toxvaldb09

August 5, 2022

Type Package

Title Builds the ToxValDB V9.2 Database

Version 1.0.1

Author Aswani Unikrishnan

Maintainer Richard Judson <judson.richard@epa.gov>

Description
ToxValDB is a database containing quantitative records form in vivo toxicologye studies from many sources (46 for this release). 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. 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.

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:

build_echa_echemportal_api_dict

code to create ECHA echemportal api dict

Description

code to create ECHA echemportal api dict

Usage

```
build_echa_echemportal_api_dict(toxval.db, filepath)
```

Arguments

toxval.db	The version of toxval into which the dictionary is loaded.
filepath	The path for all the input xlsx files ./echa_echemportal_api/echa_echemportal_api_files

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

clowder_document_list *Get a listing of all of the documents in clowder and link back to information in dev_toxval_v8*

Description

Get a listing of all of the documents in clowder and link back to information in dev_toxval_v8

Usage

```
clowder_document_list(db = "dev_toxval_v8")
```

Arguments

db	The version of toxval into which the source is loaded.

clowder_id_prep.v3 *Organize the clowder_id and document_name information*

Description

Organize the clowder_id and document_name information

Usage

```
clowder_id_prep.v3(db = "dev_toxval_v9")
```

Arguments

db	The version of toxval into which the source is loaded. File from clowder linking clowder_ids to document_names, generated by Taylor Wall clowder_doc_maps_20220608.xlsx

compare.pfas_150_sem.to.hawc *Compare teh PFAS 150 SEM and hte corresponding HAWC data*

Description

Compare teh PFAS 150 SEM and hte corresponding HAWC data

Usage

```
compare.pfas_150_sem.to.hawc(db)
```

Arguments

db	The version of toxval_source into which the source info is loaded.

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

`echemportal.prep.v2` *Process the ECHA eChemPortal data from 2020*

Description

Process the ECHA eChemPortal data from 2020

Usage

```
echemportal.prep.v2(do.load = F)
```

Arguments

`do.load`	If TRUE, laod all of the in vivo data

`echemportal.prep.v2.step2` *Process the ECHA eChemPortal data from 2020 - step 2*

Description

Process the ECHA eChemPortal data from 2020 - step 2

Usage

```
echemportal.prep.v2.step2(do.load = F)
```

Arguments

`do.load`	If TRUE, laod all of the in vivo data

`echemportal.prep.v2.step3` *Process the ECHA eChemPortal data from 2020 - step 3*

Description

Process the ECHA eChemPortal data from 2020 - step 3

Usage

```
echemportal.prep.v2.step3(do.load = F)
```

Arguments

`do.load`	If TRUE, laod all of the in vivo data

ecotox.datahub.to.file

Extract ECOTOX from the datahub to a file

Description

Extract ECOTOX from the datahub to a file

Usage

```
ecotox.datahub.to.file(toxval.db, verbose = T, do.load = 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.
do.load	If TRUE, load the data from the input file and put into a global variable

efsa.clean

prepare EFSA_combined.xlsx using dictionaries from efsa.dict.prep.R to be loaded into dev_toxval_source_v4.

Description

prepare EFSA_combined.xlsx using dictionaries from efsa.dict.prep.R to be loaded into dev_toxval_source_v4.

Usage

```
efsa.clean(dir, verbose = F)
```

Arguments

dir	The directory where the input data sits, ../efsa2/efsa2_files/

efsa.dict.prep *Create dictionaries for preparing EFSA_combined.xlsx produced from efsa.prep.R*

Description

Create dictionaries for preparing EFSA_combined.xlsx produced from efsa.prep.R

Usage

```
efsa.dict.prep(dir)
```

Arguments

dir	The directory where the input data sits, ../efsa2/efsa2_files/

efsa.prep *Prepare the new EFSA data*

Description

Prepare the new EFSA data

Usage

```
efsa.prep(dir, step1 = F, step2 = F, step3 = F, step4 = F)
```

Arguments

dir	The directory where the input data sits, ../efsa2/efsa2_files/
step1	Run the first step of the processing
step2	Run the second step of the processing
step3	Run the third step of the processing
step4	Run the fourth step of the processing

`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.by.source.for.release` *Build a data frame of the data from toxval and export by source as a series of xlsx files. This only has the columns tht someone outside might use*

Description

Build a data frame of the data from toxval and export by source as a series of xlsx files. This only has the columns tht someone outside might use

Usage

```
export.all.by.source.for.release(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.by.source.pfas`

Build a data frame of the data from toxval and export by source as a series of xlsx files. THis is a version for PFAS specific use

---

Description

Build a data frame of the data from toxval and export by source as a series of xlsx files. THis is a version for PFAS specific use

Usage

```
export.all.by.source.pfas(toxval.db)
```

Arguments

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

Value

Write a file with the results

`export.all.with.references`

Build a data frame of the PODs and exports as xlsx

Description

Build a data frame of the PODs and exports as xlsx

Build a data frame of the PODs and exports as xlsx

Usage

```
export.all.with.references(toxval.db, dir = "./export/", file.name = NA)

export.all.with.references(toxval.db, dir = "./export/", file.name = NA)
```

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

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

---

export.bcfbaf *Export the BCF / BAF data data*

Description

Export the BCF / BAF data data

Usage

```
export.bcfbaf(toxval.db, dir = "./export")
```

Arguments

toxval.db	Database version

Value

writes and Excel file with the name ../qc_export/toxval_genetox_details_[Sys.Date].xlsx

---

export.cancer.summary *Build a data frame of the Cancer calls and exports as xlsx*

Description

Build a data frame of the Cancer calls and exports as xlsx

Usage

```
export.cancer.summary(toxval.db, file.name = NA, dir = "./export")
```

Arguments

toxval.db	Database version
file.name	If not NULL, read afles of casrn from the chemical folder and only export those chemicals

Value

writes and Excel file with the name ../qc_export/toxval_pod_summary_min_quality_id_[human_eco]_[min_quality_id]_[Sys

---

`export.chemicals` *Export all chemicals in the chemical and chemical_list tables*

Description

Export all chemicals in the chemical and chemical_list tables

Usage

```
export.chemicals(toxval.db)
```

Arguments

`toxval.db`	Database version

Value

writes and Excel file with the name ../export/toxval_chemicals_[Sys.Date].xlsx

---

`export.dsstox` *Export the DSSTox chemical table*

Description

Export the DSSTox chemical table

Usage

```
export.dsstox()
```

`export.dsstox.mol.wt`
Export mol weight from DSSTox

Description

Export mol weight from DSSTox

Usage

```
export.dsstox.mol.wt()
```

`export.echa.repeats` *Export ECHA data from all four ECHA sources and look for overlaps*

Description

Export ECHA data from all four ECHA sources and look for overlaps

Usage

```
export.echa.repeats(toxval.db)
```

Arguments

`toxval.db`	Database version

Value

writes and Excel file with the name ./export/toxval_echa_repeats_[Sys.Date].xlsx

`export.final.params` *Export the final values for the character params (e.g. toxval_type).*

Description

Export the final values for the character params (e.g. toxval_type).

Usage

```
export.final.params(toxval.db)
```

Arguments

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

`export.genetox_details`
Export the detailed genetox data

Description

Export the detailed genetox data

Usage

```
export.genetox_details(toxval.db, dir = "./export")
```

Arguments

`toxval.db`	Database version

Value

writes and Excel file with the name ../qc_export/toxval_genetox_details_[Sys.Date].xlsx

---

`export.genetox_summary`
Export the summary genetox data

Description

Export the summary genetox data

Usage

```
export.genetox_summary(toxval.db, dir = "./export")
```

Arguments

`toxval.db`	Database version

Value

writes and Excel file with the name ../qc_export/toxval_genetox_summary_[Sys.Date].xlsx

export.missing.rac *Export the rows with a missing risk_assessment_class*

Description

Export the rows with a missing risk_assessment_class

Usage

```
export.missing.rac(toxval.db)
```

Arguments

toxval.db	Database version

Value

writes an Excel file with the name ./qc_export/toxval_missing_risk_assessment_class_Sys.Date().xlsx"

export.missing.rac.by.source *Export the rows with a missing risk_assessment_class*

Description

Export the rows with a missing risk_assessment_class

Usage

```
export.missing.rac.by.source(toxval.db, source)
```

Arguments

toxval.db	Database version
source	The source to be processed

Value

writes an Excel file with the name ./qc_export/toxval_missing_risk_assessment_class_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 = "2022-05-25")
```

Arguments

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

export.pod.summary *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.pod.summary(
  toxval.db,
  human_eco = "human health",
  file.name = NA,
  dir = "./export"
)
```

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.record_source *Build a data frame of the th data from record_source and export*

Description

Build a data frame of the th data from record_source and export

Usage

```
export.record_source(toxval.db, dir = "./export")
```

Arguments

toxval.db	Database version

Value

writes an Excel file with the name ../export/toxval_record_summary_[Sys.Date].xlsx

export.skin_eye *Export the skin and eye data*

Description

Export the skin and eye data

Usage

```
export.skin_eye(toxval.db, dir = "./export")
```

Arguments

toxval.db	Database version

Value

writes and Excel file with the name ../qc_export/toxval_skin_eye_[Sys.Date].xlsx

---

`export.source_chemical` *Export the source chemical table*

Description

Export the source chemical table

Usage

```
export.source_chemical(db)
```

Arguments

`db`	The name of the database String to be cleaned

`export.toxval_dictionary` *Export the toxval_dictionary table*

Description

Export the toxval_dictionary table

Usage

```
export.toxval_dictionary(toxval.db, dir = "./export")
```

Arguments

`toxval.db`	Database version

Value

writes and Excel file with the name ../qc_export/toxval_dictionary_[Sys.Date].xlsx

fill.chemical *Fill the chemical table*

Description

Fill the chemical table

Usage

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

Arguments

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

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.chemical_source_index` *Load the chemical_source_index table.*

Description

Load the chemical_source_index table.

Usage

```
## S3 method for class 'chemical_source_index'
fill(db)
```

Arguments

`db`	The version of toxval_source into which the source is loaded.

`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`
Set Toxval Defaults globally

Description

Set Toxval Defaults globally

Usage

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

Arguments

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

`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)
```

Arguments

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

`fix.all.param` *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, generation*

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, generation

Usage

```
fix.all.param(toxval.db)
```

Arguments

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

Value

The database will be altered

`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, 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.all.param.new` *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, generation*

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, generation

Usage

```
fix.all.param.new(toxval.db)
```

Arguments

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

Value

The database will be altered

`fix.all.param.new.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.new.by.source(toxval.db, source)
```

Arguments

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

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.clowder_id.by.source *Add clowder_id's to source db tables based on matching values with source table*

Description

Add clowder_id's to source db tables based on matching values with source table

Usage

```
fix.clowder_id.by.source(source.db, source_table, source)
```

Arguments

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

Value

The database will be altered

`fix.clowder_id.by.toxval`
Add clowder_id's to source db tables

Description

Add clowder_id's to source db tables

Usage

```
fix.clowder_id.by.toxval(toxval.db, source.db, source_table, source, infile)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`infile`	The input file ./clowder_id_mapping/source_doc_with_clowder_id.xlsx, new_qa_set_with_ClowderID_1 22.csv,clowderid_record_source_hash_hawc_pfas_150_430_with_source_hash.xlsx, atsdr_toxval_record_source_w_document-Aswani.csv,efsa2_toxval_record_source_w_document-Aswani.csv, original_oppt_table_w_document-Aswani.csv, subset_v8_chiu_record_source_hash-Aswani.csv,WHOIPCS_v8_v9_should_be_same_record_source_hash-Aswani.csv,missing_toxval_sourc toxval_new_qa_set_ClowderID_02-02-ASWANI.csv, del_docs_matched_20220323.xlsx, toxval_qa_set_2_ECHA-IUCLID_ClowderID_03-23_Aswani.csv

Value

The database will be altered

`fix.critical_effect`
Fix critical_effect in toxval table Fix occurances of multiple critical_effect in the toxval_critical_effect table based on values from critical_effect dictionary file

---

Description

Fix critical_effect in toxval table Fix occurances of multiple critical_effect in the toxval_critical_effect table based on values from critical_effect dictionary file

Usage

```
fix.critical_effect(toxval.db)
```

Arguments

`toxval.db`	The version of toxvaldb to use.

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

Set all empty cells in toxval to '-'

Description

Set all empty cells in toxval to '-'

Usage

```
fix.empty(toxval.db)
```

Arguments

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

Value

The database will be altered

`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)
```

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.hyphen.record_source.by.source`

Set all empty cells in record_source to 'Not Specified'

Description

Set all empty cells in record_source to 'Not Specified'

Usage

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

Arguments

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

Value

The database will be altered

`fix.empty.record_source`

Set all empty cells in record_source to '-'

Description

Set all empty cells in record_source to '-'

Usage

```
fix.empty.record_source(toxval.db)
```

Arguments

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

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)
```

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_form` *Exposure Method temporary fix to add Exposure Form*

Description

Exposure Method temporary fix to add Exposure Form

Usage

```
fix.exposure_form(toxval.db)
```

Arguments

`toxval.db`	The database version to use

`fix.exposure_form.by.source`
Exposure Method temporary fix to add Exposure Form

Description

Exposure Method temporary fix to add Exposure Form

Usage

```
fix.exposure_form.by.source(toxval.db, source)
```

Arguments

`toxval.db`	The database version to use

`fix.exposure_method.and.form.by.source`
Exposure Method temporary fix to add Exposure Form

Description

Exposure Method temporary fix to add Exposure Form

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.exposure_route.by.type`
Alter the exposure route of toxval according to an excel dictionary

---

Description

Alter the exposure route of toxval according to an excel dictionary

Usage

```
fix.exposure_route.by.type(toxval.db)
```

Arguments

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

Value

The database will be altered

`fix.exposure_route.by.type.new`
Alter the exposure route of toxval according to an excel dictionary

---

Description

Alter the exposure route of toxval according to an excel dictionary

Usage

```
fix.exposure_route.by.type.new(toxval.db)
```

Arguments

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

Value

The database will be altered

`fix.exposure_route.by.type.new.by.source`
Alter the exposure route and study_duration_class of toxval based on toxval_type

---

Description

Alter the exposure route and study_duration_class of toxval based on toxval_type

Usage

```
fix.exposure_route.by.type.new.by.source(toxval.db, source)
```

Arguments

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

Value

The database will be altered

`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` *Fix the human_eco flag*

Description

Fix the human_eco flag

Usage

```
fix.human_eco(toxval.db, reset = T)
```

Arguments

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

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.hyphen.by.source *Set all hyphen cells in toxval to 'Not Specified'*

Description

Set all hyphen cells in toxval to 'Not Specified'

Usage

```
fix.hyphen.by.source(toxval.db, source)
```

Arguments

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

Value

The database will be altered

fix.non_ascii *Flag non ascii characters in the database*

Description

Flag non ascii characters in the database

Usage

```
fix.non_ascii(df)
```

Value

The dataframe with non ascii characters replaced with XXX

`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` *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(toxval.db)
```

Arguments

`toxval.db`	The version of toxvaldb to use.

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.qa_status *Fix the qa_status flag*

Description

Fix the qa_status flag

Usage

```
fix.qa_status(toxval.db, reset = T)
```

Arguments

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

Value

The database will be altered

`fix.qa_status.by.source`
Fix the qa_status flag

Description

Fix the qa_status flag

Usage

```
fix.qa_status.by.source(toxval.db, source, reset = T)
```

Arguments

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

Value

The database will be altered

`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`

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(toxval.db, restart = T)
```

Arguments

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

`fix.risk_assessment_class.all.source`

Fix the risk assessment class for all source.

Description

Fix the risk assessment class for all source.

Usage

```
fix.risk_assessment_class.all.source(toxval.db, restart = T)
```

Arguments

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

`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, 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` *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(toxval.db, param, ignore = FALSE)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`param`	THe parameter value to be fixed
`ignore`	If TRUE allow missing values to be ignored

Value

The database will be altered

`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.single.param.new`

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.new(toxval.db, param, ignore = FALSE)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`param`	THe parameter value to be fixed
`ignore`	If TRUE allow missing values to be ignored

Value

The database will be altered

fix.single.param.new.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.new.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
ignore	If TRUE allow missing values to be ignored

Value

The database will be altered

fix.species

Update the species information. This method has to be run iteratively to find the remaining species_original names that don't match what is in the database

Description

Update the species information. This method has to be run iteratively to find the remaining species_original names that don't match what is in the database

Usage

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

Arguments

toxval.db	The version of the database to use

`fix.species.by.source`

Update the species information. This method has to be run iteratively to find the remaining species_original names that don't match what is in the database

---

Description

Update the species information. This method has to be run iteratively to find the remaining species_original names that don't match what is in the database

Usage

```
fix.species.by.source(toxval.db, source)
```

Arguments

`toxval.db`	The version of the database to use

`fix.species.ecotox.by.source`

Update the species_id in toxval from species_ecotox.

Description

Update the species_id in toxval from species_ecotox.

Usage

```
fix.species.ecotox.by.source(toxval.db, source)
```

Arguments

`toxval.db`	The version of the database to use

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 = "2022-05-25")
```

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.species_ecotox *Update the species ecotox table with habitat information from ecotox dictionary.*

Description

Update the species ecotox table with habitat information from ecotox dictionary.

Usage

```
fix.species_ecotox(toxval.db)
```

Arguments

toxval.db	The version of the database to use

fix.strain.by.source

Update the strain_group in toxval from strain_dictionary_2022-03-07.xlsx.

Description

Update the strain_group in toxval from strain_dictionary_2022-03-07.xlsx.

Usage

```
fix.strain.by.source(toxval.db, source)
```

Arguments

toxval.db	The version of the database to use

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 = "2022-08-04")
```

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.toxval_numeric_qualifier` *Fix the toxval_numeric_qualifier*

Description

Fix the toxval_numeric_qualifier

Usage

```
fix.toxval_numeric_qualifier(toxval.db)
```

Arguments

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

Value

The database will be altered

`fix.toxval_numeric_qualifier.by.source` *Fix the toxval_numeric_qualifier by source*

Description

Fix the toxval_numeric_qualifier by source

Usage

```
fix.toxval_numeric_qualifier.by.source(toxval.db, source)
```

Arguments

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

Value

The database will be altered

`fix.units` *Do all of the fixes to units*

Description

1. All of these steps operate on the toxval_units column.
2. To allow this to be run multiple times during debugging, the first step is to copy toxval_units_original into toxval_units
3. Replace variant unit names with standard ones, running fix.single.param. This fixes issues like variant names for mg/kg-day and uses the dictionary file dictionary/toxval_units_5.xlsx
4. Fix special characters in toxval_units
5. Fix issues with units containing extra characters for some ECOTOX records
6. 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
7. Run conversions from molar to mg units, using MW. This uses the dictionary file dictionary/MW conversions.xlsx
8. 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
9. 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
10. Make sure that eco studies are in mg/L and human health in mg/m3

Usage

```
fix.units(toxval.db, do.convert = T)
```

Arguments

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

`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, 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

---

`fix.units.new` *Do all of the fixes to units*

Description

1. All of these steps operate on the toxval_units column.
2. To allow this to be run multiple times during debugging, the first step is to copy toxval_units_original into toxval_units
3. Replace variant unit names with standard ones, running fix.single.param. This fixes issues like variant names for mg/kg-day and uses the dictionary file dictionary/toxval_units_5.xlsx
4. Fix special characters in toxval_units
5. Fix issues with units containing extra characters for some ECOTOX records
6. 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

7. Run conversions from molar to mg units, using MW. This uses the dictionary file dictionary/MW conversions.xlsx
8. 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
9. 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
10. Make sure that eco studies are in mg/L and human health in mg/m3

Usage

```
fix.units.new(toxval.db, do.convert = T)
```

Arguments

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

---

`fix.units.new.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.new.by.source(toxval.db, source, do.convert = T)
```

Arguments

toxval.db	The version of toxvaldb to use.
do.convert	If TRUE, so unit conversions, as opposed to just cleaning

---

fix.units.testmw *Do all of the fixes to units*

Description

Do all of the fixes to units

Usage

```
fix.units.testmw(toxval.db = "dev_toxval_v9", do.convert = T)
```

Arguments

toxval.db	The version of toxvaldb to use.
do.convert	If TRUE, so unit conversions, as opposed to just cleaning

---

format_rsl_excel *Convert complex rsl Source xlsx files into formatted rsl xlsx source files.*

Description

Convert complex rsl Source xlsx files into formatted rsl xlsx source files.

Usage

```
format_rsl_excel(x, thq_x, y, thq_y, z)
```

Arguments

x	./rsl/rsl_files/rsl_thq1_nov_2021.xlsx - The downloaded xls summary table for THQ 1 from RSL-Generic Tables (2020)
thq_x	corresponding thq for x, which is 1 here.
thq_y	corresponding thq for y, which is 0.1 here.
Y	./rsl/rsl_files/rsl_thq0.1_nov_2021.xlsx - The downloaded xls summary table for THQ 0.1 from RSL-Generic Tables (2020)
Z	./rsl/rsl_files/rsl_subchronic_nov_2021.xlsx- downloaded subchronic toxicity table

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

get.chemical.info.by.source *get chemical info from source db tables*

Description

get chemical info from source db tables

Usage

```
get.chemical.info.by.source(source.db, source_table, source, file_id)
```

Arguments

source.db	The version of toxval source to use.
source_table	The name of toxval source table to use.
source	The name of toxval source to use.
file_id	The suffixed 5 digit identifiers specified in the file names in the folder ./chemical_mapping/source_chemical_files

Value

database info collected

`get.chemical.info.by.source.combined`
get.chemical.info.by.source.combined

Description

get chemical info from source db tables for curation, create chemical table to map curated chemicals to.

Usage

```
get.chemical.info.by.source.combined(source.db, source_table, source)
```

Arguments

`source.db`	The version of toxval source to use.
`source_table`	The name of toxval source table to use.
`source`	The name of toxval source to use.

Value

database info collected

`get.cid.list.toxval`
Get chemical ids for many given CASRN/Chemical name pairs

Description

Get chemical ids for many given CASRN/Chemical name pairs

Usage

```
get.cid.list.toxval(toxval.db, chemical.list, source, verbose = F)
```

Arguments

`toxval.db`	The version of toxval that the chemical id is pulled from.
`chemical.list`	A 2-column dataframe of CAS Registry Numbers and chemical names.
`source`	The source of the chemical data
`verbose`	If TRUE, print out extra diagnostic messages

Value

A 3-column dataframe of CAS Registry Numbers, chemical names, and associated chemical IDs.

`get.cid.list.toxval.ecotox` *Get chemical ids for many given CASRN/Chemical name pairs*

Description

Get chemical ids for many given CASRN/Chemical name pairs

Usage

```
get.cid.list.toxval.ecotox(toxval.db, chemical.list, source, verbose = F)
```

Arguments

`toxval.db`	The version of toxval that the chemical id is pulled from.
`chemical.list`	A 2-column dataframe of CAS Registry Numbers and chemical names.
`source`	The source of the chemical data
`verbose`	If TRUE, print out extra diagnostic messages

Value

A 3-column dataframe of CAS Registry Numbers, chemical names, and associated chemical IDs.

`get.clowder.file.maps` *get.clowder.file.maps*

Description

Pull Clowder file information to map to toxval records

Usage

```
get.clowder.file.maps(apiKey)
```

Arguments

`apiKey`	User Clowder API key to access Clowder

Value

Dataframe list of Clowder ID maps

`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

`getPSQLDBConn` *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

```
getPSQLDBConn()
```

Value

print the database connection information

get_clowder_docList *Get Clowder Document List*

Description

This is a helper function to get a list of documents available in a Clowder dataset

Usage

```
get_clowder_docList(apiKey, clowder_dataset)
```

Arguments

apiKey	The API key required for a user to access the Clowder dataset
clowder_dataset	A character string for the dataset name

Value

Returns a dataframe with file details of: file size, date_created, file type, file id, and filename

heast.load.all *Load HEAST to toxval_source. The data to be loaded is in the file ./heast/heast_files/EPA_HEAST_Table1_ORNL for loading.xlsx*

Description

Load HEAST to toxval_source. The data to be loaded is in the file ./heast/heast_files/EPA_HEAST_Table1_ORNL for loading.xlsx

Usage

```
heast.load.all(source.db)
```

Arguments

source.db	The version of toxval_source into which the tables are loaded.

hess_record_url_from_clowder

Get record urls for hess documents from clowder

Description

Get record urls for hess documents from clowder

Usage

```
hess_record_url_from_clowder(toxval.db, source.db, apiKey, clowder_dataset)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
source.db	The version of toxval source into which clowder data is loaded.
apiKey	The API key required for a user to access the Clowder dataset, prefix the apiKey value with "?key="
clowder_dataset	A character string for the dataset name apiKey="", clowder_dataset="CCTE ToxValDB PDFs"

ig.summary.plot

Make a plot summarizing the IG flag informaiton

Description

Make a plot summarizing the IG flag informaiton

Usage

```
ig.summary.plot(to.file = F, sys.date, do.read = F)
```

Arguments

to.file	If TRUE, print the graph to a file, else write to the screen
sys.date	The date of the bin data Excel fileto read
do.read	if TRUE, read in the source file and store in a global

import.dictionary *import the toxval and toxval_type dictionaries*

Description

import the toxval and toxval_type dictionaries

import the toxval and toxval_type dictionaries by source

Usage

```
import.dictionary(toxval.db)

import.dictionary(toxval.db)
```

Arguments

toxval.db	The name of the database

import.driver *Function to run all import scripts to fill toxval_source*

Description

Function to run all import scripts to fill toxval_source

Usage

```
import.driver(
  db = "res_toxval_source_v5",
  chem.check.halt = FALSE,
  do.clean = FALSE
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
do.clean	If TRUE, delte data from all tables before reloading
chem.chek.halt	If TRUE and there are bad chemical names or casrn, stop to look at the results in indir/chemcheck.xlsx

---

`import.source.info` *Load Source Info into toxval. The information is in the file ./dictionary/source_in_2020_aug_17.xlsx*

Description

Load Source Info into toxval. The information is in the file ./dictionary/source_in_2020_aug_17.xlsx

Usage

```
import.source.info(toxval.db)
```

Arguments

`toxval.db`	The version of toxval into which the source info is loaded.

`import.source.info.by.source` *Load Source Info for each source into toxval The information is in the file ~/dictionary/source_in_2020_aug_17.xlsx*

Description

Load Source Info for each source into toxval The information is in the file ~/dictionary/source_in_2020_aug_17.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

import_actor_source

Extract ACToR1 data to toxval source

Description

Extract ACToR1 data to toxval source

Usage

```
import_actor_source(toxval.db, infile, filepath, verbose = F)
```

Arguments

toxval.db	The version of toxval source into which the tables are loaded.
infile	The input file ./ACToR replacements/ACToR_2021/assay_table_hazard prioritized for use.xlsx
filepath	The path for all the input xlsx files ./ACToR replacements/ACToR_2021
verbose	Whether the loaded rows should be printed to the console.
do.init	if TRUE, read the data in from the res_actor_2021q4 database and set up the matrix

import_atsdr_pfas_2021_source

Load ATSDR PFAS 2021 Source into toxval_source

Description

Load ATSDR PFAS 2021 Source into toxval_source

Usage

```
import_atsdr_pfas_2021_source(db, chem.check.halt = F)
```

Arguments

db	The version of toxval_source into which the source is loaded.
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping
indir	The path for all the input xlsx files ./atsdr_pfas_2021/atsdr_pfas_2021_files

import_atsdr_pfas_source

Load ATSDR PFAS Source files into toxval_source

Description

Load ATSDR PFAS Source files into toxval_source

Usage

```
import_atsdr_pfas_source(
  db,
  infile1 = "ATSDR_Perfluoroalkyls_Inhalation.xlsx",
  infile2 = "ATSDR_Perfluoroalkyls_Oral.xlsx",
  infile3 = "ATSDR_PFOA_Inhalation.xlsx",
  infile4 = ".ATSDR_PFOA_Oral.xlsx",
  infile5 = "ATSDR_PFOS_Oral.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile1	The input file ./atsdr_pfas/atsdr_pfas_files/ATSDR_Perfluoroalkyls_Inhalation.xlsx
infile2	The input file ./atsdr_pfas/atsdr_pfas_files/ATSDR_Perfluoroalkyls_Oral.xlsx
infile3	The input file ./atsdr_pfas/atsdr_pfas_files/ATSDR_PFOA_Inhalation.xlsx
infile4	The input file ./atsdr_pfas/atsdr_pfas_files/ATSDR_PFOA_Oral.xlsx
infile5	The input file ./atsdr_pfas/atsdr_pfas_files/ATSDR_PFOS_Oral.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_atsdr_source

Load atsdr Source into toxval_source

Description

Load atsdr Source into toxval_source

Usage

```
import_atsdr_source(
  db,
  infile = "ATSDR_MRLs_2020_Sept2020_Temp.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ./atsdr/atsdr_files/ATSDR_MRLs_2020_Sept2020_Temp.xls
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

---

import_caloehha_source

Load caloehha Source file into toxval_source The raw data can be exported as an Excel sheet from the web site https://oehha.ca.gov/chemicals, selecting the link "Export database as .CSV file"

Description

This method parses that file and prepares for loading into toxval source

Usage

```
import_caloehha_source(
  db,
  infile = "OEHHA-chemicals_2022-06-22T13-42-44.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ="../caloehha/caloehha_files/OEHHA-chemicals_2018-10-30T08-50-47.xlsx",
chem.check.halt	If TRUE and there are problems with chemicals CASRN checks, halt the program

---

import_chiu_source

Load chiu Source into dev_toxval_source_v3. Data from the Chiu et al. paper on RfD values

---

Description

Load chiu Source into dev_toxval_source_v3. Data from the Chiu et al. paper on RfD values

Usage

```
import_chiu_source(
  db,
  infile = "Full_RfD_databaseQAed-FINAL.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ./chiu/chiu_files/Full_RfD_databaseQAed-FINAL.xlsx
chem.check.halt	If TRUE and there are bad chemical names or casrn, stop to look at the results in indir/chemcheck.xlsx

---

import_copper_source

Load copper manufacturers Source into toxval_source

Description

Load copper manufacturers Source into toxval_source

Usage

```
import_copper_source(
  db,
  infile = "Copper Data Entry - Final.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ./copper/copper_files/Copper Data Entry - Final.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_cosmos_source

Load cosmos Source files into toxval_source

Description

Load cosmos Source files into toxval_source

Usage

```
import_cosmos_source(
  db,
  infile1 = "COSMOS_DB_v1_export_2016_04_02_study_data.xlsx",
  infile2 = "COSMOS_DB_v1_export_2016_04_02_cosmetics_inventory.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile1	The input file ./cosmos/cosmos_files/COSMOS_DB_v1_export_2016_04_02_study_data.xlsx
infile2	The input file ./cosmos/cosmos_files/COSMOS_DB_v1_export_2016_04_02_cosmetics_inventory.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_dod_ered_source

Load dod Source into toxval_source

Description

Load dod Source into toxval_source

Usage

```
import_dod_ered_source(
  db,
  infile = "USACE_ERDC_ERED_database_12_07_2018.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ./dod/dod_files/USACE_ERDC_ERED_database_12_07_2018.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

---

import_dod_source *Load DOD MEG to toxval_source. The file to be loaded are in ./dod/dod_files*

Description

Load DOD MEG to toxval_source. The file to be loaded are in ./dod/dod_files

Usage

```
import_dod_source(db, chem.check.halt = F)
```

Arguments

db	The version of toxval_source into which the tables are loaded.
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

---

import_doe_benchmarks_source *Load doe_benchmarks Source into toxval_source*

Description

Load doe_benchmarks Source into toxval_source

Usage

```
import_doe_benchmarks_source(
  db,
  infile = "DOE_Wildlife_Benchmarks_1996.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ./doe_benchmarks/doe_benchmarks_files/DOE_Wildlife_Benchmarks_1996.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_doe_source *Load DOE Source into toxval_source*

Description

Load DOE Source into toxval_source

Load doe Source into dev_toxval_source_v4.

Usage

```
import_doe_source(toxval.db, infile)

import_doe_source(toxval.db, infile)
```

Arguments

toxval.db	The version of toxval into which the source is loaded.
infile	The input file ./doe/doe_files/Revision_29.xlsx
db	The version of toxval_source into which the source is loaded.
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_echa3_source *Load ECHA TSCA POC Source into dev_toxval_source_v4.*

Description

Load ECHA TSCA POC Source into dev_toxval_source_v4.

Usage

```
import_echa3_source(
  toxval.db,
  infile,
  dict_toxval_units,
  dict_toxval_type,
  dict_study_type,
  dict_exposure_route,
  dict_exposure_method,
  dict_study_duration,
  dict_species,
  dict_critical_effects
)
```

Arguments

toxval.db	The version of toxval into which the source is loaded.
infile	The input file ./echa3/echa3_files/TSCA_POC_Chemical_Results_081220.xlsx
dict_toxval_units	The input file ./echa3/echa3_files/toxval_units_dictionary.xlsx
dict_toxval_type	The input file ./echa3/echa3_files/toxval_type_dictionary.xlsx
dict_study_type	The input file ./echa3/echa3_files/study_type_dictionary.xlsx
dict_exposure_route	The input file ./echa3/echa3_files/exposure_route_dictionary.xlsx
dict_exposure_method	The input file ./echa3/echa3_files/exposure_method_dictionary.xlsx
dict_study_duration	The input file ./echa3/echa3_files/study_duration_dictionary.xlsx
dict_species	The input file ./echa3/echa3_files/echa3_species_dict.xlsx
dict_critical_effects	The input file ./echa3/echa3_files/critical_effect_dictionary.xlsx

---

import_echa_echemportal_api_source

Load ECHA echemportal api Source into toxval_source

Description

Load ECHA echemportal api Source into toxval_source

Usage

```
import_echa_echemportal_api_source(
  db,
  filepath = "echa_echemportal_api/echa_echemportal_api_files",
  chem.check.halt = T
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
filepath	The path for all the input xlsx files ./echa_echemportal_api/echa_echemportal_api_files
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

---

`import_echa_echemportal_source`
Load ECHA echemportal 2020 Source into dev_toxval_source_v4.

Description

Load ECHA echemportal 2020 Source into dev_toxval_source_v4.

Usage

```
import_echa_echemportal_source(toxval.db, infile)
```

Arguments

`toxval.db`	The version of toxval into which the source is loaded.
`infile`	The input file ./echa_echemportal/echa_echemportal_files/eChemPortal mammalian data 2020 step 3.xlsx ,build from echemportal.prep.v2.step3.R

`import_echa_iuclid_source`
Load ECHA IUCLID Source into dev_toxval_source_v4.

Description

Load ECHA IUCLID Source into dev_toxval_source_v4.

Usage

```
import_echa_iuclid_source(toxval.db, infile, verbose = T)
```

Arguments

`toxval.db`	The version of toxval into which the source is loaded.
`infile`	The input file ./echa_iuclid/echa_iuclid_files/echa_iuclid_v8.xlsx

import_echa_source *Load ECHA Source from dev_toxval_source_v4(used in v8) saved as infile to dev_toxval_source_v4*

Description

Load ECHA Source from dev_toxval_source_v4(used in v8) saved as infile to dev_toxval_source_v4

Usage

```
import_echa_source(toxval.db, infile, verbose = T)
```

Arguments

toxval.db	The version of toxval into which the source is loaded.
infile	The input file ./echa/echa_files/echa_raw.xlsx

import_efsa2_source *Load efsa2 Source into toxval_source*

Description

Load efsa2 Source into toxval_source

Usage

```
import_efsa2_source(
  db,
  infile = "EFSA_combined_new 2022-07-19.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ./efsa2/efsa2_files/merge2/EFSA_combined_new.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

`import_efsa_source` *Process the raw excel files downloaded from EFSA version3(March 27 2020) To get the files, go to the web site https://zenodo.org/record/3693783#.XrsBMmhKjIU. At the bottom are links to a set of Excel files - download all of them into the next version V3, convert the xlsx files to csv since the xlsx files when read in R converts all symbols/special characters to certain unicode values (space to '_x0020_'). while reading the original xlsx files into R it was unsuccessful to convert enconding to UTF-8, also tried converting using stringi. Only workaround was by converting the downloaded files to csv and using the csv files as input source.*

Description

Process the raw excel files downloaded from EFSA version3(March 27 2020) To get the files, go to the web site https://zenodo.org/record/3693783#.XrsBMmhKjIU. At the bottom are links to a set of Excel files - download all of them into the next version V3, convert the xlsx files to csv since the xlsx files when read in R converts all symbols/special characters to certain unicode values (space to '_x0020_'). while reading the original xlsx files into R it was unsuccessful to convert enconding to UTF-8, also tried converting using stringi. Only workaround was by converting the downloaded files to csv and using the csv files as input source.

Usage

```
import_efsa_source(db, chem.check.halt = F)
```

Arguments

`db`	The version of toxval_source into which the source is loaded.
`chem.check.halt`	If TRUE, stop if there are problems with the chemical mapping

`import_envirotox_source`
Load EnviroTox.V2 Source data into dev_toxval_source_v4.

Description

Load EnviroTox.V2 Source data into dev_toxval_source_v4.

Load EnviroTox.V2 Source data into toxval_source

Usage

```
import_envirotox_source(
  db,
  infile = "envirotox_taxonomy clean casrn.xlsx",
  chem.check.halt = F
)

import_envirotox_source(
  db,
  infile = "envirotox_taxonomy clean casrn.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source info is loaded.
infile	The input file ./envirotox/envirotox_files/envirotox_taxonomy.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping
toxval.db	The version of toxval into which the source info is loaded.

---

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

Description

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

Usage

```
import_flex_source(
  db,
  filepath = "ACToR replacements",
  verbose = F,
  chem.check.halt = F,
  do.clean = F
)
```

Arguments

db	The version of toxval_source into which the tables are loaded.
filepath	The path for all the input xlsx files ./ACToR replacements
verbose	Whether the loaded rows should be printed to the console.

chem.check.halt
: If TRUE and there are problems with chemicals CASRN checks, halt the program

do.clean
: If true, remove data for these sources before reloading

---

import_generic_source

A generic tmpalte for adding data to toxval_source for a new source

---

Description

A generic tmpalte for adding data to toxval_source for a new source

Usage

```
import_generic_source(db, chem.check.halt = F)
```

Arguments

db
: The version of toxval_source into which the source is loaded.

chem.check.halt
: If TRUE and there are bad chemical names or casrn,

---

import_hawc_pfas_150_source

Load HAWC PFAS 150 Source into toxval_source

Description

Load HAWC PFAS 150 Source into toxval_source

Usage

```
import_hawc_pfas_150_source(
  db,
  infile1 = "hawc_pfas_150_raw3.xlsx",
  infile2 = "hawc_pfas_150_doses3.xlsx",
  infile3 = "hawc_pfas_150_groups3.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile1	The input file ./hawc_pfas/hawc_pfas_files/hawc_pfas_150_raw3.xlsx , extracted from https://hawcprd.epa.gov , assessment name - PFAS 150 (2021) and assessment id - 100500085. Data extraction using HawcClient and extraction script hawc_pfas_150.py
infile2	The input file ./hawc_pfas/hawc_pfas_files/hawc_pfas_150_doses3.xlsx
infile3	The input file ./hawc_pfas/hawc_pfas_files/hawc_pfas_150_groups3.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

---

import_hawc_pfas_430_source

Load HAWC PFAS 430 Source into toxval_source

Description

Load HAWC PFAS 430 Source into toxval_source

Usage

```
import_hawc_pfas_430_source(
  db,
  infile1 = "hawc_pfas_430_raw3.xlsx",
  infile2 = "hawc_pfas_430_doses3.xlsx",
  infile3 = "hawc_pfas_430_groups3.xlsx",
  chem.check.halt = T
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile1	The input file ./hawc_pfas_430/hawc_pfas_430_files/hawc_pfas_430_raw3.xlsx , extracted from https://hawcprd.epa.gov , assessment name - PFAS 430 (2020) and assessment id - 100500256. Data extraction using HawcClient and extraction script hawc_pfas_430.py
infile2	The input file ./hawc_pfas_430/hawc_pfas_430_files/hawc_pfas_430_doses3.xlsx
infile3	The input file ./hawc_pfas_430/hawc_pfas_430_files/hawc_pfas_430_groups3.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_hawc_source *Load HAWC Source into toxval_source*

Description

Note that the different tabs in the input sheet have different names, so these need to be adjusted manually for the code to work. This is a problem wit how the data is stored in HAWC

Usage

```
import_hawc_source(
  db,
  infile1 = "hawc_original_12_06_21.xlsx",
  infile2 = "dose_dict.xlsx",
  chem.check.halt = T
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile1	The input file ./hawc/hawc_files/hawc_original_12_06_21.xlsx
infile2	The input file ./hawc/hawc_files/dose_dict.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_health_canada_source *Load Health Canada Source Info into toxval_source*

Description

Load Health Canada Source Info into toxval_source

Usage

```
import_health_canada_source(
  db,
  infile = "HealthCanada_TRVs_2010_AppendixA v2.xlsx",
  chem.check.halt = T
)
```

Arguments

db	The version of toxval_source into which the source info is loaded.
infile	The input file ./health_canada/health_canada_files/HealthCanada_TRVs_2010_AppendixA v2.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

---

import_heast_source *Load HEAST Source into toxval_source*

Description

Load HEAST Source into toxval_source

Usage

```
import_heast_source(
  db,
  infile = "EPA_HEAST_Table1_ORNL for loading.xlsx",
  chem.check.halt = T
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ./heast/heast_files/EPA_HEAST_Table1_ORNL for loading.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

---

import_hess_source *Load HESS Source into toxval_source*

Description

Load HESS Source into toxval_source

Usage

```
import_hess_source(
  db,
  infile1 = "hess_6_16_21.xlsx",
  infile2 = "hess_record_urls_from_clowder.xlsx",
  chem.check.halt = T
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile1	The input file ./hess/hess_files/hess_6_16_21.csv, extracted by Risa Sayre(SCDCD)
infile2	The input file ./hess/hess_files/hess_record_urls_from_clowder.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

---

import_hpvis_source *Load HPVIS Source Info into toxval_source*

Description

Load HPVIS Source Info into toxval_source

Usage

```
import_hpvis_source(db, filepath = "hpvis/hpvis_files", chem.check.halt = T)
```

Arguments

db	The version of toxval_source into which the source info is loaded.
filepath	The path for all the input xlsx files ./hpvis/hpvis_files
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

---

import_iris_source *Load IRIS Source into toxval_source*

Description

Load IRIS Source into toxval_source

Usage

```
import_iris_source(
  db,
  infile1 = "IRIS_non_cancer_clean 2020-05-27.xlsx",
  infile2 = "IRIS_cancer_clean 2020-05-27.xlsx",
  chem.check.halt = T
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile1	The input file ./iris/iris_files/IRIS_non_cancer_clean 2020-05-27.xlsx
infile2	The input file ./iris/iris_files/IRIS_cancer_clean 2020-05-27.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

---

import_lanl_source *Load LANL Source into toxval_source*

Description

Load LANL Source into toxval_source

Usage

```
import_lanl_source(db, infile = "ESLs_R3.3.xlsx", chem.check.halt = T)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ./lanl/lanl_files/ESLs_R3.3.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

---

import_niosh_source *Load NIOSH Source into toxval_source*

Description

Load NIOSH Source into toxval_source

Usage

```
import_niosh_source(db, infile = "niosh_IDLH_2020.xlsx", chem.check.halt = T)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ./niosh/niosh_files/niosh_IDLH_2020.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_oppt_source *Load OPPT Source Info into toxval_source*

Description

Load OPPT Source Info into toxval_source

Usage

```
import_oppt_source(db, infile = "OPPT_data_20181219.xlsx", chem.check.halt = T)
```

Arguments

db	The version of toxval_source into which the source info is loaded.
infile	The input file ./oppt/oppt_files/OPPT_data_20181219.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_opp_source *Load OPP Source into toxval_source*

Description

Load OPP Source into toxval_source

Usage

```
import_opp_source(db, infile = "OPP RfD.xlsx", chem.check.halt = T)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ./opp/opp_files/OPP RfD.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_penn_source *Load Penn Source into toxval_source*

Description

Load Penn Source into toxval_source

Usage

```
import_penn_source(db, infile = "..enn DEP Table 5a.xlsx", chem.check.halt = T)
```

Arguments

db	The version of toxval_source into which the source is loaded.
infile	The input file ./penn/penn_files/Penn DEP Table 5a.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_pfas_150_sem_v2_source *Load PFAS 150 SEM V2 Source data into toxval_source*

Description

Load PFAS 150 SEM V2 Source data into toxval_source

Usage

```
import_pfas_150_sem_v2_source(db, chem.check.halt = F)
```

Arguments

db	The version of toxval_source into which the source info is loaded.
chem.check.halt	If TRUE and there are problems with chemicals CASRN checks, halt the program

`import_pprtv_ncea_source` *Load PPRTV NCEA Source Info into toxval_source*

Description

Load PPRTV NCEA Source Info into toxval_source

Usage

```
import_pprtv_ncea_source(
  db,
  csvfile = "../pprtv_ncea/pprtv_ncea_files/dose_reg2.csv",
  scrapepath = "../pprtv_ncea/PPRTV_scrape2020-04-08.xlsx",
  chem.check.halt = F
)
```

Arguments

`db`	The version of toxval_source into which the source info is loaded.
`csvfile`	The input csv file ./pprtv_ncea/pprtv_ncea_files/dose_reg2.csv
`scrapepath`	The path for new_pprtv_ncea_scrape_table file ./pprtv_ncea/PPRTV_scrape2020-04-08.xlsx
`chem.check.halt`	If TRUE and there are problems with chemicals CASRN checks, halt the program

`import_pprtv_ornl_source` *Load PPRTV ORNL Source into toxval_source*

Description

Load PPRTV ORNL Source into toxval_source

Usage

```
import_pprtv_ornl_source(
  db,
  infile = "new_PPRTV_ORNL cancer noncancer.xlsx",
  chem.check.halt = F
)
```

Arguments

db	The version of toxval into which the source is loaded.
infile	The input file ./pprtv_ornl/pprtv_ornl_files/new_PPRTV_ORNL cancer noncancer.xlsx
chem.check.halt	If TRUE and there are problems with chemicals CASRN checks, halt the program

---

import_rsl_source *Load RSL Source Info into toxval source database*

Description

Load RSL Source Info into toxval source database

Usage

```
import_rsl_source(
  db,
  infile1a = "final_rsl_thq_combined_nov21.xlsx",
  infile1b = "final_rsl_subchronic_nov21.xlsx",
  infile2 = "general_info_nov_21.xlsx",
  infile3 = "key_description_nov_21.xlsx",
  chem.check.halt = T
)
```

Arguments

db	The version of toxval into which the source info is loaded.
infile1a	The input file ./rsl/rsl_files/final_rsl_thq_combined_nov21.xlsx
infile1b	The input file ./rsl/rsl_files/final_rsl_subchronic_nov21.xlsx
infile2	The input file ./rsl/rsl_files/general_info_nov_21.xlsx
infile3	The input file ./rsl/rsl_files/key_description_nov_21.xlsx
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

import_rsl_source_info *Load rsl Source Info into dev_toxval_source_v2.*

Description

Load rsl Source Info into dev_toxval_source_v2.

Usage

```
import_rsl_source_info(toxval.db, infile1a, infile1b, infile2, infile3)
```

Arguments

toxval.db	The version of toxval into which the source info is loaded.
infile1a	The input file ./rsl/rsl_files/final_rsl_thq_combined_nov21.xlsx
infile1b	The input file ./rsl/rsl_files/final_rsl_subchronic_nov21.xlsx
infile2	The input file ./rsl/rsl_files/general_info_nov_21.xlsx
infile3	The input file ./rsl/rsl_files/key_description_nov_21.xlsx

import_test_source *Load TEST Source data into toxval_source*

Description

Load TEST Source data into toxval_source

Usage

```
import_test_source(
  db,
  infile1 = "TEST data.xlsx",
  infile2 = "test_chemicals_invitrodb.csv",
  chem.check.halt = T
)
```

Arguments

db	The version of toxval_source into which the source info is loaded.
infile1	The input file ./test/test_files/TEST data.xlsx
infile2	The input file ./test/test_files/test_chemicals_invitrodb.csv to map casrn to names from prod_internal_invitrodb_v3_2.chemical
chem.check.halt	If TRUE, stop if there are problems with the chemical mapping

`import_wignall_source` *Load wignall Source data into dev_toxval_source_v2.*

Description

Load wignall Source data into dev_toxval_source_v2.

Usage

```
import_wignall_source(
  db,
  infile = "BMD_Results_2014-06-17_reviewed Mar 2018 parsed.xlsx",
  chem.check.halt = T
)
```

Arguments

`db`	The version of toxval into which the source info is loaded.
`infile`	The input file ./wignall/wignall_files/BMD_Results_2014-06-17_reviewed Mar 2018.xlsx
`chem.check.halt`	If TRUE, stop if there are problems with the chemical mapping

`iris.cancer.clean` *create IRIS cancer clean source file from https://cfpub.epa.gov/ncea/iris/search/index.cfm by selecting cancer, oral and inhalation data, and toxicity value checkboxes.*

Description

create IRIS cancer clean source file from https://cfpub.epa.gov/ncea/iris/search/index.cfm by selecting cancer, oral and inhalation data, and toxicity value checkboxes.

Usage

```
iris.cancer.clean(infile)
```

Arguments

`infile`	The input file ../iris/iris_files/excelView.xls

iris.noncancer.clean

create IRIS non cancer clean source file from iris_scrape_rfd_rfc 2020-05-27.xlsx and iris_scrape_woe 2020-05-27.xlsx (build using iris.scraper.R)

Description

create IRIS non cancer clean source file from iris_scrape_rfd_rfc 2020-05-27.xlsx and iris_scrape_woe 2020-05-27.xlsx (build using iris.scraper.R)

Usage

```
iris.noncancer.clean(infile1, infile2)
```

Arguments

infile1	The input file ../iris/iris_files/iris_scrape_rfd_rfc 2020-05-27.xlsx
infile2	The input file ../iris/iris_files/iris_scrape_woe 2020-05-27.xlsx

iris.scraper

Scrape the IRIS website

Description

Scrape the IRIS website

Usage

```
iris.scraper()
```

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()
```

load_clowder_document_name *load_clowder_document_name*

Description

Function to load Clowder document information to database.

Usage

```
load_clowder_document_name(toxval.db, apiKey, clowder_dataset)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
apiKey	The API key required for a user to access the Clowder dataset, prefix the apiKey value with "?key="
clowder_dataset	A character string for the dataset name First create a table in source db with clowder document name and record source hash Update missing document names in toxval v9 record source table using data from clowder table apiKey="", clowder_dataset="CCTE ToxValDB PDFs"
source.db	The version of toxval source into which clowder data is loaded.

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

`map.chemical.to.dsstox`
Perform the DSSTox mapping

Description

Perform the DSSTox mapping

Usage

```
map.chemical.to.dsstox(toxval.db, verbose = T)
```

Arguments

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

`map.chemical.to.dsstox.by.source`
Perform the DSSTox mapping

Description

Perform the DSSTox mapping

Usage

```
map.chemical.to.dsstox.by.source(toxval.db, source, verbose = T)
```

Arguments

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

`map.chemical.to.dsstox.ecotox`
use name and dtxsid pulled from ECOTOX in source_chemical for ecotox

---

Description

use name and dtxsid pulled from ECOTOX in source_chemical for ecotox

Usage

```
map.chemical.to.dsstox.ecotox(toxval.db, source, verbose = T)
```

Arguments

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

`map.chemical_list.to.dsstox`
Perform the DSSTox mapping

Description

Perform the DSSTox mapping

Usage

```
map.chemical_list.to.dsstox(toxval.db, verbose = T)
```

Arguments

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

`map.hash.record.clowder.document_name`

Replace record_source_hash with clowder record_source_hash by mapping using document names

---

Description

Replace record_source_hash with clowder record_source_hash by mapping using document names

Usage

```
map.hash.record.clowder.document_name(toxval.db, apiKey, clowder_dataset)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`apiKey`	The API key required for a user to access the Clowder dataset, prefix the apiKey value with "?key="
`clowder_dataset`	A character string for the dataset name First create a table in source db with clowder document name and record source hash Update record_source_hash in toxval v9 record source table using data from clowder table apiKey="?key=d2547ff7-83ee-4f4e-b7b8-9875a5b18a83", clowder_dataset="CCTE ToxValDB PDFs"
`source.db`	The version of toxval source into which clowder data is loaded.

`map.hash.record_source`

Map the icf master sheet hash to record source table hash

Description

Map the icf master sheet hash to record source table hash

Usage

```
map.hash.record_source(toxval.db, source.db)
```

Arguments

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

Value

The database will be altered

`map.hash.record_source.by.source`
Map the icf master sheet hash to record source table hash

Description

Map the icf master sheet hash to record source table hash

Usage

```
map.hash.record_source.by.source(toxval.db, source)
```

Arguments

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

Value

The database will be altered

`niosh.pdf.to.excel` *niosh.pdf.to.excel*

Description

Conversion of NIOSH pdf to Excel file

Usage

```
niosh.pdf.to.excel(infile)
```

Arguments

`infile`	The input file ../niosh/niosh_files/CDC - Index of Chemicals - NIOSH Publications and Products.pdf

parse.effect.level *Process the the string that has the qualifier, value and units*

Description

Process the the string that has the qualifier, value and units

Usage

```
parse.effect.level(val)
```

Arguments

val	The string

Value

a list with three values: qualifier, value, units

pfas.by.source *Get the sources with PFAS data*

Description

Get the sources with PFAS data

Usage

```
pfas.by.source(db)
```

Arguments

db	The version of toxval into which the source is loaded.

`poc.summary.plot` *Make a scatter plot of the relationship between info availability and Scientific Domain Metric*

Description

Make a scatter plot of the relationship between info availability and Scientific Domain Metric

Usage

```
poc.summary.plot(to.file = F, sys.date, do.read = F)
```

Arguments

`to.file`	If TRUE, print the graph to a file, else write to the screen
`sys.date`	The date of the bin data Excel fileto read
`do.read`	if TRUE, read in the source file and store in a global

`pprtv.ncea.load.all` *Load NEW PPRTV (NCEA) to toxval_source. The data is found in a series of files: ../pprtv_ncea/pprtv_ncea_files/assessments.xlsx ../pprtv_ncea/pprtv_ncea_files/assessment_study.xlsx ../pprtv_ncea/pprtv_ncea_files/reference.xlsx ../pprtv_ncea/pprtv_ncea_files/study.xlsx ../pprtv_ncea/pprtv_ncea_files/dose_reg2.csv ../pprtv_ncea/PPRTV_scrape2020-04-08.xlsx*

Description

Load NEW PPRTV (NCEA) to toxval_source. The data is found in a series of files: ../pprtv_ncea/pprtv_ncea_files/assessment
../pprtv_ncea/pprtv_ncea_files/assessment_study.xlsx ../pprtv_ncea/pprtv_ncea_files/reference.xlsx
../pprtv_ncea/pprtv_ncea_files/study.xlsx ../pprtv_ncea/pprtv_ncea_files/dose_reg2.csv ../pprtv_ncea/PPRTV_scrape2020-04-08.xlsx

Usage

```
pprtv.ncea.load.all(source.db)
```

Arguments

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

`pprtv.ncea.scraper` *Prepare the PPRTV data downloaded October 2018. Go to the web site https://hhpprtv.ornl.gov/quickview/pprtv.php*

Description

Prepare the PPRTV data downloaded October 2018. Go to the web site https://hhpprtv.ornl.gov/quickview/pprtv.php

Usage

```
pprtv.ncea.scraper()
```

`pprtv.ornl.load.all.new` *Prepare all of the PPRTV data from files downloaded May 2020. Go to the web site https://hhpprtv.ornl.gov/quickview/pprtv_compare.php and download the individual files listed here. The combined data is in the file ../pprtv_ornl/pprtv_ornl_files/PPRTV_ORNL_noncancer.xlsx*

Description

Prepare all of the PPRTV data from files downloaded May 2020. Go to the web site https://hhpprtv.ornl.gov/quickview/pprtv_ and download the individual files listed here. The combined data is in the file ../pprtv_ornl/pprtv_ornl_files/PPRTV_ORNL_n

Usage

```
pprtv.ornl.load.all.new(filepath, verbose = F)
```

Arguments

`filepath`	The input files are in path ../pprtv_ornl/pprtv_ornl_files

`pprtv.ornl.scraper` *Prepare the PPRTV data downloaded October 2018. Go to the web site https://hhpprtv.ornl.gov/quickview/pprtv.php*

Description

Prepare the PPRTV data downloaded October 2018. Go to the web site https://hhpprtv.ornl.gov/quickview/pprtv.php

Usage

```
pprtv.ornl.scraper()
```

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

reset.numeric *Reset the numeric and units values to their original values*

Description

Reset the numeric and units values to their original values

Usage

```
reset.numeric(toxval.db)
```

Arguments

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

Value

The database will be altered

`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, 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

runQuery_psql — *Runs a PSQL database query and returns a result set*

Description

Runs a PSQL database query and returns a result set

Usage

```
runQuery_psql(query, db, do.halt = T, verbose = T)
```

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

`runStatement` *runStatement*

Description

Run a SQL statement, such as an ALTER or UPDATE

Usage

```
runStatement(query, db, do.halt = F, 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

Value

None. SQL statement is run.

`set.dtxsid` *Set the dtxsid values for all of the other tables*

Description

Set the dtxsid values for all of the other tables

Usage

```
set.dtxsid(toxval.db, table, verbose = T)
```

Arguments

`toxval.db`	The version of toxvaldb to use.
`table`	the Database table to be updated
`verbose`	If TRUE, print out extra diagnostic messages

set.dtxsid.by.source *Set the dtxsid values for all of the other tables*

Description

Set the dtxsid values for all of the other tables

Usage

```
set.dtxsid.by.source(toxval.db, table, source, verbose = T)
```

Arguments

toxval.db	The version of toxvaldb to use.
table	the Database table to be updated
verbose	If TRUE, print out extra diagnostic messages

set.hash.bcfbaf *Set the hash in table bcfbaf*

Description

Set the hash in table bcfbaf

Usage

```
set.hash.bcfbaf(toxval.db, do.reset = T)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
do.reset	if TRUE, reset all hashes and start rom scratch. Otherwise only update empty values

Value

The database will be altered

`set.hash.bcfbaf.by.source`
Set the hash in table bcfbaf

Description

Set the hash in table bcfbaf

Usage

```
set.hash.bcfbaf.by.source(toxval.db, source, do.reset = T)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`do.reset`	if TRUE, reset all hashes and start rom scratch. Otherwise only update empty values

Value

The database will be altered

`set.hash.genetox_details`
Set the hash in table genetox_details

Description

Set the hash in table genetox_details

Usage

```
set.hash.genetox_details(toxval.db, do.reset = T)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`do.reset`	if TRUE, reset all hashes and start rom scratch. Otherwise only update empty values

Value

The database will be altered

`set.hash.genetox_details.by.source`
Set the hash in table genetox_details

Description

Set the hash in table genetox_details

Usage

```
set.hash.genetox_details.by.source(toxval.db, source, do.reset = T)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`do.reset`	if TRUE, reset all hashes and start rom scratch. Otherwise only update empty values

Value

The database will be altered

`set.hash.record_source`
Set the hash in table record_source

Description

Set the hash in table record_source

Usage

```
set.hash.record_source(toxval.db, do.reset = T)
```

Arguments

`toxval.db`	The version of toxval in which the data is altered.
`do.reset`	if TRUE, reset all hashes and start from scratch. Otherwise only update empty values

Value

The database will be altered

set.hash.record_source.by.source *Set the hash in table record_source*

Description

Set the hash in table record_source

Usage

```
set.hash.record_source.by.source(toxval.db, source, reset = T)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
do.reset	if TRUE, reset all hashes and start from scratch. Otherwise only update empty values

Value

The database will be altered

set.hash.skin_eye *Set the hash in table skin_eye*

Description

Set the hash in table skin_eye

Usage

```
set.hash.skin_eye(toxval.db, do.reset = T)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
do.reset	if TRUE, reset all hashes and start rom scratch. Otherwise only update empty values

Value

The database will be altered

set.hash.skin_eye.by.source *Set the hash in table skin_eye*

Description

Set the hash in table skin_eye

Usage

```
set.hash.skin_eye.by.source(toxval.db, source, do.reset = T)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
do.reset	if TRUE, reset all hashes and start rom scratch. Otherwise only update empty values

Value

The database will be altered

set.hash.source *Set the hash in source table*

Description

Set the hash in source table

Usage

```
set.hash.source(source.db, source, reset = F)
```

Arguments

source.db	The source database version where data is getting altered (eg: dev_toxval_source_v4).
source	The table in source where the hash data is getting altered.
do.reset	if TRUE, reset all hashes and start from scratch. Otherwise only update empty values

Value

The database will be altered

set.hash.toxval *Set the hash in table toxval*

Description

Set the hash in table toxval

Usage

```
set.hash.toxval(toxval.db, do.reset = T)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
do.reset	if TRUE, reset all hashes and start from scratch. Otherwise only update empty values

Value

The database will be altered

set.hash.toxval.by.source *Set the hash in table toxval*

Description

Set the hash in table toxval

Usage

```
set.hash.toxval.by.source(toxval.db, source, reset = T)
```

Arguments

toxval.db	The version of toxval in which the data is altered.
do.reset	if TRUE, reset all hashes and start from scratch. Otherwise only update empty values

Value

The database will be altered

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

setPSQLDBConn *set PSQL connection to the database*

Description

set PSQL connection to the database

Usage

```
setPSQLDBConn(server, port, user, password)
```

Arguments

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

set_clowder_id *Set the clowder_id and document_name in res*

Description

Set the clowder_id and document_name in res

Usage

```
set_clowder_id(res, source)
```

Arguments

res	The input dataframe
source	The data source name

Value

Returns the input dataframe with defaults set

source.size *print out the size of each of the tables in toxval_source*

Description

print out the size of each of the tables in toxval_source

Usage

```
source.size(db = "res_toxval_source_v5")
```

Arguments

db	The version of toxval_source into which the source is loaded.

`source.table.to.DAT` *source.table.to.DAT*

Description

Convert toxval source table to DAT format for loading to DAT application

Usage

```
source.table.to.DAT(source.db, source_table, limit = 1e+06, sample_p = NA)
```

Arguments

`source.db`	The version of toxval source to use.
`source_table`	The name of toxval source table to use.
`limit`	Excel file grouping limit (default is max XLSX row limit)
`sample_p`	Percentage of records to sample down to
`source`	The name of toxval source to use.

Value

Processed source table to DAT format cached and returned.

`source_chemical.duplicates` *Find duplicated chemicals in the source_chemical table. THis will help get rid of records that have been repalced*

Description

Find duplicated chemicals in the source_chemical table. THis will help get rid of records that have been repalced

Usage

```
source_chemical.duplicates(db)
```

Arguments

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

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.process

Deal with the process of making the source_chemical information

Description

Deal with the process of making the source_chemical information

Usage

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

Arguments

db	The version of toxval into which the source info is loaded.
res	The input dataframe to which chemical information will be added
source	The source to process
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 original dataframe with a chemical_id appended

---

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

---

source_prep_and_load

Prep the source data aand load

Description

Prep the source data aand load

Usage

```
source_prep_and_load(
  db,
  source,
  table,
  res,
  do.reset = FALSE,
  do.insert = FALSE,
  chem.check.halt = FALSE
)
```

Arguments

db	The version of toxval_source into which the source is loaded.
source	Name of the source
table	Name of the database table
res	The data frame to be processed
do.reset	If TRUE, delete data from the database for this source before inserting new data. Default FALSE
do.insert	If TRUE, insert data into the database, default TRUE
chem.check.halt	If TRUE, stop the execution if there are errors in the chemical mapping

---

source_set_defaults

Set default value for NAs - jsut set NA to "-" for columns of type character

---

Description

Set default value for NAs - jsut set NA to "-" for columns of type character

Usage

```
source_set_defaults(res, source)
```

Arguments

res	The input dataframe
source	The data source name

Value

Returns the input dataframe with defaults set

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 = "2022-02-23")
```

Arguments

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

toxval.check.source_chemical *Check the status of the soruce_chemical tables*

Description

Check the status of the soruce_chemical tables

Usage

```
toxval.check.source_chemical(toxval.db, source.db)
```

Arguments

toxval.db	The version of toxvaldb to use.
source.db	The vsource database version

`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)
```

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`reset`	If TRUE, delete all content from the database

`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
)
```

Arguments

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

---

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.bcfbaf *Load the Arnot BAF / BCF data*

Description

Load the Arnot BAF / BCF data

Usage

```
toxval.load.bcfbaf(toxval.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)
```

Arguments

`toxval.db`	The version of the database to use

`toxval.load.chemical.list`
Load The chemical lists to toxval. All Excel files in the folder ../chemicals/for_load/ are loaded. The must have the columns casrn, name, list_name

---

Description

Load The chemical lists to toxval. All Excel files in the folder ../chemicals/for_load/ are loaded. The must have the columns casrn, name, list_name

Usage

```
toxval.load.chemical.list(toxval.db, verbose = T)
```

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`verbose`	If TRUE, print debugging messages

`toxval.load.chemical.list.by.source`

Load The chemical lists for corresponding sources to toxval. All Excel files in the folder ../chemicals/for_load/ are loaded. The must have the columns casrn, name, list_name

---

Description

Load The chemical lists for corresponding sources to toxval. All Excel files in the folder ../chemicals/for_load/ are loaded. The must have the columns casrn, name, list_name

Usage

```
toxval.load.chemical.list.by.source(toxval.db, source, verbose = T)
```

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`verbose`	If TRUE, print debugging messages

`toxval.load.chemical.list.ecotox`

Load The chemical lists for corresponding sources to toxval. All Excel files in the folder ../chemicals/for_load/ are loaded. The must have the columns casrn, name, list_name

---

Description

Load The chemical lists for corresponding sources to toxval. All Excel files in the folder ../chemicals/for_load/ are loaded. The must have the columns casrn, name, list_name

Usage

```
toxval.load.chemical.list.ecotox(toxval.db, source, verbose = T)
```

Arguments

`toxval.db`	The version of toxval into which the tables are loaded.
`verbose`	If TRUE, print debugging messages

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

Load new_doe_table and new_doe_benchmarks_table from toxval_source to toxval

Description

Load new_doe_table and new_doe_benchmarks_table from toxval_source to toxval

Usage

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

Arguments

toxval.db	The version of toxval into which the tables are loaded.
source.db	The source database to use.
verbose	Whether the loaded rows should be printed to the console.

`toxval.load.doe.benchmarks`
Load DOE Wildlife Benchmarksdata from toxval_source to toxval

---

Description

Load DOE Wildlife Benchmarksdata 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.ecotox *Load ECOTOX from the web services output to toxval*

Description

Load ECOTOX from the web services output to toxval

Usage

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

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`

Load the Genetox data from Grace

Description

Load the Genetox data from Grace

Usage

```
toxval.load.genetox(toxval.db, verbose = F, do.read = T)
```

Arguments

`toxval.db`	The database to use.
`verbose`	If TRUE output debug information
`do.read`	If TRUE, read in the DSSTox file

`toxval.load.genetox_details`

Load the Genetox data from Grace

Description

Load the Genetox data from Grace

Usage

```
toxval.load.genetox_details(toxval.db, verbose = F)
```

Arguments

`toxval.db`	The database to use.
`verbose`	if TRUE output debug information

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.new_ecotox *Load ecotox data from datahub to toxval*

Description

Load ecotox data from datahub to toxval

Usage

```
toxval.load.new_ecotox(toxval.db, verbose = T)
```

Arguments

toxval.db	The version of toxval into which the tables are loaded.
verbose	Whether the loaded rows should be printed to the console.

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)
```

Arguments

`toxval.db`	The database version to use
`source`	The source name
`do.convert.units`	If TRUE, convert uints, mainly from ppm to mg/kg-day. THis code is not debugged
`sourcedb`	The source database name

`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, 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.species`
Load the species table and the species_id column in toxval

Description

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

Usage

```
toxval.load.species(toxval.db, restart = F, date_string = "2022-05-25")
```

Arguments

`toxval.db`	The version of the database to use
`restart`	If TRUE, rest all of the species_id values in toxval
`date.string`	Date suffix on the input species dictionary

`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.toxrefdb3`
Load ToxRefdb data to toxval

Description

Load ToxRefdb data to toxval

Usage

```
toxval.load.toxrefdb3(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 toxrefdb database and set up the matrix
`verbose`	Whether the loaded rows should be printed to the console.

`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.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.qc.step.1` *do an initial QC of the data by comparing the current database to an old one*

Description

do an initial QC of the data by comparing the current database to an old one

Usage

```
toxval.qc.step.1(db.new = "res_toxval_v92", db.old = "dev_toxval_v9_1")
```

Arguments

`db.new`	The new database version (toxval) for the comparison
`db.old`	= The old database version for the comparison

`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)
```

Arguments

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

`toxval.source.add.timestamps` *toxval.source.add.timestamps*

Description

get chemical info from source db tables

Usage

```
toxval.source.add.timestamps(source.db)
```

Arguments

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

Value

None. SQL statements are executed to add timestamp columns.

`toxval.source.map.chemicals`
toxval.source.map.chemicals

Description

push ChemReg curated chemicals to toxval_source db tables

Usage

```
toxval.source.map.chemicals(source.db, input.path, curated.path)
```

Arguments

`source.db`	The version of toxval source database to use.
`input.path`	Path to folder with original chemical lists
`curated.path`	Path to folder with curated chemical lists

Value

None. SQL statements are executed.

`toxval.source.map.chemicals.combined`
toxval.source.map.chemicals.combined

Description

push ChemReg curated chemicals to toxval_source db tables

Usage

```
toxval.source.map.chemicals.combined(
  source.db,
  input.path,
  curated.path,
  match.raw = FALSE
)
```

Arguments

`source.db`	The version of toxval source database to use.
`input.path`	Path to folder with original chemical lists
`curated.path`	Path to folder with curated chemical lists
`match.raw`	Boolean whether to match by raw name/casrn values (Default FALSE)

Value

None. SQL statements are executed.

`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.

`toxval_source.hash.and.load` *Add the hash key to the source tables and add the new rows*

Description

Add the hash key to the source tables and add the new rows

Usage

```
toxval_source.hash.and.load(
  db = "dev_toxval_source_v5",
  source,
  table,
  do.reset = F,
  do.insert = F,
  res
)
```

Arguments

`db`	The version of toxval_source into which the source is loaded.
`source`	Name of the source
`table`	Name of the database table
`do.reset`	If TRUE, delete data from the database for this source before inserting new data. Default FALSE
`do.insert`	If TRUE, insert data into the database, default TRUE
`res`	The data frame to be processed

Index