

# SWI-Prolog interface to R

Nicos Angelopoulos

April 18, 2010

## Abstract

This article documents the package R, a library to talk to R system for Statistical Computing.

## 1 R.pl – R session

### author

- Nicos Angelopoulos
- Windows-compatibility is based on work by 'JAB'

**version** 0:0:2

**See also** `examples/R/r_demo.pl`, <http://www.r-project.org/>

**copyright** Nicos Angelopoulos

**license** YAP: Artistic License

This library facilitates interaction with an R session. On the Yap system it depends on `library(System)` and on SWI on `library(process)`- part of the `clib` package. It assumes an R executable in `$PATH` or can be given a location to a functioning R executable (see `r_open/1` for details on how R is located). R is ran as a slave with Prolog writing and reading on/from the associated streams.

Multiple session can be managed simultaneously. Each has 3 main components: a name or alias, a term structure holding the communicating streams and a number of associated data items.

The library attempts to ease the translation between prolog terms and R inputs. Thus, Prolog term `x <- c(1,2,3)` is translated to atomic '`x <- c(1,2,3)`' which is then passed on to R. That is, `<-` is a defined/recognised operator. `X <- c(1,2,3)`, where `X` is a variable, instantiates `X` to the list `[1,2,3]`. Also '`Atom <- [x1,...,xn]`' translates to R code: `Atom <- c(x1,...,xn)`. Currently only vectors can be translated in this fashion.

For example :

```
rtest :-
    % for MS Windows uncomment and change the following to point to Rterm location
    % r_bin( 'C:\\Program Files\\R\\R-2.10.1\\bin\\Rterm' ).
    r_open,
    r_in( y <- rnorm(50) ),
    r_print( y ),
    r_in( x <- rnorm(y) ),
    r_in( x11(width=5,height=3.5) ),
    r_in( plot(x,y) ),
    write( 'Press Return to continue...' ),
    nl,
```

```

write( read_line_to_codes( user, _ ) ), nl,
read_line_to_codes( user, _ ),
r_print( 'dev.off()' ),
r_in( Y <- y ),
write( y(Y) ), nl,
Z = [1,2,3,4,5,6,7,8,9],
r_in( z <- Z ),
r_print( z ),
r_close.

```

See `r_demo.pl` for more examples.

### **r\_open**

Open a new R session. Same as `r_open( [] )`.

### **r\_open(+Opts)**

Open a new R session with optional list of arguments. *Opts* should be a list of the following

#### **alias(Alias)**

Name for the session. If absent or a variable an opaque term is generated.

#### **assert(A)**

Assert token. By default session opened last is the default session (see `default_r_session/1`). Using `A = z` will push the session to the bottom of the pile.

#### **at\_r\_halt(RHAction)**

R slaves often halt when they encounter an error. This option provides a handle to changing the behaviour of the session when this happens. *RHAction* should be one of `abort`, `fail`, `call/1`, `call_ground/1`, `reinstate` or `restart`. Default is `fail`. When *RHAction* is `reinstate`, the history of the session is used to roll-back all the commands sent so far. At 'restart' the session is restarted with same name and options, but history is not replayed.

#### **copy(CopyTo, CopyWhat)**

Records interaction with R to a file/stream. *CopyTo* should be one of `null`, `stream(Stream)`, `OpenStream`, `AtomicFile`, `once(File)` or `many(File)`. In the case of `many(File)`, file is opened and closed at each write operation. *CopyWhat* should be one of `both`, `in`, `out` or `none`. Default is no recording (*CopyTo* = `null`).

#### **ssh(Host)**

#### **ssh(Host, Dir)**

Run R on Host with start directory *Dir*. *Dir* defaults to `/tmp`. Not supported on MS Windows.

#### **rbin(Rbin)**

R executable location. In non MS Windows OSes, default is 'R'. In MS Windows there is no default. If the option is not present binary registered with `r_bin/1` and environment

variable `R.BIN` are examined for the full location of the R binary. In MS windows `Rbin` should point to `Rterm.exe`. Also see `r_bin/1`.

**with(*With*)**

*With* is in `[environ,restore,save]`. The default behaviour is to start the R executable with flags `--no-environ --no-restore --no-save`. For each *With* value found in *Opts* the corresponding `--no-` flag is removed.

**r\_close**

Close the default R session.

**r\_close(+*R*)**

Close the named *R* session.

**r\_in(+*Rcmd*)**

Push *Rcmd* to the default R session. Output and Errors will be printed to the terminal.

**r\_in(+*R*, +*Rcmd*)**

As `r_in/1` but for session *R*.

**r\_push(+*Rcmd*)**

As `r_in/1` but does not consume error or output streams.

**r\_push(+*R*, +*Rcmd*)**

As `r_push/1` but for named session.

**r\_out(+*Rcmd*, -*Lines*)**

Push *Rcmd* to default R session and grab output lines *Lines* as a list of code lists.

**r\_out(+*R*, +*Rcmd*, -*Lines*)**

As `r_out/2` but for named session *R*.

**r\_err(+*Rcmd*, -*Lines*, -*ErrLines*)**

Push *Rcmd* to default R session and grab output lines *Lines* as a list of code lists. Error lines are in *ErrLines*.

**r\_err(+*R*, +*Rcmd*, -*Lines*, -*ErrLines*)**

As `r_err/3` but for named session *R*.

**r\_print(+*X*)**

A shortcut for `r_in( print(X) )`.

**r\_print(+*R*, +*X*)**

As `r_print/1` but for named session *R*.

**r\_lines\_print(+*Lines*)**

Print a list of code lists (*Lines*) to the `user_output`. *Lines* would normally be read of an R stream.

**r\_lines\_print(+*Lines*, +*Type*)**

As `r_lines_print/1` but *Type* declares whether to treat lines as output or error response. In the latter case they are written on `user_error` and prefixed with `'!'`.

**r\_lines\_print(+Lines, +Type, +Stream)**

As `r_lines_print/3` but *Lines* are written on *Stream*.

**r\_lib(+L)**

A shortcut for `r_in( library(X) )`.

**r\_lib(+R, +L)**

As `r_lib/1` but for named session *R*.

**r\_flush**

Flush default R's output and error on to the terminal.

**r\_flush(+R)**

As `r_flush/0` but for session *R*.

**r\_flush\_onto(+SAliaes, -Onto)**

Flush stream aliases to code lists *Onto*. *SAliaes* should be one of, or a list of, [output,error].

**r\_flush\_onto(+R, +SAliaes, -Onto)**

As `r_flush_onto/2` for specified session *R*.

**current\_r\_session(?R)**

True if *R* is the name of current *R* session. Can be used to enumerate all open sessions.

**current\_r\_session(?R, ?S, ?D)**

True if *R* is an open session with streams *S* and data *D* (see introduction to the library).

**default\_r\_session(?R)**

True if *R* is the default session.

**r\_streams\_data(+SId, +Streams, -S)**

True if *Streams* is an R session streams structure and *S* is its stream corresponding to identifier *SId*, which should be one of [input,output,error].

**r\_session\_data(+DId, +Data, -Datum)**

True if *Data* is a structure representing R session associated data and *Datum* is its data item corresponding to data identifier *DId*. *DId* should be in [copy\_to,copy\_this,at\_r\_halt,opts].

**r\_history**

Print on user\_output the history of the default session.

**r\_history(-H)**

*H* unifies to the history list of the Rcmds fed into the default session. Most recent command appears at the head of the list.

**r\_history(?R, -H)**

As `r_history/1` but for named session *R*. It can be used to enumerate all histories. It fails when no session is open.

**r\_session\_version(-Version)**

Installed version. *Version* is of the form Major:Minor:Fix, where all three are integers.

**r\_bin(?Rbin)**

Register the default R location, *+Rbin*, or interrogate the current location: *-Rbin*. There is no default value. The value *Rbin == retract* retracts the current default location. *Rbin == test*, succeeds if an R location has been registered.

**r\_verbosity(?Level)**

Set, *+Level*, or interrogate, *-Level*, the verbosity level. *+Level* could be false (=0), true (=3) or an integer in {0,1,2,3}. 3 being the most verbose. The default is 0. *-Level* will instantiate to the current verbosity level, an integer in {0,1,2,3}.

## Index

current\_r\_session/1, 4

current\_r\_session/3, 4

default\_r\_session/1, 4

r\_bin/1, 5

r\_close/0, 3

r\_close/1, 3

r\_err/3, 3

r\_err/4, 3

r\_flush/0, 4

r\_flush/1, 4

r\_flush\_onto/2, 4

r\_flush\_onto/3, 4

r\_history/0, 4

r\_history/1, 4

r\_history/2, 4

r\_in/1, 3

r\_in/2, 3

r\_lib/1, 4

r\_lib/2, 4

r\_lines\_print/1, 3

r\_lines\_print/2, 3

r\_lines\_print/3, 4

r\_open/0, 2

r\_open/1, 2

r\_out/2, 3

r\_out/3, 3

r\_print/1, 3

r\_print/2, 3

r\_push/1, 3

r\_push/2, 3

r\_session\_data/3, 4

r\_session\_version/1, 4

r\_streams\_data/3, 4

r\_verbosity/1, 5