Package 'rswipl'

September 26, 2026

Type Package

Title Embed 'SWI'-'Prolog'

Version 10.1.16

Date 2026-09-25

Maintainer Matthias Gondan <Matthias.Gondan-Rochon@uibk.ac.at>

Description Interface to 'SWI'-'Prolog', <https://www.swi-prolog.org/>. This package is normally not loaded directly, please refer to package 'rolog' instead. The purpose of this package is to provide the 'Prolog' runtime on systems that do not have a software installation of 'SWI'-'Prolog'.

License FreeBSD

Imports Rcpp (>= 1.0.7)

Depends R (>= 4.3)

URL https://github.com/mgondan/rswipl

BugReports https://github.com/mgondan/rswipl/issues

LinkingTo Rcpp

RoxygenNote 7.3.2

Encoding UTF-8

SystemRequirements GNU make, CMake (>= 3.20), pandoc, libarchive, libregex, libexpat, liblzma, libzstd, liblz4, libz2, libz, libbcrypt, libutf8proc

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation yes

Author Matthias Gondan [aut, com, cre] (University of Innsbruck), Jan Wielemaker [ctb, cph] (JW is the lead developer of SWI-Prolog. See https://www.swi-prolog.org/Contributors.html for a list of contributors.), European Commission [fnd] (Erasmus+ Programme, 2019-1-EE01-KA203-051708)

Repository CRAN

Date/Publication 2026-09-25 23:50:02 UTC

Contents

clear 2
query 3
submit 4
swipl 5

Index **6**

`clear` *Clear current query*

Description

Clear current query

Usage

```
clear()
```

Value

TRUE (invisible)

See Also

`query()` for opening a query

`submit()` for submitting a query

Examples

```
query(call("member", expression(X), list(quote(a), "b", 3L, 4)))
submit() # X = a
submit() # X = "b"
clear()
```

query *Create a query*

Description

Create a query

Usage

```
query(
  q = call("member", expression(X), list(quote(a), "b", 3L, 4, TRUE, expression(Y)))
)
```

Arguments

q	an R call. The R call consists of symbols, integers and real numbers, character strings, boolean values, expressions, lists, and other calls. Vectors of booleans, integers, floating point numbers, and strings with length $N > 1$ are translated to prolog compounds !/N, %/N, #/N and $$/N, respectively.

Details

SWI-Prolog does not allow multiple open queries. If another query is open, it it is closed and a warning is shown.

Value

If the creation of the query succeeds, TRUE

See Also

submit() for submitting a query

clear() to close the currently open query

Examples

```
query(call("writeln", function(x) {sin(x)}))
submit()
clear()

query(call("=", expression(X), function(x) {sin(x)}))
submit()
clear()

query(call("member", expression(X), list(quote(a), "b", 3L, 4, TRUE, expression(Y),
  NA, NaN, Inf, NULL, NULL, function(x) {y <- sin(x); y^2}, NULL)))
submit() # X = a
submit() # X = "b"
```

```
submit() # X = 3L
submit() # X = 4.0
submit() # X = TRUE
submit() # X = expression(Y) or Y = expression(X)
submit() # X = NA
submit() # X = NaN
submit() # X = Inf
submit() # X = NULL
submit() # X = NULL
submit() # X = function(x) {y <- sin(x); y^2}))
submit() # X = NULL
submit() # FALSE (no more results)
submit() # warning that no query is open

query(call("member", expression(X), list(quote(a), "b", 3L, 4)))
query(call("member", expression(X), list(TRUE, expression(Y)))) # warning that another query is open
clear()
```

---

submit *Submit a query that has been opened with* `query()` *before.*

Description

Submit a query that has been opened with `query()` before.

Usage

```
submit()
```

Value

If the query fails, `FALSE` is returned. If the query succeeds, a (possibly empty) list is returned that includes the bindings required to satisfy the query.

See Also

`query()` for a opening a query.

`clear()` for a clearing a query.

Examples

```
query(call("member", expression(X), list(quote(a), "b", 3L, 4, expression(Y))))
submit() # X = 3L
submit() # X = 4.0
submit() # X = TRUE
submit() # X = expression(Y) or Y = expression(X)
submit() # FALSE
submit() # warning that no query is open
```

```
query(call("member", expression(X), list(quote(a), "b", 3L, 4)))
submit() # X = a
submit() # X = "b"
clear()
```

`swipl` *Invoke SWI-Prolog*

Description

This function is internally used to emulate swipl -g goal using the R program: R -e "rswipl::swipl()" -q –no-echo –args -g goal

Usage

```
swipl(sigalert = NA)
```

Arguments

`sigalert`	Use a different alert signal than SIGUSR2 (ignored on Windows)

Value

nothing useful

Index

clear, 2
clear(), *3*, *4*

query, 3
query(), *2*, *4*

submit, 4
submit(), *2*, *3*
swipl, 5