toxvaldb09

July 29, 2022

Type Package

Title Builds the ToxValDB V9 Database

Version 1.0.1

Author Aswani Unikrishnan

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

Description 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

Imports DBI,
RMySQL,
openxlsx,
dplyr,
tidyr,
stringr,
tibble,
janitor,
logr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

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

Description

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

Usage

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

Arguments

`x`	chr. Input vector of values to check. Standard CAS notation using hyphens is fine, as all non-digit characters are stripped for checksum calculation. Each element of *x* should contain only one suspected CAS RN to check.
`checkLEN`	logi. Should the function check that the non-digit characters of *x* are at least 4, but no more than 10 digits long? Defaults to TRUE.

Details

This function performs a very specific type of check for CAS validity, namely whether the final digit checksum follows the CAS standard. By default, it also ensures that the digit length is compatible with CAS standards. It does nothing more.

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

Value

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

Note

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

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

Examples

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

---

chem.check

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

---

Description

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

Usage

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

Arguments

`res0`	The data frame in which chemicals names and CASRN will be replaced
`name.col`	The column name that contains the chemical names
`casrn.col`	The column name that contains the CARN values
`source`	The source to be processed. If source=NULL, process all sources
`verbose`	If TRUE, print diagnostic messages

Value

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

---

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

Description

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

Usage

```
chem.check.v2(res0, source = NULL, verbose = F)
```

Arguments

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

Value

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

---

`clean.last.character` *Clean unneeded characters from the end of a string*

Description

Clean unneeded characters from the end of a string

Usage

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

Arguments

x	String to be cleaned

Value

The cleaned string

---

clean.toxval.by.source

Delete a portion of the contents of the toxval database

Description

Delete a portion of the contents of the toxval database

Usage

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

Arguments

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

Value

The database will be altered

---

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

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

`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

---

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

---

Description

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

Usage

```
export.all.by.source(toxval.db, source = NULL)
```

Arguments

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

export.all.with.references

Build a data frame of the PODs and exports as xlsx

Description

Build a data frame of the PODs and exports as xlsx

Usage

```
export.all.with.references(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

export.dsstox

Export the DSSTox chemical table

Description

Export the DSSTox chemical table

Usage

```
export.dsstox()
```

`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.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.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

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

Description

Fix a CASRN that has one of several problems

Usage

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

Arguments

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

Value

the fixed CASRN

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

Description

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

Usage

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

Arguments

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

`fix.empty.by.source`

Set all empty cells in toxval to '-'

Description

Set all empty cells in toxval to '-'

Usage

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

Arguments

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

Value

The database will be altered

`fix.empty.record_source.by.source`

Set all empty cells in record_source to '-'

Description

Set all empty cells in record_source to '-'

Usage

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

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

---

Description

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

Usage

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

Arguments

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

Value

The database will be altered

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

Description

Fix the human_eco flag

Usage

```
fix.human_eco.by.source(toxval.db, source, 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 'not specified' before processing all records in the source

Value

The database will be altered

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

Description

Flag and fix non-ascii characters in the database

Usage

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

Arguments

df	The dataframe to be processed
The	source to be fixed

Value

The dataframe with non ascii characters replaced with cleaned versions

---

`fix.priority_id.by.source`
Fix the priority_id in the toxval table based on source

Description

Fix the priority_id in the toxval table based on source

Usage

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

Arguments

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

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

Description

Fix the qa_status flag

Usage

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

Arguments

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

Value

The database will be altered

`fix.risk_assessment_class.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.by.source *Alter the contents of toxval according to an excel dictionary*

Description

Alter the contents of toxval according to an excel dictionary

Usage

```
fix.single.param.by.source(toxval.db, param, source, ignore = FALSE)
```

Arguments

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

Value

The database will be altered

fix.species.v2 *Set the species_id column in toxval*

Description

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

Usage

```
fix.species.v2(toxval.db, source, date_string = "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.strain.v2 *Set the strain information in toxval*

Description

Set the strain information in toxval

Usage

```
fix.strain.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 ofhte latest dictinary version

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

---

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.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.

`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

`import.dictionary` *import the toxval and toxval_type dictionaries*

Description

import the toxval and toxval_type dictionaries

Usage

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

Arguments

`toxval.db`	The name of the database

import.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_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

Usage

```
import_doe_source(db, infile = "Revision_29.xlsx", chem.check.halt = F)
```

Arguments

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

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_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 toxval_source*

Description

Load EnviroTox.V2 Source data into toxval_source

Usage

```
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

---

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_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_source *Load PFAS 150 SEM Source data into toxval_source*

Description

Load PFAS 150 SEM Source data into toxval_source

Usage

```
import_pfas_150_sem_source(
  db,

    infile = "PFAS150 animal study template combined_clearance with DTXSID and CASF
  chem.check.halt = F
)
```

Arguments

db	The version of toxval_source into which the source info is loaded.
infile	The input file ./PFAS 150 SEM/PFAS 150 SEM_files/PFAS150 animal study template combined_clearance with DTXSID and CASRN.xlsx
chem.check.halt	If TRUE and there are problems with chemicals CASRN checks, halt the program

---

`import_pfas_summary_pods_source`
Load PFAS Summary PODs into toxval_source

Description

Load PFAS Summary PODs into toxval_source

Usage

```
import_pfas_summary_pods_source(
  db,
  infile1 = "PFAS 150 Study Level PODs_061920.xlsx",
  infile2 = "CompToxChemicalsDashboard-Batch-Search_2020-07-20_17_18_42.xlsx",
  chem.check.halt = T
)
```

Arguments

`db`	The version of toxval_source into which the source is loaded.
`infile1`	The input file ./PFAS Summary PODs/PFAS Summary PODs_files/PFAS 150 Study Level PODs_061920.xlsx
`infile2`	The input file ./PFAS Summary PODs/PFAS Summary PODs_files/CompToxChemicalsDashboard-Batch-Search_2020-07-20_17_18_42.xls
`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_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

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

Description

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

Usage

```
load.dsstox()
```

log_message *Function to combine output log with output message*

Description

Function to combine output log with output message

Usage

```
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

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.

printCurrentFunction *Print the name of the current function*

Description

Print the name of the current function

Usage

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

Arguments

comment.string	An optional string to be printed

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

Description

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

Usage

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

Arguments

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

Value

Returns the database table auto incremented primary key ID

runInsertTable *Inserts multiple rows into a database table*

Description

Inserts multiple rows into a database table

Usage

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

Arguments

mat	data frame containing the data, with the column names corresponding
table	name of the database table to which data will be inserted
db	the name of the database
do.halt	if TRUE, halt on errors or warnings
verbose	if TRUE, print diagnostic information

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

Description

Runs a database query and returns a result set

Usage

```
runQuery(query, 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

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_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.chiu` *Load the Chiu data from toxval_source to toxval*

Description

Load the Chiu data from toxval_source to toxval

Usage

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

Arguments

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

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

Description

Load Copper Manufacturers daa from toxval_source to toxval

Usage

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

Arguments

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

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

Description

Load teh COSMOS data from source to toxval

Usage

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

Arguments

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

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

Description

Load the DOD data from toxval_source to toxval

Usage

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

Arguments

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

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

Description

Load the DOD ERED data from toxval_source to toxval

Usage

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

Arguments

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

`toxval.load.doe.benchmarks`
Load DOE Wildlife 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.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.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`
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(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.pfas_summary_pods`
Load PFAS Summary PODs from toxval_source to toxval

Description

Load PFAS Summary PODs from toxval_source to toxval

Usage

```
toxval.load.pfas_summary_pods(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.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.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