reset = F, date_string = "2022-05-25")
```

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`version_info`	Info string describing the version.
`reset`	If TRUE, delete all content from the database
`date_string`	The date of the dictionary versions

`toxval.load.alaska_dec` *Load the alaska_dec (old ACToR - flex)data from toxval sourcedb to toxval*

Description

Load the alaska_dec (old ACToR - flex)data from toxval sourcedb to toxval

Usage

```
toxval.load.alaska_dec(toxval.db, source.db, log = F)
```

Arguments

`toxval.db`	The database version to use
`source.db`	The source database
`log`	If TRUE, output log inoformation to a file

toxval.load.all	*Load and process all information into ToxValDB. The entire process can be run with one command: toxval.load.all(toxval.db=...,source.db=..., do.all=TRUE) It can also be run in stages, but needs to be run in the order of the do.X parameters listed here. If any earlier step is run, all of the subsequent steps need to be rerun.*

Description

Load and process all information into ToxValDB. The entire process can be run with one command: toxval.load.all(toxval.db=...,source.db=..., do.all=TRUE) It can also be run in stages, but needs to be run in the order of the do.X parameters listed here. If any earlier step is run, all of the subsequent steps need to be rerun.

Usage

```
toxval.load.all(
  toxval.db,
  source.db,
  confluence_access_token,
  jira_access_token,
  log = FALSE,
  do.init = FALSE,
  do.reset = FALSE,
  do.load = FALSE,
  do.post = FALSE,
  do.extra = FALSE
)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The version of toxval_source database from which information is pulled.
confluence_access_token	A personal access token for authentication in Confluence
jira_access_token	A personal access token for authentication in Jira
log	If TRUE write the output from each load script to a log file
do.init	If True, clean out all of the database tables
do.reset	If TRUE, empty the database to restart
do.load	If TRUE, load all of the source
do.post	If TRUE, do th post-processingwork of fixing study type and risk_assessment_class
do.extra	If TRUE, load the non-toxval data (genetox, bcfbaf, skin/eye)

Value

Nothing is returned

`toxval.load.atsdr.pfas.2021`
Load data ATSDR PFAS 2021 data from toxval_source to toxval

Description

Load data ATSDR PFAS 2021 data from toxval_source to toxval

Usage

```
toxval.load.atsdr.pfas.2021(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

`toxval.db`	The database version to use
`source.db`	The source database
`log`	If TRUE, send output to a log file
`remove_null_dtxsid`	If TRUE, delete source records without curated DTXSID value

---

`toxval.load.atsdr_mrls`
Load the ATSDR MRLs data from toxval_source to toxval

Description

Load the ATSDR MRLs data from toxval_source to toxval

Usage

```
toxval.load.atsdr_mrls(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The source database to use.
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.caloehha *Load new_caloehha from toxval_source to toxval*

Description

Load new_caloehha from toxval_source to toxval

Usage

```
toxval.load.caloehha(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The source database to use.
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.cancer *prepare the cancer call data. The data comes form a series of files ../NIOSH/NIOSH_CARC_2018.xlsx ../IRIS/iris_cancer_call_2018-10-03.xlsx ../PPRTV_ORNL/PPRTV_ORNL cancer calls 2018-10-25.xlsx ../cancer_summary/cancer/NTP/NTP cancer clean.xlsx ../cancer_summary/cancer/IARC/IARC cancer 2018-10-29.xlsx ../cancer_summary/cancer/HealthCanada/HealthCanada_TRVs_2010_AppendixA v2.xlsx ../cancer_summary/cancer/EPA_OPP_CARC/EPA_CARC.xlsx ../cancer_summary/cancer/CalEPA/calepa_p65_cancer_only.xlsx*

Description

extract all of the chemicals with cancer slope factor or unit risk with appropriate units

Usage

```
toxval.load.cancer(toxval.db, source.db)
```

Arguments

toxval.db	The version of the database to use

---

toxval.load.chemidplus

Load ChemID Plus Acute data to toxval

Description

Load ChemID Plus Acute data to toxval

Usage

```
toxval.load.chemidplus(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE,
  do.init = FALSE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value
do.init	if TRUE, read the data in from the file and set up the matrix

toxval.load.copper *Load Copper Manufacturers data to toxval from toxval_source*

Description

Load Copper Manufacturers data to toxval from toxval_source

Usage

```
toxval.load.copper(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.dedup *toxval.load.dedup*

Description

Perform deduping on data before it is loaded to toxval

Usage

```
toxval.load.dedup(
  res,
  dedup_fields = NULL,
  hashing_cols = toxval.config()$hashing_cols,
  delim = " |::| "
)
```

Arguments

res	dataframe containing the source data to dedup
dedup_fields	vector containing field names to dedup, Default: NULL (all fields but hashing cols)
hashing_cols	vector containing field names of hashing columns, Default: toxval.config()$hashing_cols
delim	string used to separate collapsed values, Default: ' |::| '

Details

DETAILS

Value

dataframe containing deduped source data

See Also

select, group_by, summarise, context, filter, mutate, across, reexports, na_if, distinct

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`toxval.load.dod` *Load the DOD data from toxval_source to toxval*

Description

Load the DOD data from toxval_source to toxval

Usage

```
toxval.load.dod(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`source.db`	The source database to use.
`log`	If TRUE, send output to a log file

toxval.load.doe.benchmarks *Load DOE Wildlife Benchmarks data from toxval_source to toxval*

Description

Load DOE Wildlife Benchmarks data from toxval_source to toxval

Usage

```
toxval.load.doe.benchmarks(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.doe.pac *Load DOE Protective Action Criteria data from toxval_source to toxval*

Description

Load DOE Protective Action Criteria data from toxval_source to toxval

Usage

```
toxval.load.doe.pac(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The source database to use.
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.echa_iuclid

Loading the ECHA IUCLID data to toxval from toxval_source This method is different from most because there are multiple tables (one per study type) for this source

---

Description

Loading the ECHA IUCLID data to toxval from toxval_source This method is different from most because there are multiple tables (one per study type) for this source

Usage

```
toxval.load.echa_iuclid(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.ecotox *Load ECOTOX from the datahub to toxval*

Description

Load ECOTOX from the datahub to toxval

Usage

```
toxval.load.ecotox(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE,
  sys.date = "2024-09-19"
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value
sys.date	The version of the data to be used

toxval.load.efsa *Load EFSA data from toxval_source to toxval*

Description

Load EFSA data from toxval_source to toxval

Usage

```
toxval.load.efsa(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The source database to use.
log	If TRUE, send output to a log file

toxval.load.epa_aegl *Load the EPA AEGL data from toxval_source to toxval*

Description

Load the EPA AEGL data from toxval_source to toxval

Usage

```
toxval.load.epa_aegl(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.epa_dcap *Load EPA DCAP source from toxval_source to toxval*

Description

Load EPA DCAP source from toxval_source to toxval

Usage

```
toxval.load.epa_dcap(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.epa_etap *Load EPA ETAP source from toxval_source to toxval*

Description

Load EPA ETAP source from toxval_source to toxval

Usage

```
toxval.load.epa_etap(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.epa_hhtv *Load EPA HHTV data to toxval from toxval_source*

Description

Load EPA HHTV data to toxval from toxval_source

Usage

```
toxval.load.epa_hhtv(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.epa_ow_npdwr

Loading EPA OW NPDWR to toxval from toxval_source

Description

Loading EPA OW NPDWR to toxval from toxval_source

Usage

```
toxval.load.epa_ow_npdwr(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.epa_ow_nrwqc_hhc

Loading EPA OW NRWQC-HHC to toxval from toxval_source

Description

Loading EPA OW NRWQC-HHC to toxval from toxval_source

Usage

```
toxval.load.epa_ow_nrwqc_hhc(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.generic *Generic structure for loading to toxval from toxval_source*

Description

Generic structure for loading to toxval from toxval_source

Usage

```
toxval.load.generic(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

`toxval.load.genetox.all`

Load the Genetox data from Grace

Description

Load the Genetox data from Grace

Usage

```
toxval.load.genetox.all(
  toxval.db,
  source.db,
  sys.date = "2021-09-10",
  verbose = FALSE
)
```

Arguments

`toxval.db`	The database to use.
`source.db`	The source database to use.
`sys.date`	The input file version
`verbose`	If TRUE output debug information

`toxval.load.gestis.dnel`

Load GESTIS DNEL data from toxval_source to toxval

Description

Load GESTIS DNEL data from toxval_source to toxval

Usage

```
toxval.load.gestis.dnel(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.hawc *Load HAWC from toxval_source to toxval*

Description

Load HAWC from toxval_source to toxval

Usage

```
toxval.load.hawc(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.hawc_pfas_150 *Load HAWC PFAS 150 from toxval_source to toxval*

Description

Load HAWC PFAS 150 from toxval_source to toxval

Usage

```
toxval.load.hawc_pfas_150(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The version of toxval_source from which the tables are loaded.
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.hawc_pfas_430

Load HAWC PFAS 430 from toxval_source to toxval

Description

Load HAWC PFAS 430 from toxval_source to toxval

Usage

```
toxval.load.hawc_pfas_430(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The version of toxval_source from which the tables are loaded.
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.healthcanada

Load Health Canada data from toxval_source to toxval

Description

Generic structure for laoding to toxval from toxval_source

Usage

```
toxval.load.healthcanada(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.heast *Load the HEAST data from toxval_source to toxval*

Description

Load the HEAST data from toxval_source to toxval

Usage

```
toxval.load.heast(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.hess *Load HESS data to ToxVal*

Description

Load HESS data to ToxVal

Usage

```
toxval.load.hess(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.hpvis *Load HPVIS from toxval_source to toxval*

Description

Load HPVIS from toxval_source to toxval

Usage

```
toxval.load.hpvis(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.initial *Delete the contents of the toxval database*

Description

Delete the contents of the toxval database

Usage

```
toxval.load.initial(toxval.db)
```

Arguments

toxval.db	The version of toxval from which the data is deleted.

toxval.load.iris *Load IRIS source from toxval_source to toxval*

Description

Load IRIS source from toxval_source to toxval

Usage

```
toxval.load.iris(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The source database to use.
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.mass_mmcl *Load the Mass. Drinking Water Standards data from toxval_source to toxval*

Description

Load the Mass. Drinking Water Standards data from toxval_source to toxval

Usage

```
toxval.load.mass_mmcl(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.niosh *Load NIOSH from toxval_source to toxval*

Description

Load NIOSH from toxval_source to toxval

Usage

```
toxval.load.niosh(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.ntp.pfas *Load NTP PFAS data from toxval_source to toxval Generic structure for laoding to toxval from toxval_source*

Description

Load NTP PFAS data from toxval_source to toxval Generic structure for laoding to toxval from toxval_source

Usage

```
toxval.load.ntp.pfas(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.opp *Load EPA OPP data to toxval from toxval_source*

Description

Load EPA OPP data to toxval from toxval_source

Usage

```
toxval.load.opp(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

`toxval.load.osha_air_limits`
Load the osha_air_limits (old ACToR - flex)data from toxval sourcedb to toxval

---

Description

Load the osha_air_limits (old ACToR - flex)data from toxval sourcedb to toxval

Usage

```
toxval.load.osha_air_limits(toxval.db, source.db, log = F)
```

Arguments

`toxval.db`	The database version to use
`source.db`	The source database
`log`	If TRUE, send output to a log file

`toxval.load.ow_dwsha` *Load the ow_dwsha data from toxval_source to toxval*

Description

Load the ow_dwsha data from toxval_source to toxval

Usage

```
toxval.load.ow_dwsha(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

`toxval.db`	The database version to use
`source.db`	The source database
`log`	If TRUE, send output to a log file
`remove_null_dtxsid`	If TRUE, delete source records without curated DTXSID value

toxval.load.penn *Load Penn data from toxval_source to toxval*

Description

Load Penn data from toxval_source to toxval

Usage

```
toxval.load.penn(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The source database to use.
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.penn_dep_mscs *Load the penn_dep_mscs data from toxval_source to toxval*

Description

Load the penn_dep_mscs data from toxval_source to toxval

Usage

```
toxval.load.penn_dep_mscs(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.pfas_150_sem_v2

Load pfas_150_sem from toxval_source to toxval

Description

Load pfas_150_sem from toxval_source to toxval

Usage

```
toxval.load.pfas_150_sem_v2(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.postprocess

Do all of the post-processing steps for a source

Description

Do all of the post-processing steps for a source

Usage

```
toxval.load.postprocess(
  toxval.db,
  source.db,
  source,
  do.convert.units = FALSE,
  chem_source = NULL,
  subsource = NULL,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database name
source	The source name
do.convert.units	If TRUE, convert units, mainly from ppm to mg/kg-day. This code is not debugged
chem_source	Used only for source=ECHA IUCLID
subsource	The specific subsource to process, if desired (Default: NULL)
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.load.pprtv.cphea

Load PPRTV CPHEA from toxval_source to toxval

Description

Load PPRTV CPHEA from toxval_source to toxval

Usage

```
toxval.load.pprtv.cphea(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.rsl — *Load the RSL data from source db to toxval - the source database needs to be updated periodically*

Description

Load the RSL data from source db to toxval - the source database needs to be updated periodically

Usage

```
toxval.load.rsl(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.skin.eye — *Load the Skin eye data*

Description

Load the Skin eye data

Usage

```
toxval.load.skin.eye(toxval.db, source.db, verbose = FALSE)
```

Arguments

toxval.db	Database version
verbose	if TRUE, print diagnostic messages along the way

toxval.load.source_chemical *Perform the DSSTox mapping*

Description

Perform the DSSTox mapping

Usage

```
toxval.load.source_chemical(
  toxval.db,
  source.db,
  source = NULL,
  verbose = TRUE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The version of toxvaldb to use.
source.db	The source database version
source	The source to update for
verbose	If TRUE, print out extra diagnostic messages
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.species *Load the species table*

Description

Load the species table

Usage

```
toxval.load.species(toxval.db, date_string = "2023-05-18")
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
date_string	The date string for the dictionary files

toxval.load.test *Load TEST data from toxval_source to toxval*

Description

Load TEST data from toxval_source to toxval

Usage

```
toxval.load.test(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.toxrefdb2.1 *Load ToxRefdb data to toxval*

Description

Load ToxRefdb data to toxval

Usage

```
toxval.load.toxrefdb2.1(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.usgs_hbsl *Load the usgs_hbsl (old ACToR - flex)data from toxval source db to toxval*

Description

Load the usgs_hbsl (old ACToR - flex)data from toxval source db to toxval

Usage

```
toxval.load.usgs_hbsl(toxval.db, source.db, log = F)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file

toxval.load.ut_hb *Load the uterotophic and Hershberger data*

Description

Load the uterotophic and Hershberger data

Usage

```
toxval.load.ut_hb(toxval.db, source.db, log = FALSE, remove_null_dtxsid = TRUE)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.who_ipcs *Load the who_ipcs data from toxval source db to toxval*

Description

Load the who_ipcs data from toxval source db to toxval

Usage

```
toxval.load.who_ipcs(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

toxval.load.who_jecfa_adi *Load WHO JECFA ADI from toxval_source to toxval*

Description

Load WHO JECFA ADI from toxval_source to toxval

Usage

```
toxval.load.who_jecfa_adi(
  toxvaldb,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value
toxval.db	The database version to use

---

toxval.load.who_jecfa_tox_studies *Load WHO JECFA Tox Studies from toxval_source to toxval*

Description

Load WHO JECFA Tox Studies from toxval_source to toxval

Usage

```
toxval.load.who_jecfa_tox_studies(
  toxval.db,
  source.db,
  log = FALSE,
  remove_null_dtxsid = TRUE
)
```

Arguments

toxval.db	The database version to use
source.db	The source database
log	If TRUE, send output to a log file
remove_null_dtxsid	If TRUE, delete source records without curated DTXSID value

---

toxval.set.mw *Set the molecular weight in the toxval table, for use in unit conversions*

Description

Set the molecular weight in the toxval table, for use in unit conversions

Usage

```
toxval.set.mw(toxval.db, source = NULL, subsource = NULL)
```

Arguments

toxval.db	The database version to use
source	The source
subsource	The subsource

---

toxval.summary.stats *Generate summary statistics on the toxval database*

Description

Generate summary statistics on the toxval database

Usage

```
toxval.summary.stats(toxval.db, export = FALSE, summ_level = "source")
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
export	Boolean whether to export a file. Default FALSE.
summ_level	String of "source" or "supersource" how to group the summary

Value

DataFrame of record qc_status summary by source.

---

toxvaldb.view *Produce a view of the ToxValDB Data*

Description

'toxvaldb.view' Produces a view for ToxValDB by merging specified tables

Usage

```
toxvaldb.view(toxval.db, user, password, count = 10)
```

Arguments

toxval.db	Database version
user	The username for the MySQL database. The database instance is hard-coded in the function setDBConn().
password	The user's MySQL database password.
count	If count>0, only select this number of records from each source, used for debugging

Value

Write a file with the results: data/view/ToxValDB View toxval.db Sys.Date().xlsx

---

`toxvaldb_release_comparison_stats` *toxvaldb_release_comparison_stats.R*

Description

Pull comparison stats between toxvaldb versions.

Usage

```
toxvaldb_release_comparison_stats(repoDir)
```

Arguments

`repoDir`	Path to Repo/ folder

Value

None. RData file is saved.

`toxval_type.species.mismap` *Find species and toxval_type mismaps, e.g. species other than human fro RfD, RfC, cancer sloper, MSL, etc.*

Description

Find species and toxval_type mismaps, e.g. species other than human fro RfD, RfC, cancer sloper, MSL, etc.

Usage

```
toxval_type.species.mismap(toxval.db)
```

Arguments

`toxval.db`	Database version
`source`	The source to be updated

Value

Write a file with the results

`%>%` *Pipe operator*

Description

See `magrittr::%>%` for details.

Usage

```
lhs %>% rhs
```

Arguments

`lhs`	A value or the magrittr placeholder.
`rhs`	A function call using the magrittr semantics.

Value

The result of calling ‘rhs(lhs)‘.

Index