SWI-Prolog interface to R

Nicos Angelopoulos

August 13, 2008

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
version 0:0:1
See also examples/R/`r_demo.pl`, `http://www.r-project.org/`

license YAP: Artistic
To be done Fix starting the R process on Windows.

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. Currently it only works on Linux systems. It assumes an R executable in $PATH or can be given a location to a functioning R executable. R is run as a slave with Prolog writing and reading on/off 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]`. Currently only vectors can be translated in this fashion.

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`. In all cases apart from when CopyTo is `null`, error stream is copied to CopyTo. Default is no recording (CopyTo = `null`).

ssh(*Host***)**

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

Run R on Host with start directory Dir. Dir defaults to /tmp.

rbin(*Rbin***)**

R executable location. Default is 'R'.

with(*With***)**

With is in [environ,restore,save]. The default behaviour is to start the R executable is started 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(+*SAliases, -Onto*)

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

r_flush_onto(+*R, +SAliases, -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.

Index