

Package: srppphist (via r-universe)

September 29, 2026

Type Package

Title Historical Data from the Swiss Register of Plant Protection
Products as an R Package

Version 2.0.3

Date 2026-09-16

Description Data objects extracted from historical XML versions of the Swiss Register of Plant Protection Products 'SRPPP'. Starting from 2011, one such data object is contained for each year. Note that the information contained in this project is outdated. An online version of the current register can be accessed via <<https://www.psm.admin.ch/de/produkte>>. An XML dump of the current register can be read in using the 'srppp' package. There is no guarantee of correspondence of the data contained in or read in using this package with any online version, or with the original registration documents. Also, the Federal Food Safety and Veterinary Office, coordinating the authorisation of plant protection products in Switzerland, does not answer requests regarding this package.

Depends R (>= 4.1.0), srppp (>= 2.0.5)

Imports cli, dplyr, data.tree, stringr, purrr, rlang

Suggests knitr, rmarkdown, here, DiagrammeR, ggplot2, parallel, testthat (>= 3.0.0), tibble, tidyr, flextable, kableExtra

BugReports <https://github.com/agroscope-ch/srppphist/issues>

URL <https://agroscope-ch.github.io/srppphist>

License GPL (>= 3)

LazyData true

LazyDataCompression xz

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

Encoding UTF-8

Language en-GB
Config/testthat/edition 3
Config/roxygen2/version 8.1.0
Config/pak/sysreqs libicu-dev libxml2-dev libssl-dev
Repository https://agroscope-ch.r-universe.dev
Date/Publication 2026-09-29 05:16:26 UTC
RemoteUrl https://github.com/agroscope-ch/srpphist
RemoteRef HEAD
RemoteSha 3735b6d362825979c8a6f473370010eb08fe18ab

Contents

composition	2
last_approved	3
pNbr	4
product_categories	4
srppp_active_substances	5
srppp_active_substances_merged	6
srppp_compositions	7
srppp_ingredients	7
srppp_list	8
srppp_obligations_spe3	10
srppp_pests	11
srppp_products	11
srppp_substances	12
srppp_substances_merged	13
srppp_xml_dates	14
srppp_xml_get.numeric	14
srppp_xml_product_use_not_authorized	15
srppp_xml_zip_files	17
Index	18

composition	<i>Retrieve one or more product compositions</i>
-------------	--

Description

Retrieve one or more product compositions

Usage

composition(pNbrs, lang = c("de", "fr", "it"))

Arguments

pNbrs A numeric vector of one or more P-Numbers
lang The language for the returned table

Examples

```
pNbr("Karate", select = FALSE) |>  
  composition()
```

last_approved	<i>Find the year of the last approved use of an active substance</i>
---------------	--

Description

Find the year of the last approved use of an active substance

Usage

```
last_approved(pattern, lang = c("de", "fr", "it"))
```

Arguments

pattern Search pattern for the substance name, passed to [grepl](#).
lang Should the German names be searched in (default), or the French or the Italian ones?

Value

A tibble containing the matched substance names and the latest occurrence in [srppp_active_substances_merged](#).

Examples

```
last_approved("Imidacloprid")  
last_approved("Quin")
```

pNbr	<i>Retrieve and select a P-Number for a product name</i>
------	--

Description

Retrieve and select a P-Number for a product name

Usage

```
pNbr(string, select = interactive())
```

Arguments

string	The search string
select	If TRUE, then the user is asked to select one product if there is more than one match in interactive mode, otherwise the first match is returned. If FALSE, more than one number is returned in the case of multiple matches

Examples

```
pNbr("Karate")
```

product_categories	<i>Get product categories for a pNbr</i>
--------------------	--

Description

Get product categories for a pNbr

Usage

```
product_categories(
  pNbr,
  year = format(Sys.Date(), "%Y"),
  lang = c("de", "fr", "it")
)
```

Arguments

pNbr	The P-Number of the product
year	The year, defaults to current year
lang	The language to be used for the return value

Value

A character string with the product categories in the SRPPP pasted together, separated by a comma.

Examples

```
product_categories(7811, year = "2024")
product_categories(7811, year = "2024", lang = "it")
```

srppp_active_substances

Table of active substance keys extracted from all years starting 2011

Description

Please note that products whose authorisation has expired remain in the files until the end of the period during which use by the end user is still permitted. Therefore, the latest year given in this dataset means the latest year where a use up period ('exhaustionDeadline') of at least one product containing the substance had not expired in the beginning of the respective year.

Usage

```
srppp_active_substances
```

Format

A [tibble](#), resolving the primary keys ('pk') of the active substances to the latest entry in any of the 'substances' tables in [srppp_list](#). The earliest and the latest year of occurrence are given in the columns 'earliest' and 'latest'.

Details

Please see [srppp_substances](#) and [srppp_substances_merged](#) for details on the different primary keys used in these tables.

Note

Note that a substance occurring in the register in a certain year does not necessarily mean that a product containing the substance was authorized. In particular, a number of products with such comments are present in the register used as representative for 2011. Also note that some known errors in the original XML data are corrected when the data are read using the srppp package. For details, see [Corrections made to the data](#).

See Also

[srppp_active_substances_merged](#), [srppp_substances](#), [srppp_products](#)

Examples

```
print(srppp_active_substances, n = Inf)
```

`srppp_active_substances_merged`*Table of active substance names extracted from all years starting 2011*

Description

Please note that products whose authorisation has expired remain in the files until the end of the period during which use by the end user is still permitted. Therefore, the latest year given in this dataset means the latest year where a use up period ('exhaustionDeadline') of at least one product containing the substance had not expired in the beginning of the respective year.

Usage

`srppp_active_substances_merged`

Format

A [tibble](#)

Details

Please see [srppp_substances](#) and [srppp_substances_merged](#) for details on the different primary keys used in these tables.

Note

Note that a substance occurring in the register in a certain year does not necessarily mean that a product containing the substance was authorized. In particular, a number of products with such comments are present in the register used as representative for 2011. Also note that some known errors in the original XML data are corrected when the data are read using the srppp package. For details, see [Corrections made to the data](#).

See Also

[srppp_active_substances](#), [srppp_substances_merged](#), [srppp_products](#)

Examples

```
print(srppp_active_substances_merged, n = Inf)
```

srppp_compositions	<i>Table of the latest product compositions</i>
--------------------	---

Description

This table contains only one composition for each P-Number. In cases, where several versions of the composition are given in different years, the composition in the latest year of occurrence is used, assuming that it is the correct one.

Usage

```
srppp_compositions
```

Format

A [tibble](#) with the columns P-Number 'pNbr', substance primary key 'pk', 'type' and product composition columns 'percent', 'g_per_L', 'ingredient_de', 'ingredient_fr' and 'ingredient_it', and the column 'latest', indicating the year entry of that combination.

See Also

[srppp_ingredients](#) for all unique compositions of a product
[srppp_products](#)

Examples

```
print(srppp_compositions, n = Inf)
```

srppp_ingredients	<i>Table of product ingredients extracted from all years starting 2011</i>
-------------------	--

Description

This table contains all unique ingredient information extracted from the XML files. Some products have different ingredient information in different years. In this table, all unique compositions occurring in one of the years are collected.

Usage

```
srppp_ingredients
```

Format

A [tibble](#) of all unique combinations of P-Number, substance primary key 'pk', 'type' and product composition columns 'percent', 'g_per_L', 'ingredient_de', 'ingredient_fr' and 'ingredient_it', and the column 'latest', indicating the latest entry of that combination.

See Also

[srppp_compositions](#) for one product composition per product
[srppp_products](#)

Examples

```
print(srppp_ingredients, n = Inf)
```

srppp_list

List of 'srppp_dm' objects for all years starting 2011

Description

For each year, the first XML dump published by the registration authority is used, with few exceptions, where a corrected dump was published shortly after the first one. Please note the use conditions set out by the registration authority for the XML dumps currently published at their website.

Usage

```
srppp_list
```

Format

A named list of srppp::srppp_dm objects created with the companion package 'srppp'. The list elements are named with the years from 2011 to the current year as a character vector

Use conditions set out by the registration authority

Please consult the [use conditions of the XML data files](#) currently published by the Federal Food Safety and Veterinary Office (FSVO). For the the historical data contained in this package, the following points are of particular importance:

- In cases of doubt, the definitive source of information are always the original registration documents, for present as well as past authorisations.
- Commercial use of the data provided as XML files is not permitted without the written consent of the FSVO.

Additional notes regarding proper use of the data

As we include only historical, not current authorisation data in this package, please note the following:

- The descriptions of products and their authorised uses contained in this package refer to past authorisations. Regarding current authorisation, please refer to the [Swiss Register of Plant Protection Products](#), or use the srppp package which facilitates reading in the current registration data into R.

- Products whose authorisation has expired or which have been withdrawn from the parallel import list are present in the historical data until the end of the period during which use by the end user is still permitted ('exhaustionDeadline'). This date and the sell-out period ('soldout-Deadline') are indicated in the products table of each srppp_dm object.
- If you use the historical registration data in the form provided by this package, please cite the package as described by the output of `citation("srppphist")`.
- Please note that some known errors in the original XML data are corrected when the data are read using the srppp package. For details, see [Corrections made to the data](#).

Examples

```
names(srppp_list)

# In case you are interested in the registered uses of products containing
# a certain active substance, here is some example code
library(dplyr, warn.conflicts = FALSE)
# Step 1: Get the pk number of a certain active substance
pk_active <- srppp_active_substances |>
  filter(substance_de == "Cyproconazole") |>
  pull(pk)

# Step 2: Get the products (pNbrs) containing that substance in 2018
products_2018 <- srppp_list[["2018"]]$ingredients |>
  filter(pk == pk_active)

# Step 3: Get the associated uses
uses_2018 <- products_2018 |>
  left_join(srppp_list[["2018"]]$uses, by = "pNbr")

# Step 4: Add additional information, e.g. the cultures
uses_x_cultures_2018 <- uses_2018 |>
  left_join(srppp_list[["2018"]]$cultures, by = c("pNbr", "use_nr"))

# Step 5: Application rate in g/ha
uses_x_cultures_2018_rate <- uses_x_cultures_2018 |>
  srppp::application_rate_g_per_ha() |>
  select(pNbr, use_nr, application_area_de, culture_de, rate_g_per_ha)

# If this should be repeated for all available years, it is convenient
# to define a function that extracts the desired information, apply it
# to the list of yearly product registers, and combine the results in a
# table.
uses_cultures_rates <- function(sr, pk_active) {
  sr$ingredients |>
    filter(pk == pk_active) |>
    left_join(sr$uses, by = "pNbr") |>
    left_join(sr$cultures, by = c("pNbr", "use_nr")) |>
    srppp::application_rate_g_per_ha() |>
    select(pNbr, use_nr, application_area_de,
           culture_de, rate_g_per_ha)
}
```

```

# Test the function
uses_cultures_rates(srppp_list[["2018"]], 116L)

# Create a list of tables
uses_cultures_rates_list <- lapply(srppp_list, uses_cultures_rates, 116L)

# Combine the tables for all years
uses_cultures_rates_all_years <- bind_rows(uses_cultures_rates_list,
  .id = "year")
print(uses_cultures_rates_all_years)

# Find names of original products and sales permissions (W-Numbers with dash)
uses_cultures_rates_all_years |>
  select(year, pNbr) |>
  unique() |>
  left_join(srppp_products[c("pNbr", "wNbr", "name")], by = "pNbr",
    relationship = "many-to-many")

```

srppp_obligations_spe3

Table of all SPe 3 obligations and corresponding mitigation measures

Description

The quantitative mitigation measures extracted from the text as explained in the documentation to [srppp::srppp_dm](#).

Usage

```
srppp_obligations_spe3
```

Format

A [tibble](#) with the relevant obligation texts in German, and its corresponding mitigation measures

Examples

```
print(srppp_obligations_spe3, n = Inf)
```

srppp_pests	<i>Table of pesticide targets ("pests") extracted from all years starting 2011</i>
-------------	--

Description

Table of pesticide targets ("pests") extracted from all years starting 2011

Usage

```
srppp_pests
```

Format

A [tibble](#) with all pesticide target names in German, French and Italian, a scientific name if available and the columns 'earliest' and 'latest', indicating the earliest and latest year of occurrence.

Examples

```
print(srppp_pests, n = Inf)
```

srppp_products	<i>Table of products extracted from all years starting 2011</i>
----------------	---

Description

Please note that products whose authorisation has expired remain in the files until the end of the period during which use by the end user is still permitted. The end of the sell-out period ('soldout-Deadline') and the end of the use period ('exhaustionDeadline') are given in the respective columns of the products table.

Usage

```
srppp_products
```

Format

A [tibble](#) with all unique combinations of P-Number, W-Number, Swiss registration number and product name over all years, and the columns 'earliest' and 'latest', indicating the earliest and latest year of occurrence of that combination. Finally, the columns 'categories_de', 'categories_fr', and 'categories_it' contain concatenations of the product categories in the respective language, with the product categories separated by a comma and a space.

Details

As of version 1.0.0.9004, parallel imports are included in the table, and the Swiss registration number ("Eigenössische Zulassungsnummer") is given in the column `chNbr`. For regular products, this column lists the W-Number in the format W-XXXX. For a parallel import, the format is D-XXXX for a product from the German market, F-XXXX for a product from the French market and so on. The W in the W-Number stands for Wädenswil, the first location of the Swiss registration office for plant protection products ("Zulassungsstelle").

Note

Note that comments like "(Bew. beendet/Aut. révoquée/Aut. revocata)" were removed from the product names using the function `srppp::srppp_xml_clean_product_names`. Therefore, if a product occurs in a certain year, it does not necessarily mean that it was authorized in that year. In particular, a number of products with such comments are present in the register used as representative for 2011.

Examples

```
print(srppp_products[1:7], n = Inf)
```

<code>srppp_substances</code>	<i>Table resolving substance primary keys extracted from all years starting 2011</i>
-------------------------------	--

Description

A table of all primary keys ('pk') of the active substances, resolving them to the latest entry in any of the 'substances' tables in `srppp_list`. As the primary key scheme was changed in the XML files published by the registration authorities in the end of 2025, some substances in this table are listed twice, once with the old integer primary key, and once with the new primary key in the form of a **UUID**. For each primary key, the earliest and the latest year of occurrence are given in the columns 'earliest' and 'latest'. A substance list without these duplicates is supplied as `srppp_substances_merged`.

Usage

```
srppp_substances
```

Format

A [tibble](#).

Note

Note that a substance occurring in the register in a certain year does not necessarily mean that a product containing the substance was authorized. In particular, a number of products with such comments are present in the register used as representative for 2011.

See Also

[srppp_substances_merged](#), [srppp_products](#)

Examples

```
print(srppp_substances, n = Inf)
```

srppp_substances_merged

Table of substance names extracted from all years starting 2011

Description

A table of all unique substance names (strictly speaking, of all unique combinations of German and French names) occurring in any of the 'substances' tables in [srppp_list](#).

Usage

```
srppp_substances_merged
```

Format

A [tibble](#). The different types of primary keys (integer and uuid) are listed in the columns pk_v1 and pk_v2, respectively. The earliest and the latest year of occurrence are given in the columns 'earliest' and 'latest'.

Note

Note that a substance occurring in the register in a certain year does not necessarily mean that a product containing the substance was authorized. In particular, a number of products with such comments are present in the register used as representative for 2011.

See Also

[srppp_substances](#), [srppp_products](#)

Examples

```
print(srppp_substances_merged, n = Inf)
```

srppp_xml_dates	<i>Publication dates of the available zip files</i>
-----------------	---

Description

Publication dates of the available zip files

Usage

```
srppp_xml_dates
```

Format

A character vector of publication dates in the format YYYY-MM-DD

Examples

```
print(srppp_xml_dates)
```

srppp_xml_get.numeric	<i>Additional methods to read an XML version of the Swiss Register of Plant Protection Products</i>
-----------------------	---

Description

Additional methods to read an XML version of the Swiss Register of Plant Protection Products

Usage

```
## S3 method for class 'numeric'
srppp_xml_get(from, ...)
```

```
## S3 method for class 'Date'
srppp_xml_get(from, ...)
```

Arguments

from	A number giving a year starting from 2011 up to the current year, or one of the dates in srppp_xml_dates .
...	Currently not used

Value

An object inheriting from 'srppp_xml', 'xml_document', 'xml_node'

Examples

```
## Not run:
# The following only works if you have a collection of zipped XML dumps in a directory
# specified in the environment variable R_srppphist_idir
srppp_2015 <- srppp_xml_get(2015)
print(srppp_2015)
class(srppp_2015)

# This is the method for dates used behind the scenes
srppp_2017 <- srppp_xml_get(as.Date("2017-01-13"))
print(srppp_2017)

## End(Not run)
```

```
srppp_xml_product_use_not_authorized
```

Check if a product name indicates that its use is not authorized at the time

Description

In most cases, products are not present in the XML file (source data), if the "exhaustionDeadline" (German "Aufbrauchsfrist"), i.e. the date until stocks of the product can still be rightfully used is earlier than the date of the publication the srppp.

Usage

```
srppp_xml_product_use_not_authorized(names, year, exhaustionDeadline)
```

Arguments

names	Character vector of product names that should be checked
year	The publication years
exhaustionDeadline	The exhaustion deadlines

Details

However, in some cases, comments are amended to the product name that show that they are not authorized. This is especially true for the year 2011, where a number of products contain comments such as

- (Bew. suspendiert/Aut. suspendue/Aut. sospesa)
- (Bew. beendet/Aut. révoquée/Aut. revocata)

If such a comment is detected, the function returns TRUE.

Other comments indicate that authorisation to use the product is not clarified, such as

- [Bewilligung beendet: Wegen hängigem Gerichtsverfahren gilt einstweilen folgende Aufbrauchfrist: 31.05.2022. Verkauf ist einstweilen weiterhin möglich.]

In all but one of these cases, the date of the exhaustion deadline in the comment is later than the date of the publication, therefore the function returns FALSE. In the specific case of the product Allegro (W-5479) in the year 2022, the exhaustion deadline given in the comment is before the publication of the register. However, in that case the exhaustionDeadline in the dedicated field is 31 August 2022, so it is assumed the use of the product was authorized at least in a part of 2022.

Comments indicating that an authorization renewal is still in process are not considered to indicate that the use was not authorized at the time of publication. Examples for such comments are

- [Erneuerungsgesuch in Bearbeitung]
- (demande de renouvellement en cours)

Such comments are frequently found in the registers published in 2016 and 2017, but also occur in later years.

Finally, in the register for 2020 there are a number of products where an exhaustion date of 31 December 2019 is indicated in the comment, and in the register for 2022, there are two products where an exhaustion date of 31 December 2021 is indicated. In these cases, the function returns TRUE.

Value

Boolean (Does a comment in the name indicate that the use of the product was not authorized at the time of publication of the file?)

Examples

```
library(dplyr, warn.conflicts = TRUE)
srppp_list[["2011"]] %>% products |>
  mutate(nonauth = srppp_xml_product_use_not_authorized(name, 2011, exhaustionDeadline)) |>
  select(pNbr:soldoutDeadline, nonauth) |>
  head(20)

srppp_list[["2016"]] %>% products |>
  mutate(nonauth = srppp_xml_product_use_not_authorized(name, 2016, exhaustionDeadline)) |>
  select(pNbr:soldoutDeadline, nonauth) |>
  head(20)

# Special case Allegro: There is a conflict of information (see text)
srppp_list[["2022"]] %>% products |>
  filter(grepl("Allegro", name)) |>
  select(pNbr, wNbr, name, exhaustionDeadline) |>
  as.data.frame()
example_years <- as.character(2019:2022)
names(example_years) <- example_years
lapply(example_years, function(year) {
  sr <- srppp_list[[year]]
  sr$products |>
  mutate(nonauth = srppp_xml_product_use_not_authorized(name, year, exhaustionDeadline)) |>
  select(pNbr:soldoutDeadline, nonauth) |>
```

```
      filter(grepl("Allegro", name))
    }) |>
    purrr::list_rbind(names_to = "year")

lapply(example_years, function(year) {
  sr <- srppp_list[[year]]
  sr$products |>
  mutate(nonauth = srppp_xml_product_use_not_authorized(name, year, exhaustionDeadline)) |>
  select(pNbr:soldoutDeadline, nonauth) |>
  filter(grepl("Aufbrauch", name))
}) |>
purrr::list_rbind(names_to = "year") |>
as.data.frame() |>
head(2)
```

srppp_xml_zip_files	<i>Relative paths of the available zip files</i>
---------------------	--

Description

Relative paths of the available zip files

Usage

```
srppp_xml_zip_files
```

Format

A character vector of paths relative to 'srppp_xml_idir', named with their publication dates in the format YYYY-MM-DD

Examples

```
print(srppp_xml_zip_files)
```

Index

* datasets

- srppp_active_substances, [5](#)
- srppp_active_substances_merged, [6](#)
- srppp_compositions, [7](#)
- srppp_ingredients, [7](#)
- srppp_list, [8](#)
- srppp_obligations_spe3, [10](#)
- srppp_pests, [11](#)
- srppp_products, [11](#)
- srppp_substances, [12](#)
- srppp_substances_merged, [13](#)
- srppp_xml_dates, [14](#)
- srppp_xml_zip_files, [17](#)
- srppp_xml_product_use_not_authorized, [15](#)
- srppp_xml_zip_files, [17](#)
- tibble, [5–7](#), [10–13](#)

composition, [2](#)

grepl, [3](#)

last_approved, [3](#)

pNbr, [4](#)

product_categories, [4](#)

srppp::srppp_dm, [10](#)

srppp::srppp_xml_clean_product_names, [12](#)

srppp_active_substances, [5](#), [6](#)

srppp_active_substances_merged, [3](#), [5](#), [6](#)

srppp_compositions, [7](#), [8](#)

srppp_ingredients, [7](#), [7](#)

srppp_list, [5](#), [8](#), [12](#), [13](#)

srppp_obligations_spe3, [10](#)

srppp_pests, [11](#)

srppp_products, [5–8](#), [11](#), [13](#)

srppp_substances, [5](#), [6](#), [12](#), [13](#)

srppp_substances_merged, [5](#), [6](#), [12](#), [13](#), [13](#)

srppp_xml_dates, [14](#), [14](#)

srppp_xml_get.Date

(srppp_xml_get.numeric), [14](#)

srppp_xml_get.numeric, [14](#)