toxvaldbmain

April 19, 2023

Type Package

Title Builds the ToxValDB V9.x Database

Version 1.0.1

Author Taylor Wall

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

Description

ToxValDB is a database containing quantitative records form in vivo toxicologye 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 project moves data from toxval_soruce 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. The ToxValDB SOPs describe in more detail how to run the code. This version is setup to build ToxValDB v9.5

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

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`cas_checkSum` *Check CAS RN validity via checksum method*

Description

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

Usage

```
cas_checkSum(x, checkLEN = TRUE)
```

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.
checkLEN	logi. Should the function check that the non-digit characters of *x* are at least 4, but no more than 10 digits long? Defaults to TRUE.

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. It does nothing more.

This means that there is no check for valid CAS format. Use the `cas_detect` function to check CAS format beforehand, or write your own function if necessary.

Value

A `logical` vector of length *x* denoting whether each *x* is a valid CAS by the checksum method. `NA` input values will remain `NA`.

Note

This is a vectorized, reasonably high-performance version of the is.cas function found in the webchem package. The functionality encompasses only the actual checksum checking of `webchem::is.cas`; as mentioned in `details`, use `cas_detect` to recreate the CAS format + checksum checking in `webchem::is.cas`. See examples.

Short of looking up against the CAS registry, there is no way to be absolutely sure that even inputs that pass the checksum test are actually registered CAS RNs. The short digit length of CAS IDs combined with the modulo 10 single- digit checksum means that even within a set of randomly generated validly-formatted CAS entities, ~10% will pass checksum.

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))
```

chem.check	*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(
  res0,
  name.col = "name",
  casrn.col = "casrn",
  source = NULL,
  verbose = F
)
```

Arguments

res0	The data frame in which chemicals names and CASRN will be replaced
name.col	The column name that contains the chemical names
casrn.col	The column name that contains the CARN values
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

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, source = NULL, verbose = F)
```

Arguments

res0	The data frame in which chemicals names and CASRN will be replaced
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 unneeded characters from the end of a string

Description

Clean unneeded characters from the end of a string

Usage

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

Arguments

x	String to be cleaned

Value

The cleaned string

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

contains

Find out if one string contains another

Description

Find out if one string contains another

Usage

```
contains(x, query, verbose = F)
```

Arguments

x	The string to be searched in
query	the second string
verbose	if TRUE, the two strings are printed

Value

if x contains query, return TRUE, FALSE otherwise

count.source_hash *Look for duplicated source_hash*

Description

Look for duplicated source_hash

Usage

```
## S3 method for class 'source_hash'
count(toxval.db)
```

Arguments

toxval.db	The version of toxval in which the data is altered.

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)
```

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.all.with.references.v93` *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.all.with.references.v93(
  toxval.db = "res_toxval_v94",
  file.name = NA,
  pfas = T
)
```

Arguments

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

Value

writes an Excel file with the name ../export/toxval_pod_summary_[human_eco]_Sys.Date().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 = "res_toxval_v94", 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
`human_eco`	Either 'human health' or 'eco'

Value

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

---

`export.for.study_type`
Export data required for setting the study type

Description

Export data required for setting the study type

Usage

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

Arguments

`toxval.db`	Database version

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.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)
```

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

Value

An excel file in dictionaries with the missing entries "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 = "2022-08-15")
```

Arguments

`toxval.db`	The version of the database to use
`date_string`	The date of the latest dictinary 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)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.

Value

An excel file in dictionaries with the missing entries "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

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

Description

Fill the chemical table

Usage

```
## S3 method for class 'chemical.by.source'
fill(toxval.db, source, verbose = T)
```

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

```
## S3 method for class 'toxval.defaults'
fill(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

```
## S3 method for class 'toxval.defaults.global.by.source'
fill(toxval.db, source = NULL)
```

Arguments

`toxval.db`	The version of toxval from which to set defaults.
`source`	The source to be fixed

`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, media, 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, media, toxval_subtype

Usage

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

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.casrn` *Fix a CASRN that has one of several problems*

Description

Fix a CASRN that has one of several problems

Usage

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

Arguments

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

Value

the fixed CASRN

`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)
```

Arguments

`toxval.db`	The version of toxvaldb to use.
`source`	THe source to be fixed

`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)
```

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.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)
```

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.exposure_method.and.form.by.source`

Update the exposure route, method and form from a dictionary

Description

Update the exposure route, method and form from a dictionary

Usage

```
fix.exposure_method.and.form.by.source(toxval.db, source)
```

Arguments

`toxval.db`	The database version to use
`source`	The source to process

`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)
```

Arguments

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

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, reset = T)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`source`	The source to be fixed. If NULL, fix all sources
`reset`	If TRUE, reset all values to 'not specified' before processing all records in the source

Value

The database will be altered

fix.non_ascii.v2 *Flag and fix non-ascii characters in the database*

Description

Flag and fix non-ascii characters in the database

Usage

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

Arguments

df	The dataframe to be processed
The	source to be fixed

Value

The dataframe with non ascii characters replaced with cleaned versions

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)
```

Arguments

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

`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 = NULL, reset = T)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`source`	The source to be fixed
`reset`	If TRUE, reset all values to 'pass' before setting

Value

The database will be altered

`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, restart = T)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`source`	The source to be updated
`restart`	If TRUE, delete all values and start from scratch

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, ignore = FALSE)
```

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
ignore	If TRUE allow missing values to be ignored

Value

The database will be altered

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, date_string = "2023-02-14")
```

Arguments

toxval.db	The version of the database to use
source	The source to be fixed
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, date_string = "2023-04-03")
```

Arguments

toxval.db	The version of the database to use
source	The source to be fixed. If NULL, fix for all sources
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)
```

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.study_type.manual` *Fix the study_type using manual curation*

Description

Fix the study_type using manual curation

Usage

```
fix.study_type.manual(toxval.db, source = NULL, sys.date = "2023-04-10")
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.

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 = F
)
```

Arguments

`toxval.db`	The version of toxvaldb to use.
`source`	Source to be fixed
`do.convert.units`	If TRUE, so unit conversions, as opposed to just cleaning

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

Description

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

Usage

```
getDBConn()
```

Value

print the database connection information

hello *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

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 2023-02-15.xlsx*

Description

Load Source Info for each source into toxval The information is in the file ~/dictionary/source_info 2023-02-15.xlsx

Usage

```
import.source.info.by.source(toxval.db, source = NULL)
```

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

---

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()
```

---

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

printCurrentFunction *Print the name of the current function*

Description

Print the name of the current function

Usage

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

Arguments

comment.string	An optional string to be printed

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

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	if TRUE, add the auto increment primary key even if not part of the query

Value

Returns the database table auto incremented primary key ID

runInsertTable *Inserts multiple rows into a database table*

Description

Inserts multiple rows into a database table

Usage

```
runInsertTable(mat, table, db, do.halt = T, verbose = F, get.id = T)
```

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

runQuery *Runs a database query and returns a result set*

Description

Runs a database query and returns a result set

Usage

```
runQuery(query = NULL, db, do.halt = T, verbose = 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

setDBConn *set SQL connection to the database*

Description

set SQL connection to the database

Usage

```
setDBConn(server = "ccte-mysql-res.epa.gov", user, password)
```

Arguments

server	SQL server on which relevant database lives
user	SQL username to access database
password	SQL password corresponding to username

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 = F
)
```

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 = F
)
```

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

---

species.mapper *Map the species to the ECOTOX species dictionary and export the missing species to add to the dictionary*

Description

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

Usage

```
species.mapper(toxval.db, date_string = "2023-02-14")
```

Arguments

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

`toxval.config` *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) and the urls for the ACToR web services.*

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) and the urls for the ACToR web services.

Usage

```
toxval.config()
```

Value

Returns a set of parameters to be used throughout the package

`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, reset = F, date_string = "2022-05-25")
```

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`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=T) 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=T) 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,
  log = F,
  do.init = F,
  do.reset = F,
  do.load = F,
  do.post = F,
  do.extra = F
)
```

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.
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 *Load the ATSDR MRLs 2020 data from toxval_source to toxval*

Description

Load the ATSDR MRLs 2020 data from toxval_source to toxval

Usage

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

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.atsdr.pfas`
Load the original ATSDR PFAS from toxval_source to toxval

Description

Load the original ATSDR PFAS from toxval_source to toxval

Usage

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

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.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 = F)
```

Arguments

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

`toxval.load.atsdr_mrl_2020`
Load the ATSDR MRLs 2020 data from toxval_source to toxval

Description

Load the ATSDR MRLs 2020 data from toxval_source to toxval

Usage

```
toxval.load.atsdr_mrl_2020(toxval.db, source.db, log = F)
```

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.atsdr_mrl_2022`
Load the ATSDR MRLs 2022 data from toxval_source to toxval

Description

Load the ATSDR MRLs 2022 data from toxval_source to toxval

Usage

```
toxval.load.atsdr_mrl_2022(toxval.db, source.db, log = F)
```

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.bcfbaf` *Load the Arnot BAF / BCF data*

Description

Load the Arnot BAF / BCF data

Usage

```
toxval.load.bcfbaf(toxval.db, source.db, verbose = F)
```

Arguments

`toxval.db`	The database to use.
`verbose`	If TRUE, print out extra diagnostic messages

`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 = F)
```

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.cal_dph` *Load the California DPH data (old ACToR - flex)data from toxval sourcedb to toxval*

Description

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

Usage

```
toxval.load.cal_dph(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.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 data to toxval

Description

Load ChemID Plus Acute data data to toxval

Usage

```
toxval.load.chemidplus(toxval.db, source.db, log = F, do.init = F)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
log	If TRUE, send output to a log file
do.init	if TRUE, read the data in from the file and set up the matrix
verbose	Whether the loaded rows should be printed to the console.

toxval.load.chiu

Load the Chiu data from toxval_source to toxval

Description

Load the Chiu data from toxval_source to toxval

Usage

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

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.copper *Load Copper Manufacturers daa from toxval_source to toxval*

Description

Load Copper Manufacturers daa from toxval_source to toxval

Usage

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

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.cosmos *Load teh COSMOS data from source to toxval*

Description

Load teh COSMOS data from source to toxval

Usage

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

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.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 = F)
```

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.dod.ered *Load the DOD ERED data from toxval_source to toxval*

Description

Load the DOD ERED data from toxval_source to toxval

Usage

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

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 = F)
```

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.ecorisk`
Load the DOE ECORISK data (also called LANL) data from toxval_source to toxval

---

Description

Load the DOE ECORISK data (also called LANL) data from toxval_source to toxval

Usage

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

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

`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 = F)
```

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.echa.echemportal.api`

Load ECHA eChemPortal API data from toxval_source to toxval

Description

Load ECHA eChemPortal API data from toxval_source to toxval

Usage

```
toxval.load.echa.echemportal.api(toxval.db, source.db, log = F)
```

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.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 = F, reset = 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.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 = F,
  do.load = F,
  sys.date = "2023-01-26"
)
```

Arguments

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

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 = F)
```

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.efsa2 *Load EFSA2 data from toxval_source to toxval*

Description

Load EFSA2 data from toxval_source to toxval

Usage

```
toxval.load.efsa2(toxval.db, source.db, log = F)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The source databse from which data should be loaded
log	If TRUE, send output to a log file

`toxval.load.envirotox` *Load EnviroTox data from toxval_source to toxval*

Description

Load EnviroTox data from toxval_source to toxval

Usage

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

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 (old ACToR - flex)data from toxval sourcedb to toxval*

Description

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

Usage

```
toxval.load.epa_aegl(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.fda_cedi` *Load the FDA CEDI (old ACToR - flex)data from toxval sourcedb to toxval*

Description

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

Usage

```
toxval.load.fda_cedi(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.flex` *Load the FLEX data (old ACToR data) from files to toxval. This will load all Excel file in the folder ACToR replacements/*

Description

Load the FLEX data (old ACToR data) from files to toxval. This will load all Excel file in the folder ACToR replacements/

Usage

```
toxval.load.flex(toxval.db, verbose = F, only.new = F)
```

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`verbose`	Whether the loaded rows should be printed to the console.
`only.new`	if TRUE only files where the sources is not alrady in the database will be loaded

toxval.load.generic

Generic structure for laoding to toxval from toxval_source

Description

Generic structure for laoding to toxval from toxval_source

Usage

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

Arguments

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

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.
verbose	If TRUE output debug information
do.read	If TRUE, read in the DSSTox file

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 = F)
```

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

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 = F)
```

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

`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 = F)
```

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

`toxval.load.healthcanada`
Load Health Canada data from toxval_source to toxval

Description

Load Health Canada data from toxval_source to toxval

Usage

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

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

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 = F)
```

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.hess *Load the HESS data from toxval_source to toxval*

Description

Load the HESS data from toxval_source to toxval

Usage

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

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.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 = F)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The source databse from which data should be loaded
log	If TRUE, send output to a log file

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 new_iris_noncancer and new_iris_cancer from toxval_source to toxval*

Description

Load new_iris_noncancer and new_iris_cancer from toxval_source to toxval

Usage

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

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.mass_mmcl *Load the mass_mmcl (old ACToR - flex)data from toxval sourcedb to toxval*

Description

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

Usage

```
toxval.load.mass_mmcl(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.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 = F)
```

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.opp *Load opp from toxval_source to toxval*

Description

Load opp from toxval_source to toxval

Usage

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

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

`toxval.load.oppt` *Load new_oppt_table from toxval_source to toxval*

Description

Load new_oppt_table from toxval_source to toxval

Usage

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

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.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 (old ACToR - flex) data from toxval sourcedb to toxval*

Description

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

Usage

```
toxval.load.ow_dwsha(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.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 = F)
```

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

---

Description

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

Usage

```
toxval.load.penn_dep(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.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 = F)
```

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.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 = F,
  chem_source,
  subsource = NULL
)
```

Arguments

`toxval.db`	The database version to use
`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
`sourcedb`	The source database name

`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 = F)
```

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

`toxval.load.pprtv.ncea`
Load PPRTV (NCEA) from toxval source to toxval

Description

Load PPRTV (NCEA) from toxval source to toxval

Usage

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

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

`toxval.load.pprtv.ornl`
Load PPRTV (ORNL) from toxval_source to toxval

Description

Load PPRTV (ORNL) from toxval_source to toxval

Usage

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

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`source.db`	The source databse from which data should be loaded
`log`	If TRUE, send output to a log file

`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 = 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.skin.eye` *Load the Skin eye data*

Description

Load the Skin eye data

Usage

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

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 = T)
```

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

`toxval.load.source_chemical.echa_iuclid`
Perform the DSSTox mapping

Description

Perform the DSSTox mapping

Usage

```
toxval.load.source_chemical.echa_iuclid(
  toxval.db,
  source.db,
  source = "ECHA IUCLID",
  verbose = T,
  chem_source
)
```

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
`chem_source`	The source_table name - this is the soruce in chemical source

toxval.load.species *Load the species table*

Description

Load the species table

Usage

```
toxval.load.species(toxval.db, date_string = "2023-02-14")
```

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 = F)
```

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.toxrefdb2.1`
Load ToxRefdb data to toxval

Description

Load ToxRefdb data to toxval

Usage

```
toxval.load.toxrefdb2.1(toxval.db, source.db, log = F, do.init = T)
```

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`log`	If TRUE, send output to a log file
`do.init`	if TRUE, read the data in from the toxrefdb database and set up the matrix
`verbose`	Whether the loaded rows should be printed to the console.

`toxval.load.tri` *Load TRI data data to toxval*

Description

Load TRI data data to toxval

Usage

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

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`log`	If TRUE, send output to a log file
`verbose`	Whether the loaded rows should be printed to the console.
`do.init`	if TRUE, read the data in from the file and set up the matrix

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 teh uterotophic and Hershberger data*

Description

Load teh uterotophic and Hershberger data

Usage

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

Arguments

toxval.db	The version of toxval into which the tables are loaded.
log	If TRUE, send output to a log file
verbose	Whether the loaded rows should be printed to the console.

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

---

Description

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

Usage

```
toxval.load.who_ipcs(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.wignall`
Load Wignall from toxval_source to toxval

Description

Load Wignall from toxval_source to toxval

Usage

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

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

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)
```

Arguments

toxval.db	The database version to use
source	The source

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

Description

Generate summary statistics on the toxval database

Usage

```
toxval.summary.stats(toxval.db)
```

Arguments

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

Index