

# Package ‘TinnR’

April 2, 2012

**Type** Package

**Title** Resources of Tinn-R GUI/Editor for R Environment

**Version** 1.0-5

**OS\_type** windows

**Date** 2012-04-02

**Author** Jose Claudio Faria <joseclaudio.faria@gmail.com>, based on the sources of Philippe Grosjean <phgrosjean@sciviews.org>

**Maintainer** Jose Claudio Faria <joseclaudio.faria@gmail.com>

**Depends** R (>= 2.6.0), utils, tcltk, Hmisc, R2HTML

**Description** Implements a set of customized functions, adapted from svFamily packages of Philippe Grosjean and co-authors, necessary to Tinn-R GUI/Editor for R environment

**License** GPL (>= 2)

**Repository** CRAN

**Date/Publication** 2012-04-02 17:47:19

## R topics documented:

TinnR-package . . . . .	2
trCallTip . . . . .	2
trComplete . . . . .	3
trCopy . . . . .	4
trDDEInstall . . . . .	5
trExport . . . . .	7
trObjBrowse . . . . .	8
trStartIDE . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

TinnR-package                      *Resources of Tinn-R GUI/Editor for R Environment*

---

### Description

Implements a set of customized functions, adapted from svIDE, svMisc and svIO packages of Philippe Grosjean, necessary to Tinn-R GUI/Editor for R environment.

### Author(s)

Adaptations: Jose Claudio Faria <joseclaudio.faria@gmail.com>

Original author: Philippe Grosjean <phgrosjean@sciviews.org>

### References

URL:<http://www.sciviews.org/SciViews-R>

---

trCallTip                              *Show function arguments in a human-readable way - get a call tip*

---

### Description

Code tips in Tinn-R under DDE protocol.

### Usage

```
trCallTip(code, file=NULL, onlyargs=FALSE, width=60, location=FALSE)
```

### Arguments

code	A piece of R code (in a character string) to analyze.
file	A file where to return the result ("", or NULL for none). You can use "clipboard" to send the result to the clipboard under Windows only.
onlyargs	Do we return the whole calltip or only the function arguments?
width	Reformat the calltip to with (use 0 for not reformatting it).
location	If TRUE then the location (in which package the function resides) is appended to the calltip between square brackets.

### Value

A string with the calling syntax of the function

### Note

Args() is supposed to display S3/S4 methods, and primitives adequately,... but this is not implemented yet in the current version!

**Author(s)**

Adaptations: Jose Claudio Faria <joseclaudio.faria@gmail.com>

Original author: Philippe Grosjean <phgrosjean@sciviews.org>

**See Also**

[CallTip](#), [Complete](#) [trComplete](#) [trDDEInstall](#)

---

trComplete

*Get a completion list for a R code fragment*

---

**Description**

A list with matching items is returned in a string.

**Usage**

```
trComplete(code, file=NULL, givetype=FALSE, sep="|")
```

**Arguments**

code	A piece of R code (in a character string) to analyze.
file	A file where to return the result ("", or NULL for none). You can use "clipboard" to send the result to the clipboard under Windows only.
givetype	Return also the type of each object in the completion list (possibly to use set a different icon in the list displayed by the external program).
sep	Separator to use between items in the file (if file is not NULL).

**Details**

The completion list is context-dependent, and it is calculated as if the code was entered at the command line.

If the code ends with \$, then the function look for items in a list or data.frame whose name is the last identifier.

If the code ends with @, then the function look for slots of the corresponding S4 object.

If the code ends with ::, then it looks for objects in a namespace.

If the code ends with a partial identifier name, the function returns all visible matching keywords.

There are other situations where the completion can be calculated, see the help of `rc.settings()`.

**Value**

A string with the completion list (items separated by 'sep'), or an unique string completing unambiguously the current code, starting with 'sep'.

**Note**

This function uses `utils:::win32consoleCompletion()` as completion engine, then, it makes the result compatible with old version of `trComplete()`.

**Author(s)**

Adaptations: Jose Claudio Faria <joseclaudio.faria@gmail.com>  
Original author: Philippe Grosjean <phgrosjean@sciviews.org>

**See Also**

[Complete CallTip](#), [trCallTip](#) [trDDEInstall](#)

**Examples**

```
## Not run:
## a data.frame
data(iris)
trComplete("item <- iris$")
trComplete("item <- iris[[" , TRUE)

## An S4 object
setClass("track", representation(x = "numeric", y = "numeric"))
t1 <- new("track", x=1:20, y=(1:20)^2)
trComplete("item2 <- t1@", TRUE)

## A namespace
trComplete("utils::")

## A partial identifier
trComplete("item3 <- va", TRUE)

## Otherwise, a list with the content of .GlobalEnv
trComplete("item4 <- ")

## End(Not run)
```

---

trCopy

---

*Convert an R object and copy it to the clipboard*


---

**Description**

This is a wrapper around `trExport`, using `file="clipboard"`.

**Usage**

```
trCopy(x, type="raw", objname=deparse(substitute(x)), ...)
```

**Arguments**

x	The object to copy.
type	type gives the format in which the object should be converted. The list of recognized formats for this object is obtained with <code>type = "typelist"</code> .
objname	The name of the copie object (by default, it is the name of the variable that holds it).
...	Further arguments to use with the corresponding <code>export</code> method.

**Value**

The result returned by the corresponding `export` method.

**Author(s)**

Adaptations: Jose Claudio Faria <joseclaudio.faria@gmail.com>  
Original author: Philippe Grosjean <phgrosjean@sciviews.org>

**See Also**

[trExport](#),

**Examples**

```
## Not run:  
data(iris)  
trCopy(iris, type = "ascii")  
  
## End(Not run)
```

---

trDDEInstall

*Install a DDE server (Windows only) for Tinn-R*

---

**Description**

Install and manage a DDE server to return context-sensitive calltips or completion lists to Tinn-R GUI/Editor under Windows.

**Usage**

```
trDDEInstall()
```

**Note**

This function should be used to interact with Tinn-R. Their result is returned invisibly for debugging purposes and is not intended to be use in R.

DDE is a communication protocol that exists only under Windows. Consequently, those functions cannot be used (yet) on other platforms.

On loading of the package, if the option(`use.DDE = TRUE`) is defined, the DDE server (`trDDEInstall()`) is automatically installed when the package is loaded. Also if options(`IDE = "[path.to.exe]"`) is defined, then that IDE is automatically started afterward.

**Author(s)**

Adaptations: Jose Claudio Faria <joseclaudio.faria@gmail.com>  
Original author: Philippe Grosjean <phgrosjean@sciviews.org>

**See Also**

[CallTip](#), [trCallTip](#), [Complete](#) [trComplete](#)

**Examples**

```
## Not run:
## DDE exchange protocol is available ONLY under Windows!

## Also, this cannot be run by examples() but should be OK when pasted
## into an interactive R session with the tcltk package loaded

## Here is how you can test these features under Windows
options(use.DDE = TRUE)
library(TinnR) # This should automatically start the
# DDE server named 'TclEval Tinn-R', according to the option set

# Get some data in the user workspace
data(trees)
a <- 1
b <- "some text in the first instance of R"

#####
# To test these functions in an external program, we need now
# to start a second instance of R. In this second instance, enter:
library(tcltk)
.Tcl("package require dde")
.Tcl("dde services TclEval {}")
# You should get 'TclEval Tinn-R' in the list
# if the server in the first instance is running

# Now, request a calltip for the function 'ls'
# This is done in two steps:
# 1) Execute the command 'trCallTip' with this code chunk as argument
.Tcl("dde execute TclEval Tinn-R {trCallTip {res <- ls({})}")
```

```

# 2) Retrieve the calltip from the variable tr_CallTip
.Tcl("dde request TclEval Tinn-R tr_CallTip")

# Another way to trigger DDE commands (for programs that do not support
# DDE is to use 'execdde.exe' of the tcltk2 package (see ?tk2dde)

# It is also possible to copy the calltip to a file, or to the clipboard
# by specifying it after the command (also the two additional arguments
# have their default values changed)
.Tcl("dde execute TclEval Tinn-R {trCallTip {library()} clipboard TRUE 40 TRUE}")
# Look at what is in the clipboard
cat(readClipboard(), "\n")

# The process is similar to get completion lists
.Tcl("dde execute TclEval Tinn-R {trComplete {iris$}}")
.Tcl("dde request TclEval Tinn-R tr_Complete")

# Get the list of variables in the user workspace of the first R instance
# into the clipboard (use also the other arguments to get the type of objects)
.Tcl("dde execute TclEval Tinn-R {trComplete {} clipboard TRUE { - }}")
# Look at what is in the clipboard
cat(readClipboard(), "\n")

## End(Not run)

```

---

trExport

*Export the content of an object in a foreign format*


---

## Description

trExport is a generic function (S3 based) to convert objects into various foreign formats and write it to a file or a connection.

## Usage

```

trExport(x, ...)
## Default S3 method:
trExport(x, type="raw", file, append=FALSE,
         objname=deparse(substitute(x)), ...)
## S3 method for class 'data.frame'
trExport(x, type="raw", file=NULL, append=FALSE,
         objname=deparse(substitute(x)), ...)
## S3 method for class 'matrix'
trExport(x, type="raw", file, append=FALSE,
         objname=deparse(substitute(x)), ...)

```

**Arguments**

x	The object to trExport.
type	type gives the format in which the object should be converted. The list of recognized formats for this object is obtained with type="typelist".
file	The path to the file where the converted object should be extracted.
append	Do we append this object to this file?
objname	The name of the object (by default, it is the name of the variable that holds it).
...	Further argument passed to specific trExport methods.

**Value**

Return the result from specif trExportation methods (invisibly if the file is written).

**Author(s)**

Adaptations: Jose Claudio Faria <joseclaudio.faria@gmail.com>  
Original author: Philippe Grosjean <phgrosjean@sciviews.org>

**See Also**

[trCopy](#)

**Examples**

```
## Not run:
data(iris)
trExport(iris, type="ascii", file="iris.txt")
# Inspect the file 'iris.txt', and then...
unlink("iris.txt")

## End(Not run)
```

---

trObjBrowse

*Functions to implement the an object browser (R explorer) in Tinn-R*


---

**Description**

These functions provide features required to implement a complete object browser in Tinn-R as GUI client.

**Usage**

```
trObjSearch(sep="\t", path=NULL)
trObjList(id="default", envir=.GlobalEnv, all.names=TRUE,
  pattern="", group="", all.info=FALSE, sep="\t", path=NULL)
```

**Arguments**

sep	Separator to use between items in the file (if path is not NULL).
path	The path where to write a temporary file with the requested information. Use path=NULL (default) if you don't want to pass this data by mean of a file.
id	The id of the object browser (you can run several ones concurrently, providing you give them different ids).
envir	An environment, or the name of the environment, or the position in the search() path.
all.names	Does it display all names (including hidden variables starting with '.')?
pattern	A pattern to match for selecting variables.
group	A group to filter.
all.info	Do we return all the information (envir as first column or not (by default)).

**Details**

trObjSearch() lists the search path. trObjList() lists objects in a given environment.

**Value**

Depending on the function, a list, a string, a reference to an external, temporary file or TRUE in case of success or FALSE otherwise is returned invisibly.

**Author(s)**

Adaptations: Jose Claudio Faria <joseclaudio.faria@gmail.com>  
Original author: Philippe Grosjean <phgrosjean@sciviews.org>

**See Also**

[objSearch](#), [objList](#),

---

trStartIDE	<i>Starts IDE (Windows only)</i>
------------	----------------------------------

---

**Description**

Starts the IDE under Windows.

**Usage**

```
trStartIDE()
```

**Note**

If options(IDE = "[path.to.exe]") is defined, then that IDE is automatically started afterward.

**Author(s)**

Adaptations: Jose Claudio Faria <joseclaudio.faria@gmail.com>  
Original author: Philippe Grosjean <phgrosjean@sciviews.org>

**Examples**

```
## Not run:  
## The bet way to use this function is add it in the Rprofile.site file  
## located in the folder 'etc' where R is installed.  
trStartIDE()  
  
## End(Not run)
```

# Index

## \*Topic **IO**

trCopy, 4  
trExport, 7

## \*Topic **file**

trCopy, 4  
trExport, 7

## \*Topic **misc**

trObjBrowse, 8

## \*Topic **package**

TinnR-package, 2

## \*Topic **utilities**

trCallTip, 2  
trComplete, 3  
trDDEInstall, 5  
trStartIDE, 9

CallTip, 3, 4, 6

Complete, 3, 4, 6

objList, 9

objSearch, 9

TinnR-package, 2

trCallTip, 2, 4, 6

trComplete, 3, 3, 6

trCopy, 4, 8

trDDEInstall, 3, 4, 5

trExport, 5, 7

trObjBrowse, 8

trObjList (trObjBrowse), 8

trObjSearch (trObjBrowse), 8

trStartIDE, 9