



Editor - GUI for R Language and Environment

Tinn-R Team

User Guide

Reprint III

TINN-R
EDITOR - GUI FOR R LANGUAGE AND ENVIRONMENT

AUTHOR:
FARIA, JOSÉ CLÁUDIO

COLLABORATORS:
GROSJEAN, PHILIPPE
JELIHOVSCHI, ENIO GALINKIN
PIETROBON, RICARDO
FARIAS, PHILIPE SILVA
KRAMER, PHILIPHE ALEXANDRE RIBEIRO
FARNESE, BALTHAZAR MATTOS

DEDICATION

*This user guide is dedicated to all those who
have assisted us with the Tinn-R project,
either by sending suggestions or
by contributing to its development.*

CONTENTS

DEDICATION	III
CONTENTS	IV
LIST OF FIGURES	X
LIST OF TABLES	XII
1 OVERVIEW	1
1.1 <i>Quick start</i>	2
1.2 <i>What is Tinn-R?</i>	3
1.3 <i>Why Tinn-R?</i>	4
1.4 <i>What do you get by using Tinn-R?</i>	4
1.5 <i>Do I have to pay for Tinn-R?</i>	6
1.6 <i>What was the motivation to start and maintain the Tinn-R project?</i>	6
1.7 <i>What is the sentence that we, from the development team, most like to hear?</i>	6
1.8 <i>Acknowledgment</i>	7
1.9 <i>Feedback, suggestions and bug reports</i>	7
1.10 <i>What tools were used to make this guide?</i>	7
2 BASICS	9
2.1 <i>Configuration</i>	10
Uninstall Tinn-R	10
Install and configure Tinn-R and R	10
Rgui.exe	14
Rterm.exe	14
Term interface and debug package	15
Speller installation	15
Inverse DVI search	15
Inverse PDF search	17
Ruby and Deplate	18

Pandoc	20
Python and Txt2tags	21
2.2 <i>App shortcuts (default)</i>	23
Main menu (systematically)	27
Call Tip	31
Code (or data) Completion	31
R Explorer	31
Alphabetical List of App Shortcuts	32
Shortcuts Not User Configurable	35
2.3 <i>FAQ</i>	36
What is Tinn-R?	36
Feedback, suggestions and bug report	36
Tinn-R installation	36
Hotkeys (operational system)	43
Miscellaneous	44
3 WORKING WITH	51
3.1 <i>Application options</i>	52
Main	52
Editor	54
R	57
Processing	57
3.2 <i>Main dialogs</i>	58
About	59
ASCII chart	61
Card (R)	63
Code completion	64
Comments	66
Differences	67
Find	68
Highlighters (settings)	69
Hotkeys (Send/Control/Custom)	71
Keystrokes (Editor)	73
License manager	74
Mirrors (R)	75
Open URL	76
Pandoc	77
Print	78
Print preview	79
Find and replace	79
Search in files	82
Search (trigger snippet)	84
Shortcuts (Application)	86
S/K/H manager	87

Snippets	88
Tinn-R updater	89
3.3 <i>Database</i>	90
Shortcuts (dataset - app)	90
Keystrokes (dataset - editor)	92
Hotkeys (dataset - send R)	93
Hotkeys (dataset - control R)	94
Hotkeys (dataset - custom R)	95
Snippets (dataset)	96
Comments (dataset)	97
Card (dataset R)	98
Mirrors (dataset R)	99
3.4 <i>Term interface</i>	101
IO	103
LOG	105
3.5 <i>Tools interface</i>	106
Misc (Tools)	106
Markup (Tools)	108
Results (Tools)	109
Spell (Tools)	113
Database (Tools)	114
R (Tools)	119
3.6 <i>Projects</i>	124
Overview	124
Opening the demo project	125
Creating your projects	126
Working with the project in graphical mode	126
Working with the project in text mode	126
Submit entire project (or parts) to R interpreter	127
Sets the main L ^A T _E X project file	127
Closing your projects	128
3.7 <i>Tools bar</i>	129
Show/Hide	130
Disposition	130
3.8 <i>File tabs</i>	131
3.9 <i>Selection mode</i>	132
Normal	132
Line	132
Column	133
3.10 <i>Working with marks</i>	134
Overview	134
3.11 <i>Regular expressions</i>	136
What are regular expressions?	136
Simple Matches	136

Escape Sequences	136
Character Classes	137
Metacharacters	138
Regular Expression Tutorials	142
4 MENU DESCRIPTION	143
4.1 File	144
Template	144
Copy full path to clipboard	146
4.2 Project	147
Project	147
Group(s)	148
File(s)	148
Copy full path to clipboard	148
4.3 Edit	149
4.4 Format	150
Selection	150
Word	150
Encoding	151
EOL (line ending)	151
UTF-8 (special case)	153
4.5 Marks	154
Block	154
4.6 Insert	155
R (assignment)	155
Latex	155
4.7 Search	159
4.8 Options	160
Syntax (highlighter)	160
Selection mode	162
4.9 Tools	164
Processing	164
Database	169
Backup	169
Restore	170
Macro	170
Sort	170
4.10 R	171
Toogle (start/close)	171
Toogle (active R)	171
Send	172
Control	173
Term	174
4.11 View	180

R	180
Tools	184
Toolbars	187
Line wrap	188
Split	188
Font of active control (not permanent)	188
4.12 Window	189
4.13 Web	190
R (search)	190
R (information)	190
R (Editors, Gui's and IDEs)	191
Tinn-R	191
Statistics	191
Processing	193
4.14 Help	194
File conversion (introduction)	194
5 SOME SECRETS FOR AN EFFICIENT USE	195
5.1 Introduction	196
5.2 After installation	197
5.3 Shortcuts that can make you lose the visibility of a resource	200
5.4 I'm panicking: what do I do?	201
5.5 Application options	202
General	202
Appearance	202
View	203
Editor options	204
R	206
Processing	208
5.6 Highlighters (settings)	209
5.7 Tools	210
Project	210
Database	211
R	213
5.8 Bars	218
5.9 Spell checker	219
5.10 Split	220
5.11 Editor: how to?	221
Macros: how to use?	221
Marks: how to use?	221
Editor, Tools and Term: how to arrange?	222
Active page using the keyboard: how to change?	222
Tab order: how to change?	222
Auto completion: how to use?	223

A	WHAT IS NEW?	225
A.1	<i>Versions released in 2024 (01)</i>	226
A.2	<i>Versions released in 2023 (05)</i>	230
A.3	<i>Versions released in 2022 (02)</i>	231
A.4	<i>Versions released in 2021 (15)</i>	232
A.5	<i>Versions released in 2020 (05)</i>	239
A.6	<i>Versions released in 2019 (04)</i>	241
A.7	<i>Versions released in 2018 (06)</i>	243
A.8	<i>Versions released in 2016 (01)</i>	245
A.9	<i>Versions released in 2015 (05)</i>	249
A.10	<i>Versions released in 2014 (08)</i>	252
A.11	<i>Versions released in 2013 (10)</i>	255
A.12	<i>Versions released in 2012 (03)</i>	260
A.13	<i>Versions released in 2010 (10)</i>	263
A.14	<i>Versions released in 2009 (16)</i>	266
A.15	<i>Versions released in 2008 (14)</i>	273
A.16	<i>Versions released in 2007 (26)</i>	282
A.17	<i>Versions released in 2006 (64)</i>	290
A.18	<i>Versions released in 2005 (25)</i>	312
A.19	<i>Versions released in 2004 (16)</i>	328
B	R MANUALS ON CRAN	335
C	GNU GENERAL PUBLIC LICENSE	337
INDEX		353
ABOUT THE AUTHOR AND COLLABORATORS		361

LIST OF FIGURES

1.1	Tinn-R screenshot.	3
3.1	Tinn-R: Main resources.	52
3.2	Main (Options/Application).	53
3.3	Editor options: Display.	54
3.4	Editor options: Advanced.	55
3.5	Editor options: encoding/EOL.	57
3.6	R (Options/Application): Installed, Path and options, Basic and Resources.	58
3.7	R (Options/Application): Rterm.exe and Rgui.exe.	59
3.12	Main screens of the About dialog.	59
3.8	R (Options/Application/Packages): Library, TinnRcom and sv-Socket.	60
3.9	R (Options/Application/Packages): formatR and knitr.	60
3.10	Conversion (Options/Application/Processing): Pandoc, Txt2tags, Deplate and More.	61
3.13	ASCII chart dialog.	61
3.11	Latex (Options/Application/Processing): L ^A T _E X PDE, DVI and More.	62
3.14	R card (dataset).	63
3.15	Code completion dialog.	64
3.16	Comments (dataset).	66
3.17	Differences dialog.	67
3.18	Find dialog.	68
3.19	Highlighter preferences dialog.	69
3.20	Hotkeys (database).	71
3.21	Keystrokes (dataset).	73
3.22	License manager.	74
3.23	R mirrors (dataset).	75
3.24	Open URL dialog.	76
3.25	Pandoc (document converter).	77
3.26	Print dialog.	78
3.27	Print preview dialog.	79

3.28 Find dialog	80
3.29 Find and replace dialog	80
3.30 Search in files dialog	82
3.31 Search (trigger snippets) dialog	84
3.32 Shortcuts (app) (dataset)	86
3.33 S/K/H manager	87
3.34 Snippets (dataset)	88
3.35 Tinn-R updater dialog	89
3.36 Shortcuts (dataset)	90
3.37 Keystrokes (dataset)	92
3.38 Hotkeys (dataset)	93
3.39 Hotkeys (dataset)	94
3.40 Hotkeys (dataset)	95
3.41 Snippets (dataset)	96
3.42 Comments (dataset)	97
3.43 R card (dataset)	98
3.44 R mirrors (dataset)	100
3.45 Term interface	101
3.46 IO (Term interface)	103
3.47 LOG (Term interface)	105
3.48 Tools interface	106
3.49 Markups (Tools)	108
3.50 Inilog (Tools/Results)	110
3.51 Search (Tools/Results)	111
3.52 Hex (Tools/Hex viewer)	112
3.53 Spelling (Tools)	113
3.54 Database (Tools)	114
3.55 R (Tools)	119
3.56 R card (dataset)	121
3.57 R mirrors (dataset)	122
3.58 Tinn-R: Projects	124
3.59 Tinn-R: Projects, groups and files	125
3.60 Tinn-R: Project demo	126
3.61 Tinn-R: Projects in text mode	127
3.62 Tools bar	129
3.63 Tools bar menu	130
3.64 File tabs	131
3.65 File tabs menu	131
3.66 Normal (selection mode)	132
3.67 Line (selection mode)	132
3.68 Column (selection mode)	133
3.69 Tinn-R: Marks	134

LIST OF TABLES

2.1	R interface keyboard shortcuts	23
2.2	Visualization keyboard shortcuts	24
2.3	Navigation keyboard shortcuts	24
2.4	Search/Replace and go keyboard shortcuts	24
2.5	Function keys	25
2.6	Edition keyboard shortcuts	25
2.7	Marks and go to marks keyboard shortcuts	25
2.8	Project keyboard shortcuts	26
2.9	R script edition keyboard shortcuts	26
2.10	Selection keyboard shortcuts	26
2.11	Compilation keyboard shortcuts	26
2.12	Conversion and visualization keyboard shortcuts	26
2.13	Main menu keyboard shortcuts	27
2.14	File menu keyboard shortcuts	27
2.15	Project menu keyboard shortcuts	27
2.16	Edit menu keyboard shortcuts	28
2.17	Format menu keyboard shortcuts	28
2.18	Marks menu keyboard shortcuts	28
2.19	Insert menu keyboard shortcuts	28
2.20	Search menu keyboard shortcuts	29
2.21	Options menu keyboard shortcuts	29
2.22	Tools menu keyboard shortcuts	29
2.23	R menu keyboard shortcuts	30
2.24	View menu keyboard shortcuts	31
2.25	Call tip keyboard shortcuts	31
2.26	Code completion keyboard shortcuts	31
2.27	R explorer keyboard shortcuts	31
2.28	ALT keyboard shortcuts	32
2.29	CTRL keyboard shortcuts	33
2.30	DEL keyboard shortcut	34
2.31	END keyboard shortcut	34
2.32	Function + keyboard shortcuts	34
2.33	HOME keyboard shortcut	34

2.34 SHIFT + keyboard shortcuts	35
2.35 Shortcuts Not User Configurable	35

CHAPTER 1

OVERVIEW

This chapter provides a brief overview of the Tinn-R project.

1.1 QUICK START

Let's say you don't have time to read the full user guide just yet. That's OK, we know it is huge, so let's just give you a few tips on how to get started:

- If your version is the same or above 3.0.1.0, Tinn-R does not require any special configuration. That is, the program is ready to be used. One important thing to be done before using it: set a R mirror as close as possible to where you work. For that, first click on CTRL + F8. This opens the Tools window, then click on R/Mirrors. Select the R mirror and push the button that shows an hourglass in the taskbar. The chosen repository will be the new default for all actions dependent repository (install packages, upgrade packages, etc);
- If your version is below the 3.0.1.0, read the basic instructions to [install and configure](#) R and Tinn-R: it's a small and easy to follow document;
- Choose either `Rterm.exe` or `Rgui.exe`: it takes one mouse click;
- Open *Help/Example of script (R)*: another mouse click;
- Use the *R toolbar* to send e to control R: it only takes one mouse click for each action;
- Have fun!

If you have any questions we suggest you consult this user guide.

1.2 WHAT IS TINN-R?

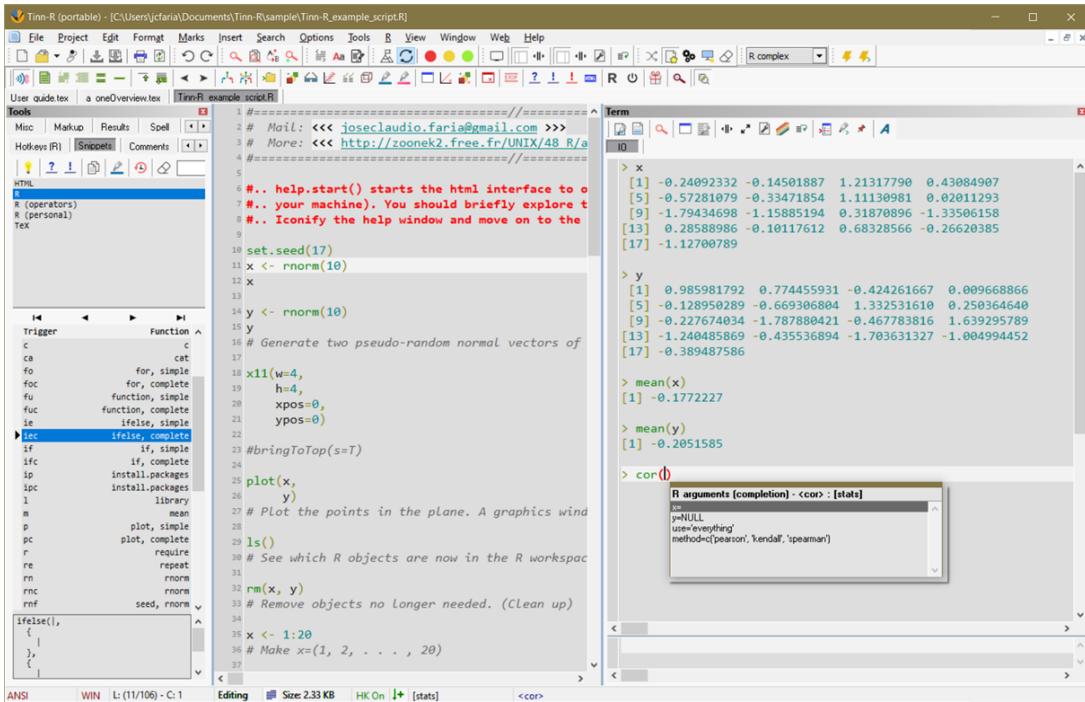


FIGURE 1.1: Tinn-R screenshot.

Tinn is a small ASCII file editor primarily intended as a better replacement for the default Notepad running under the Windows OS. The name is the recursive acronym: Tinn is not Notepad.

Tinn-R (Figure 1.1) is an extension of the original Tinn editor (ASCII and UNICODE), providing additional functionality to control R running as Rterm.exe or Rgui.exe (in SDI mode) and a whole lot of additional resources.

Tinn-R can also be thought of as feature-rich replacement of the basic script editor provided with Rgui.exe. It provides syntax-highlighting, code submission as a whole or line-by-line, in addition to many other useful tools to ease the writing and debugging of R code.

Both Tinn and Tinn-R are distributed under the **GPL 2** license or above.

1.3 WHY TINN-R?

Do you:

- like the open source initiative?
- need a simple but powerful GUI/Editor for the R environment?
- enjoy the ability to have syntax highlighting in your source code?
- need a tool that is simple to use but with the capabilities of a mighty editor?
- need a tool to work with plain text files?
- need a tool with simple commands for working with L^AT_EX, Sweave and Txt2tags?
- want to have access to the functionality of commercial and professional products?

If you have answered YES to any of the questions above, then Tinn-R is a good option for you!

1.4 WHAT DO YOU GET BY USING TINN-R?

- The ability to communicate with the R environment by sending instructions, controlling its processing, and receiving results:
 - Rterm.exe
 - Rgui.exe. From Tinn-R version 8.02.01.01 onwards, Rgui.exe must be launched from within Tinn-R so that it is properly identified at startup
- Projects:
 - Create project files to organize your work including one level of sub-folders and automatic file name sorting
 - Easy project management in graphical and text modes
- Work with unlimited length files
- Work on multiple documents at the same time, choosing between multiple-document interface (MDI) and tabbed document interface (TDI)
- Single-document window splitting
- Support to macros (volatile):

- Record
- Playback commonly used sequences
- Search and replace not restricted to your active file, but also extendable to all open files, all project files, or any folder
- View file differences with color highlighting
- Syntax highlighting, which can be set by file type
- Spell checking
- Multiple undo/redo
- Highlighted color syntax with print preview
- Ability to select:
 - Normal
 - Column
 - Lines
- Ability to bookmark:
 - Line
 - Block
- Line numbers
- Special characters
- Sort multiple variable types:
 - String
 - Data
 - Number
- Count:
 - Character
 - Words
 - Spaces
- ASCII chart
- Export with highlight to clipboard RTF
- Matching bracket highlighting

- Conversion tools:
 - Pandoc
 - Deplate
 - Txt2tags
- L^AT_EX support:
 - Edition
 - Compilation
 - Inverse DVI and PDF search.

We are constantly on the move!

1.5 DO I HAVE TO PAY FOR TINN-R?

The Tinn-R Team have been developing Tinn-R project under the GPL since june of 2003. All this time it was distributed as free software. Currently, there are corporative alternatives (paid or free as RStudio) to lead with R environment. So, to cover the costs, we are evaluating the possibility that it will be distributed as shareware in the near future.

1.6 WHAT WAS THE MOTIVATION TO START AND MAINTAIN THE TINN-R PROJECT?

Motivation to start Tinn-R: We could not find a GUI/Editor for R running in the Windows OS that would give us all the ease of use and flexibility we wanted. So, we started this project using an open source editor called [Tinn](#) as our initial platform.

Motivation to maintain Tinn-R: The most difficult phase of the project was getting started: choosing the editor; all the preliminary performance and stability tests; understanding source structure; among many other struggles. Making it to run more and more smoothly and according to our daily needs was then a natural consequence. This is all to say that the open source movement has substantially changed our lives for better. We strongly believe in making software more widely available so that more people can benefit from it. We consider Tinn-R to be our small contribution to this fantastic open source initiative.

1.7 WHAT IS THE SENTENCE THAT WE, FROM THE DEVELOPMENT TEAM, MOST LIKE TO HEAR?

Tinn-R made my life easier... thanks for creating it.

1.8 ACKNOWLEDGMENT

We would like to thank those who have assisted us with the Tinn-R project, either by sending suggestions or by contributing to its development.

1.9 FEEDBACK, SUGGESTIONS AND BUG REPORTS

Please submit feedback to [José Cláudio Faria](#). If you submit a bug report, please provide as much detail as possible. This includes indicating the Tinn-R version, your operating system (Windows 10, Windows 11, etc) and language (English, French, Portuguese). If the bug is related to an interface with R, please also indicate which version of R you are using, as well as whether you are running Rterm.exe or Rgui.exe. Ideally, please also add the content of the *Tools/Results/Ini log* interface since this will help us address the issue more promptly.

1.10 WHAT TOOLS WERE USED TO MAKE THIS GUIDE?

It were used mainly:

1. [Tinn-R](#) to edit, manager (as project) and submit to compilation the \LaTeX source files;
2. [TeXstudio](#) was also used in some parts where, to improve the productivity, it was necessary a more specialized \LaTeX editor;
3. [Vim](#) and [\LaTeX Box plugin](#) when working under Linux;
4. [WinSnap](#) to get (and edit) all images from this PDF file;
5. [FastStone Image Viewer](#) to browser and manager images;
6. [R](#) as interpreter;
7. [TeX Live](#) and [Mik\TeX](#) to compile the \LaTeX source files to final PDF format;
8. [SumatraPDF](#) as PDF viewer.

CHAPTER 2

BASICS

This chapter provides the basics about the Tinn-R project.

2.1 CONFIGURATION

This section provides information on Tinn-R configuration and associated applications.

Uninstall Tinn-R

- **ALWAYS UNINSTALL ANY PRIOR VERSION OF Tinn-R BEFORE INSTALLING A NEW ONE!** Tinn-R has its own uninstall option.
- The folder where Tinn-R project stores the ini files will not be removed when uninstalling it. Why? Because whenever you install a different version all your preferences will be preserved.
- You can check where these files are located by checking *Help/Ini files (path information)*. If you prefer, you may delete these settings by removing the entire folder manually. **All your preferences will be lost forever if you don't have a backup file.**

Install and configure Tinn-R and R

R basic configuration:

- Starting from version 1.18.XX, Tinn-R requires Rgui.exe to run in SDI mode. So, Tinn-R is not compatible neither with Rgui.exe in MDI mode (only SDI) nor with S-PLUS. The latest compatible version was the historic 1.17.2.4.
- Starting from version 2.0.0.0, Tinn-R requires R to run either Rterm.exe or Rgui.exe (in SDI mode).
- Starting from version 8.02.01.01, Rgui.exe must be launched from within Tinn-R so that it is properly identified at startup.
- You have three basic options in order to switch Rgui.exe from MDI to SDI:
 1. In Rgui.exe, select **Edit/GUI preferences...**, set SDI and click on **Save**, then **OK** without changing the name of the proposed file. Then, click **OK** or **Cancel** in the **Rgui.exe Configuration Editor** (ignore any eventual messages), and restart Rgui.exe (changes will not be taken into account in the current session).
 2. Manually edit the file **Rconsole**:

```
## Style
# This can be yes (for MDI) or no (for SDI).
MDI = no
```

3. Create a shortcut to R on your desktop (or anywhere that is convenient), and type in the switch -sdi after the . . . \Rgui.exe in the Target box. To do this, right click on your shortcut, select Properties and navigate to the Shortcut tab.

If you have any version of Tinn-R (>= 3.0.1.0) installed:

If your version is the same or above 3.0.1.0, Tinn-R does not require any special configuration. That is, the program is ready to be used. One important thing to be done before using it: set a R mirror as close as possible to where you work. For that, first click on CTRL + F8. This opens the Tools window, then click on R/Mirrors. Select the R mirror and push the button that shows an hourglass in the taskbar. The chosen repository will be the new default for all actions dependent repository (install packages, upgrade packages, etc).

If you have any version of Tinn-R (<= 2.2.0.2) installed:

1. Uninstall previous versions of Tinn-R
2. Edit the file Rprofile.site (folder *etc* where R is installed) and comment (or remove) all prior configuration scripts RELATED TO TINN-R
3. Start R
4. Install the following packages:
 - a) *TinnR*
 - b) *svSocket*
 - c) *formatR*
5. Close R
6. Install the new version of Tinn-R
7. Start Tinn-R
8. From the Tinn-R main menu, choose the option R/Configure/Permanent (*Rprofile.site*). It will write the following text to the file *Rprofile.site*:

```
##=====
## Tinn-R: necessary packages and functions
## Tinn-R: >= 2.4.1.1 with TinnR package >= 1.0.3
##=====
## Set the URL of the preferred repository, below some examples:
options(repos='http://cran.at.r-project.org/')      # Austria/Wien
#options(repos='http://cran.r.c3sl.ufpr.br/')        # Brazil/PR
#options(repos='http://cran.fiocruz.br/')            # Brazil/RJ
#options(repos='http://www.vps.fmvz.usp.br/CRAN/')  # Brazil/SP
#options(repos='http://brieger.esalq.usp.br/CRAN/') # Brazil/SP

library(utils)

## Check necessary packages
necessary <- c('TinnR',
              'svSocket',
              'formatR')

installed <- necessary %in% installed.packages()[, 'Package']
if (length(necessary[!installed]) >=1)
install.packages(necessary[!installed])

## Load packages
library(TinnR)
library(svSocket)

## Uncomment the two lines below if you want Tinn-R to always start R at start-up
## (Observation: check the path of Tinn-R.exe)
#options(IDE='C:/Tinn-R/bin/Tinn-R.exe')
#trStartIDE()

## Option
options(use.DDE=T)

## Start DDE
trDDEInstall()

## Short paths
.trPaths <- paste(paste(Sys.getenv('APPDATA'),
                           '\\\\Tinn-R\\\\tmp\\\\',
                           sep=''),
                    c('',
                      'search.txt',
                      'objects.txt',
                      'file.r',
                      'selection.r',
                      'block.r',
                      'lines.r',
                      'reformat-input.r',
                      'reformat-output.r'),
                    sep=''))
```

9. Start Rgui.exe or Rterm.exe from within Tinn-R

10. Read the content from the links below:

- *Card*: to know the shortcuts related with R and all others
- *What is new*: to know the news.

Information about how to customize the `Rprofile.site` file can be obtained from [Initialization at Start of an R Session](#) and an example of this file from [SourceForge](#).

If you have any version of Tinn-R ($\geq 2.2.0.2$) installed and configured:

1. Uninstall the prior version of Tinn-R 2.X.XX
2. Install the new version of Tinn-R
3. Run it.

If you want to install any old version of Tinn-R ($\leq 2.0.0.0$):

- **Downgrading**: rename (or delete) the folder where Tinn-R stores the ini files. The uninstall is necessary since Tinn-R does not downgrade automatically. If you encounter any problems while downgrading, check the ini folder and respective files.
- Download and install Tinn-R.
- Install the SciViews bundle, then type `guiDDEInstall()` in R and that's all!

```
> install.packages('SciViews', dep=T)
> guiDDEInstall()
```

- Perhaps the best way to get R to communicate with Tinn-R from the time it is started is to add the following commands to `./etc/Rprofile.site` in the R install directory:

```
=====
# Tinn-R: necessary packages and functions
=====
library(utils)
necessary = c('svIDE', 'svIO', 'svSocket', 'R2HTML')
installed = necessary %in% installed.packages()[, 'Package']
if (length(necessary[!installed]) >=1)
  install.packages(necessary[!installed], dep = T)
```

```
library(svIDE)
library(svIO)
library(svSocket)
library(R2HTML)
guiDDEInstall()
```

- If you chose the latter option `.../etc/Rprofile.site`, a nice additional functionality is provided by adding the two lines below BEFORE the `library(svIDE)` command:

```
options(IDE = 'C:/Tinn-R/bin/Tinn-R.exe')
options(use.DDE = T)
```

The first line tells R that you want to use Tinn-R as your IDE (Integrated Development Environment). To make this happen, you should change the path that leads to where `Tinn-R.exe` is installed if it happens to be different from the default configuration. The second line indicates that you want to start the DDE server automatically.

By doing this, Tinn-R will start automatically once you invoke R.

Rgui.exe



- Tinn-R has an icon within the *Options toolbar* containing the hint `Options: return focus to editor after send/control Rgui.exe` which enables the user to configure the focus control. When checked:
 - If the editor has the focus: it will go back to the editor after any *send to* or *R control* action;
 - Otherwise, the focus will be set to the Rgui.exe interface.

Rterm.exe

- The above-mentioned icon will be disabled with Term interface. The following will then happen:
 - If the focus is placed on the editor it will go back to the editor after any *send to* or *R control* action;
 - If the focus is placed on the Term (*IO* or *LOG*), it will be maintained in this interface (*IO*);
 - Situations above are also the case when working with two monitors.

Term interface and debug package

- Several changes were made to the debug package (1.0.2) regarding the messaging system (*stdout* and *stderr*). The default option is no longer compatible with Term interface implementation.
- The best way to make it compatible again is to add the option below to the Rprofile.site file:

```
options(debug.catfile = 'stdout')
```

- This option is automatically sent to the R interpreter from version 3.0.3.0 of Tinn-R project.

Speller installation

- To install this resource:
 1. Close Tinn-R (if it is running);
 2. Download the **dictionaries** you would like to add to Tinn-R;
 3. Install the file (for example ISpEnFrGe.exe);
- Upon start, Tinn-R will recognize all installed dictionaries. You should choose one as your default.
- Before installing new dictionaries, it is strongly recommended that you close Tinn-R.
- Another useful tool is the **UserDicEditor** which enables the editing of dictionaries.

Inverse DVI search

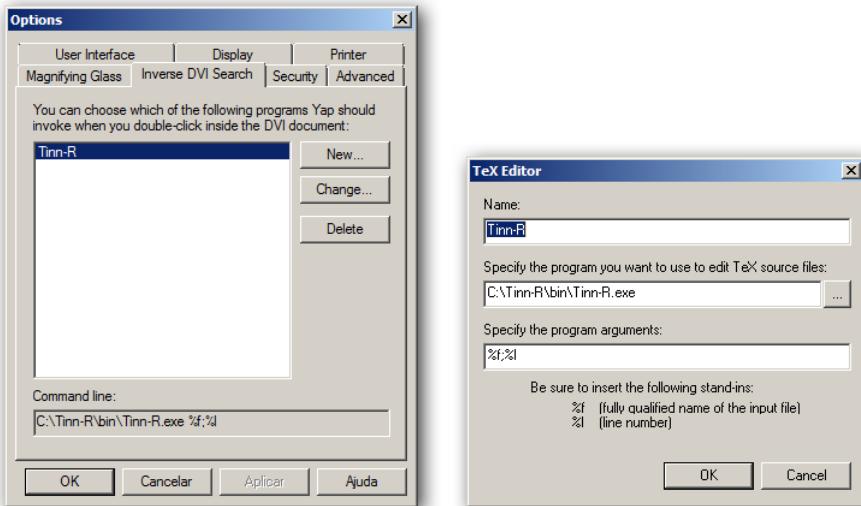
- Tinn-R is able to perform **inverse DVI search**. To get this function to work, include in your DVI previewer the path of the binary executable file for Tinn-R along with the parameters for file and line. For example, using YAP under Miktex, the configuration would be (assuming a default path for Tinn-R):

```
C:\Tinn-R\bin\Tinn-R.exe "%f;%l"
```

Note:

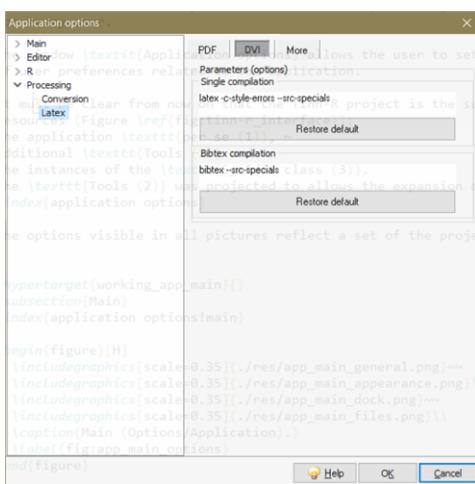
%f: file
%l: line

- Please make sure that there is no space between the parameters %f(related to file) and %l(related to line);



- It is necessary to add the parameter for TeX Live or MikTeX compilation within Tinn-R (*Options/Application/Processing/Latex/DVI*): `latex -c-style-errors -src-specials;`

```
latex -c-style-errors --src-specials
and
bibtex --src-specials
```



Inverse PDF search

- Tinn-R is able to perform inverse PDF search. To get this function to work, include in your PDF previewer (we recommend Sumatra) the path of the binary executable file for Tinn-R along with the parameters for file and line. For example, using Sumatra, the configuration would be (assuming a default path for Tinn-R):

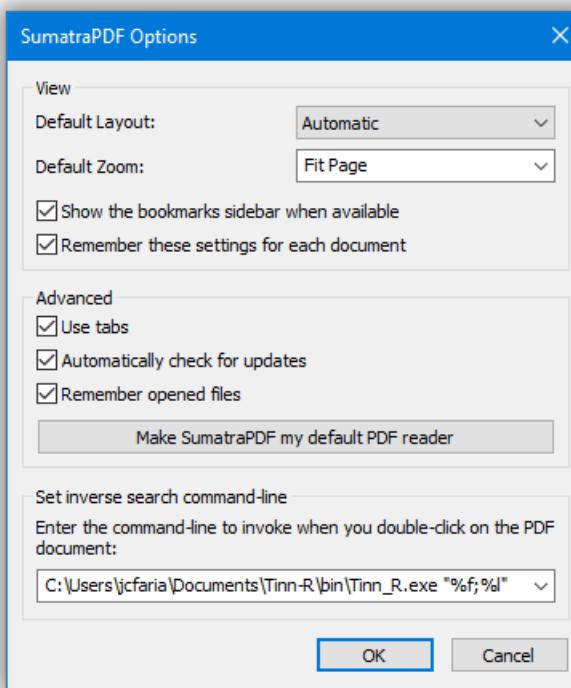
```
C:\Tinn-R\bin\Tinn-R.exe "%f;%l"
```

Note:

%f: file

%l: line

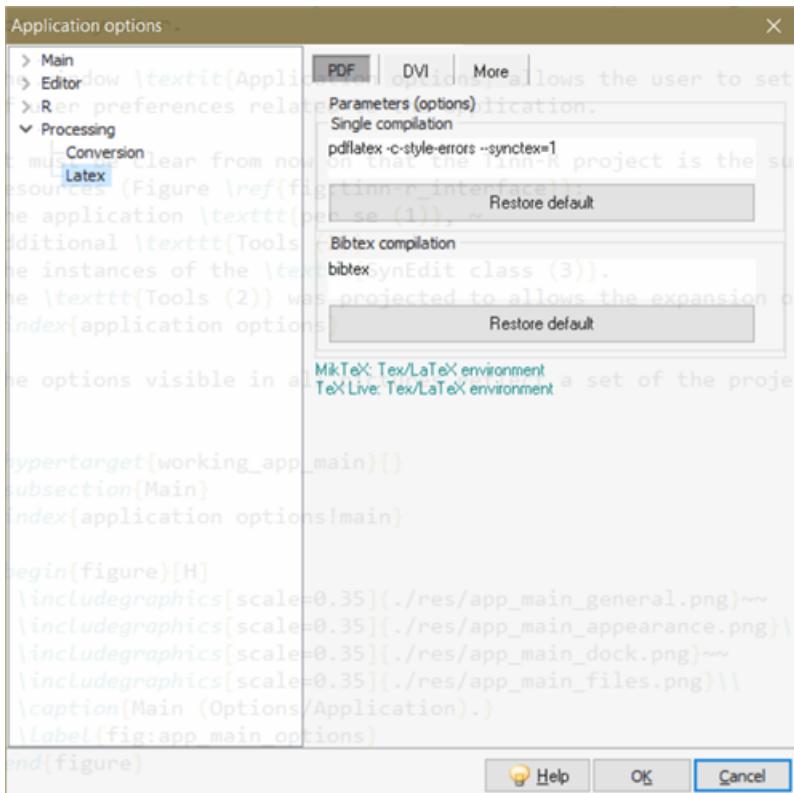
- Please make sure that there is no space between the parameters %f (related to file) and %l (related to line);



- It is necessary to add the parameter for Miktex compilation within Tinn-R (*Options/Application/Processing/Latex/PDF*):
pdflatex -c-style-errors -synctex=1;

- Tinn-R can do all of this automatically by setting the option
Restore default:

```
pdflatex -c-style-errors --synctex=1
and
bibtex
```

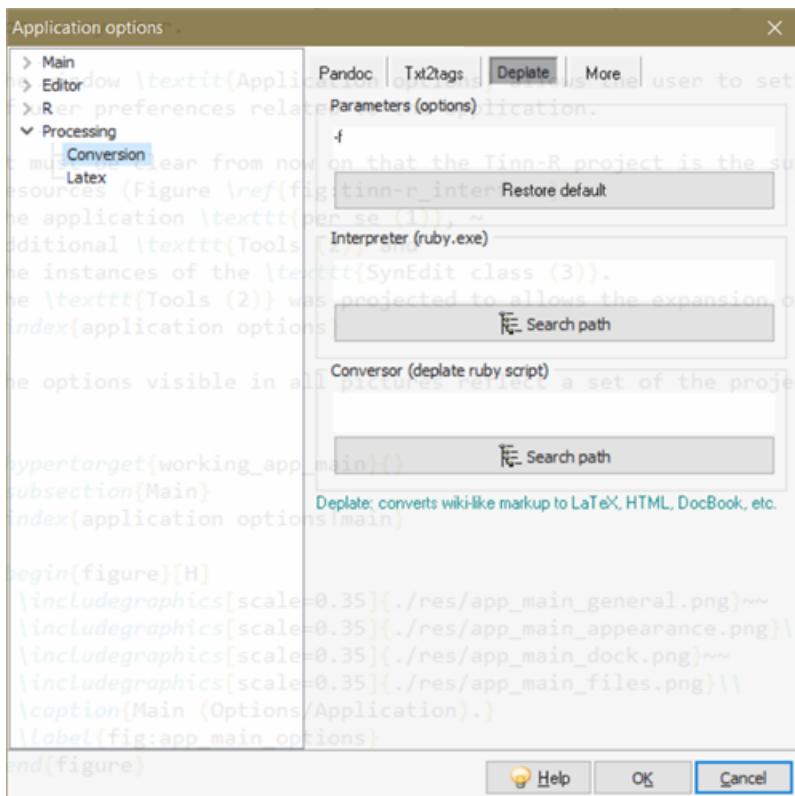


Ruby and Deplate

- **Deplate** ([user guide here](#)) is a remote Ruby based tool for converting documents written in wiki-like markup to L^AT_EX, HTML, HTML slides, or DocBook format. Deplate supports page templates, embedded L^AT_EX code, footnotes, citations, bibliographies, automatic generation of indices, tables of contents, among others. Deplate can also be used to create Web pages and, via L^AT_EX or DocBook, high-quality printouts.
- Tinn-R works with the Ruby interpreter for Windows (ruby.exe) and Ruby scripts to generate file conversation within deplate.

- To install and configure these resources follow these steps:

1. Download and unzip the **Ruby** interpreter anywhere in your computer;
2. Download and unzip **Deplate** anywhere in your computer;
3. Within Tinn-R, go to Options/Application/Processing/Conversion/Deplate and add information on parameters (-f is the default), the interpreter path (ruby.exe), and the converter (deplate.rb ruby script);

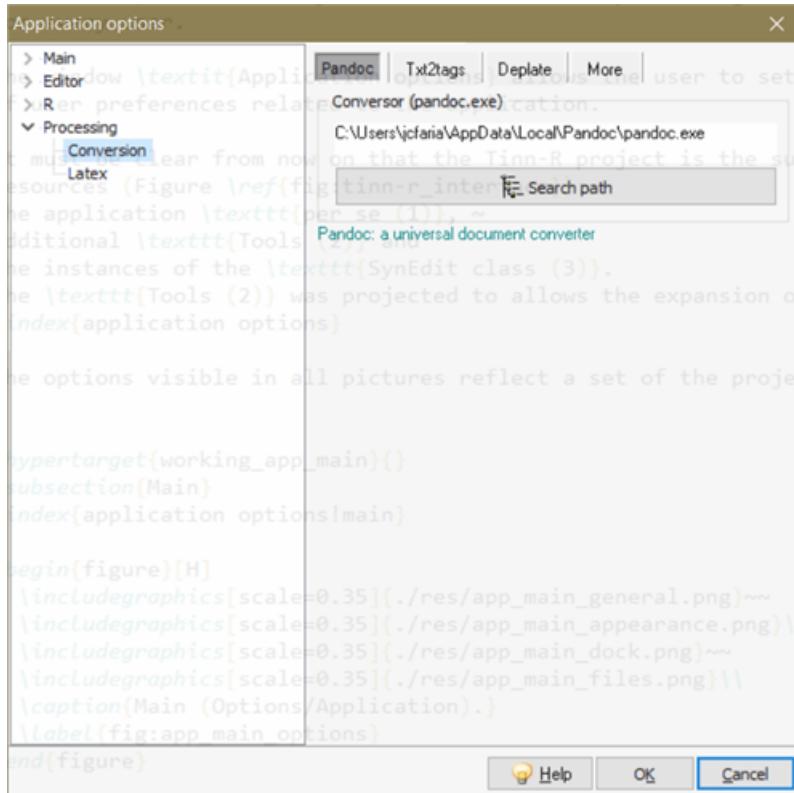


The conversion options using Deplate depends of the file extension. Are recognized: .dp, .dpt, .dplt, .deplate and .txt.

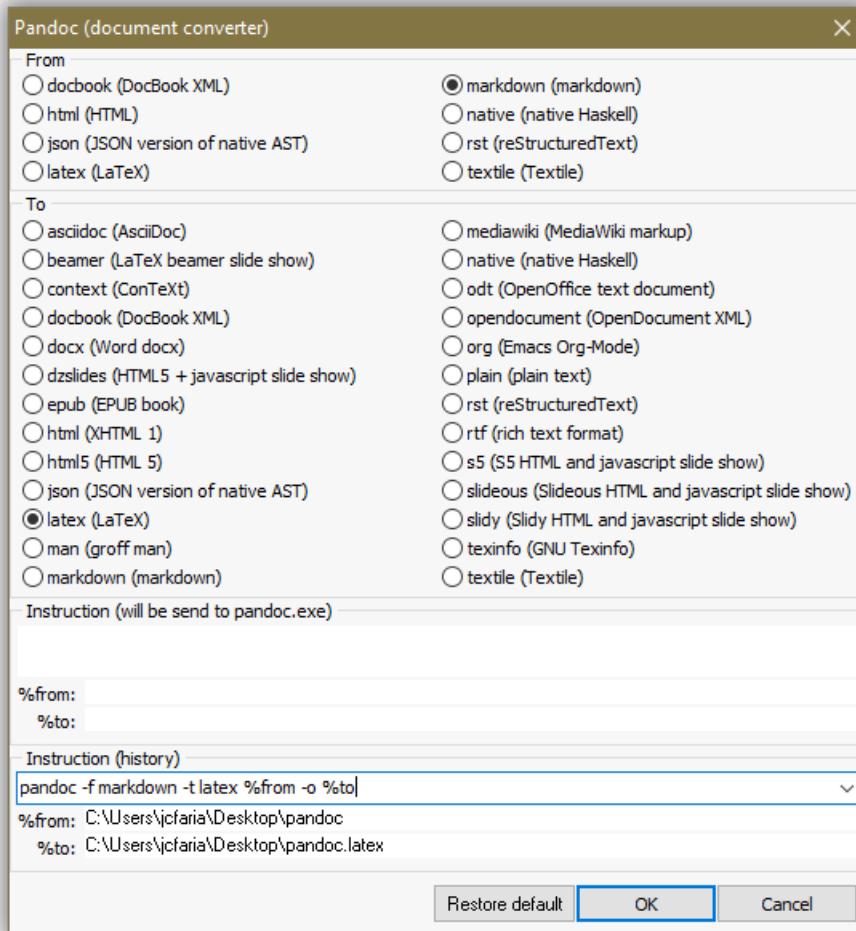
We recently observed a problem when converting files with file names with an underscore. For example deplate_intro.dplt. In these cases the file conversion is completed, but Tinn-R won't open the file since it can't find it. This pattern is caused by Deplate (a ruby script) generating a file named deplate_intro.html. *Observe that this file name contains a*

double underscore. In sum, for the time being, avoid underscores in file names when you intend to convert them later through Deplate.

Pandoc



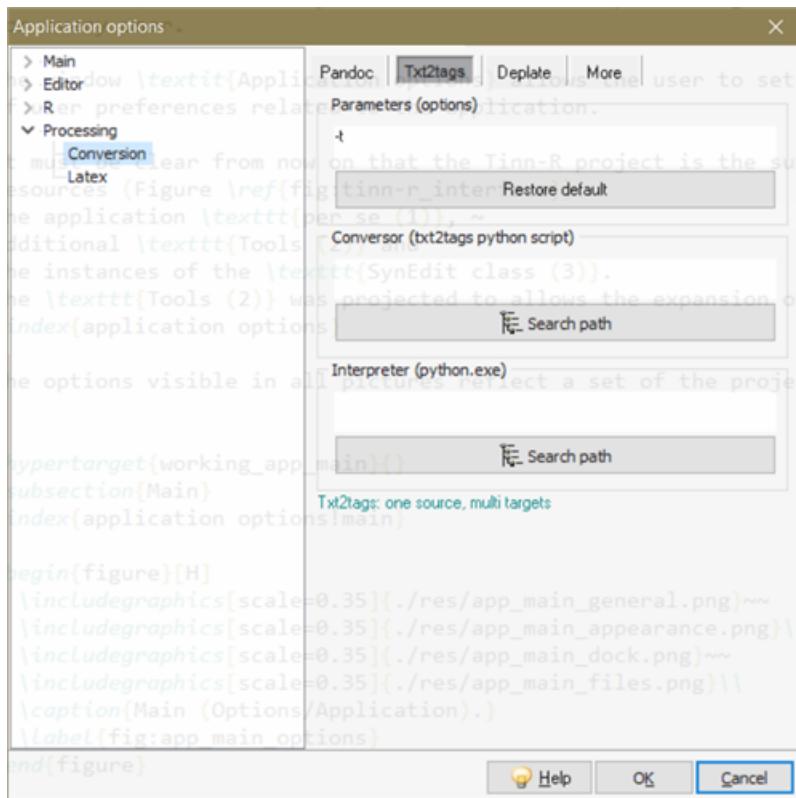
- If you need to convert files from one markup format into another, **Pandoc** is your swiss-army knife.
- To install and configure Pandoc resources, just follow these steps:
 1. **Download** and install the converter Pandoc anywhere in your computer;
 2. Within Tinn-R, go to Options/Application/Processing/Conversion/Pandoc and add information on path (`pandoc.exe`).
- Pandoc can **convert** documents:



Python and Txt2tags

- **Txt2tags** ([user guide here](#)) converts a text file with minimal and human readable markup to: HTML, XHTML, SGML, L^AT_EX, Lout, UNIX man page, Wikipedia, Google Code Wiki, DokuWiki, MoinMoin, MagicPoint (mpg), and PageMaker. It is simple and fast, featuring automatic TOC, macros, filters, include, tools, GUI, CLI, Web interfaces, translations, and extensive documentation.
- Tinn-R works with the Python interpreter for Windows (python.exe), using Python scripts to make the conversion (txt2tags).
- To install and configure Python resources, just follow these steps:

1. Download and install the **Python** interpreter anywhere in your computer;
2. Download and unzip **Txt2tags** anywhere in your computer;
3. Within Tinn-R, go to Options/Application/Processing/Txt2tags and add information on parameters (- t is the default), interpreter path (python.exe) and the converter (txt2tags python script);



The conversion options using **Txt2tags** depends of the file extension. Are recognized: **.t2**, **.t2t**, **.txt2tags**. and **.txt**.

2.2 APP SHORTCUTS (DEFAULT)

This section provides information about application shortcuts.

R interface:

LISTING 2.1: R INTERFACE KEYBOARD SHORTCUTS

```
ALT + BKSP      : Clear (History)
ALT + CTRL + SPACE : R history completion window (IO)
ALT + DOWN      : R history: down (IO)
ALT + E          : R echo (on/off)
ALT + LEFT       : Editor: set focus
ALT + PGDN       : LOG: set focus
ALT + RIGHT      : IO: set focus
ALT + UP          : R history: up (IO)
CTRL + -          : Insert <-
CTRL + +          : Insert ->
CTRL + ALT + L    : Clear (LOG)
CTRL + DELETE     : Delete the pair of active (highlighted) brackets
CTRL + ENTER      : Send line or selection (Editor and IO)
CTRL + F1          : Example on selected word*
CTRL + F2          : Open example on selected word*
CTRL + L           : Clear (IO and LOG)
CTRL + SPACE        : Tip after '(' and completion after '$'
CTRL + TAB          : Change seq. IO and LOG to the right**
CTRL + TIL          : Send current file to Sweave
F1                 : Help on selected word*
SHIFT + CTRL + B    :
SHIFT + CTRL + F1    : R help (HTML)*
SHIFT + CTRL + TAB   : Change seq. IO and LOG to the left**
* (requires an R session)
** (requires focus)
```

Visualization:

LISTING 2.2: VISUALIZATION KEYBOARD SHORTCUTS

ALT + L	: Line number (show/hide)
CTRL + F8	: Tools: visible (show/hide)
CTRL + F9	: R interface: visible (show/hide)
CTRL + F10	: R interface: minimize
CTRL + F11	: R interface: optimize
CTRL + F12	: R interface: maximize
CTRL + ALT + G	: Gutter (show/hide)
CTRL + ALT + K	: Special characters (show/hide)
CTRL + ALT + V	: Tool bars: all (show/hide)
CTRL + TAB	: Change seq. the active page (Editor, Term an Help) to the right*
CTRL + SHIFT + TAB	: Change seq. the active page (Editor, Term an Help) to the left*
SHIFT + CTRL + +	: Increase font size**
SHIFT + CTRL + -	: Decrease font size**

* (requires more than one)
** (text within the main interface)

Navigation:

LISTING 2.3: NAVIGATION KEYBOARD SHORTCUTS

CTRL + END	: End of a doc
CTRL + HOME	: Beginning of a doc
END	: End of a line
HOME	: Beginning of a line

Search/Replace and Go:

LISTING 2.4: SEARCH/REPLACE AND GO KEYBOARD SHORTCUTS

CTRL + F	: Find
ALT + SPACE	: Find again
SHIFT + CTRL + F	: Search in files
CTRL + R	: Replace
CTRL + G	: Go to line

Function Keys:

LISTING 2.5: FUNCTION KEYS

```
F1 : Help on selected word*
F2 : List the structure of selected object*
F7 : Macro record
F8 : Macro play
F9 : Clear the R console*
F10 : Close all graphic devices*
F11 : Remove all objects*
F12 : Clear all*
* (requires an R session)
```

Edition:

LISTING 2.6: EDITION KEYBOARD SHORTCUTS

```
ALT + C      : Block comment
ALT + N      : Block uncomment (first occurrence)
ALT + Z      : Block uncomment (all occurrence)
CTRL + (     : Insert (or replace) (|)
CTRL + )     : Insert (or replace) ()
CTRL + A     : Select all
CTRL + C     : Copy
CTRL + DELETE : Delete the pair of active (highlighted) brackets
CTRL + END   : End doc
CTRL + HOME  : Beginning of a doc
CTRL + I     : Block indent
CTRL + T     : Delete word
CTRL + U     : Block unindent
CTRL + V     : Paste
CTRL + X     : Cut
CTRL + Y     : Delete line
CTRL + Z     : Undo
END          : End of a line
HOME         : Beginning of a line
SHIFT + CTRL + Z : Redo
```

Marks and Go to Marks:

LISTING 2.7: MARKS AND GO TO MARKS KEYBOARD SHORTCUTS

```
CTRL + NUMBER[0..9]      : Go to mark (no numeric keypad)
SHIFT + CTRL + NUMBER[0..9] : Mark (no numeric keypad)
```

Project:**LISTING 2.8: PROJECT KEYBOARD SHORTCUTS**

```
CTRL + INS      : Add current file to selected group
SHIFT + CTRL + INS : Add file(s) to selected group (with dialog)
DEL           : Delete selected group or file
```

R Script Edition:**LISTING 2.9: R SCRIPT EDITION KEYBOARD SHORTCUTS**

```
CTRL + - : Insert <-
CTRL + - : Insert <- (numeric keypad)*
CTRL + + : Insert ->
CTRL + + : Insert -> (numeric keypad)*
* (not configurable by user)
```

Selection:**LISTING 2.10: SELECTION KEYBOARD SHORTCUTS**

```
CTRL + ALT + S : Mark block
CTRL + ALT + Z : Unmark block
CTRL + ALT + X : Unmark all
SHIFT + CTRL + C : Selection: set to column mode
SHIFT + CTRL + L : Selection: set to line mode
SHIFT + CTRL + N : Selection: set to normal mode
```

Compilation:**LISTING 2.11: COMPILE KEYBOARD SHORTCUTS**

```
CTRL + ALT + P      : Compilation: LaTeX to PDF (single)
SHIFT + CTRL + ALT + P : Compilation: LaTeX to PDF (bibtex)
CTRL + ALT + I      : Compilation: Make index (makeindex)
```

Conversion and Visualization:**LISTING 2.12: CONVERSION AND VISUALIZATION KEYBOARD SHORTCUTS**

```
SHIFT + CTRL + P : Conversion: Pandoc*
SHIFT + CTRL + H : Open current HTML files**
SHIFT + CTRL + O : Open current file (generic)
* (a universal document converter)
** (with system default browser)
```

Main menu (systematically)

LISTING 2.13: MAIN MENU KEYBOARD SHORTCUTS

```
ALT + F : File
ALT + P : Project
ALT + E : Edit
ALT + A : Format
ALT + M : Marks
ALT + I : Insert
ALT + S : Search
ALT + O : Options
ALT + T : Tools
ALT + R : R
ALT + V : View
ALT + D : Window
ALT + B : Web
ALT + H : Help
```

File:

LISTING 2.14: FILE MENU KEYBOARD SHORTCUTS

```
CTRL + N : New file
CTRL + O : Open file
CTRL + ALT + O : Open URL
SHIFT + CTRL + R : Reload file
CTRL + S : Save file
SHIFT + CTRL + S : Save file as
CTRL + W : Close file
SHIFT + CTRL + W : Close all files
CTRL + P : Print
CTRL + X : Exit;
```

Project:

LISTING 2.15: PROJECT KEYBOARD SHORTCUTS

```
CTRL + INS : Add current file to selected group
SHIFT + CTRL + INS : Add file(s) to selected group (with dialog)
```

Edit:

LISTING 2.16: EDIT MENU KEYBOARD SHORTCUTS

CTRL + Z	: Undo
CTRL + SHIFT + Z	: Redo
CTRL + C	: Copy
CTRL + X	: Cut
CTRL + V	: Paste
CTRL + A	: Copy all
ALT + C	: Block/line comment
ALT + N	: Block/line uncomment first occurrence
ALT + Z	: Block/line uncomment all occurrences
CTRL + DELETE	: Delete the pair of active (highlighted) brackets

Format:

LISTING 2.17: FORMAT MENU KEYBOARD SHORTCUTS

CTRL + F5	: Reformat R (file or selection)
CTRL + I	: Block indent
CTRL + U	: Block unindent

Marks:

LISTING 2.18: MARKS MENU KEYBOARD SHORTCUTS

CTRL + NUMBER[0..9]	: Go to mark (no numeric keypad)
SHIFT + CTRL + NUMBER[0..9]	: Mark (no numeric keypad)
CTRL + ALT + S	: Mark block
CTRL + ALT + Z	: Unmark block
CTRL + ALT + X	: Unmark all

Insert:

LISTING 2.19: INSERT MENU KEYBOARD SHORTCUTS

CTRL + -	: Insert <-
CTRL + +	: Insert ->
CTRL + J	: Snippets
SHIFT + CTRL + I	: Insert dimensional element (LaTeX)

Search:

LISTING 2.20: SEARCH MENU KEYBOARD SHORTCUTS

```
CTRL + F      : Search text
CTRL + G      : Go to
CTRL + R      : Replace text
ALT + SPACE   : Search again
```

Options:

LISTING 2.21: OPTIONS MENU KEYBOARD SHORTCUTS

```
CTRL + ALT + M : Shortcuts/Keystrokes/hotkeys (map)
ALT + E        : R echo (on/off)
CTRL + ALT + C : Auto completion \verb{([ { ' ")}
CTRL + ALT + N : Enable notification
CTRL + ALT + U : Update silently
SHIFT + CTRL + N : Selection: set to normal mode
SHIFT + CTRL + C : Selection: set to column mode
SHIFT + CTRL + L : Selection: set to line mode
```

Tools:

LISTING 2.22: TOOLSMENU KEYBOARD SHORTCUTS

```
SHIFT + CTRL + Down    : Sort strings
SHIFT + CTRL + P       : Conversion: Pandoc*
SHIFT + CTRL + H       : Open current HTML files**
SHIFT + CTRL + O       : Open current file (generic)
CTRL + ALT + P        : Compilation: LaTeX to PDF (single)
SHIFT + CTRL + ALT + P : Compilation: LaTeX to PDF (bibtex)
CTRL + ALT + I        : Compilation: Make index (makeindex)
CTRL + B              : Match bracket
F7                   : Macro/Record
F8                   : Macro/Play
* (a universal document converter)
** (with system default browser)
```

R:

LISTING 2.23: R MENU KEYBOARD SHORTCUTS

```
ALT + BKSP      : Clear (History)
ALT + DOWN      : History (next)
ALT + LEFT      : Editor: set focus
ALT + PGDN      : LOG: set focus
ALT + RIGHT     : IO: set focus
ALT + UP         : History (prior)
CTRL + ALT + D  : Set work directory (choose)
CTRL + ALT + TIL : Rterm.exe (start)
CTRL + D        : Get work directory (current)
CTRL + F1       : Example (selected)
CTRL + F10      : R interface: minimize
CTRL + F11      : R interface: optimize
CTRL + F12      : R interface: maximize
CTRL + F2       : Open example (selected)
CTRL + F9       : R interface: visible (show/hide)
CTRL + K        : Knit to LaTeX (.Rnw)
CTRL + L        : Clear (IO and LOG)
CTRL + TIL      : Sweave
CTRL + ALT + L  : Clear (LOG)
F1              : Help (selected)
F10             : Close all graphic device
F11             : Remove all objects
F12             : Clear all
F2              : List structure (selected)
F9              : Clear console
SHIFT + CTRL + B : Clear (IO and LOG)
SHIFT + CTRL + D : Set work directory (to current file path)
SHIFT + CTRL + F1 : Help
SHIFT + CTRL + K : Knit to HTML (.Rmd, Rhtml)
```

View:

LISTING 2.24: VIEW MENU KEYBOARD SHORTCUTS

```
ALT + L      : Line number (show/hide)
ALT + LEFT   : Editor: set focus
ALT + PGDN   : LOG: set focus
ALT + RIGHT  : IO: set focus
ALT + W      : Editor: line wrap (show/hide)
CTRL + ALT + K : Special characters (show/hide)
CTRL + ALT + V : Tool bars: all (show/hide)
CTRL + F10    : R interface: minimize
CTRL + F11    : R interface: optimize
CTRL + F12    : R interface: maximize
CTRL + F8     : Tools: visible (show/hide)
CTRL + F9     : R interface: visible (show/hide)
```

Call Tip

LISTING 2.25: CALL TIP KEYBOARD SHORTCUTS

```
CTRL + SPACE : Tip after '(' and completion after '$'
```

Code (or data) Completion

LISTING 2.26: CODE COMPLETION KEYBOARD SHORTCUTS

```
CTRL + SPACE : Tip after '(' and completion after '$'
```

R Explorer

LISTING 2.27: R EXPLORER KEYBOARD SHORTCUTS

```
CTRL + E      : Refresh R environment
SHIFT + CTRL + E : Refresh R explorer or filter
```

Alphabetical List of App Shortcuts

ALT + : A-Z:

LISTING 2.28: ALT KEYBOARD SHORTCUTS

ALT + A	: Format
ALT + B	: Web
ALT + C	: Block comment
ALT + CTRL + SPACE	: R history completion window (I0)
ALT + D	: Window
ALT + DOWN	: R history: down (I0)
ALT + E	: Edit
ALT + E	: R echo (on/off)
ALT + F	: File
ALT + H	: Help
ALT + I	: Insert
ALT + L	: Clear (LOG)
ALT + LEFT	: Editor: set focus
ALT + M	: Marks
ALT + N	: Block uncomment (first occurrence)
ALT + O	: Options
ALT + P	: Project
ALT + PGDN	: LOG: set focus
ALT + R	: R
ALT + RIGHT	: I0: set focus
ALT + S	: Search
ALT + SPACE	: Search again
ALT + T	: Tools
ALT + UP	: R history: up (I0)
ALT + V	: View
ALT + W	: Editor: line wrap (show/hide)
ALT + Z	: Block uncomment (all occurrence)

CTRL + :A-Z, 0-9, /, (and):

LISTING 2.29: CTRL KEYBOARD SHORTCUTS

```

CTRL + -          : Insert <-
CTRL + -          : Insert <- (numeric keypad)*
CTRL + (          : Insert (or replace) (|)
CTRL + )          : Insert (or replace) ()
CTRL + +
CTRL + +          : Insert ->
CTRL + +          : Insert -> (numeric keypad)*
CTRL + A          : Select all
CTRL + ALT + C    : Auto completion \verb{({ [ { ' "
CTRL + ALT + D    : Set work directory (choose)
CTRL + ALT + G    : Gutter (show/hide)
CTRL + ALT + I    : Compilation: Make index (makeindex)
CTRL + ALT + K    : Special characters (show/hide)
CTRL + ALT + L    : Clear (LOG)
CTRL + ALT + M    : Shortcut/keystrokes/hotkeys (map)
CTRL + ALT + N    : Enable notification
CTRL + ALT + P    : Compilation: LaTeX to PDF (single)
CTRL + ALT + S    : Mark block
CTRL + ALT + U    : Update silently
CTRL + ALT + V    : Tool bars: all (show/hide)
CTRL + ALT + X    : Unmark all
CTRL + ALT + Z    : Unmark block
CTRL + B          : Match bracket
CTRL + C          : Copy
CTRL + D          : Get work directory (current)
CTRL + DELETE     : Delete the pair of active (highlighted) brackets
CTRL + E          : Refresh R environment
CTRL + END         : End doc
CTRL + ENTER       : Send line or selection (Editor and IO)
CTRL + F          : Find
CTRL + F1         : Example on selected word**
CTRL + F2         : Open example on selected word*
CTRL + F10        : R interface: minimize
CTRL + F11        : R interface: optimize
CTRL + F12        : R interface: maximize
CTRL + F5          : Reformat R (file or selection)
CTRL + F8          : Tools: visible (show/hide)
CTRL + F9          : R interface: visible (show/hide)
CTRL + G          : Go to line
CTRL + HOME        : Beginning doc
CTRL + I          : Block indent
CTRL + INS         : Add current file to selected group
CTRL + L          : Clear (IO and LOG)
CTRL + N          : New file
CTRL + NUMBER[0..9] : Go to mark (no numeric keypad)
CTRL + O          : Open file
CTRL + P          : Print
CTRL + R          : Replace text
CTRL + S          : Save file
CTRL + SPACE       : Tip after '(' and completion after '$'
CTRL + T          : Delete word

```

```
CTRL + TAB      : Change seq. the active page
                  (Editor, Term an Help) to the right ***
CTRL + TIL      : Send current file to Sweave
CTRL + U        : Block unindent
CTRL + V        : Paste
CTRL + W        : Close file
CTRL + X        : Cut
CTRL + Y        : Delete line
CTRL + Z        : Undo
* (not configurable by user)
** (requires an R session)
*** (require more than one)
```

DEL:

LISTING 2.30: DEL KEYBOARD SHORTCUT

```
DEL (Project) : Delete selected group or file
```

END:

LISTING 2.31: END KEYBOARD SHORTCUT

```
END : End line
```

Function + : 1-12:

LISTING 2.32: FUNCTION + KEYBOARD SHORTCUTS

```
F1  : Help on selected word*
F2  : List structure of selected object*
F7  : Macro record
F8  : Macro play
F9  : Clear the R console*
F10 : Close all graphic devices*
F11 : Remove all objects*
F12 : Clear all*
* (requires an R session)
```

HOME:

LISTING 2.33: HOME KEYBOARD SHORTCUT

```
HOME : Beginning line
```

SHIFT + : A-Z, 0-9 and /:

LISTING 2.34: SHIFT + KEYBOARD SHORTCUTS

SHIFT + ALT + D	: Set work directory (to current file path)
SHIFT + CTRL + -	: Decrease font size
SHIFT + CTRL + +	: Increase font size*
SHIFT + CTRL + ALT + P	: Compilation: LaTeX to PDF (bibtex)
SHIFT + CTRL + B	: Clear (IO and LOG)
SHIFT + CTRL + C	: Selection: set to column mode
SHIFT + CTRL + D	: Set work directory (to current file path)
SHIFT + CTRL + Down	: Sort strings
SHIFT + CTRL + E	: Refresh R explorer or filter
SHIFT + CTRL + F	: Search in files
SHIFT + CTRL + F1	: R help (HTML)*
SHIFT + CTRL + I	: Insert dimensional element (LaTeX)
SHIFT + CTRL + INS	: Add file(s) to selected group (with dialog)
SHIFT + CTRL + L	: Selection: set to line mode
SHIFT + CTRL + N	: Selection: set to normal mode
SHIFT + CTRL + NUMBER[0..9]	: Mark (no numeric keypad)
SHIFT + CTRL + H	: Open current HTML files****
SHIFT + CTRL + P	: Conversion: Pandoc***
SHIFT + CTRL + R	: Reload file
SHIFT + CTRL + S	: Save file as
SHIFT + CTRL + TAB an Help) to the left	: Change seq. the active page (Editor, Term
SHIFT + CTRL + O	: Open current file (generic)
SHIFT + CTRL + W	: Close all files
SHIFT + CTRL + Z	: Redo
* (extensive to text of the main interface)	
** (requires an R session)	
*** (a universal document converter)	
**** (with system default browser)	

Shortcuts Not User Configurable

LISTING 2.35: SHORTCUTS NOT USER CONFIGURABLE

CTRL + ENTER (Editor, IO)	: Send current line or selection to R
CTRL + SPACE	: R Tip after '(' and completion after '\$'
CTRL + ALT + SPACE	: R history

CTRL + D : Get work directory (current) SHIFT + CTRL + D : Set work directory (to current file path) CTRL + ALT + D : Set work directory (choose)

2.3 FAQ

This section provides information on Frequently Asked Questions (FAQ).

What is Tinn-R?

- **Tinn** is a small ASCII file editor primarily intended as a better replacement for the default Notepad running under the Windows OS. The name is the recursive acronym: Tinn is not Notepad.
- **Tinn-R** is an extension of the original Tinn editor (ASCII and UNICODE), providing additional functionality to control **R** running as Rgui.exe (in SDI mode), Rterm.exe and a whole lot of additional resources.
- Tinn-R can also be thought of as feature-rich replacement of the basic script editor provided with Rgui.exe. It provides syntax-highlighting, code submission as a whole or line-by-line, in addition to many other useful tools to ease the writing and debugging of R code.
- Both Tinn and Tinn-R are distributed under the **GPL 2** license or above.

Feedback, suggestions and bug report

Please send your feedback to **JosÃ© ClÃ¡udio Faria**. If you submit a bug report, please provide as much detail as possible. This includes indicating the Tinn-R version, your operating system (Windows XP, Windows 7, etc), and language (English, French, Portuguese). If the bug is related to an interface with R, please also indicate which version of R you are using, as well as whether you are running Rterm.exe or Rgui.exe. Ideally, please also add the content of the *Tools/Results/Ini log* interface since this will help us address the issue more promptly.

Tinn-R installation

See details ...

Where can I get the latest version of Tinn-R?

- The latest version of Tinn-R can be downloaded from [Web page](#) or [SourceForge](#).

How do I install Tinn-R?

- Tinn-R uses a classical method of installation and runs on all versions of the Windows OS. You need administrative rights to install, although you can install it as a regular user provided you have write permission on the directory where you will perform the installation. If you have problems, please contact your computer or network administrator.
- Note that if you install Tinn-R you will likely want to use it along with R, and so R must be installed separately. R can be obtained from [here](#).

.paths? (a very recurrent FAQ)

If your version is the same or above 3.0.1.0, Tinn-R does not require any special configuration. That is, the program is ready to be used. One important thing to be done before using it: set a R mirror as close as possible to where you work. For that, first click on CTRL + F8. This opens the Tools window, then click on R/Mirrors. Select the R mirror and push the button that shows an hourglass in the taskbar. The chosen repository will be the new default for all actions dependent repository (install packages, upgrade packages, etc).

If you have any version of Tinn-R (<= 2.2.0.2) installed, you need to define your .paths correctly in the file R/etc/Rprofile.site. With the "new" versions of Windows (Vista, 7 e 8) write permissions are more annoying to deal with, and you have to manually give yourself permission to write to the file/directory, before you can change and save the files. So, to fix it, you have to enable write access to the Rprofile.site file and then configure R to run.

- Locate your R install. The default is C:\Program Files\R\R-X.X\.
- The file you need write access to is Rprofile.site, located in the folder etc. A good option is to give write access to all of C:\Program Files\R\R-X.X\.
- Select the directory you want to give yourself permission to write in, right click, properties, security, and then it depends on your version of Windows. Once you've enabled writing and saved, you'll need to go back to Tinn-R. In the main menu choice: R/Configure/Permanent(Rprofile.site).
 - It should autoload and fix your Rprofile.site file.
 - The one additional change you would make is to either comment out or change your default repository.

- Useful links about:
 1. Hidden (files and folders)
 - Microsoft
 - About.com
 - Computer Hope
 2. Permission (files and folders)
 - Microsoft
 - Seven Forums
 - Addictive Tips

Can I get the source code?

- Yes. You can get and modify the source code of Tinn-R as well as redistribute your changes as long as you respect the terms of the GPL license. The source code is available from [SourceForge](#).

How can I add a shortcut to Tinn-R in the start menu or in the desktop?

- This is automatically done by the installer. If you want to do it manually later on, here are the steps:
 1. Under object explorer, right-click the file Tinn-R.exe and select Create shortcut;
 2. Drag & drop this shortcut to the desktop or wherever you might want to place it.

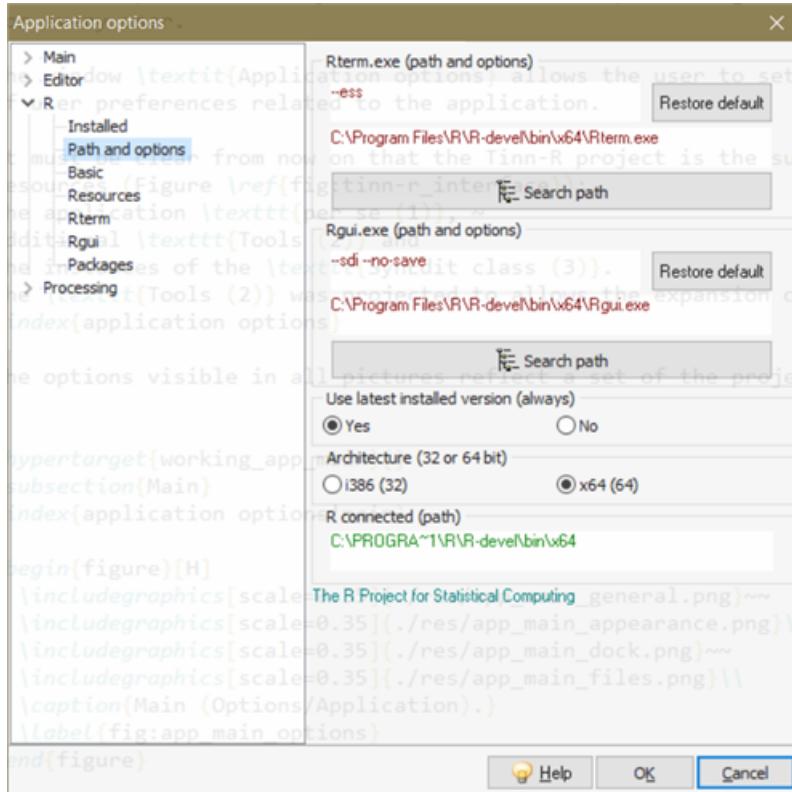
Can I save or reuse my preferences on another computer?

- **Desktop version:**
 1. Copy the folder where Tinn-R stores all user preferences (main menu: Help/Ini files (path information));
 2. Paste it inside the directory with the Tinn-R folder;
- **Portable version.** The subfolder app_data stores all user preferences. In this case, you can load the application on the memory device or make it available in the cloud.

How can I open a file in Tinn-R by double-clicking it under Windows Explorer?

- You need to register Tinn-R as the default program to open files with a given extension. You can either check this option during installation or follow the steps below:
 1. In order to open *.R files (R scripts) with Tinn-R, locate one such file in your disk;
 2. Right-click this file and select `Open with/Choose program...` in the context menu;
 3. Click `Browse` in the `Open with` dialog box and then select `Tinn-R.exe`;
 4. Make sure the option `Always use the selected program to open this kind of file` is selected;
 5. Click `OK`.
- Now, when you double-click on a *.R file in the Windows explorer, it will be opened in Tinn-R.

How to define the starting Rgui.exe from within Tinn-R?



- You must start your preferred Rgui.exe directly from Tinn-R. To do that, go to Options/Application/R/Path. **From Tinn-R version 8.02.01.01, Rgui.exe must be launched from within Tinn-R so that it is properly identified at startup.**
- At the bottom of the dialog box, you can determine the path of the Rgui.exe executable to start from within Tinn-R. Select Rgui.exe from, for instance, C:\Program Files\R\R-X.X.X\bin\Rgui.exe).

Note: to use R from within Tinn-R, you must first install it from
<http://cran.r-project.org>

- With Rgui.exe, you must choose SDI mode at Edit/GUI preferences.

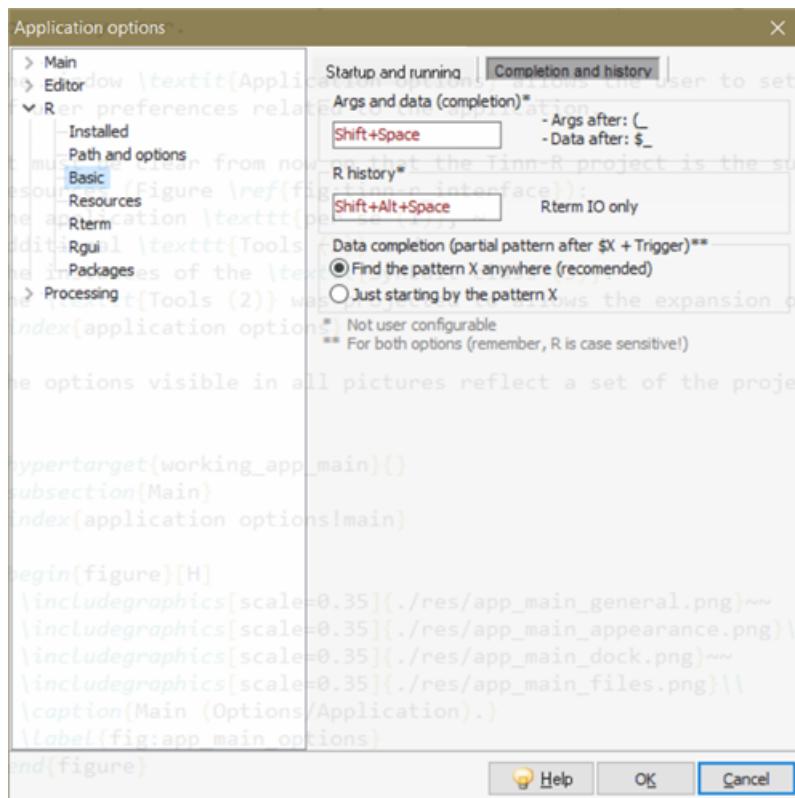
Can I define Tinn-R as the default editor for R objects?

- No, currently, it does not have that capability. In order to do that, just use the internal script editor of Rgui.exe to edit() or fix() R objects.

Can I use Emacs or WinEdt style for syntax highlighting color?

- Just set your preferred color scheme in Options/Colors (preference). To change color scheme on other computers, just use the Options/Backup/Restore system options configuration functions (*See details ...*).

What does "Args and data" mean in Options/Application/R/General/Basic?



- Args are tooltips displaying the syntax of the currently used R function.
- By default, if you enter the name of a function followed by an open bracket such as `'sd(|)`' in a R code document, then Tinn-R recognizes that you call the `sd` R function and reminds you of its syntax by showing the following tip: `x, na.rm=FALSE`, that is, `sd` accepts two

arguments: `x`, and `na.rm` with the latter having `FALSE` as the default value.

- Tinn-R uses a database with the syntax of most common functions in R. However, neither functions in additional packages nor your custom functions are cached in this database. Adding them all manually is tedious.
- Tinn-R therefore offers a second mechanism: direct requests to R. This is accomplished through TCP/IP protocols, using functions automatically loaded when you start the TinnR package you downloaded from CRAN. ([See details ...](#)).

On some computers, the delay for synchronization might need to be adjusted. If Tinn-R seems to freeze while querying R for tips and you get no results, increase the value a bit by setting Options/Application/Main/-General/Computational synchronization (delay).

Can I start R and Tinn-R all at once?

- There are many ways to accomplish this, but here is one: first, configure R so that it understands that you want to use Tinn-R as your IDE (Integrated Development Environment). In order to do that, start a new R session and add the following command:

```
> options(IDE = "C:/Tinn-R/bin/Tinn-R.exe")
```

Replace the path by the present location of Tinn-R.exe on your computer if different from the location above. Then you will indicate that you want to start the DDE server automatically by setting (valid only for versions prior to 3.0.1.0, because this communication protocol was deleted from the project, the only one remaining is TCP/IP):

```
> options(use.DDE = TRUE)
```

At this point, Tinn-R will be automatically started when you load svIDE, at the same time as the R call-tip server is installed (valid only for versions prior to 3.0.1.0):

```
> library(TinnR)
```

If those steps work well in manual mode, but you now want them to run whenever you start R, edit the `Rprofile.site` file (located in the `\etc\` subdirectory of R). File location varies, but it should be under something like `C:\Program Files\R\R-X.X.X\etc\Rprofile.site`. Add the above-mentioned three lines of code at the end of the `Rprofile` file (valid only for versions prior to 3.0.1.0).

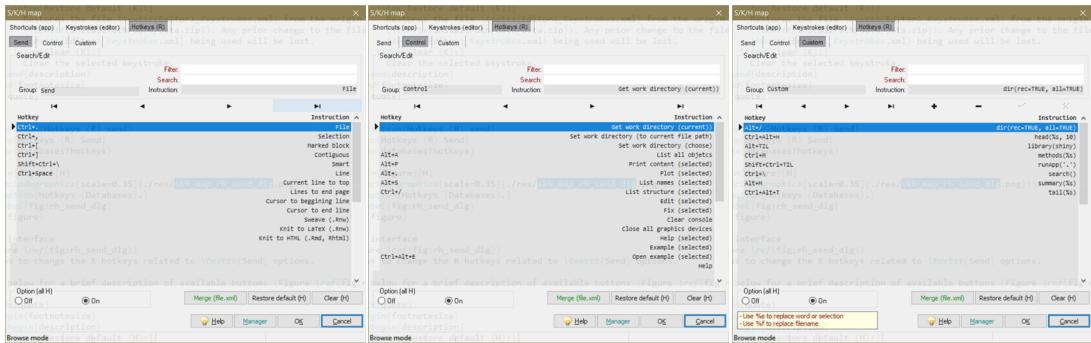
```
options(IDE = "C:/Tinn-R/bin/Tinn-R.exe")
options(use.DDE = TRUE)
library(TinnR)
```

A copy of `Rprofile.site` file created by JosÃ© ClÃ¡udio Faria can be obtained from [SourceForge](#), which you can adapt according to your needs. To make sure that everything works well and smoothly, close both R and Tinn-R and restart R. Tinn-R should start concomitantly. Now, create a very simple function in R such as:

```
> cube <- function(x) x^3
```

Switch to Tinn-R and type: `'cube(1)'`. You should get a call-tip displaying `x` if the R call-tip server was correctly installed.

Hotkeys (operational system)



What is the difference between hotkeys (operational system) and shortcuts in Tinn-R?

- The hotkeys are related to the operational system. In other words, they work without the focus on Tinn-R, whereas the shortcuts will only work with the focus on the Tinn-R interface.

How do I define hotkeys for R tools in Tinn-R?

- Go to Tools/Database/Hotkeys (R). Once there, define your favorite hotkeys for the various R tools and make sure to activate them (Option -> Active).

Is there a shortcut for cycling through opened files?

- Yes, you can use CTRL + TAB to go to next file, and CTRL + SHIFT + TAB to go to previous ones when several files are loaded simultaneously in Tinn-R.

Is there a shortcut for <- and -> for the S/R languages?

- The (non user configurable) shortcut for -> is CTRL + ADD key (numeric keypad). Similarly, CTRL + SUBTRACT (numeric keypad) is a shortcut for <-..-> and <- , all being assignment symbols in the S/R languages.

Miscellaneous

I am editing a table. Can I select text in column mode?

- Yes you can, but you must first make sure that this option is selected. Go to Options/Application/Editor/Advanced options tab and check (x) Alt sets column modes. Once this is done, by pressing Alt key while selecting your text with the mouse in Tinn-R, the selection will be done in column mode.
- Another option is to change the selection mode to column in a permanent way. This is done through the menu Options/Selection mode or by clicking on the selection mode place at the status bar. The available options are: smNormal, smLine and smColumn.

Can I define bookmarks to facilitate the navigation through my documents?

- Yes, you can define up to 10 bookmarks in each of your opened documents. To define the bookmark, use **CTRL + SHIFT + [0-9]** (a key from 0 to 9). Then, to go to the corresponding bookmark just use **CTRL + [0-9]**. A visual indicator appears in the right margin at the location of your bookmarks to remind you where they are.

What is the left gutter used for?

- In Tinn-R, bookmarks are visually displayed in the left gutter (use **CTRL + SHIFT + [0-9]** to set bookmarks and then use **CTRL + [0-9]** to navigate to them). It also displays the respective line numbers. You must set **gutter Visible** in **Options/Application/Editor/Display** tab (and also **Show line numbers**) to activate this feature.

Can I run my code step-by-step?

- Yes, but for more convenient use of this function, you must place Tinn-R and R side by side on your screen and click on the 'Send line' icon with the mouse (seventh button from the left on the R toolbar).
- If you use a shortcut, you can just submit one line since the R console gets the focus when code is sent to R. Alternatively, you can set Tinn-R as a topmost window on top of R using **Options/On top**. The downside is that Tinn-R will permanently hide the R console and there is a chance that you won't see a part of the output generated in R during your step-by-step code execution.

Is there a graphical debugger for my R functions?

- Not yet, but you can download the excellent **debug** package from CRAN and use the **mttrace** function available from there.

What is the Tools panel?

- It is a panel you can open at either the left or the right side of your text. It helps you to manage large projects with multiple documents. The **Computer** tab allows you to explore your computer disks and

open one or several files without using File/Open, or switching to the Windows file explorer. The Project tab is a convenient manager for all files collected in a given project.

Can I copy and paste syntax highlighted R code in Word, Writer or other RTF editor?

- Syntax highlighted code enhances the code's visibility. It is convenient in the code editor, but could also be useful for pieces of code presented elsewhere. Tinn-R allows you to copy the code while keeping syntax highlighting color through Edit/Copy selection to clipboard (RTF formatted).

How can I fix incorrect icon displays on Windows after I have installed a new version of Tinn-R?

- If you get an incorrect icon displayed on Windows after installing a new version of Tinn-R, just proceed as follows:
 - In order to accelerate the display of program or file icons, Windows stores images in the ICON CACHE (ShellIconCache), a hidden icon cache file in your Windows directory.
 - Sometimes the icon of the object changes, but Windows still shows the old icon instead of the new one. To solve this problem, use the shareware program called [IconChanger](#).
 - If you have just installed Tinn-R with a new icon but Windows has not changed the image yet, use IconChanger and select REBUILD ICON CACHE. If that still doesn't work, then select REMOVE ICON CACHE.
 - If you have selected REBUILD the icon cache will start rebuilding from scratch. If you have selected REMOVE, you will see a warning message. Select YES and then restart your computer.

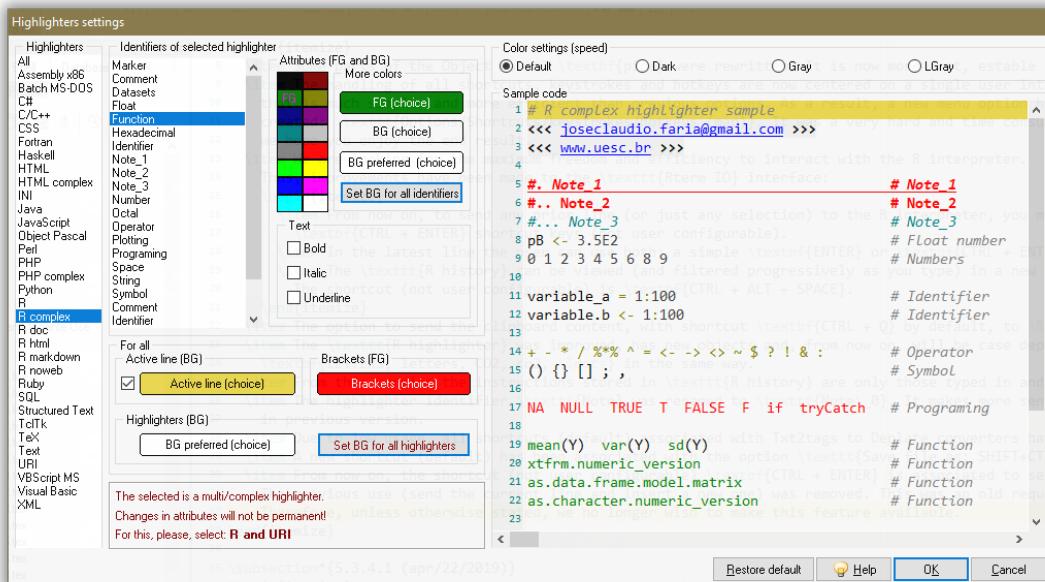
Basic instructions about focus control:



- Tinn-R has a button within the *Options toolbar* with the hint (*Options: return focus to editor after send/control Rgui.exe*) that enables the user to configure out the focus control. When this option is checked Tinn-R will display the following behavior:

- If the editor has the focus, it will go back to the editor after any *send to* or *R control* action, otherwise it will remain on Rgui.exe. This is also true when working with a dual-monitor display.
- If the Term has the focus, it will be maintained in this interface (*IO*), disregarding the *Options: return focus to editor after send/control Rgui.exe*.

Why Tinn-R doesn't remember my syntax color preferences?



Tinn-R has seven multi-highlighters: *HTML complex*, *PHP complex*, *R complex*, *R doc*, *R html*, *R markdown* and *R noweb*, with each one behaving as follows:

1. *HTML complex* = *HTML & JavaScript*
2. *PHP complex* = *HTML & JavaScript & PHP*
3. *R complex* = *R & URI ('<<<' begin URI; '>>>' end URI)*
4. *R doc* = *TeX & R ('>>=' begin R; '@' end R)*
5. *R html* = *HTML & R ('!!--begin.rcode' begin R; 'end.rcode-->' end R)*
5. *R markdown* = *URI & R ('`{` begin R; `}` end R)*
6. *R noweb* = *TeX & R ('>>=' begin R; '@' end R)*

URI : Uniform Resource Identifiers.

R complex : The main syntax is R, '<<<' and '>>>' are the tags enabling the user to insert a block of URI syntax.

R doc : The main syntax is TeX, '>>=' and '@' are the tags enabling

the user to insert a block of R syntax.

R html : The main syntax is HTML, '<!--begin.rcode' and 'end.rcode-->' are the tags enabling the user to insert a block of R syntax.

R markdown: The main syntax is URI, '```{' and '```' are the tags enabling the user to insert a block of R syntax.

R noweb : The main syntax is TeX, '>>=' and '@' are the tags enabling the user to insert a block of R syntax.

These highlighters do not establish priorities when you set the syntax color preferences. Thus, if you change the color preferences for any of these multi-highlighters these settings will be valid only in the current Tinn-R session and will not be saved when Tinn-R is closed. If you want to make these changes permanent, just set the preferences from all simple highlighters.

How do I set a block as marked?

- **If the file has no marks:** the option will not be available (grayed out);
- **If the file has one or more marks and the cursor is either above the first mark or below the last mark:** all text (above or below this mark) will be submitted in relation to the cursor position (above or below) the mark;
- **If the cursor is between any two adjacent marks:** all text between those two marks will be submitted.

How can I find errors in my script using Term interface?

- The Application options/R/Term is split in two tabs: Error and Options. The tab Error has an option: Trying to find code errors (at the editor)*. It enables the user to set Tinn-R in order to find code errors at the editor when sending instructions to Rterm.exe.
- It may happen that the error will not be found at the right place. For example, the error might be the same word appearing in a comment which comes before the actual code. In that case the user should use the shortcut F3 (Find again). The word will appear selected,

than just press *OK* until finding the right error. The first search done internally by Tinn-R has *Case sensitive* and *Whole word only* as default, but, this is not passed to the search interface, therefore the user should just select them if convenient. If the error has numbers among letters *Whole word only* is not a good option.

The communication between Tinn-R and Rgui.exe seems to freeze!

- **First, it is not necessary to reinstall the Tinn-R nor R!**
- In some Windows flavors the communication between Rgui.exe and Tinn-R sporadically seems to freeze. The cause of this bug is still unknown to us and seems to be related to the new features of some Windows flavors. However, the solution is very simple: rest your mouse (without pressing for a few seconds) on the icon of the Tinn-R on the taskbar of windows...: the communication should be restored automatically.

The communication between Tinn-R and Rterm.exe seems to freeze using knit function!

- To all `knit` procedures it will be added the argument `quiet=TRUE`. It is very difficult to synchronize the `txtProgressBar` used in these functions and Term interface. So, if you want more control, we suggest (for while) to use the `knit` with Rgui.exe instead of Rterm.exe.

How to make Ctrl+Shift+0 mark works?

For Windows 10, do the following (it may be a bit different for Windows 8):

1. In the control panel, click Language. This brings up the "Language" panel.
2. Choose "Advanced Settings" in the left area. This brings up the "Advanced Settings" panel.
3. Choose "Change language bar hot keys". This brings up the "Text Services & Input Language" panel.
4. Select that item and click the "Change Key Sequence" button. This brings up the "Change Key Sequence" panel.
5. Set the Ctrl + Shift in both radio buttons to "Not Assigned" and click OK.

CHAPTER 3

WORKING WITH

This chapter provides information on how to work using Tinn-R.

3.1 APPLICATION OPTIONS

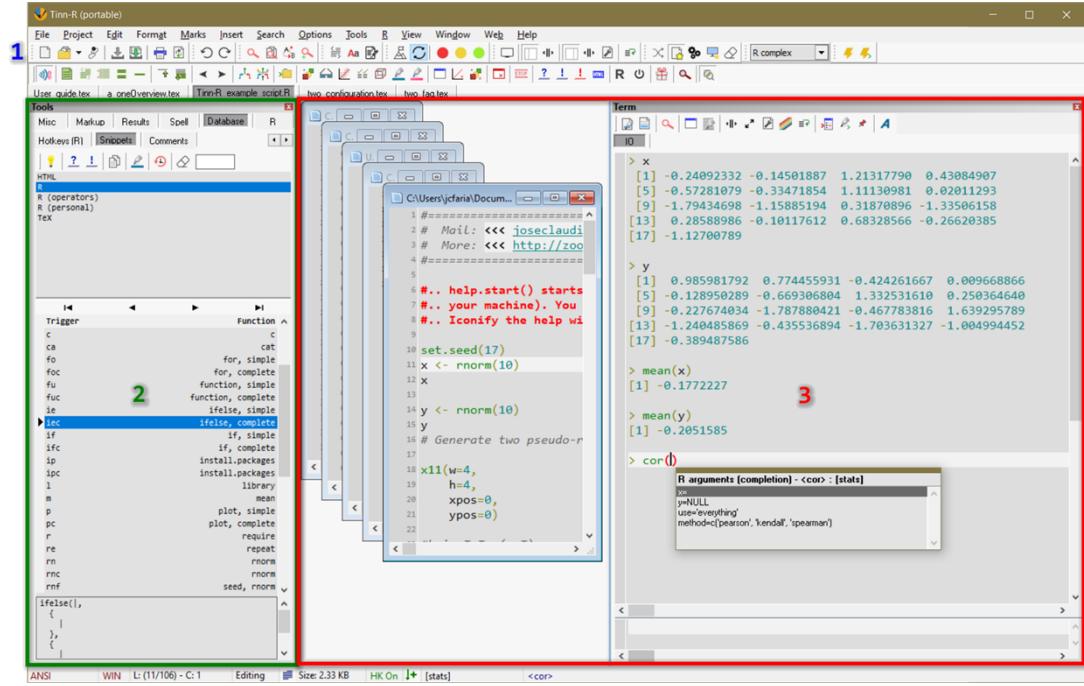


FIGURE 3.1: Tinn-R: Main resources.

Tinn-R interface (Figure 3.1) is very flexible and user configurable. It is necessary time to know all available resources and to configure this out (according to your preferences) in a nice way. The default set of options might not be suitable for every user.

The window *Application options* (Options/Application) and *Highlighters settings* (Options/Highlighters (settings)) allows the user to set the major piece of user preferences related to the application.

It must be clear from now on that the Tinn-R project is the sum of three main resources (Figure 3.1): The application per se (1), additional Tools (2) and the instances of the SynEdit class (3). The Tools (2) was projected to allows the expansion of resources.

The options visible in all pictures reflect a set of the project leader preferences.

Main

Since the options are self-explanatory, Table 3.1 only gives some details about the most difficult options to understand.

Option	Description
Computational synchronization (delay)	Several processes are dependent on synchronization between applications (R, converters, compilers). The optimal value of the delay is determined by the following characteristics: user habits, hardware and software available. The ideal value is unique to the various possible combinations of those three characteristics. Try to reduce to the minimum value (50 ms) and test it: if something does not work, increase it gradually and keep testing until getting to the optimal value. The default value (100 ms) may not be optimal for all users.
Remove extension for Save as	All file extensions will be removed in the <i>Save as Windows</i> interface

Table 3.1: Same main options

General:

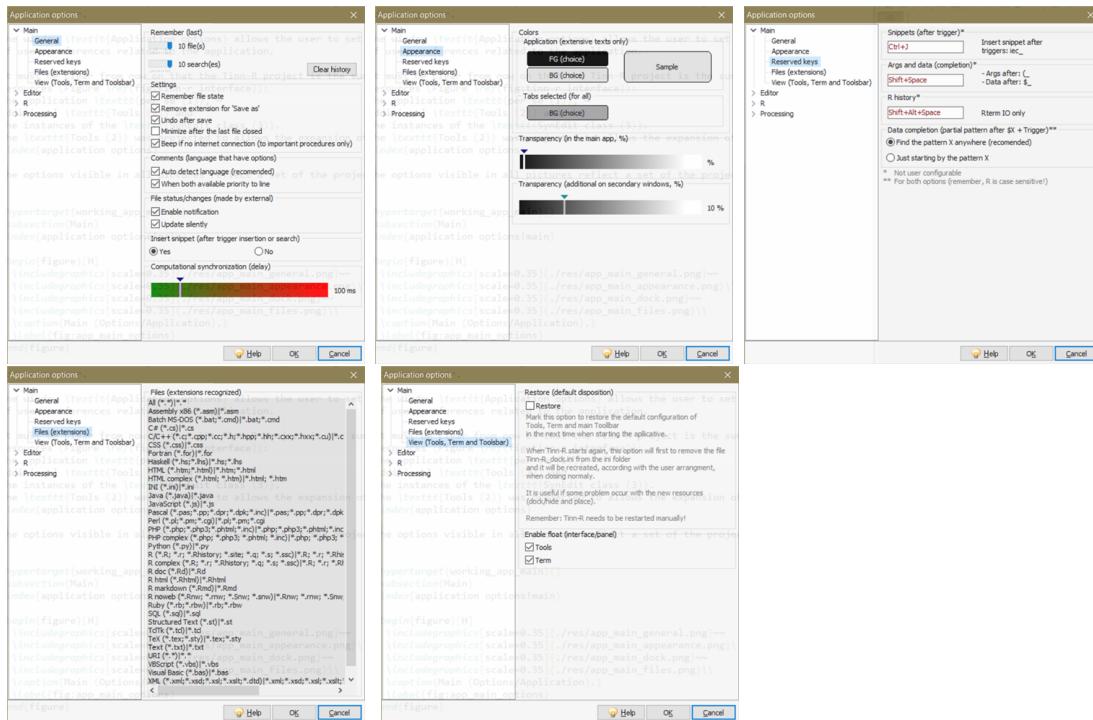


FIGURE 3.2: Main (Options/Application).

Figure 3.2 and Table 3.1 show the options related to this topic.

Appearance:

Option	Description
Application colors (extensive text only)	Dark colors (low level of radiation) for the background, and pale light (high level of radiation) for the characters are recommended for people who work with the computer/monitor for long periods. Pictures of this user guide like this

Table 3.2: Appearance options

Table 3.2 give some information for who work with the computer/monitor for long periods.

Editor

The *Editor* window (Figures 3.3, 3.4) was adapted from the sources of the *SynEdit* component, mainly related to the general appearance and standard options. The set of options available complement the *Application options* and allows high level of customization.

Display:

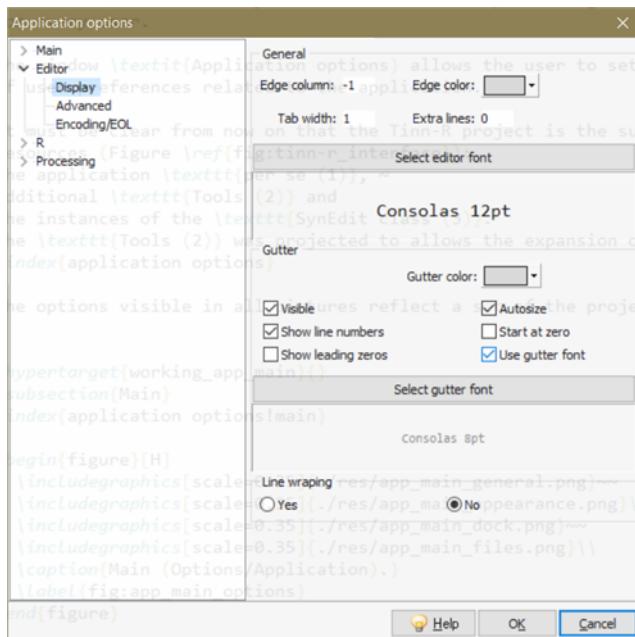


FIGURE 3.3: Editor options: Display.

Figure 3.3 and Table 3.3 show the main resources.

Option	Description
Edge column	Will be showed as a vertical line in the editor and the default is 80 characters. Set it to 0 or a negative value (-1) to make the edge column not visible
Edge color	Choice of the edge color
Tab width	Set the number of characters that will be inserted when typing the <i>Tab</i> key
Extra lines	Set the width which each single line will be displayed
Font	Will open the Windows interface for choosing installed fonts
Gutter color	Will open the Windows interface to choose a color
Visible	Visibility option
Autosize	Autosize option
Show line number	Show line number option
Start at zero	Start at zero option
Show leading zeros	Show leading zeros option
Use gutter font	Use gutter font option

Table 3.3: Display (Options/Editor).

Advanced:

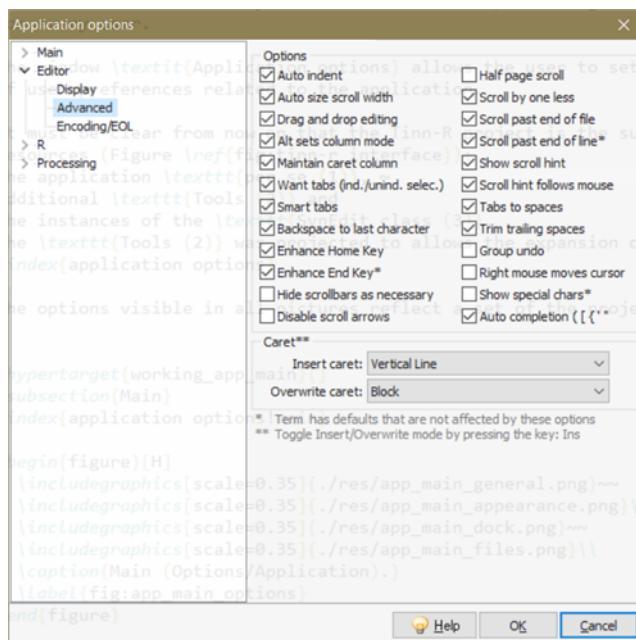


FIGURE 3.4: Editor options: Advanced.

Figure 3.4 and Table 3.4 show the main resources.

Encoding/EOL:

Option	Description
Auto indent	Will indent the caret (position of the cursor in the current line) on new lines with the same amount of leading white space as the preceding line
Auto size scroll width	Automatically resizes the MaxScrollWidth property when inserting text
Drag and drop editing	Allows you to select a block of text and drag it within the document to another location
Alt sets column mode	Holding down the <ALT> key will put the selection mode into column format
Maintain caret column	When moving through lines w/o cursor past EOL, keeps the X position of the cursor
Want tabs (ind./unind. select.)	When tabbing (if there is a selection) <TAB> and <SHIFT><TAB> act as block indent, unindent
Smart tabs	When tabbing, the cursor will go to the next non-white space character of the previous line
Backspace to last character	The cursor will go to the next non-white space character of the line
Enhance home key	Enhances HOME key positioning, similar to visual studio
Enhance end Key	Enhances END key positioning, similar to JDeveloper
Hide scrollbars as necessary	If enabled, then the scrollbars will only show when necessary. If you have ScrollPastEOL, then the horizontal bar will always be there (it uses MaxLength instead)
Disable scroll arrows	Disables the scroll bar arrow buttons when you can't scroll in that direction any more
Half page scroll	When scrolling with page-up and page-down commands, only scroll a half page at a time
Scroll by one less	Forces scrolling to be one less
Scroll past end of file	Allows the cursor to go past the end of file marker
Scroll past end of line	Allows the cursor to go past the last character into the white space at the end of a line
Show scroll hint	Shows a hint of the visible line numbers when scrolling vertically
Scroll hint follows mouse	The scroll hint follows the mouse when scrolling vertically
Tabs to spaces	Converts a tab character to a specified number of space characters
Trim trailing spaces	Spaces at the end of lines will be trimmed and not saved
Group undo	When undoing/redoing actions, handle all continuous changes of the same kind in one call instead undoing/redoing each command separately
Right mouse moves cursor	When clicking with the right mouse for a pop-up menu, move the cursor to that location
Show special chars	Shows the special characters
Insert caret	A list with four options: Vertical line, Horizontal line, Half block and block
Overwrite caret	A list with options: Vertical line, Horizontal line, Half block and block

Table 3.4: Display (Options/Editor).

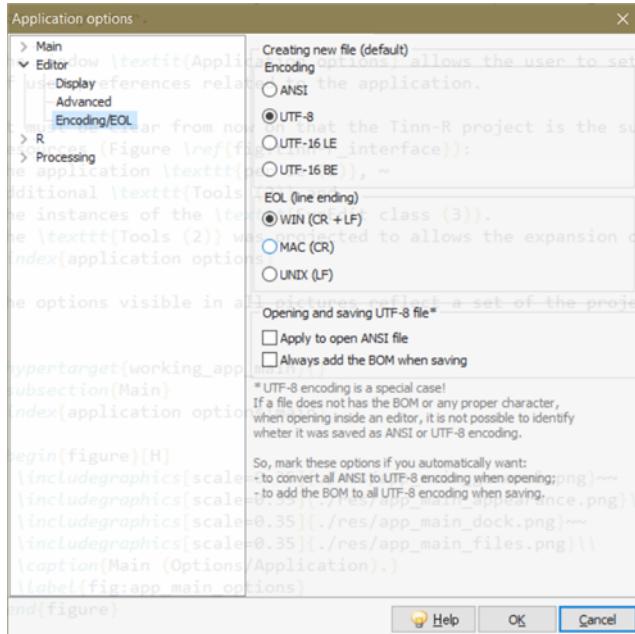


FIGURE 3.5: Editor options: encoding/EOL.

This interface (Figure 3.5) allows to change the default encoding and EOL when creating new files and also the user option related to UTF-8 files.

R

Figures 3.6, 3.8, 3.9 and 3.7 shows a set of options available. As you can see, it allows a high level of customization with the R environment.

Processing

There are resources (Figure 3.10 and 3.11) related to conversion (Deplate, Pandoc and Txt2tags) and compilation (Miktex).

Conversion:

Tinn-R project makes it easy to work with these nice conversion tools: Deplate, Pandoc and Txt2tags. (Figure 3.10).

\LaTeX :

Tinn-R is not a specific editor to \LaTeX , but it has the basic resources (Figure 3.11) allowing the user to use the main resources of this environment.

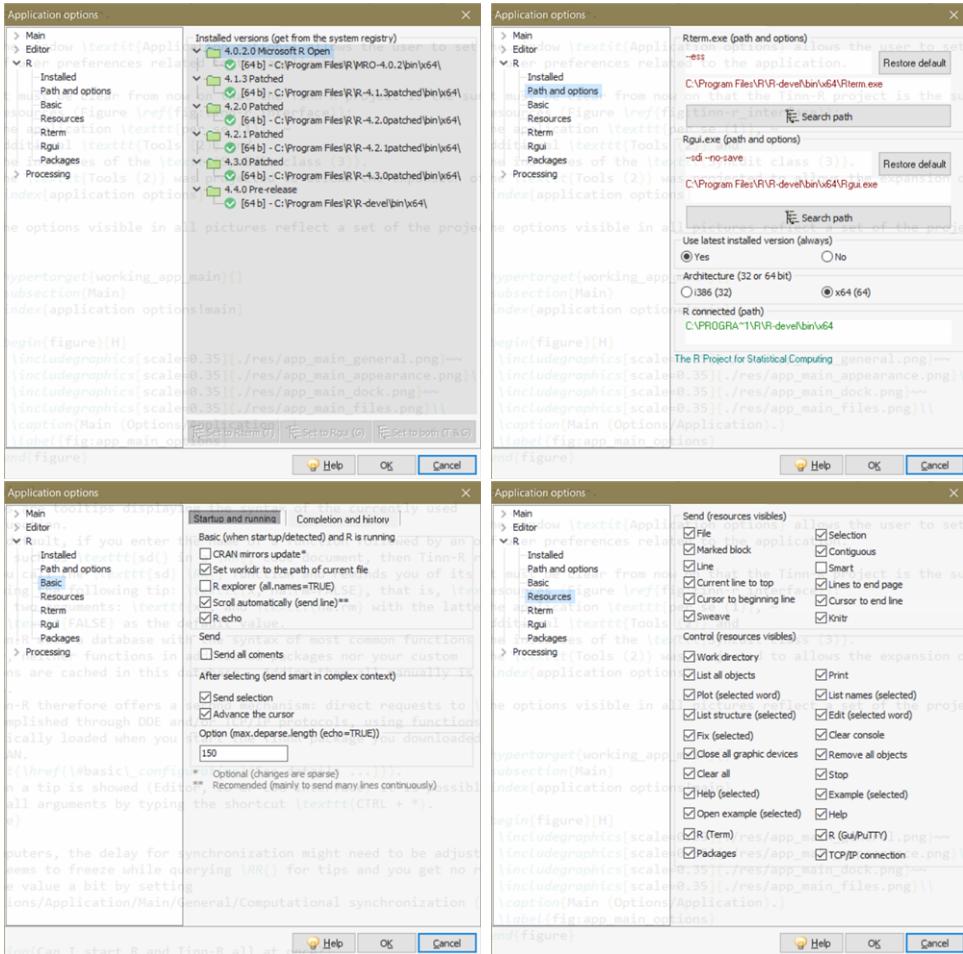


FIGURE 3.6: R (Options/Application): Installed, Path and options, Basic and Resources.

3.2 MAIN DIALOGS

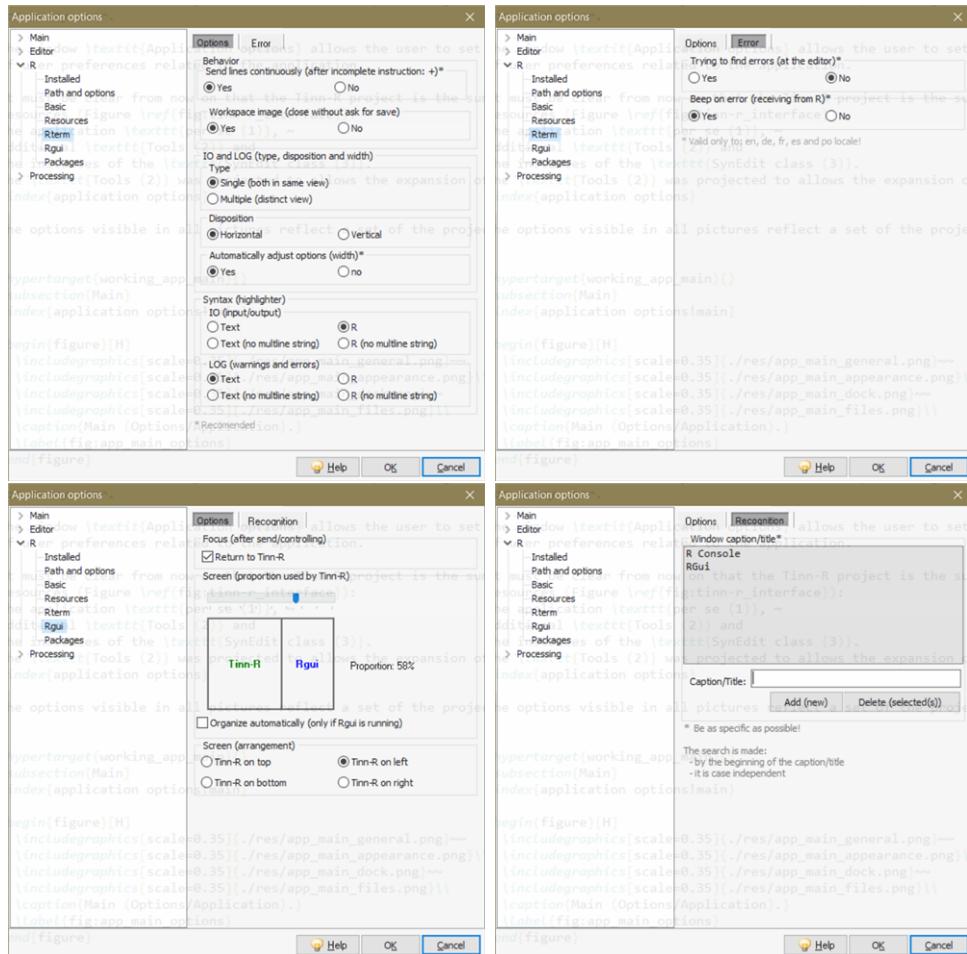


FIGURE 3.7: R (Options/Application): Rterm.exe and Rgui.exe.

About

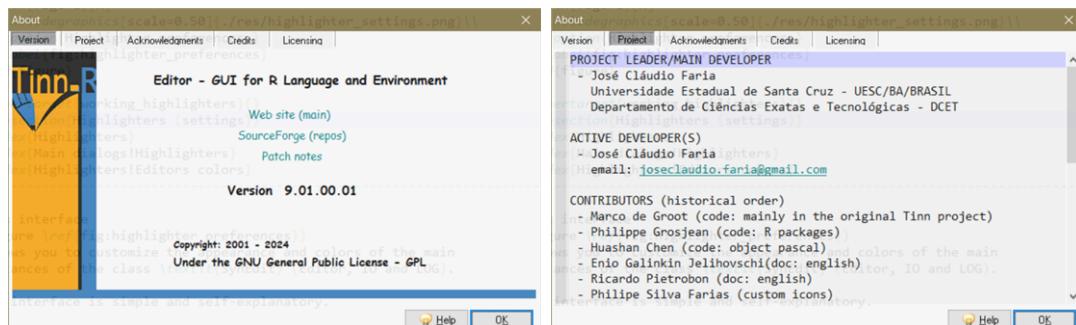


FIGURE 3.12: Main screens of the About dialog.

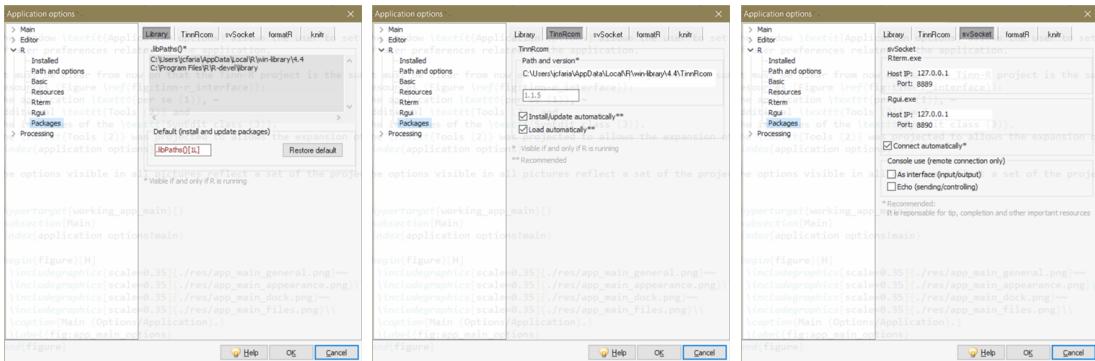


FIGURE 3.8: R (Options/Application/Packages): Library, TinnRcom and svSocket.

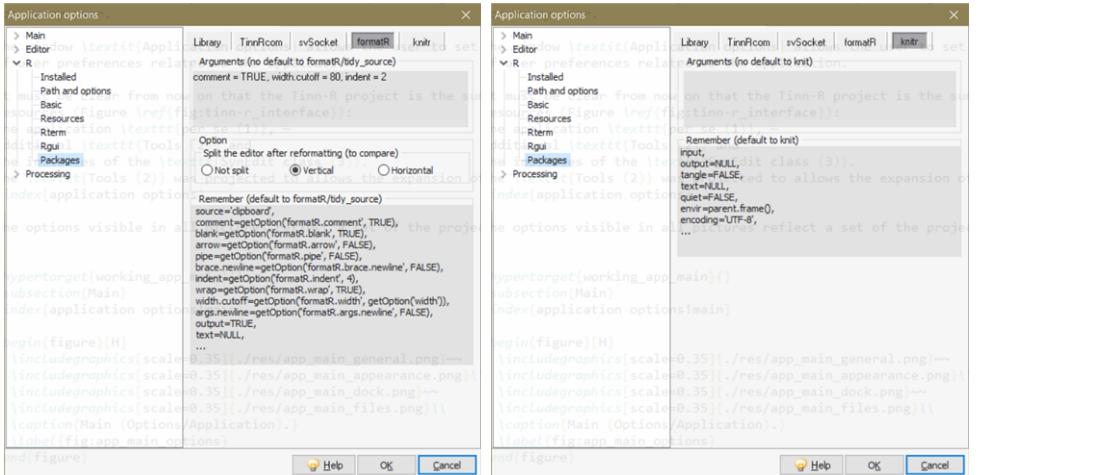


FIGURE 3.9: R (Options/Application/Packages): formatR and knitr.

This dialog (Figure 3.12) displays generic and important information about the program. The interface is simple and self-explanatory.

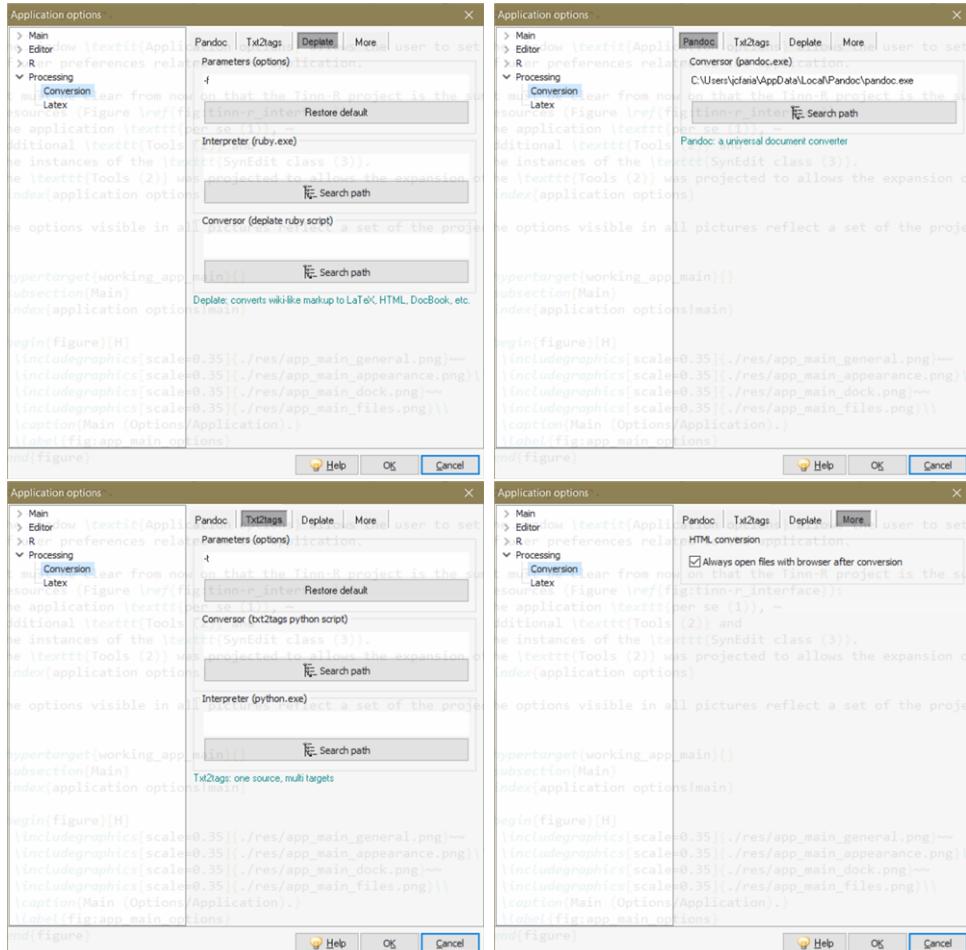


FIGURE 3.10: Conversion (Options/Application/Processing): Pandoc, Txt2tags, Deplate and More.

ASCII chart

ASCII chart														
	Low	High	Dec	Hex	Value	Char	Console	8						
0	NUL	16	DLE	32	48	0	64	@	80	P	96	`	112 p	
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113 q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114 r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115 s
4	EOT	20	DC4	36	\$	52	4	68	D	84	T	100	d	116 t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117 u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118 v
7	BEL	23	ETB	39	*	55	7	71	G	87	W	103	g	119 w
8	BS	24	CAN	40	(56	8	72	H	88	X	104	h	120 x
9	TAB	25	EM	41)	57	9	73	I	89	Y	105	i	121 y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122 z
11	VT	27	ESC	43	+	59	;	75	K	91	[107	k	123 {
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124
13	CR	29	GS	45	-	61	=	77	M	93]	109	m	125 }
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126 ~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127 \

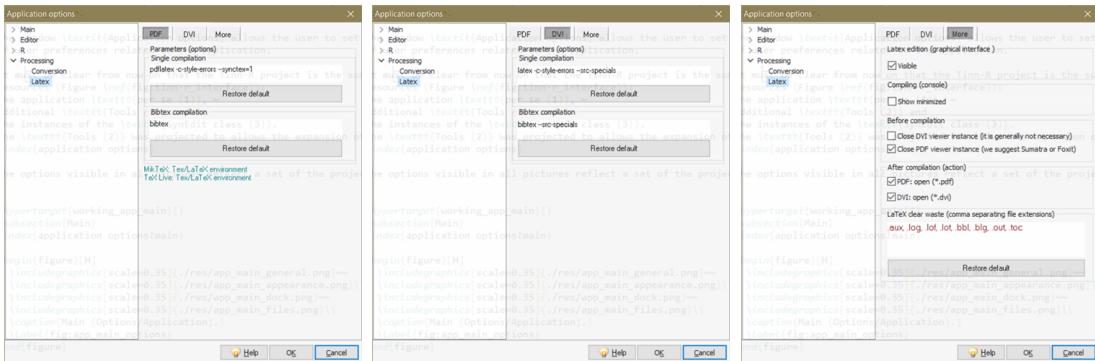


FIGURE 3.11: Latex (Options/Application/Processing): L^AT_EX PDF, DVI and More.

This dialog (Figure 3.13) displays a traditional ASCII character table. When you click on a chosen character, it is inserted into the editor. It's not available - we don't see a need for it - for the Term interface.

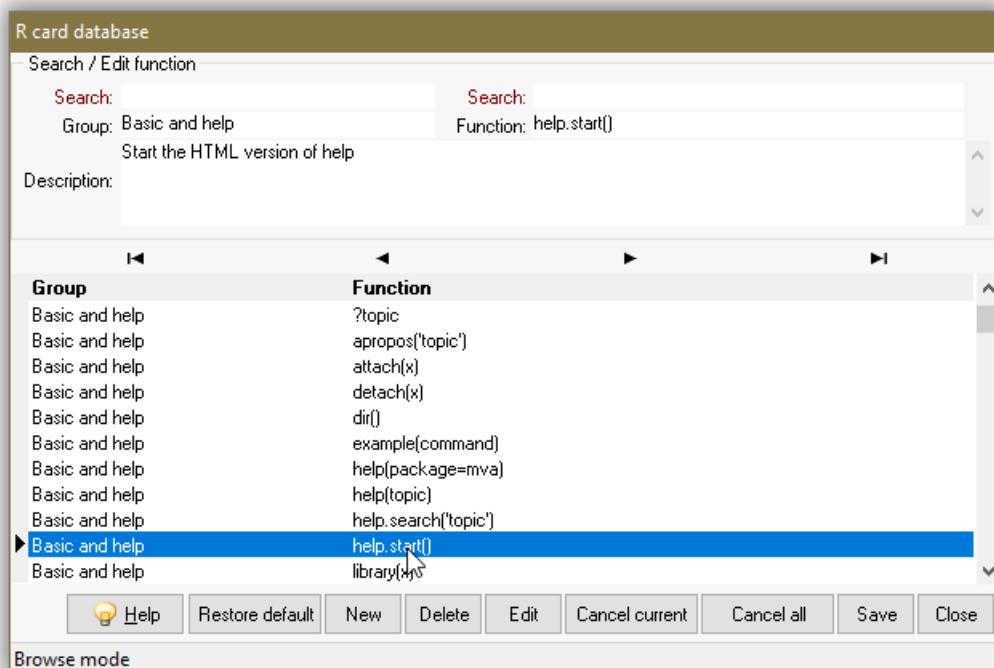
Card (R)

FIGURE 3.14: R card (dataset).

The *card* was based on two R cards already published: R/Rpad Reference Card by Tom Short and R reference card by Jonathan Baron.

Read below a brief description of available buttons (Figure 3.43):

Restore default: Restores the file `Rcard.xml` from the origin at (`InstallPath/data/data.zip`). Any prior changes in the file `Rcard.xml` currently being used will be lost.

New: Places the table in insertion mode.

Delete: Delete the current registry from the table.

Edit: Places the table in edition mode.

Cancel current: Cancels any change made to the current edition.

Cancel all: Cancels all changes made to the database prior to *Save*.

Save: Overwrites the text file (XML) saving all changes made to the current table.

Close: Closes the dialog. All non-saved changes will be lost.

Code completion

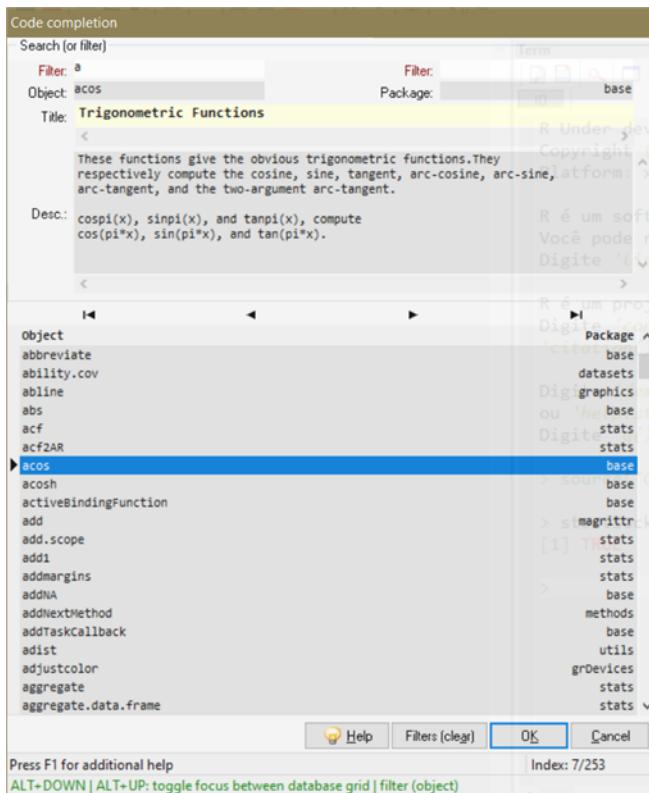


FIGURE 3.15: Code completion dialog.

Code completion (Figure 3.15) is a dialog window used to search R objects displaying their basic documentation.

The strategy designed for this interface is - in our view - innovative. It is an alternative to what is conventional in the main IDEs used in programming. These generally try to guess - with drop-down windows filled with information - what the user wants as they type. Paradoxically, they are poor in the information needed to make a decision. In our understanding, drop-down windows generally have a low success rate, pollute the editing space and do not have features that make it possible to apply filters to different groups or comfortably browse the options (which in many cases can be 200 or more) in certain cases.

The R code completion dialog window (Figure 3.15) allows you to apply filters (non-retroactive) to the object name, the package name and allows you to comfortably navigate the dataset grid seeing as much information as possible about the objects to really enable precise location .

Due to the search mechanism being relatively complex, it was decided not to allow retroactive searches. In other words, if you don't have detailed information about the object you want to locate in R, the best option is to start the search with the smallest possible number of characters and apply filters, and/or, navigate the grid to locate the desired object. In other words, start with a large number and logically reduce the options until you find what you are looking for.

The shortcuts (ALT+DOWN | ALT+UP) allow you to switch the cursor focus between the editing and filtering boxes and thedatagrid, making it easier to choose the object sought, as a greater amount of documentation (title and description) is displayed in the interface.

Once located, simply type ENTER (or OK) and the object will be inserted in the Editor, IO or LOG where the cursor was located when the dialog was called.

Use this dialog as follows:

1. Place the cursor where you want to use the code completion
2. Enter the default shortcut: (CTRL+Q) to open the dialog window
3. Use edit boxes for object or package filtering
4. Locate the desired object
5. ENTER or OK to insert the object

As the search database - however small it is - brings a relatively large volume of information from the R documentation and is not very fast, searches should only be done progressively (adding more characters to the initial(s)) in the filter box. In other words, reverse search is not performed in this interface.

Comments

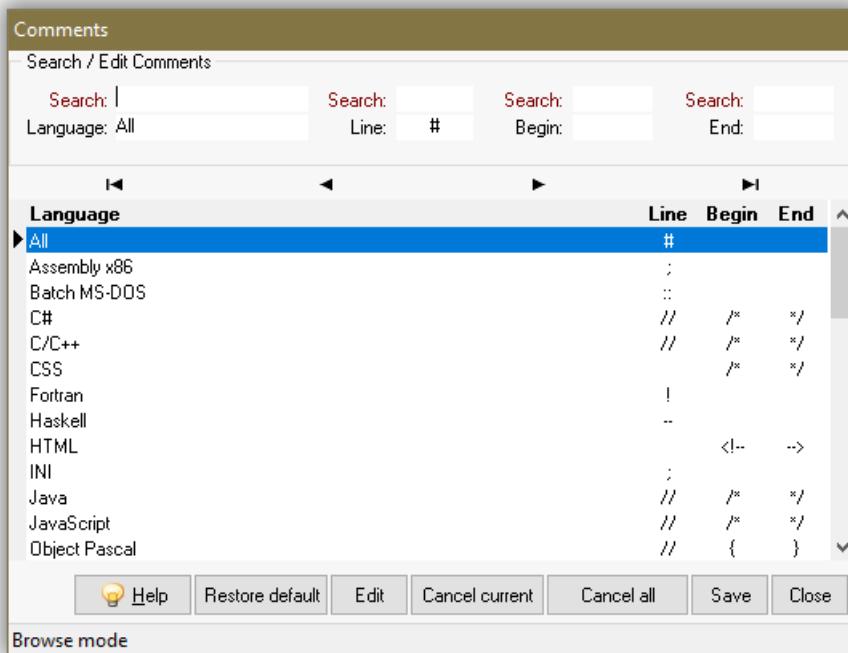


FIGURE 3.16: Comments (dataset).

The *Comments* resource is very simple and allows high level of user customization.

From version 3.0.1.0 Tinn-R automatically recognizes the language of the file on focus. Further, inside the file - if it is a syntax a multi-highlighter (complex syntax) - which language of the line where the cursor (or selection) is found.

This identification is done automatically if (and only if) the option (*x*) *Auto detect language (recomended)* is checked. Otherwise the user is forcing the application to use the comments of the selected language (indicator arrow).

Selected code snippets involving more than one language will not be commented/uncommented and a warning message is issued. That is, you must select only the snippet of a single language.

Read below a brief description of available buttons (Figure 3.42):

Restore default: Restores the file `Comments.xml` from the origin at (InstallPath/data/data.zip).

Any prior changes in the file `Comments.xml` currently being used will be lost.

Edit: Places the table in edition mode.

Cancel current: Cancels any change made to the current edition.

Cancel all: Cancels all changes made to the database prior to *Save*.

Save: Overwrites the text file (XML) saving all changes made to the current table.

Close: Closes the dialog. All non-saved changes will be lost.

Differences

```
File Edit Search Options Actions Help  
C:\Users\jcfaria\Documents\Tinn-Rsample\Tinn-R_example_script.R  
1 #####  
2 # Mail: <>< joseclaudio.faria@gmail.com >>  
3 # More: <>< http://zoonek2.free.fr/UNIX/48_R/all.html >>  
4 #####  
5  
6 ##. help.start() starts the html interface to on-line he  
7 ##.. your machine). You should briefly explore the featur  
8 ##.. Iconify the help window and move on to the next part  
9  
10 set.seed(17)  
11 x <- rnorm(10)  
12 x  
13 # Main dialogs! ASC chart  
14 y <- rnorm(10)  
15 y  
16 # Generate two pseudo-random normal vectors of x and y c  
17 x11(w=4, index>Main dialogs! Search in files  
18 h=4, Find  
19 xpos=0,  
20 ypos=0, index>Main dialogs! Find)  
21  
22 #bringToTop(s=T)  
23  
24 # Open URL  
25 plot(x,  
26 y) index>Main dialogs! Open URL  
27 # Plot the points in the plane. A graphics window will a  
28 # Print  
29 ls()  
30 # See which R objects are now in the R workspace  
31  
32 rm(x, y)  
33 # Remove objects no longer needed. (Clean up)  
34  
+ - 0 lines added, 4 lines modified, 0 lines deleted.
```

FIGURE 3.17: Differences dialog.

This dialog (Figure 3.17) allows comparison between two directories or between two files.

As it is software developed by third parties - adapted and incorporated into Tinn-R - the documentation (`textdiff.hlp`) is in the old Windows documentation system. Therefore, if necessary, you must install the appropriate Help viewer to read the documentation.

The interface of this dialog is friendly and very intuitive. We are evaluating whether to continue with this (third-party) application and update the

documentation or replace it with something more modern for the next versions. It - as it is - has served our basic needs well over many years, as it was a job well done in Object Pascal.

Find

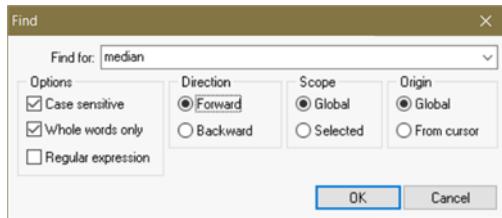


FIGURE 3.18: Find dialog.

This dialog (Figure 3.28) and *Replace* (Figure 3.29) are very similar.

When you call up the *Find* dialog the *Find for* box will be prefilled with the word under the cursor. You can type over the entry if you are looking for another word. There is also a dropdown list of phrases previously searched.

There are options for Case sensitive, Whole words only, Regular expression, Direction (Forward and Backward), Scope (Global and Selected) and Origin (Global and From cursor).

Highlighters (settings)

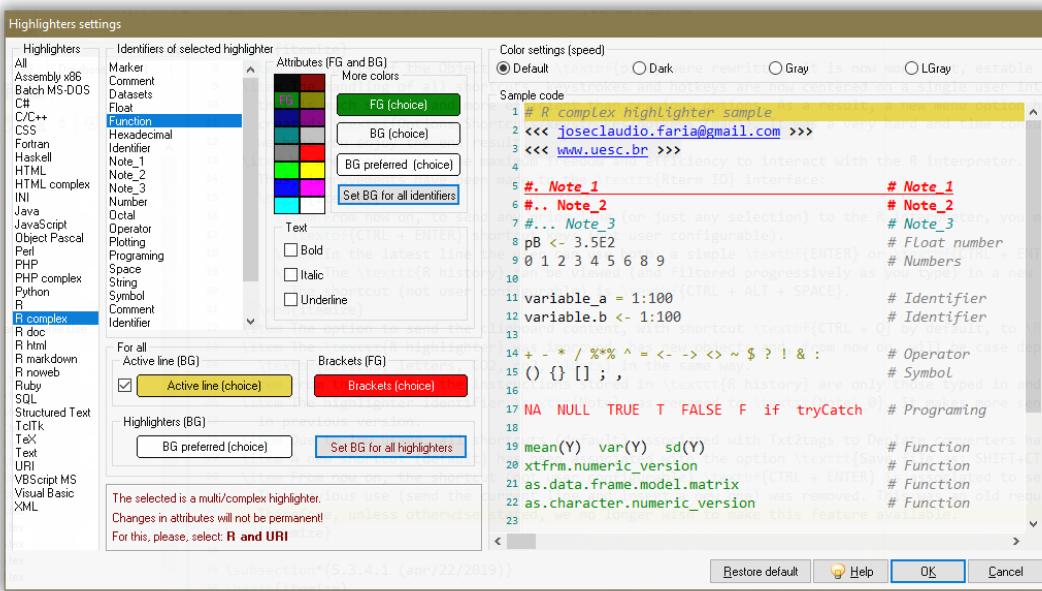


FIGURE 3.19: Highlighter preferences dialog.

This interface (Figure 3.19) allows you to customize the appearance and colors of the main instances of the class *SynEdit* (Editor, IO and LOG).

The interface is simple and self-explanatory.

Basically, make a choice between the set of highlighters available from the *Highlighters* list. The identifier of the selected highlighter will be updated. It is possible to set only one foreground attribute each time. But it is possible to set the background for all attributes of the selected highlighter and also the background of all attributes of all highlighters.

It is also possible to set the color brackets and the active line background.

Observation:

Tinn-R has seven multi-highlighters: *HTML complex*, *PHP complex*, *R complex*, *R doc*, *R html*, *R markdown* and *R noweb*, with each one behaving as follows:

1. *HTML complex* = *HTML & JavaScript*
2. *PHP complex* = *HTML & JavaScript & PHP*
3. *R complex* = *R & URI ('<<' begin URI; '>>' end URI)*
4. *R doc* = *TeX & R ('>=' begin R; '@' end R)*
5. *R html* = *HTML & R ('<!--begin.rcode' begin R; 'end.rcode-->' end R)*

5. R markdown = URI & R ('```{' begin R; '```' end R)
6. R noweb = TeX & R ('>=' begin R; '@' end R)

URI : Uniform Resource Identifiers.

R complex : The main syntax is R, '<<<' and '>>>' are the tags enabling the user to insert a block of URI syntax.

R doc : The main syntax is TeX, '>=' and '@' are the tags enabling the user to insert a block of R syntax.

R html : The main syntax is HTML, '<!--begin.rcode' and 'end.rcode-->' are the tags enabling the user to insert a block of R syntax.

R markdown: The main syntax is URI, '```{' and '```' are the tags enabling the user to insert a block of R syntax.

R noweb : The main syntax is TeX, '>=' and '@' are the tags enabling the user to insert a block of R syntax.

These highlighters haven't priorities when you set the syntax color preferences. Thus, if you change the colors' preferences of any of these multi-highlighters these settings will be valid only in the current Tinn-R session and will not be saved when Tinn-R is closed. So, if you want to make permanent changes, set the preferences from all simple highlighters.

From version 3.0.1.0 a warning message is displayed whenever a multi-highlighter is selected. It shows which highlighters the user must change the characteristics so that they are properly stored and henceforth always displayed.

Hotkeys (Send/Control/Custom)

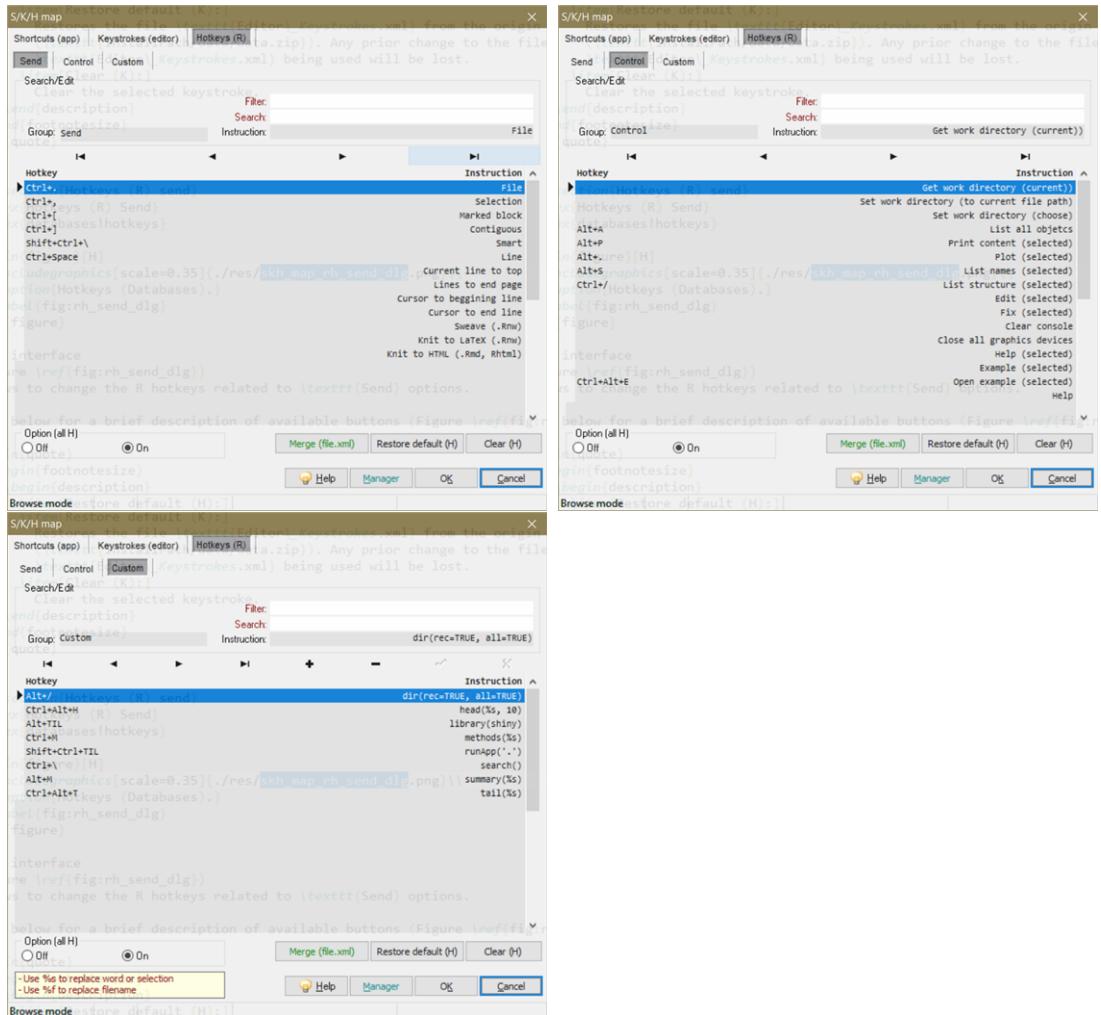


FIGURE 3.20: Hotkeys (database).

The *Hotkeys (operational system)* (Figure 3.20) allow setting the hotkeys related to the operational system. The difference between those hotkeys and *Shortcuts customization* is that the latter works only with the focus in Tinn-R, whereas the hotkeys work with the focus anywhere.

The interface is self-explanatory. Basically you first make a choice from the *R/Hotkeys (operational system)* and set the desired Hotkey.

The set of hotkeys will perform actions only if the option *Active* is checked. The objective of these options (*Inactive* and *Active*) is to avoid conflict with

others applications allowing to enable/disable the set of hotkeys quickly and easily.

The R/Hotkeys interface was deeply reworked in the version 5.05.01.01 and it now has three tabs, Send, Control and Custom:

- **Send:** Contains the already traditional Send instructions of Tinn-R;
- **Custom:** Contains the already traditional Control instructions of Tinn-R;
- **Custom:** **Allows the user to customize any instructions** to be send to R interpreter (thanks to Philemon Lenherr for the suggestion). The instructions must be as follows:
 - Simple: `search()`. The R interpreter will receive > `search();`
 - Replace word or small selection: `View(%s, title='View of iris dataset')`. If the editor cursor is over the word `iris` or it is selected, the R interpreter will receive > `View(iris, title='View of iris dataset')`
 - Replace whole file: `source(%f, echo=TRUE, verbose=TRUE)`. The R interpreter will receive > `source(.trPaths[4], echo=TRUE, verbose=TRUE)`. All rules related to send file are preserved.

Starting with version 5.05.01.01, the number of Custom statements the user can define is unlimited.

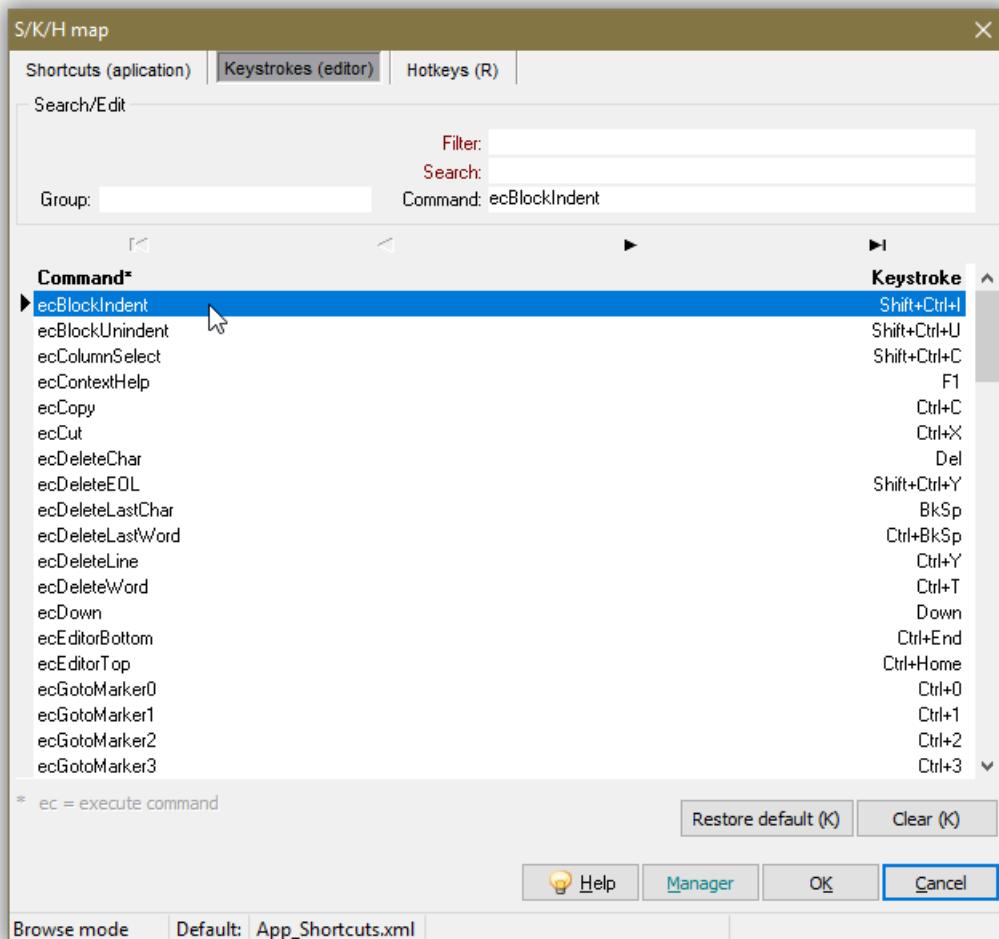
Keystrokes (Editor)

FIGURE 3.21: Keystrokes (dataset).

This interface (Figure 3.21) allows to change the default SynEdit keystrokes. It is possible only change the keystroke associaed to any ecAction (execute command action). A set of user friendly keystrokes gives high productivity leading with all instances of the class *SynEdit*: Editor, IO and LOG.

Read below for a brief description of available buttons (Figure 3.21):

Restore default (K): Restores the file *Editor_Keystrokes.xml* from the origin at (*InstallPath*/data/data.zip). Any prior change to the file ini

file `Editor_Keystrokes.xml` being used will be lost.

Clear (K): Clear the selected keystroke.

License manager

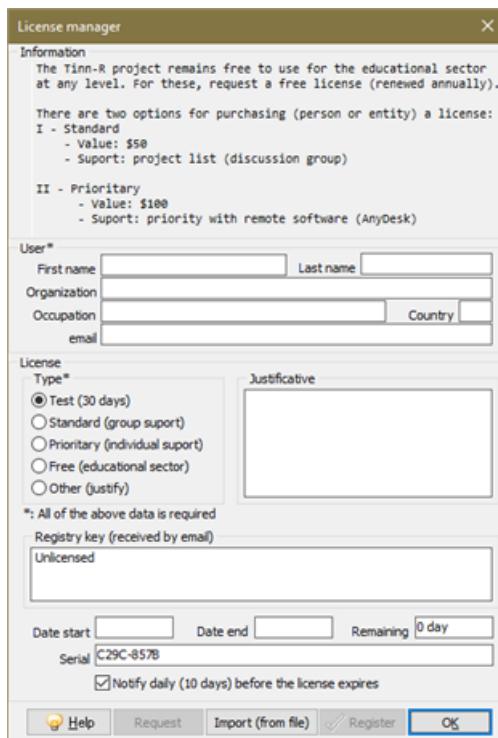


FIGURE 3.22: License manager.

This interface (Figure 3.22) allows the license manager dialog.

This dialog must be used to manage program licenses. There you can request [Request], Import [Import (from file)] and Register [Register] the usage license.

All use licenses are valid for one year from the date of issue.

Tinn-R does not restrict any of its features, even for unlicensed versions. However, at each period of time determined by the developers, the user with an unlicensed version (or an expired version) observes an informative dialog about licensing on their monitor screen.

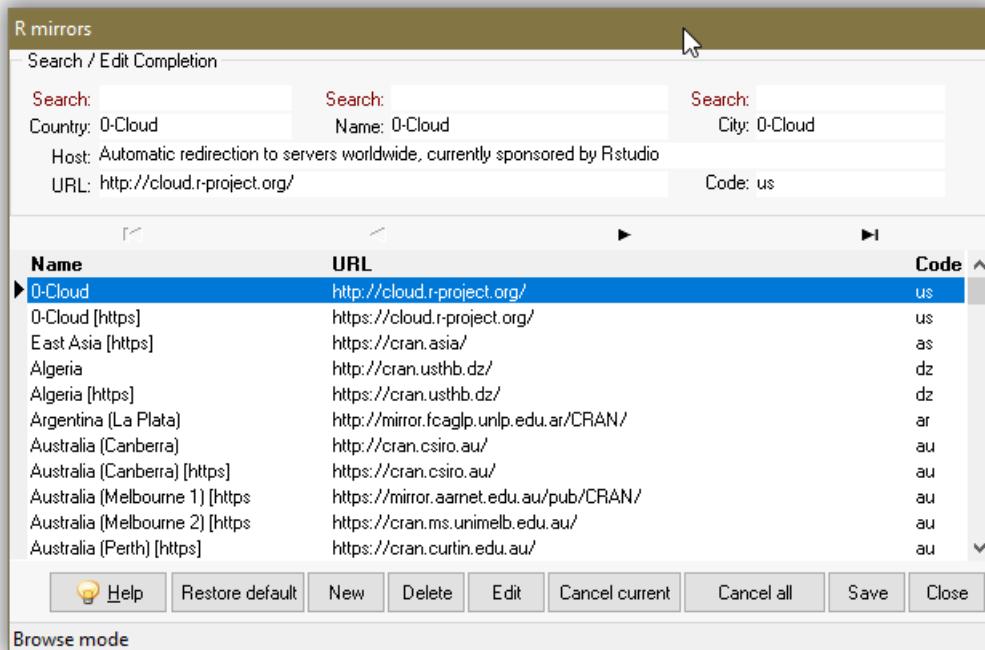
Mirrors (R)

FIGURE 3.23: R mirrors (dataset).

The *Mirrors* is an interface that allows the user to manage the repositories (or mirrors) of R.

You should always choose a repository physically closest to where you are, so that, the Web communication tends to be faster and more efficient.

The default mirror is the University [Wien](#) (Austria). Consider that this is the central mirror of CRAN.

The reasons for the Tinn-R always set a repository are two:

- Prevent R keep asking which repository you want to use in each session;
- Workaround of intermittency (only Rterm.exe) display the dialog for selecting the repository. That is, sometimes the dialog is displayed and not others. The cause of this intermittency is still unknown.

Read below a brief description of available buttons (Figure 3.44):

Restore default: Restores the file `Rmirrors.xml` from the origin at (`InstallPath/data/data.zip`).

Any prior change to the file `Rmirrors.xml` while being used will be lost.

New: Places the table in insertion mode.

Delete: Deletes the current registry from the table.

Edit: Places the table in edition mode.

Cancel current: Cancels any change made during the current editing session.

Cancel all: Cancels all changes made to the database prior to *Save*.

Save: Overwrites the text file (XML) while saving all changes made to the current table.

Close: Closes the dialog. All changes not previously saved will be lost.

Open URL

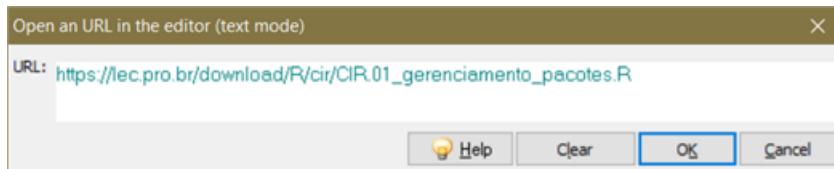


FIGURE 3.24: Open URL dialog.

This dialog (Figure 3.24) allows the user to inform a Uniform Resource Locator (URL) of a file available on the WEB that they want to open in Tinn-R for viewing and/or editing.

A copy of the file saved and opened in the operating system's temporary folder with a suffix added to its name [hh_mm_ss] to make it unique. If desired, the user can use the Save as option, choose a more suitable location to store it and remove the suffix to simplify the file name.

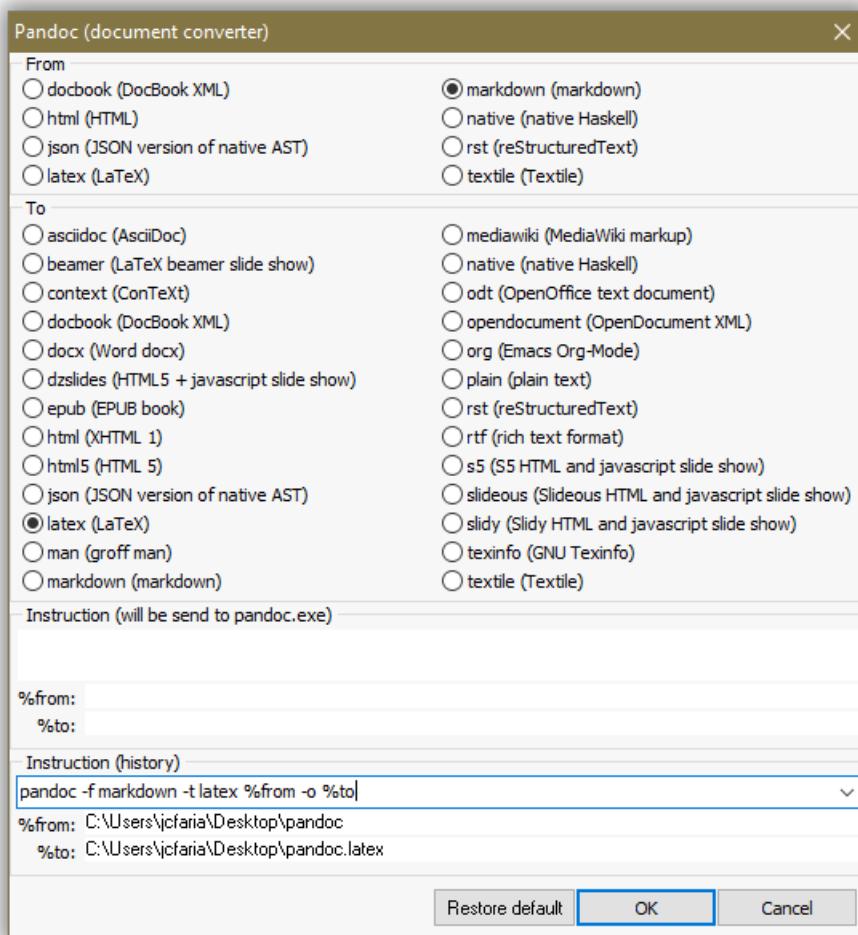
Pandoc

FIGURE 3.25: Pandoc (document converter).

This dialog (Figure 3.25) displays a simple GUI made to make more easy to convert files from one markup format into another using [Pandoc](#). The interface is simple and self-explanatory. Be aware that Pandoc requires that the file to be converted is in UTF-8 format.

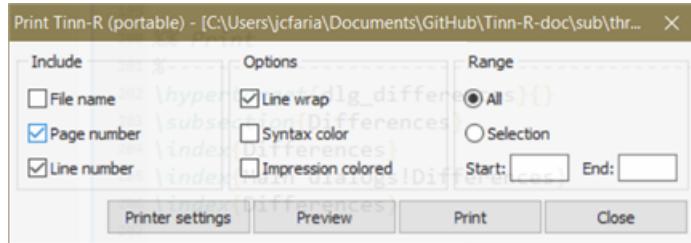
Print

FIGURE 3.26: Print dialog.

This dialog (Figure 3.26) allows the user to define the main options related to printing the active document. It also allows you to define the printer properties [Printer settings], see in advance how the print will be [Preview], print directly if you have already defined the preferences and close the dialog [Close].

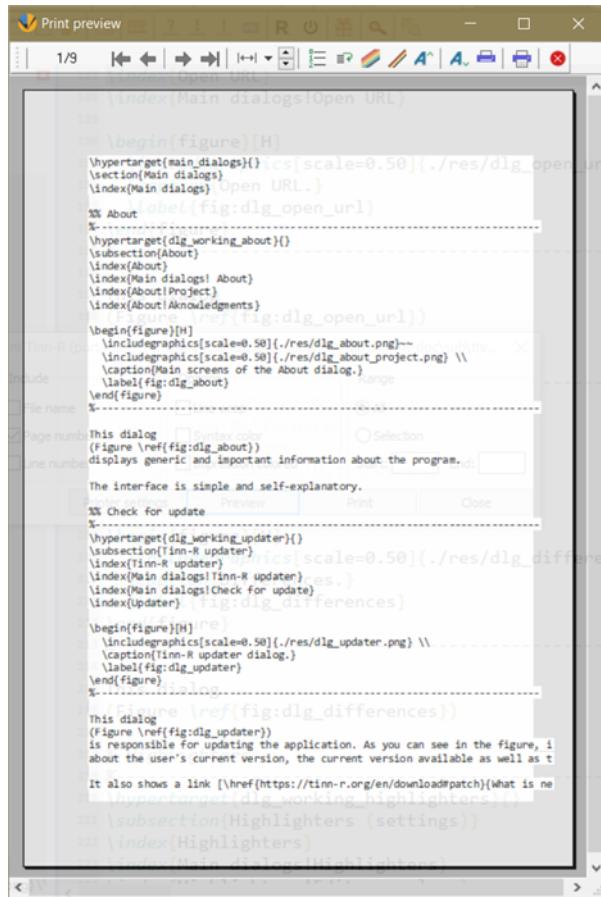
Print preview

FIGURE 3.27: Print preview dialog.

This dialog (Figure 3.27) is opened - at the user's option - if they want to see in advance what the printout of the open document will look like. It is a simple interface that allows you to define or reset printer and printing properties.

Find and replace

Find:

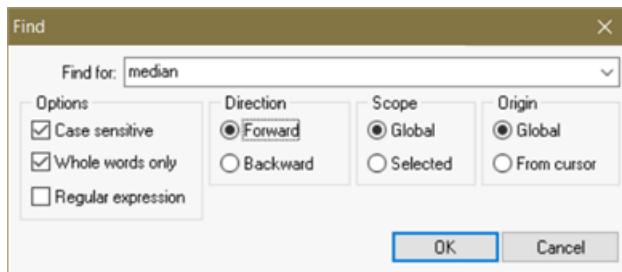


FIGURE 3.28: Find dialog.

The dialogs for *Find* and *Replace* (Figure 3.28 and 3.29) are very similar, so this session will just discuss the *Replace* dialog and will point out the changes when necessary.

When you call up the *Find* dialog the *Find for* box will be prefilled with the word under the cursor. You can type over the entry if you are looking for another word. There is also a dropdown list of phrases previously searched.

Replace:

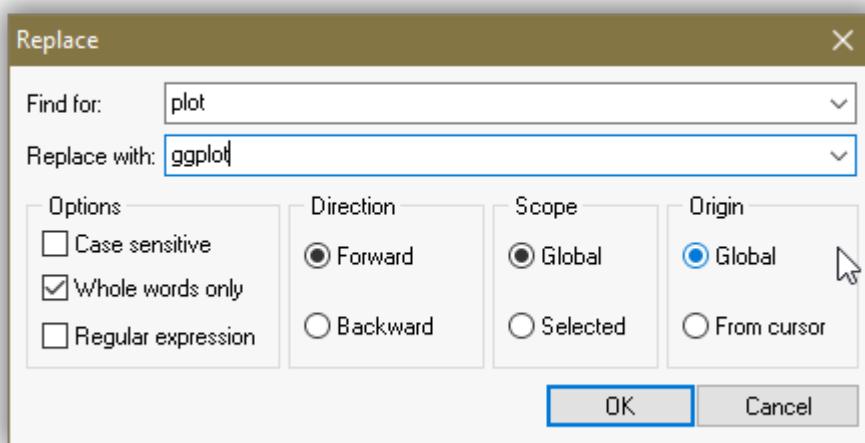


FIGURE 3.29: Find and replace dialog.

When you call up the *Replace* dialog (Figure 3.29) the *Replace with* box will be filled with the last string you entered in it. If this is the first time you have called the *Replace* dialog since starting Tinn-R then the Replace box

will be empty. You can type over any text in box. There is also a dropdown list of strings previously used.

Options:

Case sensitive: When this option is set the search is done case sensitively. For instance, Ab, AB and ab are all treated as different words whereas they are not if the option is not set.

Whole words only: When this option is set the system will only find complete words matching the search criteria. So, for example, if ab is the search string the system will not match occurrences of words like abc or cab.

Regular expressions: See regular expressions ...

Direction:

The direction to search. This option is ignored if searching in selected text.

Forward: Search from the cursor position to the end of the file.

Backward: Search from the cursor position to the beginning of the file.

Scope:

Global: Search the entire file.

Selected Text: Search just the selected text.

Origin:

Global: Search from the beginning of the file.

From cursor: Search just from the position of the cursor.

Search in files

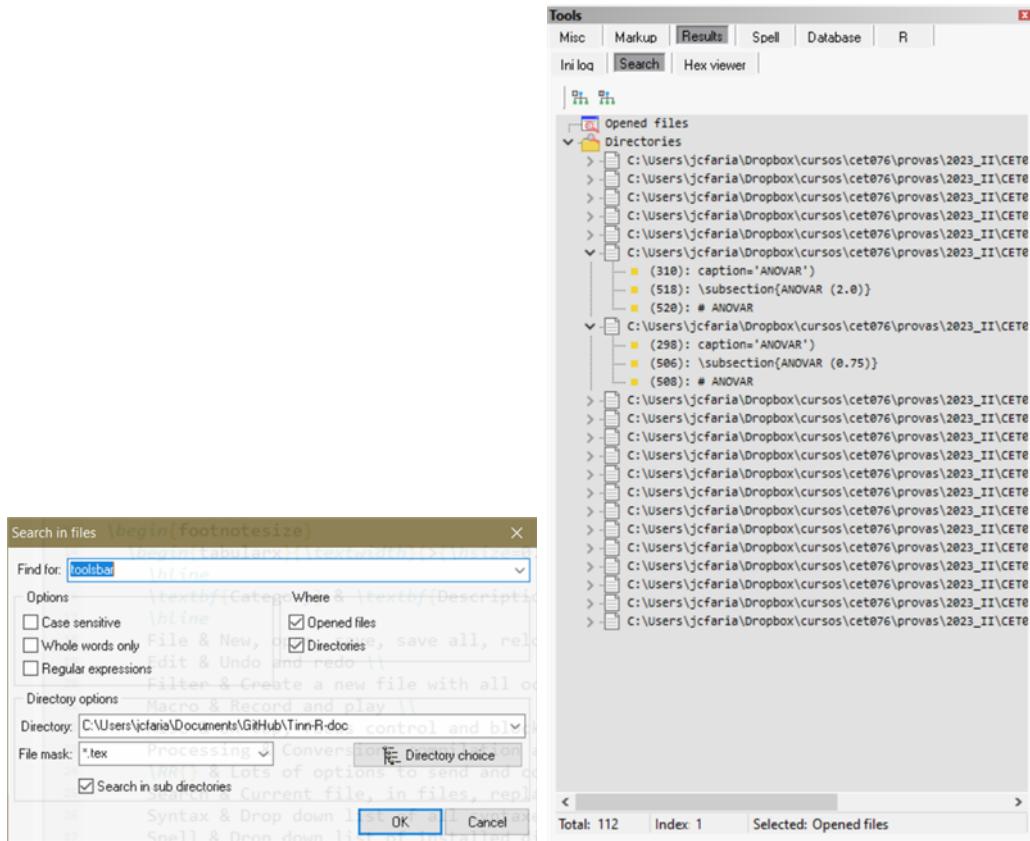


FIGURE 3.30: Search in files dialog.

This dialog (Figure 3.30) allows you to match a criteria in all opened files and/or in files on disk.

Options:

Case sensitive: When this option is set the search is case sensitive. For example, Ab, AB and ab are all treated as different words.

Whole words only: When this option is set the system will only find complete words matching the search criteria. For example, if ab is the search string the system will not match occurrences of words such as abc or cab.

Regular expressions: See regular expressions ...

Where:

Opened files: When this option is set the search is performed on all opened files.

Directories: When this option is set the search is performed in disk files.

Directory options:

Directory: A dropdown list of previously searched directories.

File mask: A dropdown list of the previously searched file mask.

Search in sub directories: When this option is set the search is performed on all sub directories of the main directory.

Results interface:

The associated results interface (Figure ??) shows the results.

A double click in a single occurrence (or dragging and dropping it into the editor interface) will open the file and results will be placed in the first line of the editor window.

Search (trigger|snippet)

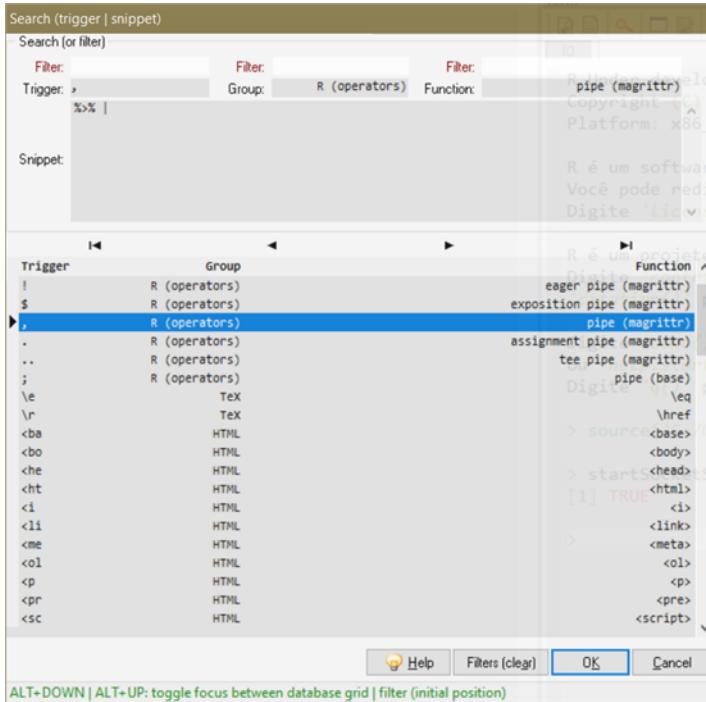


FIGURE 3.31: Search (trigger|snippets) dialog.

Snippets (Figure 3.31) are a way to efficiently edit previously-formed, frequently used blocks of text.

As the user creates personalized snippets (and associated triggers) in the database, it can become difficult to remember all the triggers. This dialog - Search (trigger|snippet) - (Figure 3.31) is intended to make it easier to find the trigger, if users don't remember it.

Allows you to filter table occurrences by trigger, group and function, individually or simultaneously.

The shortcuts (ALT+DOWN | ALT+UP) allow you to switch the cursor focus between the editing and filtering boxes and the datagrid, making it easier to choose the trigger associated with the snippet.

Once located, simply type ENTER (or OK) and the trigger will be inserted in the Editor, IO or LOG where the cursor was located when the dialog was called.

What happens next is dependent on a user option: Options/Insert code snippet (after trigger search or insertion from DB). If it is not selected, only the trigger will be inserted and the user needs to fire the

trigger (CTRL+J) - by default - for the snippet to be inserted in place of the trigger. If the above option is checked, the trigger will be inserted followed by the snippet. In both cases (CTRL+Z) undoes what was done.

Use this dialog as follows:

1. Place the cursor where you want to insert a snippet
2. Enter the default shortcut: (SHIFT+CTRL+Q) to open the dialog window
3. Use edit boxes for trigger/snippet filtering
4. Locate the desired trigger/snippet
5. ENTER or OK to insert the trigger

As the search database - however large it may be - is relatively small and the search and filtering mechanism is very fast, searches can be made progressive (adding more characters to the initial(s)) or retroactive (completely changing content) in the trigger filter box.

Shortcuts (Application)

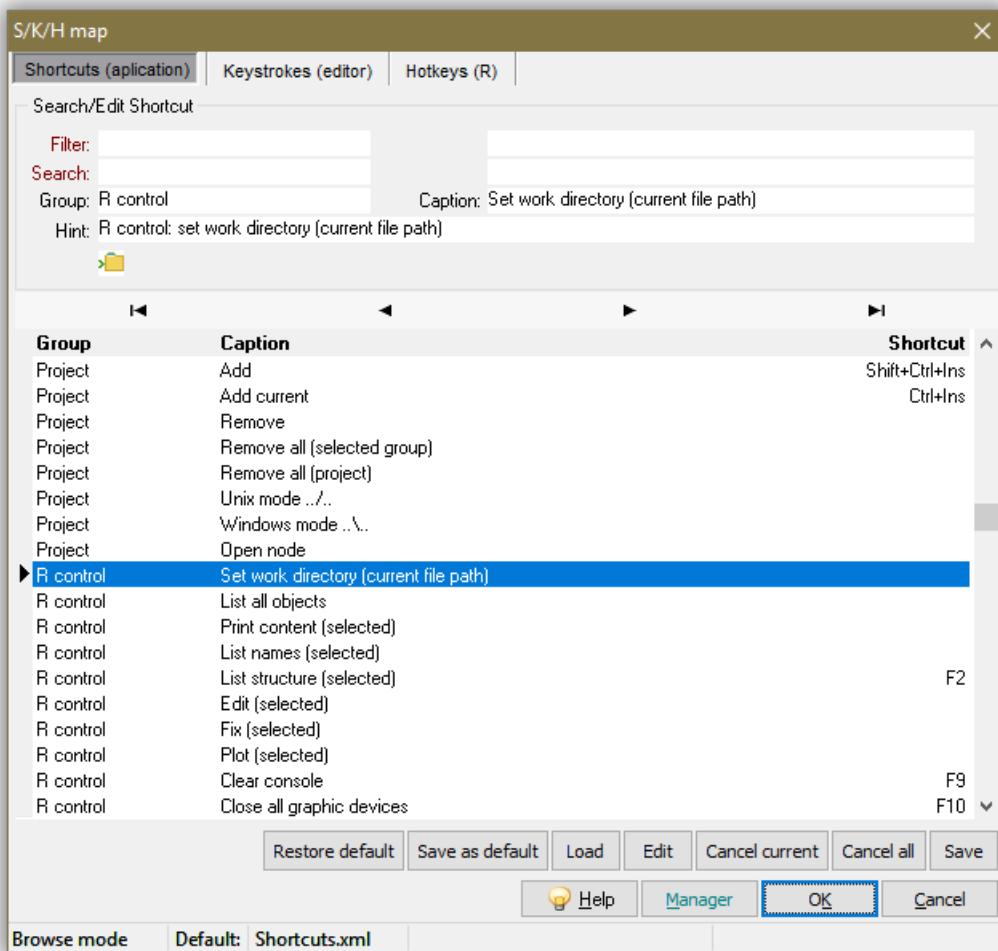


FIGURE 3.32: Shortcuts (app) (dataset).

The *Shortcuts (app)* (Figure 3.32) allows the user to set the shortcuts related to the application, it works together with the *Editor keystrokes*, and allows for high level of customization.

The difference between *Shortcuts* and *Hotkeys (operational system)* is that the former works only with the focus on Tinn-R, whereas hotkeys work with the focus anywhere.

Read below a brief description of available buttons.

Restore default: Restores the file `Shortcuts.xml` from the origin (Install-Path/data/data.zip). Any prior changes to the file `Shortcuts.xml` in use will be lost.

Save as default: Opens the save dialog allowing to save the file. From this point, this file will be the new default shortcuts.

Load: Opens the open dialog allowing to load a shortcut file. From this point on, this file will be the new default shortcuts.

Edit: Sets the table in edition mode.

Cancel current: Cancels any changes made to the current edition.

Cancel all: Cancels all changes made to the database prior to *Save* or *Save as default*.

Save: Saves to text file (XML) all changes made to the current table.

Close: Closes the dialog. All changes not saved will be lost.

S/K/H manager

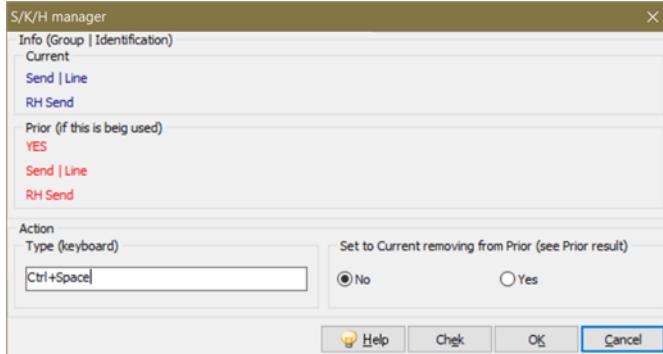


FIGURE 3.33: S/K/H manager.

S/K/H manager (Figure 3.33) is the dialog responsible for checking, replacing and assigning shortcuts to the application itself, through the `SynEdit` class instances (editor, Term/IO and Term/LOG) and also through the hotkeys associated with R: send, control and custom.

In other words, it is the dialog that centralizes the checks and assignments of shortcuts and hotkeys. Access to it is via the [Manager] button in the S/K/H map dialog ((Figure 3.36, 3.21 and 3.20)).

Snippets

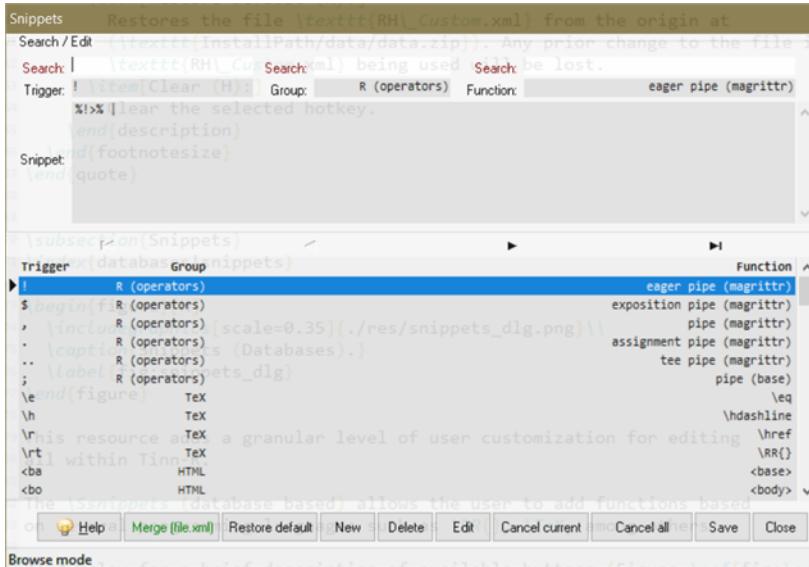


FIGURE 3.34: Snippets (dataset).

This interface (Figure 3.41) allows to change the default SynEdit keystrokes. It is possible only change the keystroke associaed to any ecAction (execute command action). A set of user friendly keystrokes gives high productivity leading with all instances of the class *SynEdit*: Editor, IO and LOG.

Read below for a brief description of available buttons (Figure 3.37):

Restore default (K): Restores the file *Editor_Keystrokes.xml* from the origin at (*InstallPath*/data/data.zip). Any prior change to the file ini file *Editor_Keystrokes.xml* being used will be lost.

Clear (K): Clear the selected keystroke.

Tools/Completion has been replaced by Tools/Snippets. The *Completion.xml* table is now named *Snippets.xml*. Users who have made (and saved) an extensive dataset of completion (before) (snippets from now on) will be able to notice a new button [Merge (file .xml)] that has been added in the new Snippets dialog window. When a new dataset (or any update to the source) is made, the user's dataset (*Completion.xml* or *Snippet.xml*, respectively) is saved in the **bkp** folder which is located in the Tinn-R startup folders (*Help/Ini files (path information)*). The new button allows the user's old base (completion or snippets) to be

merged with the current one distributed with the new versions and eventually modified at the source. Thanks to Ivan B. Allaman for the helpful suggestion.

Tinn-R updater

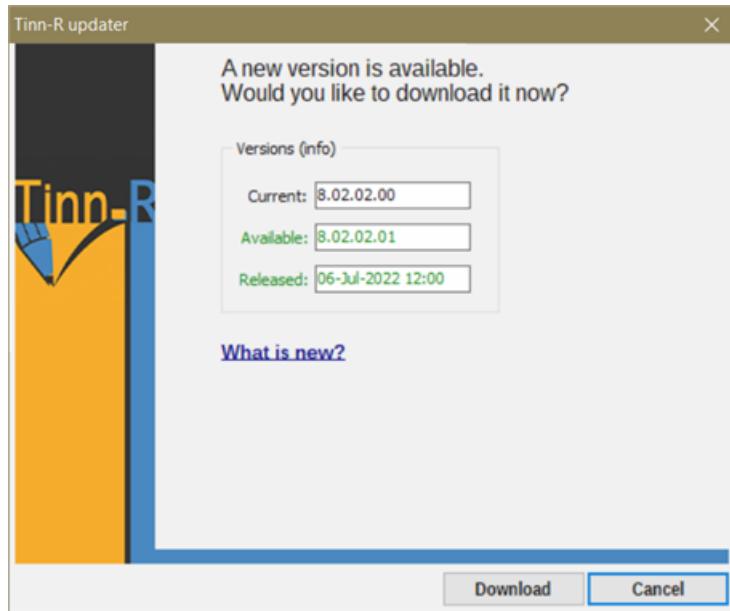


FIGURE 3.35: Tinn-R updater dialog.

This dialog (Figure 3.35) is responsible for updating the application. As you can see in the figure, information is displayed about the user's current version, the current version available as well as the date of its release. It also shows a link [[What is new?](#)] so that the user can read the new features implemented on the project website and decide whether to update or not. The [DOWNLOAD] button starts the update process, just follow the subsequent instructions for the program to be updated.

3.3 DATABASE

The database uses the native XML engine provided by Embarcadero.

Shortcuts (dataset - app)

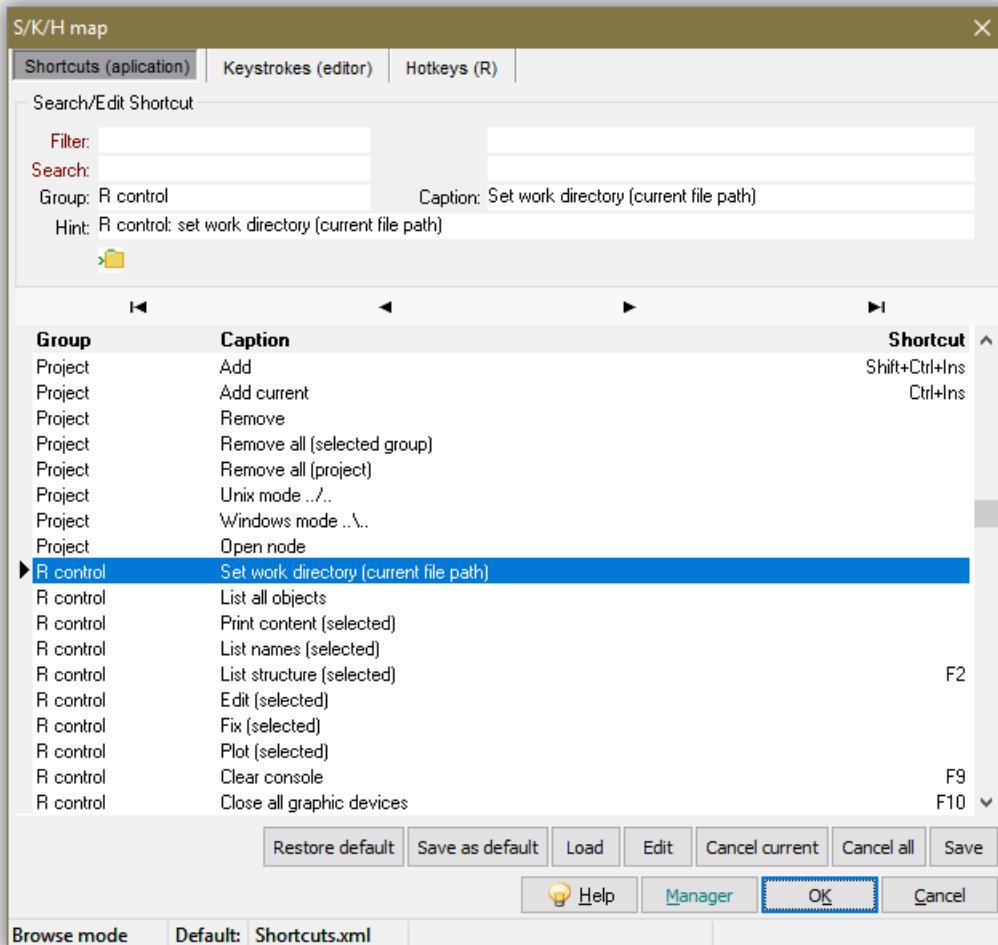


FIGURE 3.36: Shortcuts (dataset).

Read below for a brief description of available buttons (Figure 3.36):

Restore default: Restores the file `Shortcuts.xml` from the origin (Install-Path\data\data.zip). Any prior changes to the file `Shortcuts.xml` currently being used will be lost.

Save as default: Opens the save dialog allowing you to save the file. From this point on, this file will be the new default shortcut.

Load: Opens the open dialog allowing you to load a shortcut file. From this point on, this file will be the new default shortcut.

Edit: Places the table in edition mode.

Cancel current: Cancels any change made to the current edition.

Cancel all: Cancels all changes made to the database prior to *Save* or *Save as default*.

Save: Overwrites the text file (XML) saving all changes made to the current table.

Close: Closes the dialog. All non-saved changes will be lost.

Keystrokes (dataset - editor)

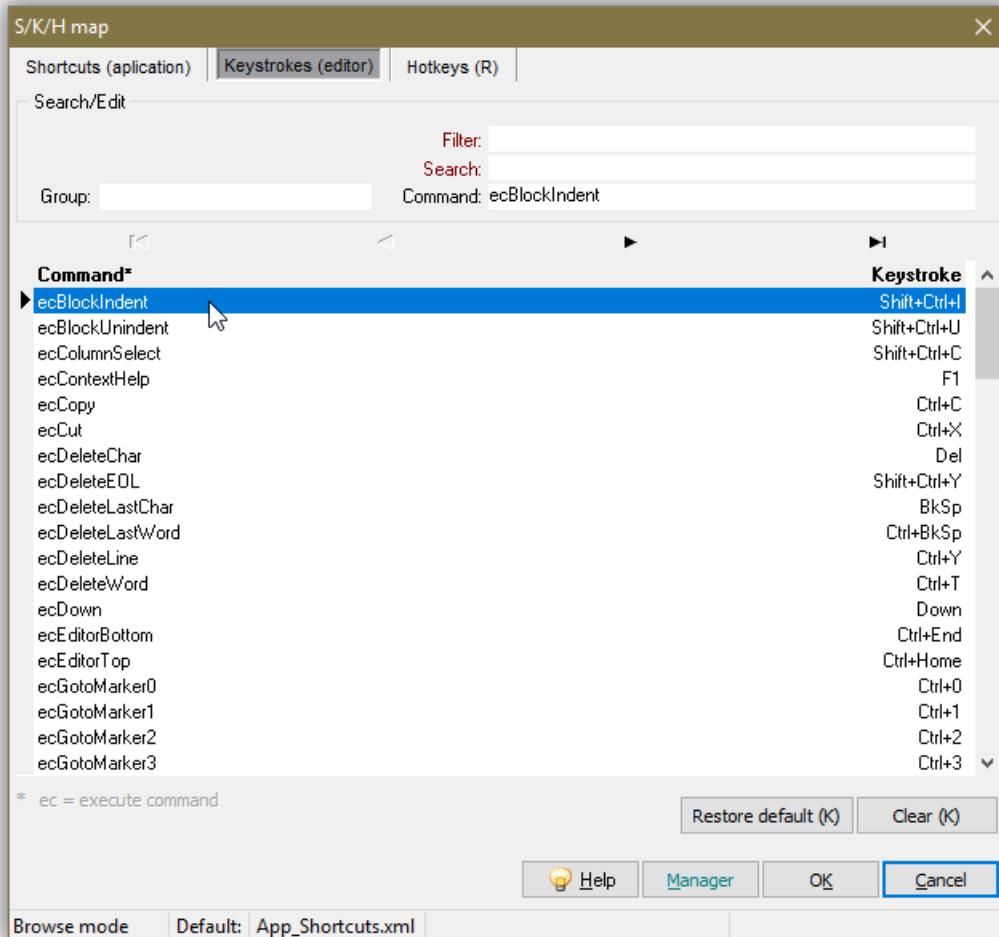


FIGURE 3.37: Keystrokes (dataset).

This interface (Figure 3.37) allows to change the default SynEdit keystrokes. It is possible only change the keystroke associaed to any ecAction (execute command action). A set of user friendly keystrokes gives high productivity leading with all instances of the class *SynEdit*: Editor, IO and LOG.

Read below for a brief description of available buttons (Figure 3.37):

Restore default (K): Restores the file `Editor_Keystrokes.xml` from the origin at (`InstallPath/data/data.zip`). Any prior change to the file ini

file `Editor_Keystrokes.xml` being used will be lost.

Clear (K): Clear the selected keystroke.

Hotkeys (dataset - send R)

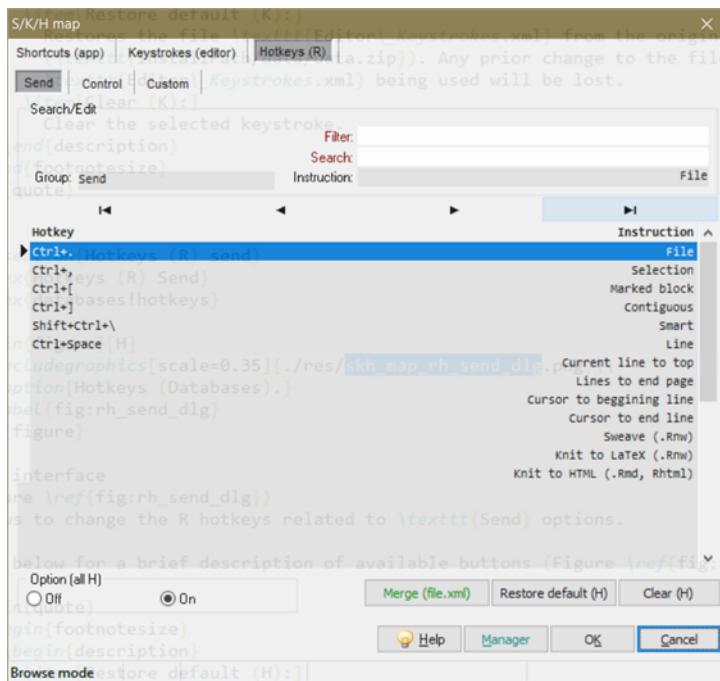


FIGURE 3.38: Hotkeys (dataset).

This interface (Figure 3.38) allows to change the R hotkeys related to Send options.

Read below for a brief description of available buttons (Figure 3.38):

Merge (file.xml): When - in new versions - any update is made to the `RH_Send.xml` file database, which accompanies the new version, the user's original file will be stored in the `bkp` folder of the mother folder (Tinn-R) of the preference files and initialization. This button allows you to retrieve (merge) all the information that the user has registered and that is not included in the new dataset.

Restore default (H): Restores the file `RH_Send.xml` from the origin at (`InstallPath/data/data.zip`). Any prior change to the file ini file `RH_Send.xml` being used will be lost.

Clear (H): Clear the selected hotkey.

Hotkeys (dataset - control R)

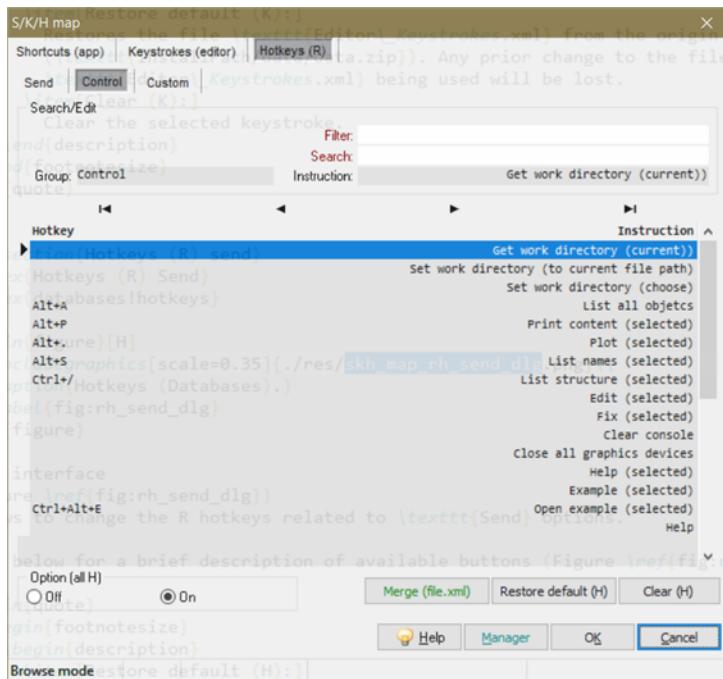


FIGURE 3.39: Hotkeys (dataset).

This interface (Figure 3.39) allows to change the R hotkeys related to Send options.

Read below for a brief description of available buttons (Figure 3.39):

Merge (file.xml): When - in new versions - any update is made to the RH_Control.xml file database, which accompanies the new version, the user's original file will be stored in the bkp folder of the mother folder (Tinn-R) of the preference files and initialization. This button allows you to retrieve (merge) all the information that the user has registered and that is not included in the new dataset.

Restore default (H): Restores the file RH_Control.xml from the origin at (InstallPath\data\data.zip). Any prior change to the file ini file RH_Control.xml being used will be lost.

Clear (H): Clear the selected hotkey.

Hotkeys (dataset - custom R)

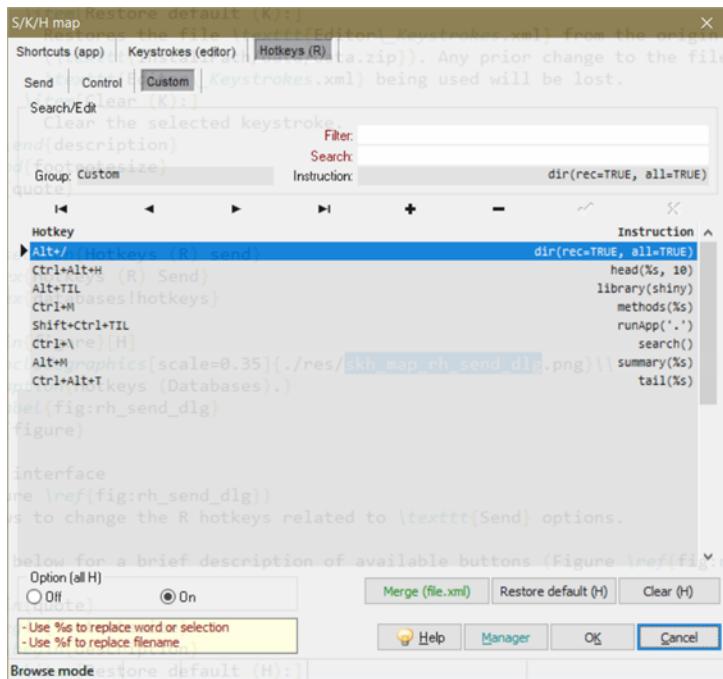


FIGURE 3.40: Hotkeys (dataset).

This interface (Figure 3.40) allows to change the R hotkeys related to Send options.

Read below for a brief description of available buttons (Figure 3.40):

Merge (file.xml): When - in new versions - any update is made to the RH_Custom.xml file database, which accompanies the new version, the user's original file will be stored in the bkp folder of the mother folder (Tinn-R) of the preference files and initialization. This button allows you to retrieve (merge) all the information that the user has registered and that is not included in the new dataset.

Restore default (H): Restores the file RH_Custom.xml from the origin at (InstallPath/data/data.zip). Any prior change to the file ini file RH_Custom.xml being used will be lost.

Clear (H): Clear the selected hotkey.

Snippets (dataset)

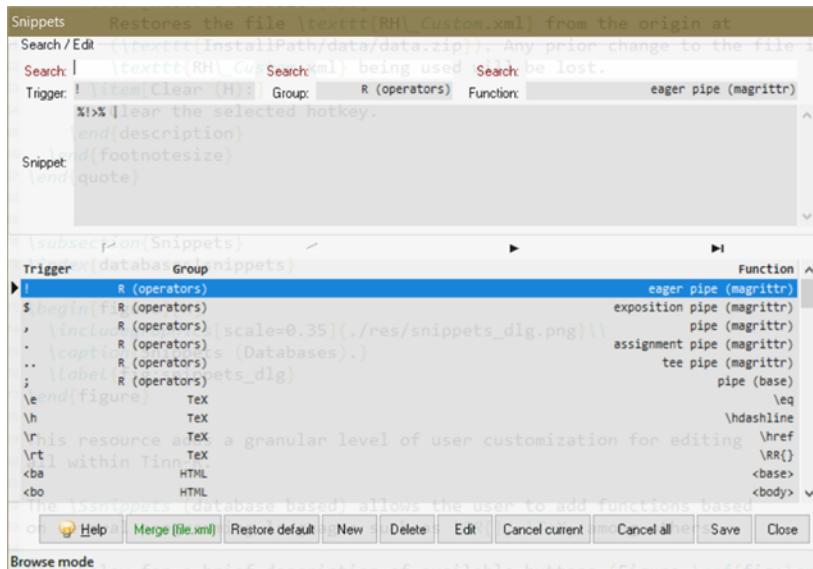


FIGURE 3.41: Snippets (dataset).

This resource adds a granular level of user customization for editing all within Tinn-R.

The Snippets (database based) allows the user to add functions based on several programming languages such as R, TeX, among others.

Read below for a brief description of available buttons (Figure 3.41):

Restore default: Restores the file `Snippets.xml` from the origin at (`InstallPath/data/data.zip`). Any prior change to the file `Snippets.xml` being used will be lost.

New: Places the table in insertion mode.

Delete: Deletes the current registry from the table.

Edit: Places the table in edition mode.

Cancel current: Cancels any change made to the current edition.

Cancel all: Cancels all changes made to the database prior to *Save*.

Save: Overwrites the text file (XML), saving all changes made to the current table.

Close: Closes the dialog. All non-saved changes will be lost.

Comments (dataset)

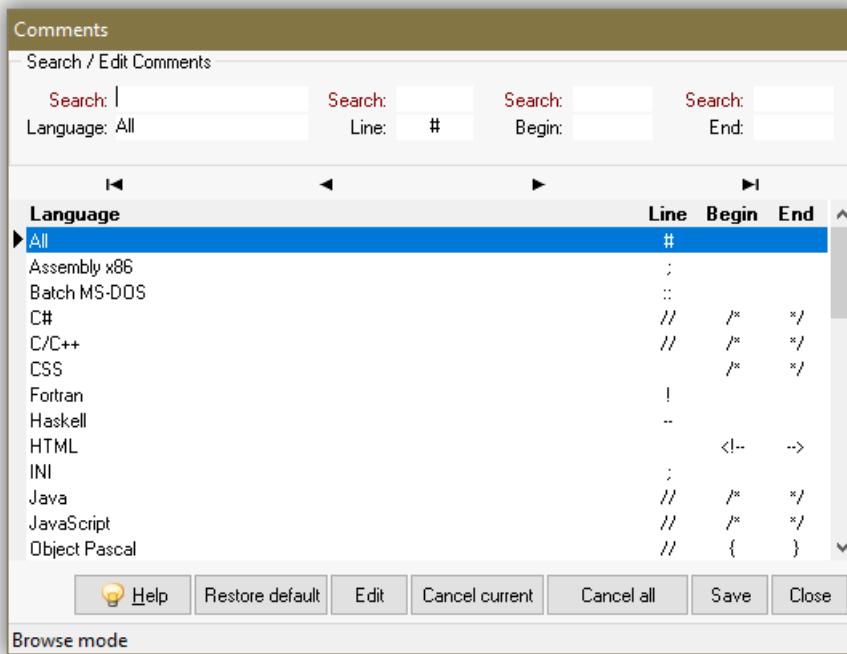


FIGURE 3.42: Comments (dataset).

The *Comments* resource is very simple and allows high level of user customization.

From version 3.0.1.0 Tinn-R automatically recognizes the language of the file on focus. Further, inside the file - if it is a syntax a multi-highlighter (complex syntax) - which language of the line where the cursor (or selection) is found.

This identification is done automatically if (and only if) the option (*x*) *Auto detect language (recomended)* is checked. Otherwise the user is forcing the application to use the comments of the selected language (indicator arrow).

Selected code snippets involving more than one language will not be commented/uncommented and a warning message is issued. That is, you must select only the snippet of a single language.

Read below a brief description of available buttons (Figure 3.42):

Restore default: Restores the file `Comments.xml` from the origin at (InstallPath/data/data.zip).

Any prior changes in the file `Comments.xml` currently being used will be lost.

- Edit:** Places the table in edition mode.
- Cancel current:** Cancels any change made to the current edition.
- Cancel all:** Cancels all changes made to the database prior to *Save*.
- Save:** Overwrites the text file (XML) saving all changes made to the current table.
- Close:** Closes the dialog. All non-saved changes will be lost.

Card (dataset R)

The *card* was based on two R cards already published: R/Rpad Reference Card by Tom Short and R reference card by Jonathan Baron.

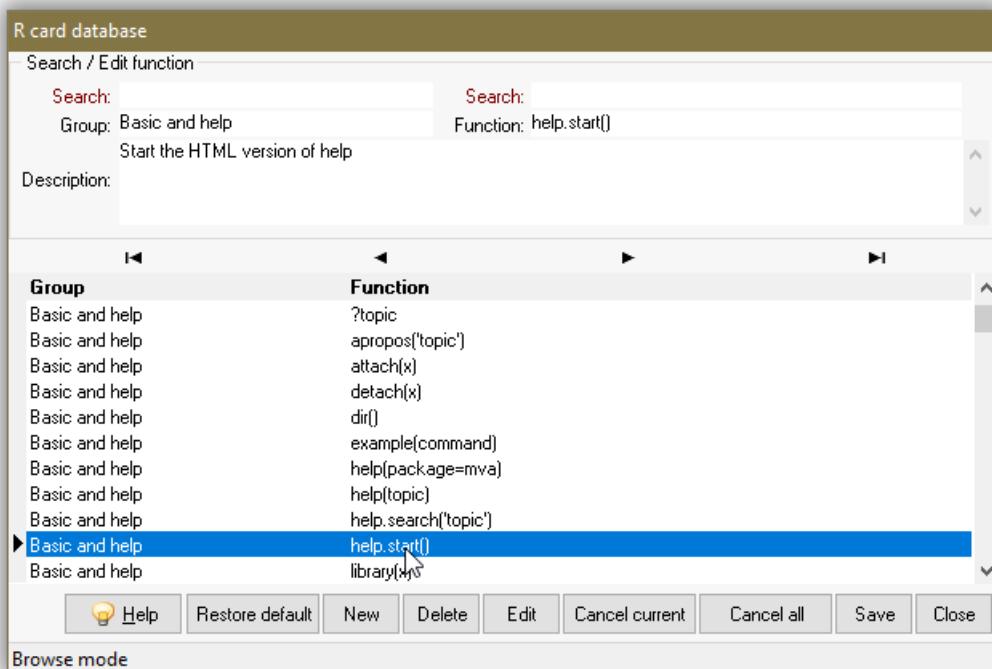


FIGURE 3.43: R card (dataset).

Read below a brief description of available buttons (Figure 3.43):

Restore default: Restores the file `Rcard.xml` from the origin at (`InstallPath/data/data.zip`). Any prior changes in the file `Rcard.xml` currently being used will be lost.

New: Places the table in insertion mode.

Delete: Delete the current registry from the table.

Edit: Places the table in edition mode.

Cancel current: Cancels any change made to the current edition.

Cancel all: Cancels all changes made to the database prior to *Save*.

Save: Overwrites the text file (XML) saving all changes made to the current table.

Close: Closes the dialog. All non-saved changes will be lost.

Mirrors (dataset R)

The *Mirrors* is an interface that allows the user to manage the repositories (or mirrors) of R. You should always choose a repository physically closest to where you are, so that, the Web communication tends to be faster and more efficient.

The default mirror is the University [Wien](#) (Austria). Consider that this is the central mirror of CRAN.

The reasons for the Tinn-R always set a repository are two:

- Prevent R keep asking which repository you want to use in each session;
- Workaround of intermittency (only Rterm.exe) display the dialog for selecting the repository. That is, sometimes the dialog is displayed and not others. The cause of this intermittency is still unknown.

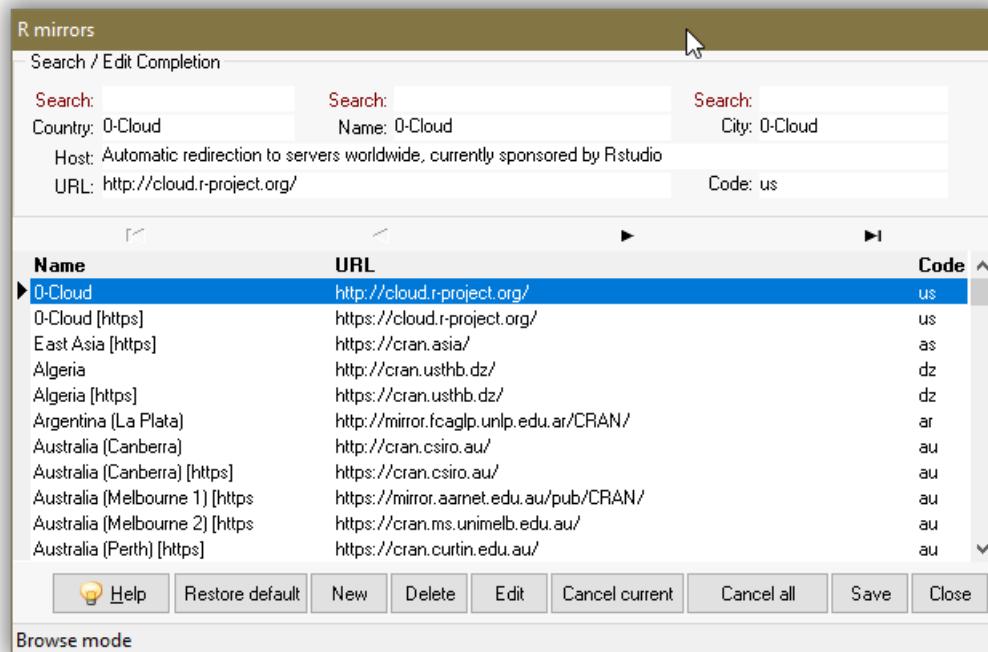


FIGURE 3.44: R mirrors (dataset).

Read below a brief description of available buttons (Figure 3.44):

Restore default: Restores the file `Rmirrors.xml` from the origin at (`InstallPath/data/data.zip`).

Any prior change to the file `Rmirrors.xml` while being used will be lost.

New: Places the table in insertion mode.

Delete: Deletes the current registry from the table.

Edit: Places the table in edition mode.

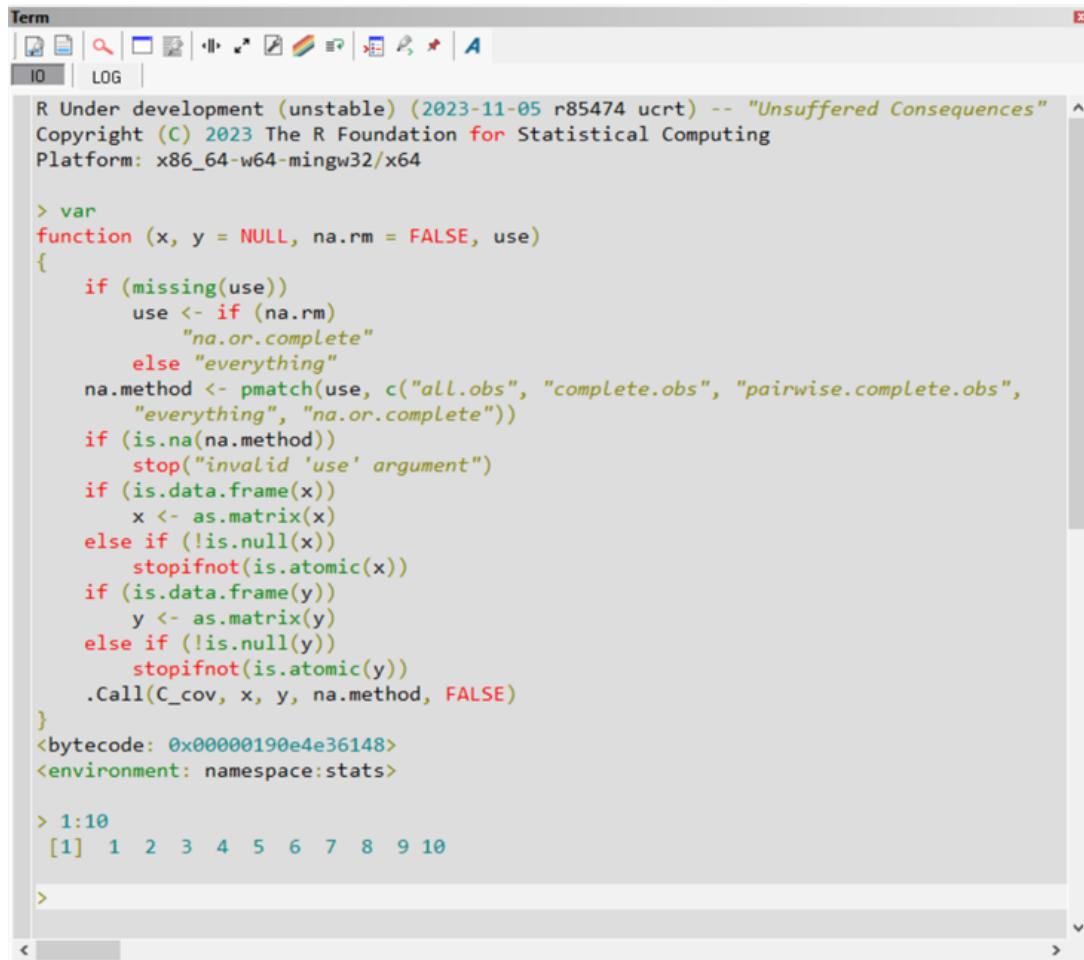
Cancel current: Cancels any change made during the current editing session.

Cancel all: Cancels all changes made to the database prior to `Save`.

Save: Overwrites the text file (XML) while saving all changes made to the current table.

Close: Closes the dialog. All changes not previously saved will be lost.

3.4 TERM INTERFACE



The screenshot shows the Tinn-R Term interface window. The title bar says "Term". The menu bar includes "File", "Edit", "View", "Help", and "About". Below the menu is a toolbar with icons for file operations like Open, Save, Find, Copy, Paste, and Print. A tab bar shows "IO" and "LOG". The main area displays R session output:

```
R Under development (unstable) (2023-11-05 r85474 ucrt) -- "Unsuffered Consequences"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

> var
function (x, y = NULL, na.rm = FALSE, use)
{
  if (missing(use))
    use <- if (na.rm)
      "na.or.complete"
    else "everything"
  na.method <- pmatch(use, c("all.obs", "complete.obs", "pairwise.complete.obs",
    "everything", "na.or.complete"))
  if (is.na(na.method))
    stop("invalid 'use' argument")
  if (is.data.frame(x))
    x <- as.matrix(x)
  else if (!is.null(x))
    stopifnot(is.atomic(x))
  if (is.data.frame(y))
    y <- as.matrix(y)
  else if (!is.null(y))
    stopifnot(is.atomic(y))
  .Call(C_cov, x, y, na.method, FALSE)
}
<bytecode: 0x00000190e4e36148>
<environment: namespace:stats>

> 1:10
[1] 1 2 3 4 5 6 7 8 9 10

>
```

FIGURE 3.45: Term interface.

The implementation of a Term interface (Figure 3.45) in Tinn-R has the following aims:

- To address some limitations (edition, navigation and control) imposed by the Rgui.exe interface;
- To add more flexibility and power to the GUI/Editor;
- To maintain the prior user knowledge associated with Tinn-R editor and the Rgui.exe console;

- To maintain the structural simplicity of the application;
- To use a more efficient engine of Inter Process Communication (IPC) than the Windows clipboard used in previous versions.

The *IO* (Figure 3.46) and *LOG* (Figure 3.47) interfaces are instances of the class SynEdit. In other words, all prior user knowledge of the resources associated with the editor were preserved:

- Free navigation with keyboard keys;
 - Marks;
 - Shortcuts;
 - Syntax;
 - Match brackets;
 - Tips;
 - Data completion;
 - Edition: copy, paste, cut, etc;
 - Selection/copy/paste in column mode;
 - etc.
1. *IO* (Figure 3.46): The aim was to add flexibility and power, i.e., joining the power of SynEdit (editor) and the functionality of a common console.
 2. *LOG* (Figure 3.47): Has three basic objectives:
 - a) To receive and show warnings and error messages;
 - b) To make the *IO* interface cleaner;
 - c) To avoid synchronization difficulties with the inter process communication (IPC) called *pipe* used.

10

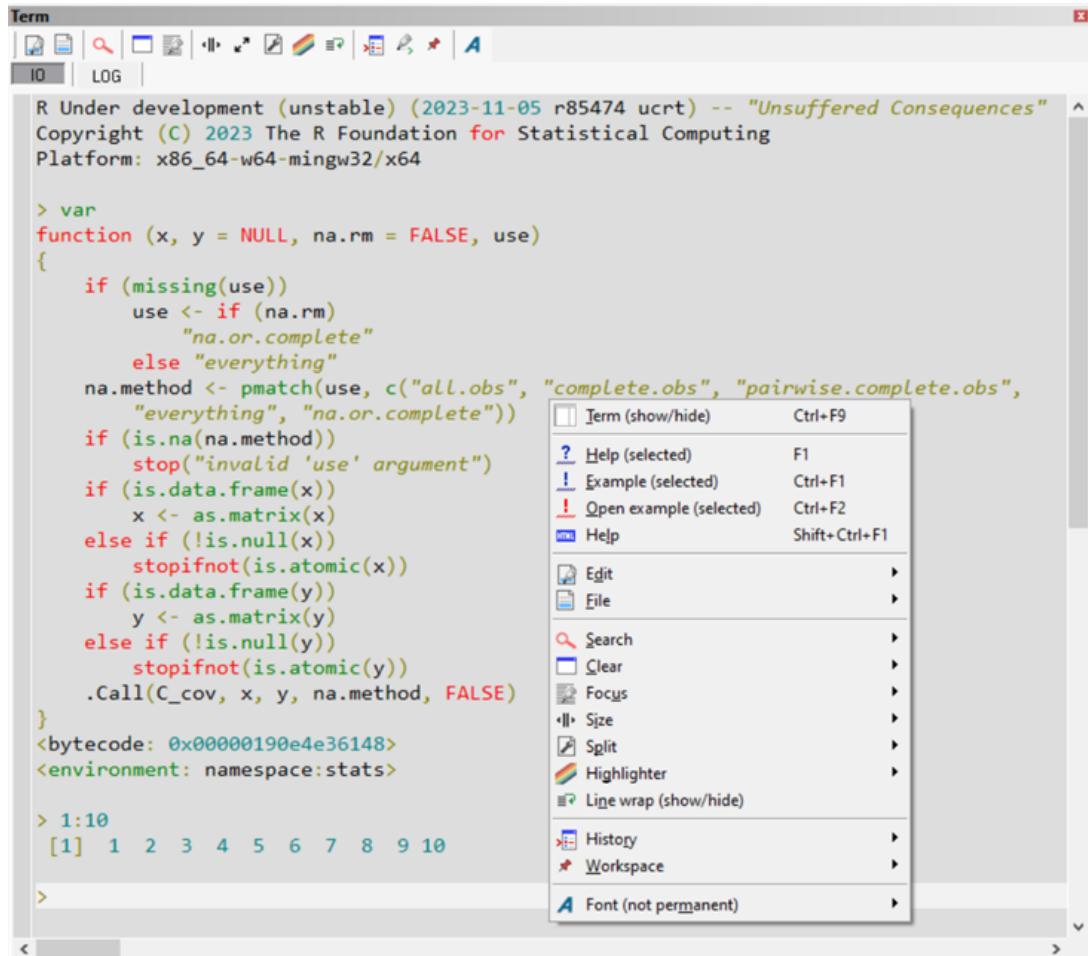


FIGURE 3.46: IO (Term interface).

The *IO* interface (Figure 3.46 and Table 3.5) is used to receive output (SDTOUT) from the R environment.

It is necessary to adjust some R options (for example: `options(width=70)`) to obtain a suitable number of characters in each single line, according to hardware and user preferences (side of *IO*, place of *IO*, length of *IO*, width of *IO*, type and size of font). Once you get a suitable result, it is a good practice to add this option to the `Rprofile.site` (located inside of the folder *etc* where the R was installed) file. Thus, your option will always be set when starting R.

Resource	Description
Edition	All resources available to the editor (copy, paste, cut, etc) can be used
Free navigation	Using keyboard keys: Home, Page Up, Page Down, End, Left, Top, Right and Bottom
Marks	CTRL + [0..9] can be used to mark, SHIFT + CTRL + [0..9] to go to prior marks
Shortcuts	All shortcuts available to the editor are also to the IO
Syntax	Two options: Text and R
Match brackets	Makes it easier to build more complex instructions like <code>plot(sqrt(rnorm(1e3)), pch='.', cex=3)</code>
Tips	Are invoked using the same trigger as the editor
Data completion	Are invoked using the same trigger as the editor

Table 3.5: IO interface, main resources available.

The IO is an instance of SynEdit. Therefore, it can be edited and used like the editor, allowing the tasks showed in the Table 3.5.

If the *IO* has the focus, all actions of the R toolbar and main menu associated with control R can be used in the IO interface.

The *IO* interface has a special pop-up menu allowing the most common tasks. It is self-explanatory. So, make a small tour (right mouse button inside of Term/IO) to find out about its options.

Some details:

- Shortcuts and pop-up menu make it easy to change among the interfaces: *Editor*, *IO* and *LOG*:
 1. if *IO* and *LOG* are in distinct tabs (views), the common Windows shortcut CTRL + TAB changes the active page (*IO*-*LOG*).
 2. Any prior line can be sent another time by just putting the cursor in any place of it and typing: CTRL + ENTER;
- The last line of the *IO* interface (the prompt) has special features:
 1. CTRL+ENTER must be used to send any line or selection (single line only) to R interpreter;
 2. ALT+DOWN and ALT+UP are the shortcuts (prior/later) for command R history. The R history is continuous, cyclic, and have a 100 line limit.

LOG

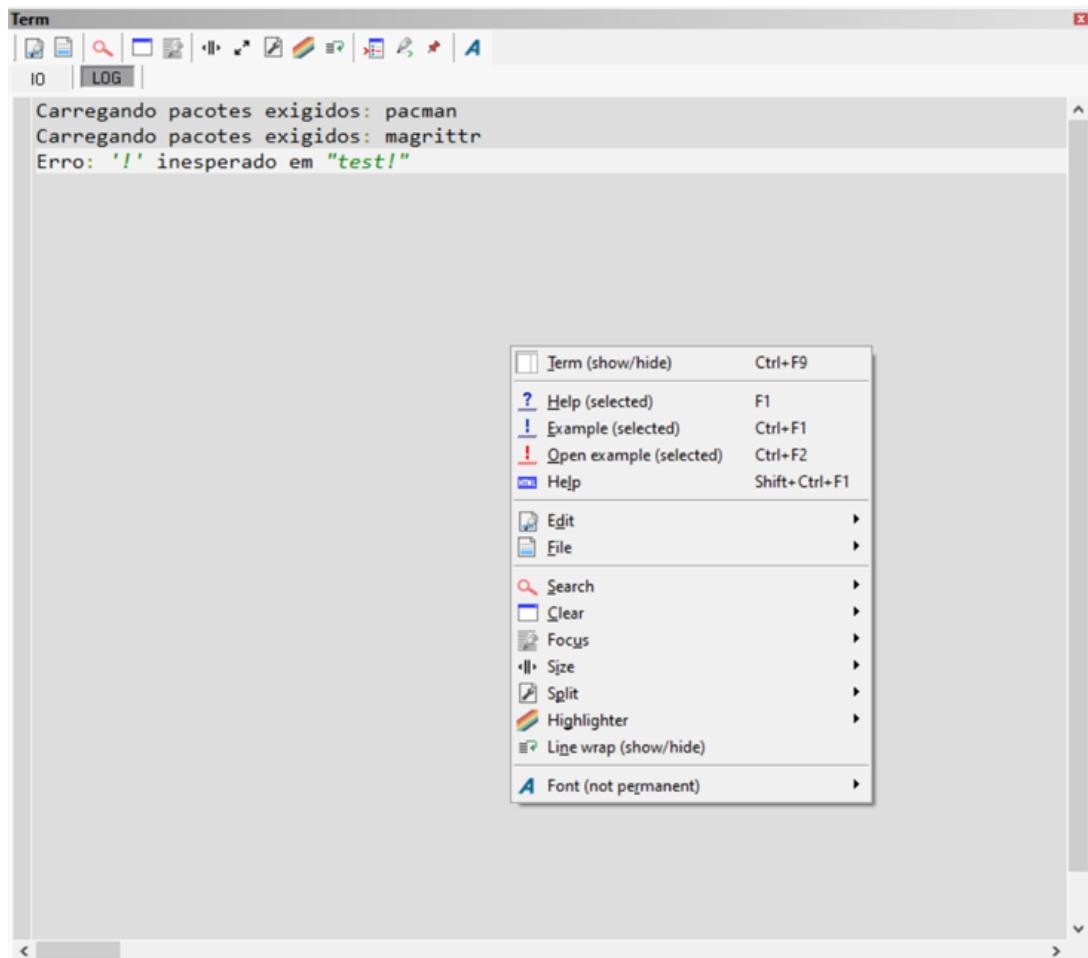


FIGURE 3.47: LOG (Term interface).

The *LOG* interface (Figure 3.47) is used to receive warnings and error messages (SDTERR) from the R environment.

It has a special pop-up menu that allows the most common tasks. It is self-explanatory. So, take a small tour (right mouse button inside of Term/LOG) to know all options.

Most of the resources available to the *IO* are also available to this interface.

3.5 TOOLS INTERFACE

This graphical interface (Figure 3.48) was projected to allow access to Tinn-R resources and also to accommodate future growth of related news resources.

Position: starting from version 2.1.1.1 (Oct/15/2008) this interface is dockable. It can float or be docked on the left, top, right, or bottom sides of the main interface.

Misc (Tools)

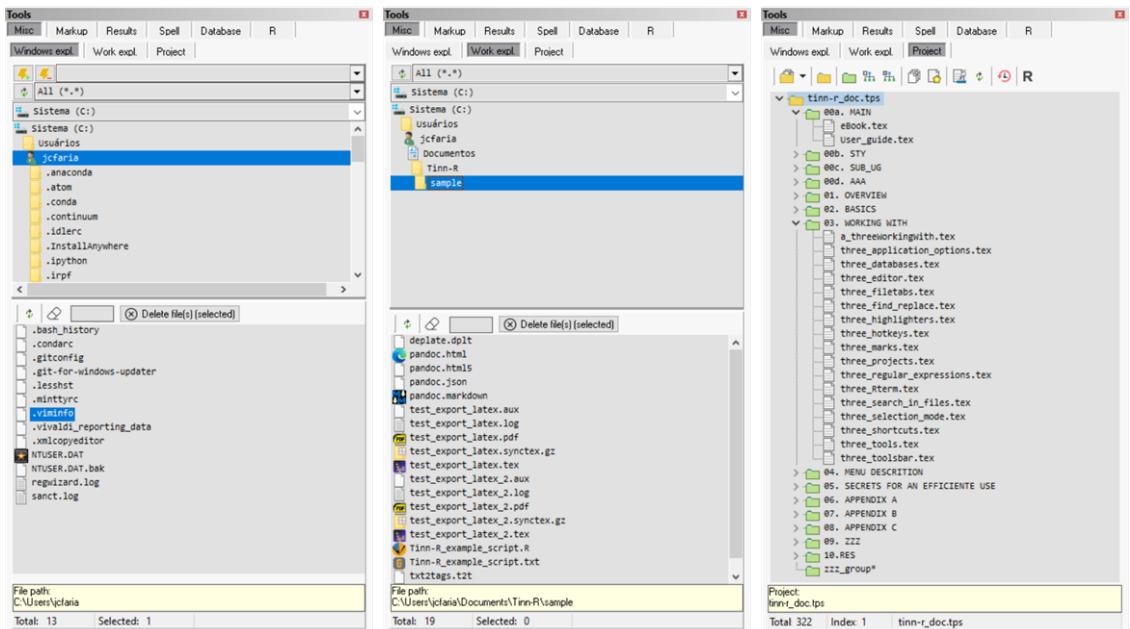


FIGURE 3.48: Tools interface.

Tool	Description
Windows expl.	See details ...
Work expl.	See details ...
Project	See details ...

Table 3.6: Misc (tools)

The resources are showed in Figure 3.48 and Table 3.6.

Windows expl.:

See Figure 3.48.

- Allows manager favorites (add and remove);
- Allows filter by file extension;
- Has pop-up menus similar to Windows explorer;
- Supports drag and drop actions (it is possible to drag any file and drop it on the editor interface to be opened).

Work expl.:

See Figure 3.48.

- Always shows the folder related to the latest file opened;
- Does not have a pop-up menu;
- Supports drag and drop actions, i.e., it is possible to drag any file and drop it in the editor interface that will be opened.

Project:

See Figure 3.48.

- Allows for project management using a graphical interface;
- Supports drag and drop actions, i.e., it is possible to drag the entire project, groups, or any file and then drop them into the editor interface that will be opened:
 - Project: will open all files related to the current project;
 - Group: will open all files for the selected group;
 - File: will open the selected file.
- It is possible to send an entire project, a selected group, or an individual file to the R environment through a pop-up menu.
- Source file of project:
 - It is possible to edit the project in text mode (with the button *Project: edit (as text file)* of the specific toolbar). After any change, save the text file (it contains the textual description of the project structure) and reload the file to the graphical interface (with the button *Project: reload (from text file)* of the specific toolbar).

Tool	Description
Txt2tags	Sets marks, macros and settings of Txt2tags convertor
TeX	Sets TeX symbols settings in a customizable manner

Table 3.7: Markup (Tools).

- Any changes to the graphical interface will be reflected in the text file for the project, after it is saved.
- The best way to work with projects (graphics of textual mode) depends on the complexity of the actions and the user preference. For single tasks, we suggest that you use the graphical mode. For complex actions, it is faster to use the textual mode with all editor resources.

Markup (Tools)

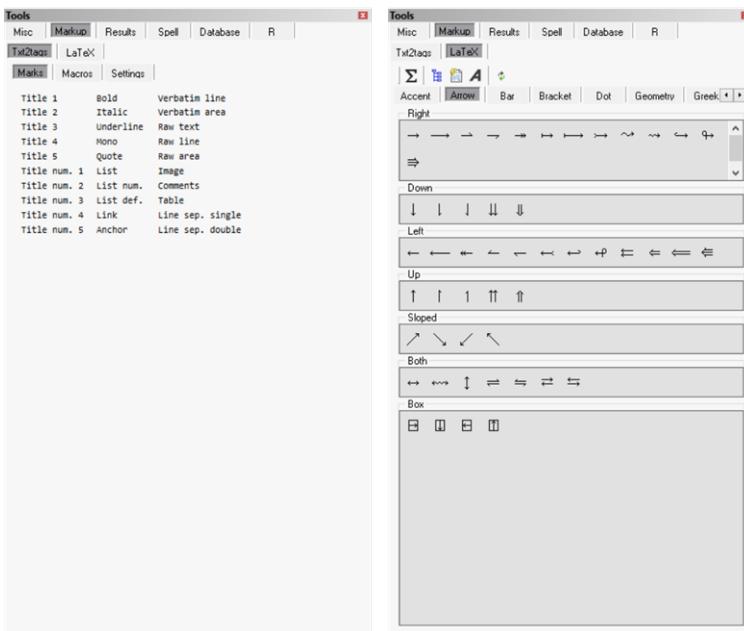


FIGURE 3.49: Markups (Tools).

It contains resources (Figure 3.49 and Table 3.7) related to the Txt2tags and TeX.

Txt2tags:

Tool	Description
Ini log	Displays useful results when starting Tinn-R
Search	Interface for <i>Search</i> results associated with <i>Search in files</i>
Hex viewer	Interface for hex viewer

Table 3.8: Results (Tools).

Sets (Figure 3.49) marks, macros, and settings for the Txt2tags convertor. A single click over any graphical will add it to the current editor.

\LaTeX :

Set (Figure 3.49) of \LaTeX symbols. A single click over any graphical object will add it to the current editor;

The symbols, place and order of all symbols are customizable. To customize them, open the folder *latex* and edit *ini* path. At the end of the edition, update the interface using the button *Latex: reload symbols (from ini)*. Be careful when editing the symbols to maintain the name structure. For example: *Number_SymbolName.FileExtension,001_alpha.gif*, *002_beta.gif*. The number will be used to order symbols in the graphical interface, while the name will be used (if recognized) as a \LaTeX symbol.

Results (Tools)

It contains resources (Table 3.8) related to *Ini log*, *Search* and *Hex viewer*.

Ini log:

Topic	Description
Path of executable and sources (origin)	Lists executable files and resources
Path of ini folders	Lists the path of all folders of the ini
Verification of necessary folder and files	Lists the status of folders and files of ini
Tinn-R, app, bkp, colors, editor, syntax, syntax bkp and tmp	Lists the status of these folders
Data (version)	Lists the status of this folder and files
Latex (version)	Lists the status of this folder and files
Project (version)	Lists the status of this folder and files
Editor options	Lists the status of this folder and files
Shortcuts (version)	Lists the status of this folder and files
Unihighlighter (version)	Lists the status of this folder and files
Tmp	Lists the status of this folder

Table 3.9: Ini log

```

Tools | Misc | Markup | Results | Spell | Database | R |
Inilog | Search | Hex viewer |
Version - 9.01.00.01
Started - 2024-03-10 10:29:23

1. Path of executable and sources (origin)
Tinn-R - C:\Users\jcfaria\Documents\Tinn-R\bin\Tinn_R
  data - C:\Users\jcfaria\Documents\Tinn-R\data\data.
  latex - C:\Users\jcfaria\Documents\Tinn-R\latex\late
  syntax - C:\Users\jcfaria\Documents\Tinn-R\syntax\syn
  utils - C:\Users\jcfaria\Documents\Tinn-R\utils\%
  \example - C:\Users\jcfaria\Documents\Tinn-R\utils\%
  \info - C:\Users\jcfaria\Documents\Tinn-R\utils\%
  \install - C:\Users\jcfaria\Documents\Tinn-R\utils\%
  \mirrors - C:\Users\jcfaria\Documents\Tinn-R\utils\m

2. Path of ini folders
Tinn-R - C:\Users\jcfaria\Documents\Tinn-R\app_data
  \app
  \app
  \colors
  \data
  \editor
  \project
  \latex
  \license
  \syntax
  \syntax_bkp

3. Verification of necessary folder and files
3.1. Tinn-R, app, bkp, colors and editor
Main folder - C:\Users\jcfaria\Documents\Tinn-R\app_d
  \app : OK
  \bkp : OK
  \colors : OK
  \editor : OK

3.2. Syntax
Main folder - C:\Users\jcfaria\Documents\Tinn-R\app_d
  \Defn(version = 5.04.03.01): OK
  \Dark(version = 5.04.03.01): OK
  \Gray(version = 5.04.03.01): OK
  \LGray(version = 5.04.03.01): OK

3.3. Data
Main folder - C:\Users\jcfaria\Documents\Tinn-R\app_d
  \SColor.App.xml(version = 9.00.00.02): OK
  \Cache.web2dinson - 5.04.01.011: OK
  < >
  
```

FIGURE 3.50: Inilog (Tools/Results).

Displays (Figure 3.50 and Table 3.9) useful results when starting Tinn-R. If you submit a bug report, please also send the results for the respective page by copying & pasting.

Search:

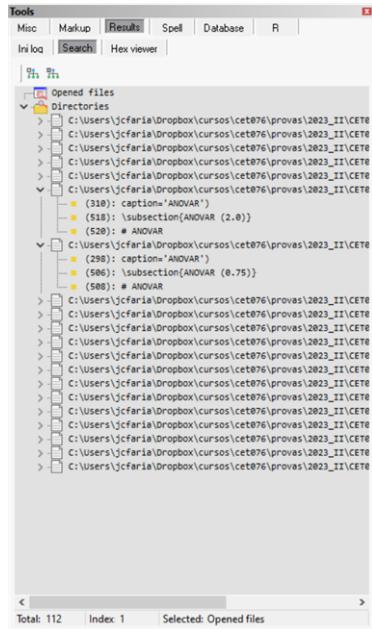


FIGURE 3.51: Search (Tools/Results).

The interface (Figure 3.51) for *Search* results associated with *Search in files*.

The results for the *Search in files* actions are displayed as a tree with all files. Double click the file to open it in the editor interface.

Hex viewer:

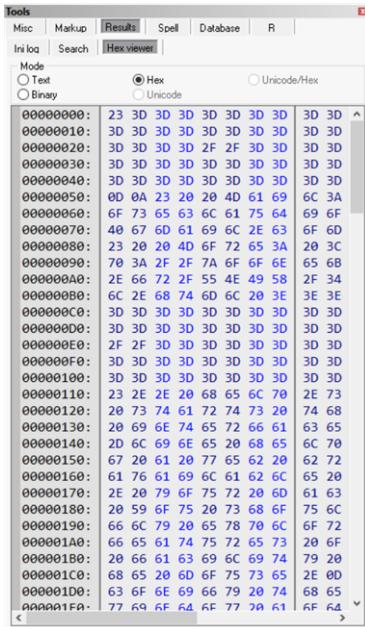
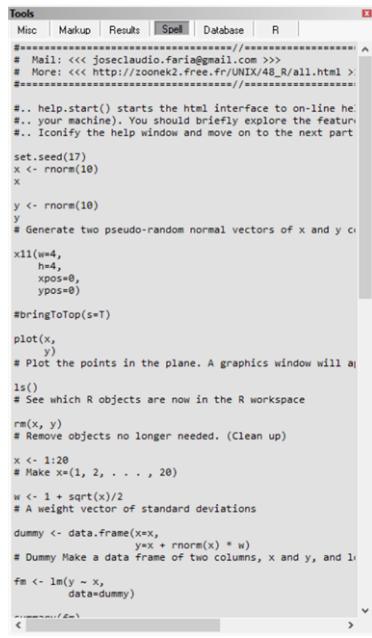


FIGURE 3.52: Hex (Tools/Hex viewer).

The interface (Figure 3.52) for *Hex viewer* results associated to any active file.

Spell (Tools)



```

Tools
Misc | Markup | Results | Spell | Database | R |
=====
#<> Mail: <<> josedclaudio.faria@gmail.com >>
#<> More: <<> http://zonnek2.free.fr/UNIX/48_R/aall.html >>
=====//=====

#. help.start() starts the html interface to on-line help
#. on your machine). You should briefly explore the features
#. Iconify the help window and move on to the next part

set.seed(17)
x <- rnorm(10)
x

y <- rnorm(10)
y
# Generate two pseudo-random normal vectors of x and y c

x11(w=4,
     h=4,
     xpos=0,
     ypos=0)

#bringToTop(s=T)
plot(x,
      y)
# Plot the points in the plane. A graphics window will appear
is()
# See which R objects are now in the R workspace

rm(x, y)
# Remove objects no longer needed. (Clean up)

x <- 1:20
# Make x=(1, 2, . . . , 20)

v <- 1 + sqrt(x)/2
# A weight vector of standard deviations

dummy <- data.frame(x=x,
                     y=x + rnorm(x) * v)
# Dummy Make a data frame of two columns, x and y, and 1

fm <- lm(y ~ x,
          data=dummy)

```

FIGURE 3.53: Spelling (Tools).

Tool	Description
Spell	Interface to speller

To enable spellchecking with Tinn-R it is necessary to install at least one dictionary of the list of [available](#) ones. It is also a good idea to install the dictionary manager. See [instructions](#)

Database (Tools)

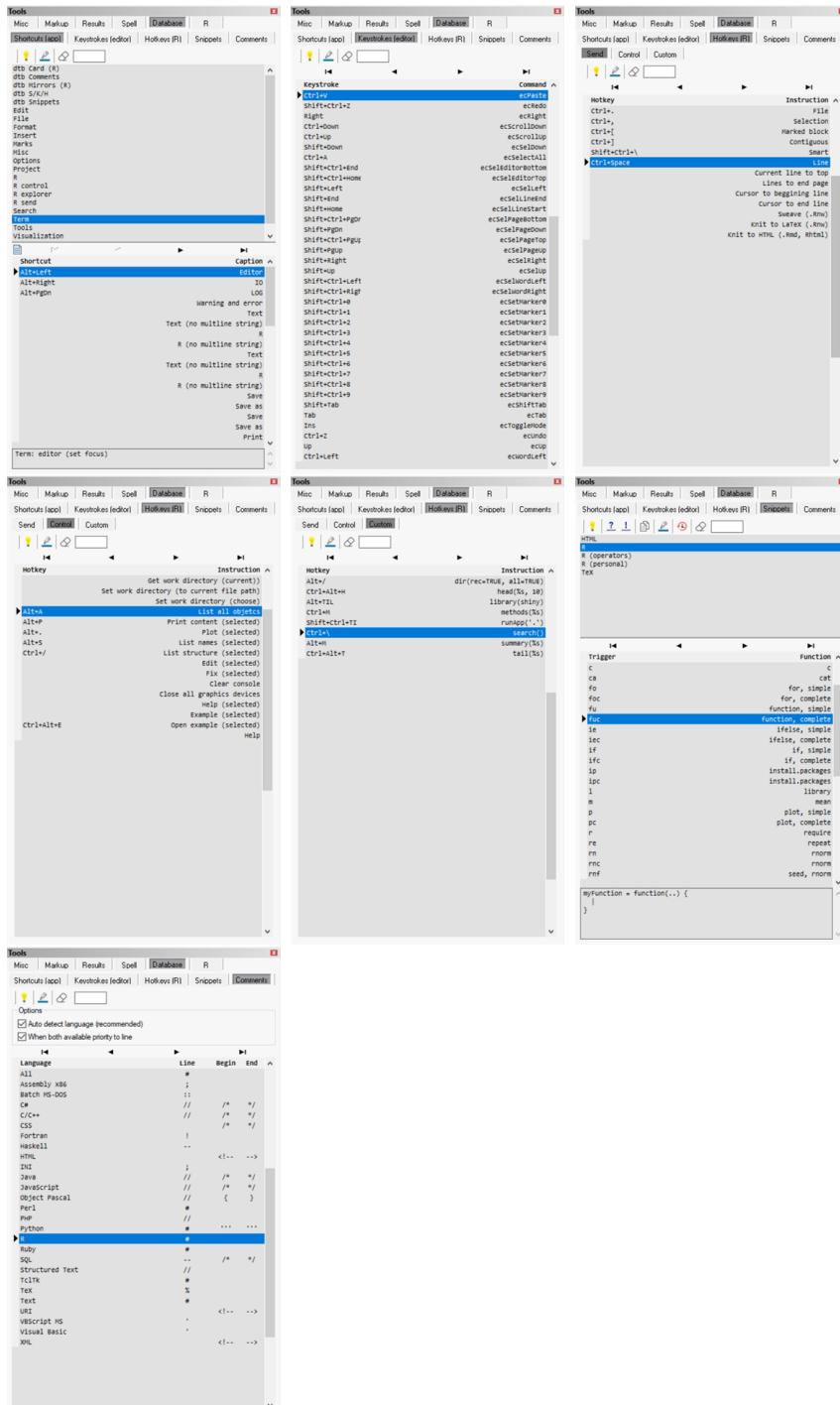


FIGURE 3.54: Database (Tools).

Tool	Description
Shortcuts	A digital shortcuts interface based in a XML database
Snippets	A digital snippets interface based in a XML database
Comments	A digital comments interface based in a XML database

Table 3.10: Database (Tools)

The database (Figure 3.54 and Table 3.10) uses the native XML engine provided by Embarcadero. Each tab (*Shortcuts*, *Snippets* and *Comments*) has its own pop-up menus and toolbars.

Shortcuts:

The *Shortcuts* interface allows the user to find out about the internal organization of Tinn-R and also to customize all shortcuts related to the application. It is our intention, in the near future, to add additional keystrokes related to the editor and to the R hotkeys.

The available buttons (Figure 3.54) are:

Help: Opens the User guide.

Edit: Opens the dialog *Snippets dataset (xml based)*.

Clear: Clears the filter edit box.

The *Edit* button opens the dialog shown in the Figure 3.36.

Keystrokes (editor):

This window allows the user to have an overview of the keystrokes related to the SynEdit class (Editor, IO and LOG). The filter edit box allows you to quickly find shortcuts.

You need to be aware that some shortcuts are superimposed by Shortcuts (app). In these cases, the shortcuts displayed in this window can be designed and used, for the most part, as secondary shortcuts for methods of the SynEdit class.

The available buttons (Figure 3.54) are:

Help: Opens the User guide.

Edit: Opens the dialog *Keystrokes (editor) dataset (xml based)*.

Clear: Clears the filter edit box.

The *Edit* button opens the dialog shown in the Figure 3.21.

Hotkeys:

The main tab called Hotkeys has three daughter tabs: Send, Control and Custom.

The purpose of these windows is to enable quick consultation of each one as well as allowing you to enter the editing modes for each one. The filter edit box makes it easy to find hotkeys.

Hotkeys are very interesting resources to use when communicating with R, since the triggered action is independent of whether or not the focus is in the Tinn-R interface and whether it is also independent of the location.

Read more about the subject at:

- [FAQ: Hotkeys](#)
- [Hotkeys: editor](#)
- [Some Secrets for an Efficient Use](#)

The available buttons (Figure 3.54) are:

Help: Opens the User guide.

Edit: Opens the dialog *Hotkeys dataset (xml based)*.

Clear: Clears the filter edit box.

The *Edit* button opens the dialog shown in the Figure 3.20.

Snippets:

The *Snippet* resource is very simple and allows high level of user customization related to edition.

This resource adds a granular level of user customization for editing all within Tinn-R.

The snippets (database based) allows the user to add functions based on several programming languages such as R, TeX, among others.

The available buttons (Figure 3.54) are:

Help: Sends the following instruction to R: `help('selected function')`.

Example: Sends the following instruction to R: `example('selected function')`.

Copy function: Places the selected function in the clipboard.

Copy description: Places the description of the selected function in the clipboard.

Edit: Opens the dialog *Snippets dataset (xml based)* below.

Insert: Inserts the selected function in the active editor. A Double click or Enter performs the same function. The default shortcut is CTRL + J, but this can be customized under *Options/Shortcuts* or *Tools/Database/Shortcuts*. To use it just push the keystrokes after any valid word:

```

if<CTRL + J> to obtain:
if (| < )

ifc<CTRL + J> to obtain:
if (| < ) {

}

fo<CTRL + J> to obtain:
for (i in 1:i|)

foc<CTRL + J> to obtain:
for (i in 1:|) {

}

sw<CTRL + J> to obtain:
switch(|,
a = ' ',
b = ' ',
)

wh<CTRL + J> to obtain:
i = 0
while (i < |) {

    i = i + 1
}

eq<CTRL + J> to obtain:
\begin{equation}\label{eq_01}
|
\end{equation}

```

Observations:

1. Only two letters were used to define the functions (for example: *fo* = *for*, *fu* = *function*);
2. Therefore, we added the letter c for more complex structures (for example: *foc*, *fuc*);
3. The | symbol is used to define where the cursor will first stop after snippets. After being selected the | symbol marks the point where the user can start typing.

Clear: Clears the filter edit box.

The *Edit* button opens the dialog showed in the Figure 3.41.

Comments:

The *Comments* resource is very simple and allows high level of user customization.

From version 3.0.1.0 Tinn-R automatically recognizes the language of the file on focus. Further, inside the file - if it is a syntax a multi-highlighter (complex syntax) - which language of the line where the cursor (or selection) is found.

This identification is done automatically if (and only if) the option (*x*) *Auto detect language (recomended)* is checked. Otherwise the user is forcing the application to use the comments of the selected language (indicator arrow).

Selected code snippets involving more than one language will not be commented/uncommented and a warning message is issued. That is, you must select only the snippet of a single language.

The available buttons (Figure 3.54) are:

Help: It opens the *User Guide* on the section Database.

Edit: It opens the dialog *R card dataset (xml based)*.

Clear: Clears the filter edit box.

The *Edit* button opens the dialog shown in the Figure 3.42.

R (Tools)

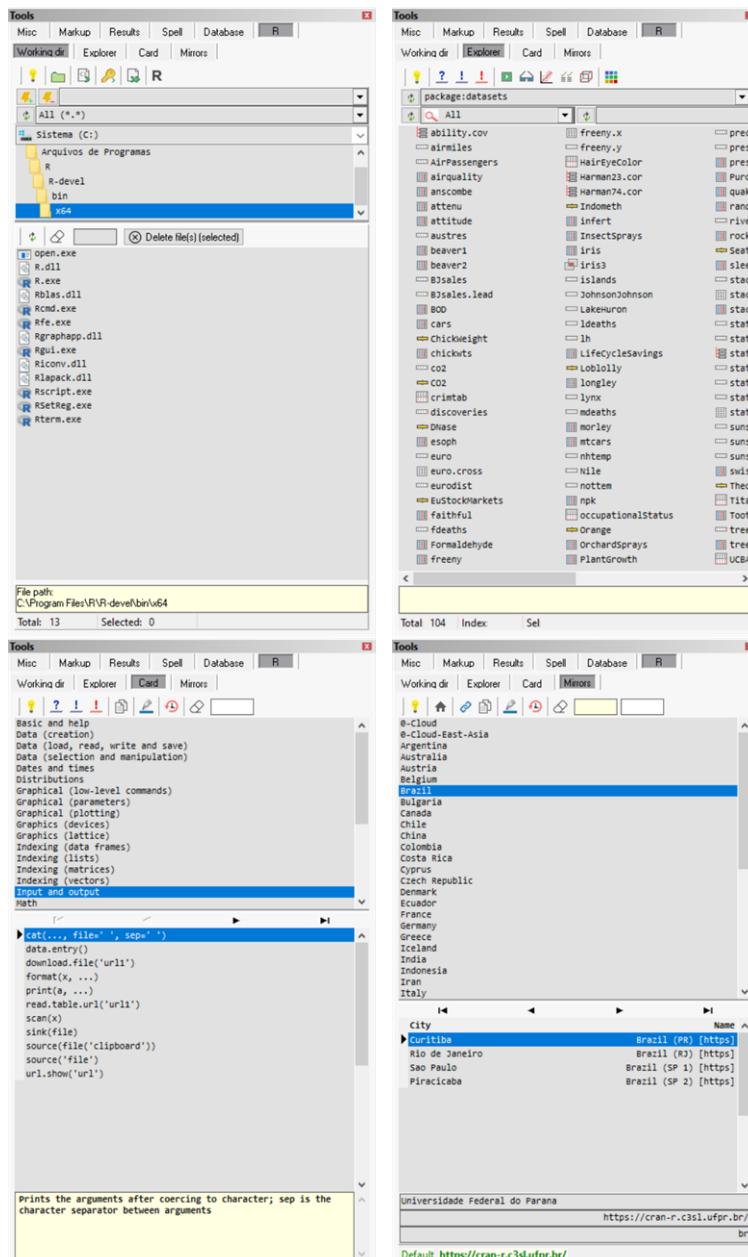


FIGURE 3.55: R (Tools).

Tool	Description
Working dir	A graphical interface to make it easy to deal with the R interpreter's Working Directory
Explorer	Simple and functional graphical interface of objects of the R environment
Card	A digital and simple R card based in a XML database
Mirrors	A digital and simple R mirrors management based in a XML database

Table 3.11: R (Tools).

A simple and functional graphical interface (Figure 3.55 and Table 3.11) of objects, card and mirrors management of the R environment.

Working dir:

This interface (Figure 3.55) allows a graphical visualization of the active R working directory. It also allows storing favorites and indicating the work folder to R with several interesting options.

This interface is experimental, but we believe it is of great importance for novice users.

Explorer:

This interface (Figure 3.55) has its own pop-up menu, toolbar and three combo box. The pop-up menu and toolbar contain the most common actions related to an object explorer.

The button *R explorer: refresh environment* sends an instruction to R environment requesting the list of all loaded packages in the current session. The result is shown inside a graphical classified list. When one of these is selected, the graphical list (and structure) of the objects are shown.

There are two options of filter: type of objects and any sequence of characters associated with the names of the objects.

It is possible to remove visible objects of the user workspace (.GlobalEnv) using the key *Delete*. To do this, select an object and type *Delete*.

A double click in any selected object will add its name to the editor. If the object is dragged to the editor interface, the textual description of the object is always shown in a new file. It is useful to know the sources of functions and to see data objects (vectors, frames, list, etc).

If view mode is grid, you can see more details about each object. In grid mode, in the *Dim* column, the description **ns** and **nd** mean nonsense and undefined, respectively.

Card:

The *card* was based on two R cards already published: R/Rpad Reference Card by Tom Short and R reference card by Jonathan Baron.

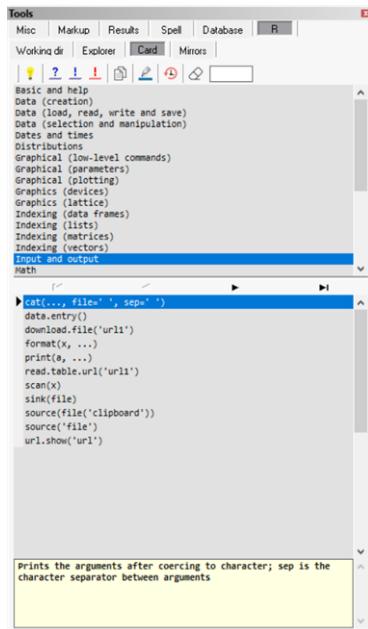


FIGURE 3.56: R card (dataset).

The available buttons (Figure 3.56) are:

Help: Sends the following instruction to R: `help('selected function')`.

Example: Sends the following instruction to R: `example('selected function')`.

Copy function: Places the selected function in the clipboard.

Copy description: Places the description of the selected function on the clipboard.

Edit: Opens the dialog *R card dataset (xml based)* below.

Insert: Inserts the selected function in the active editor. A Double click or Enter performs the same function.

Clear: Clears the filter edit box.

The *Edit* button opens the dialog shown in the Figure 3.43.

Mirrors:

The *Mirrors* is an interface that allows the user to manage the repositories (or mirrors) of R. You should always choose a repository physically closest

to where you are, so that, the Web communication tends to be faster and more efficient.

The default mirror is the University [Wien](#) (Austria). Consider that this is the central mirror of CRAN.

The reasons for the Tinn-R always set a repository are two:

- Prevent R keep asking which repository you want to use in each session;
- Workaround of intermittency (only Rterm.exe) display the dialog for selecting the repository. That is, sometimes the dialog is displayed and not others. The cause of this intermittency is still unknown.

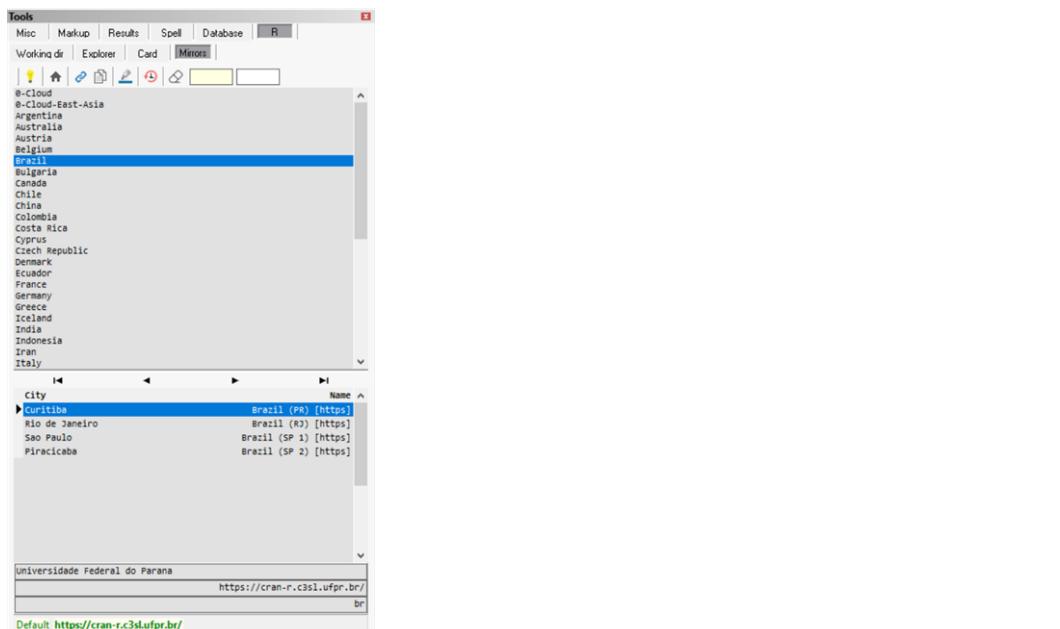


FIGURE 3.57: R mirrors (dataset).

The available buttons (Figure 3.57) are:

Help: Opens the User guide.

Update: Updates the available R mirrors.

Copy host: Places the selected mirror host in the clipboard.

Copy URL: Places the selected mirror URL in the clipboard.

Edit: Opens the dialog *R mirrors (xml based)* below.

Insert: Sends the following instruction to R: `options('repos='URL of selected mirror')`.

Clear: Clears the filter edit box.

The *Edit* button opens the dialog shown in the Figure 3.44.

3.6 PROJECTS

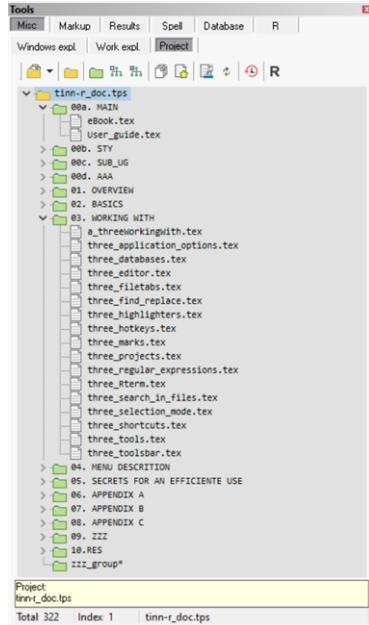


FIGURE 3.58: Tinn-R: Projects.

Overview

A Tinn-R project (Figure 3.58) is a container for different types of editable files associated with a single task, for example program files, data files, and text files. Files in similar categories can be grouped as Tinn-R groups within the project much like folders and subfolders. This saves you looking for and opening each of the files individually every time you start working. Simply double click on any file in the project or group (Figure 3.59) visible in the Tools/Misc/Project pane and it will open under a new(s) tab(s) in Tinn-R.

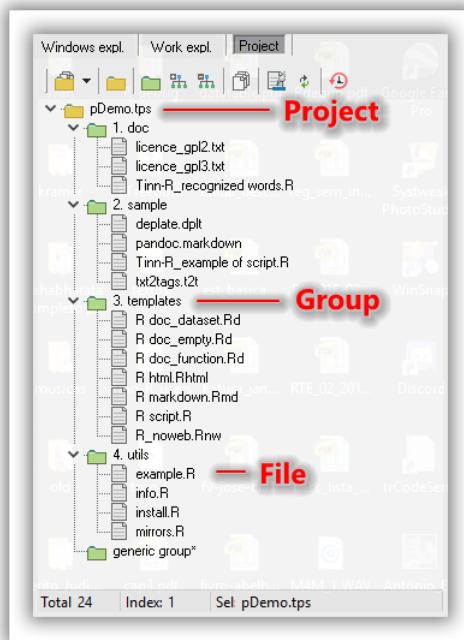


FIGURE 3.59: Tinn-R: Projects, groups and files.

You can also drag the object (entire project, group or individual file) to the editor. Doing so will open the corresponding files for viewing in the editor.

Uneditable files (not ASCII/ANSI/UTF-X, such as PDF, PNG, etc) can be included in a project if you will benefit from the listing, but they are not correctly viewed in the editor.

Opening the demo project

First, to see a project, you must open the Tools pane: View/Tools/Tools (CTRL+F8 by default) and choose the page Misc/Project.

You should find a demonstration project (pDemo.tps) at the second (left to right) yellow-brown project icon. The option Open demo (Figure 3.60) will open a didactic demo project. You can play around with it to get the general idea.

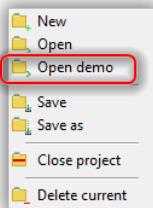


FIGURE 3.60: Tinn-R: Project demo.

Once you have opened the first project, you can use the first yellow-brown file-card icon whenever you wish to open a project from the displayed list.

Creating your projects

To start your own project, click the smaller yellow-brown file-card icon, then New.

The new project will be stored as a ".tps" file. Add groups to the project with the green file-card icon. Add files to a group, or directly to the project itself, with the multiple-sheets icon, then click on Add. Files can be selected and dragged and dropped to change the groupings. Groups can be renamed by highlighting the group name, then right clicking. Groups can be expanded and contracted all at once with two other icons. You can close the Tools pane if it is getting in the way with |X| at the top right. Later, re-open it and your project will still be present.

Working with the project in graphical mode

For small projects, changing the project structure is preferable, for example, to add or exclude files, change groupings, rename objects directly in graphical mode using the taskbar buttons of the project and popup menus available. They are self explanatory.

Working with the project in text mode

To create or edit projects with many groups, and/or files, editing in text mode is usually more productive than in graphical mode (Figure 3.61).

The screenshot shows the Tinn-R application window. On the left, there's a 'Tools' menu with options like Misc, Markup, Results, Spell, and Data. Below that is a toolbar with buttons for Windows expl., Work expl., and Project. The main area has a tree view under 'Project' showing a folder 'pDemo.tps' containing files 1.doc, 2.sample, 3.templates, 4.utils, and generic group. To the right is a large text editor window displaying a series of numbered lines representing R code or project configuration. The lines include file paths such as C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\licence_gpl2.txt, C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\licence_gpl3.txt, and C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\Tinn-R_recognized words. The text editor has a yellow header bar with the number 1. A status bar at the bottom shows 'Total 24', 'Index 1', and 'Sel pDemo.tps'. A watermark for 'KIKO FREITAS' is visible in the background.

```

1 \1. doc
2 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\licence_gpl2.txt
3 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\licence_gpl3.txt
4 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\Tinn-R_recognized words
5 \2. sample
6 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\deplate.dplt
7 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\pandoc.markdown
8 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\Tinn-R_example of script
9 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\txt2tags.t2t
10 \3. templates
11 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\R_doc_dataset.Rd
12 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\R_doc_empty.Rd
13 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\R_doc_function.Rd
14 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\R_html.Rhtml
15 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\R_markdown.Rmd
16 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\R_script.R
17 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\R_noweb.Rnw
18 \4. utils
19 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\example.R
20 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\info.R
21 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\install.R
22 C:\Users\jcfaria\AppData\Roaming\Tinn-R\project\mirrors.R
23
24

```

FIGURE 3.61: Tinn-R: Projects in text mode.

For this, with a project (eg pDemo.tps) open in the GUI, the third button from the right on the project: edit (as text file) project taskbar. This opens the text file in the editor. It can then be viewed and edited within the rules for a project file. After saving it (Ctrl+S), the second button from left to right will reconstruct the graphical interface of the project from its textual description.

Submit entire project (or parts) to R interpreter

If you organize your R scripts into projects with a proper group structure, a project popup menu option allows the individual file, group, and entire project to be sent to the interpreter. This can be very productive in R package development and more complex data analyses.

Sets the main L^AT_EX project file

Once you choose a project file as the main one, you can be editing in any project file that, when compiling (generate the pdf or dvi), the instruction will be sent correctly for the compiler. This new feature speeds up the process of creating files (PDF and DVI) as the user does not need to navigate to the main file each time to submit the project to the compilation. For projects with many files, this feature (already present in more specialized L^AT_EX editors) makes work much easier. The feature is functional for all

options associated with compilation (.pdf, .dvi, makeindex and bibitex). This resource is volatile with each session and project, that is, it is not stored with the project information.

When there are two main file indications for L^AT_EX (one in the project and the other in the main interface for any file), this option, from within the project, takes precedence, that is, it is mandatory.

Closing your projects

When finished with the project, or just to back it up as it is, click the smaller yellow-brown file-card icon, then Save.

3.7 TOOLS BAR



FIGURE 3.62: Tools bar.

Category	Description
File	New, open, save, save all, reload and print
Edit	Undo and redo
Filter	Create a new file with all occurrences of typed sequence of characters
Macro	Record and play
Misc	On top, focus control and block marks
Processing	Conversion, compilation and viewer
R	Lots of options to send and control R
Search	Current file, in files, replace and go to line
Syntax	Drop down list of all syntaxes available
Spell	Drop down list of installed dictionaries and a bottom to start the speller
View	Organize screen, Tools (show/hide), Tools (size), Term (show/hide, Term (size), options to <i>IO</i> and <i>LOG</i> and word wrap

Table 3.12: Tools bar.

Unlike most applications of this category, this interface (Figure 3.62) was designed to be as small and simple as possible. In other words, the full access to all resources of Tinn-R are available at the main menu and associated shortcuts (it takes time to learn all and most are user configurable).

Two groups are available: main and R tool bar.

The main toolbar interface is categorized (Table 3.12) and contains the most common tasks:

The R toolbar has two basic divisions: Send (left side, finishing in the *Set work directory button*) and Control (right side, starting in the *List all objects button*).

Show/Hide

FIGURE 3.63: Tools bar menu.

The Tools bar has its own pop-up (Figure 3.63) menu enabling the user to choose what resources will be visible (show/hide). To see the pop-up menu, press the right mouse button inside any place of the main tool bar.

Disposition

The interface also allows drag and drop. In other words, you can organize the order of the individual tool bar inside of the main container. It is better to do that with the main interface not maximized to avoid screen flicker (a small nuance related to some version of the Windows and Borland engine).

3.8 FILE TABS

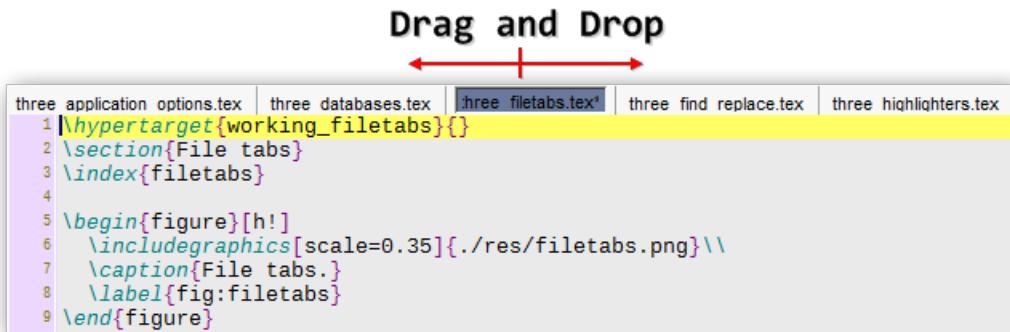


FIGURE 3.64: File tabs.



FIGURE 3.65: File tabs menu.

The position of the file tabs (Figure 3.64) can be changed by drag and drop. It allows to put the file tab in a desirable order making it suitable for the user interface.

The file tabs have their own pop-up menu (Figure 3.65) allowing fast control of the most common tasks.

3.9 SELECTION MODE

Allows the setting of the current selection mode (Figure 3.66, 3.67 and 3.68).

Select text by clicking and dragging with the left mouse button held down or moving the cursor with the shift key held down. The status bar will display an icon indicating the current selection mode.

Normal

```
35 fm <- lm(y ~ x, data = dummy)
36 summary(fm)
37 # Fit a simple linear regression of y on x and Look
38
39 fm1 <- lm(y ~ x, data = dummy, weight = 1/w^2)
40 summary(fm1)
41 # Since we know the standard deviations, we can do
42
43
```

FIGURE 3.66: Normal (selection mode).

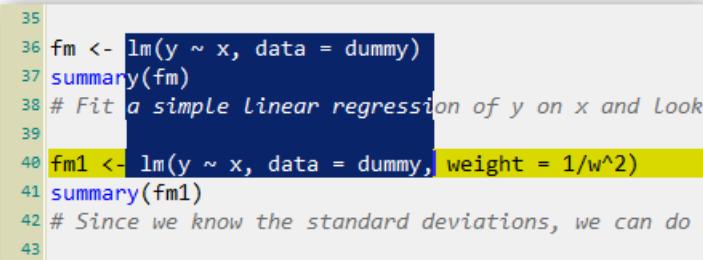
This is the standard selection mode (Figure 3.66) found in many Windows applications.

Line

```
35 fm <- lm(y ~ x, data = dummy)
36 summary(fm)
37 # Fit a simple linear regression of y on x and Look
38
39 fm1 <- lm(y ~ x, data = dummy, weight = 1/w^2)
40 summary(fm1)
41 # Since we know the standard deviations, we can do
42
43
```

FIGURE 3.67: Line (selection mode).

This selection mode (Figure 3.67) allows only for complete lines to be selected.

Column

A screenshot of an R code editor window. The code is as follows:

```
35 fm <- lm(y ~ x, data = dummy)
36 summary(fm)
37 # Fit a simple linear regression of y on x and look
38
39 fm1 <- lm(y ~ x, data = dummy, weight = 1/w^2)
40 summary(fm1)
41 # Since we know the standard deviations, we can do
```

The line `fm1 <- lm(y ~ x, data = dummy, weight = 1/w^2)` is highlighted with a yellow background, indicating it is selected.

FIGURE 3.68: Column (selection mode).

This selection mode (Figure 3.68) allows vertical blocks of text to be selected. The option ALT sets column mode allowing the selection mode to be switched to Column Mode when selecting with the mouse by simply holding down the ALT key. [See details at editor \(advanced options\)](#).

3.10 WORKING WITH MARKS

The screenshot shows the Tinn-R editor interface with the following R code:

```
1 #=====//=====
2 # $ Mail: <<< joseclaudio.faria@gmail.com >>>
3 # More: <<< http://zoonek2.free.fr/UNIX/48_R/all.html >>>
4 #=====//=====
5
6 #.. help.start() starts the html interface to on-line help (using
7 #.. your machine). You should briefly explore the features of this
8 #.. Iconify the help window and move on to the next part.
9
10 0 set.seed(17)
11 1 x <- rnorm(10)
12 2 x
13 3
14 4 y <- rnorm(10)
15 5 y
16 6 # Generate two pseudo-random normal vectors of x and y coordinates
17
18 7 x11(w=4,
19 8 h=4,
20 9 xpos=0,
21 10 ypos=0)
22
23 11 bringToTop(s=T)
24
25 12 plot(x,
26 13 y)
27 14 # Plot the points in the plane. A graphics window will appear auto-
28 15 #matically
29 16 ls()
30 17 # See which R objects are now in the R workspace
```

A red rectangular box highlights the first five lines of code (set.seed, x <- rnorm(10), x, y <- rnorm(10), y), which are associated with mark numbers 0 through 4. Line 16, which contains the command 'ls()', is highlighted with a yellow background, indicating it is the current line of focus.

FIGURE 3.69: Tinn-R: Marks.

Overview

When working with long text or program files, moving from one part of the file to another can be done with the scroll bars, the GoTo line number function (CTRL+G), or under Search, or by searching for specific text strings (CTRL+F). However, these methods become laborious for frequent moves. Placing marks in the text at points you will want to return is more efficient. Marks are visible as small, circled numbers in the left-side column of the editor, at the start of the line that is marked (Figure 3.69).

Use marks as follows:

1. Insert a new mark at the cursor position: **CTRL + SHIFT + 1 + ...+ 9.** This allows up to 9 separate marks to be inserted
2. Go to a mark from somewhere else in the text: **CTRL + 1 + ...+ 9.** Translocation is immediate
3. Move a mark from one location to another: eg. for Mark 5. Position the cursor at the new position, then **CTRL+SHIFT+5**, as in (1). The original position of Mark 5 is lost, and the new position stored.
4. Delete a mark: position the cursor at the existing position using, eg. **CTRL+5**. Then re-mark 5, as in (1), with **CTRL+SHIFT+5** The number 5 beginning of the line disappears to show that mark 5 has been lost and is free for use later.

Basic options are also available in the **Misc** taskbar and in the main menu **Marks**.

3.11 REGULAR EXPRESSIONS

This session is an adaptation. It is based on the help of the freeware **PSPad** editor.

What are regular expressions?

Regular expressions are widely-used method to specify text patterns to be searched for. Special metacharacters allow you to specify, for instance, that a particular string you are looking for occurs at the beginning or end of a line, or contains n recurrences of a certain character.

Regular expressions may look ugly to novices, but are actually a very simple, handy and powerful tool.

Simple Matches

Any single character matches itself, unless it is a metacharacter with a special meaning described below.

A series of characters matches that series of characters in the target string, so the pattern bluh would match bluh in the target string.

You can cause characters that normally function as metacharacters or escape sequences to be interpreted literally by escaping them. Do this by preceding them with a backslash \. For instance: metacharacter ^ match beginning of string, but \^ match character ^, \\ match \ and so on.

Examples:

ER	Matches
foobar	foobar
\^FooBarPtr	^FooBarPtr

Escape Sequences

Characters may be specified using a escape sequences syntax much like that used in C and Perl: \n matches a newline, \t a tab, etc. More generally, \xnn, where nn is a string of hexadecimal digits, matches the character whose ASCII value is nn. If you need wide (Unicode) character code, you can use \x{nnnn}, where nnnn is one or more hexadecimal (base 16) digits (1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). Hex digit letters may be in upper or lower case.

ER	Description
\xnn	Char with hex code nn
\x{nnnn}	Char with hex code nnnn (one byte for plain text and two bytes for Unicode)
\t	Tab (HT/TAB), same as \x09
\n	Newline (NL), same as \x0a
\r	Carriage return (CR), same as \x0d
\f	Form feed (FF), same as \x0c
\a	Alarm (bell) (BEL), same as \x07
\e	Escape (ESC), same as \x1b

Examples:

ER	Matches
foo\x20bar	foo bar (note space in the middle)
\tfoobar	foobar predefined by tab

Character Classes

You can specify a character class by enclosing a list of characters in [], which will match any one character from the list. If the first character after the [is ^, the class matches any character not in the list.

Examples:

ER	Matches
foob[aeiou]r	foobar, foober, etc. But not foobbr, foobcr, etc
foob[^aeiou]r	foobbr, foobcr, etc. But not foobar, foober, etc

Within a list, the - character is used to specify a range, so that a-z represents all characters between a and z, inclusive.

If you want - itself to be a member of a class, put it at the start or end of the list, or escape it with a backslash. If you want] you may place it at the start of list or escape it with a backslash.

Examples:

ER	Matches
[-az]	a, z and -
[az -]	a, z and -
[a \ -z]	a, z and -
[a -z]	All twenty six small characters from a to z
[\n -\x0D]	Any of ASCII #10(Lf), #11, #12(Ff), #13(Cr)
[\d -t]	Any digit, - or t
[] -a]	Any char from].a

Metacharacters

Metacharacters are special characters which are the essence of regular expressions. There are different types of metacharacters, described below.

Metacharacters - Line Separators:

ER	Description
^	Start of line
\$	End of line
\A	Start of text
\Z	End of text
.	Any character in line

Examples:

ER	Matches
^foobar	foobar only if it's at the beginning of line
foobar\$	foobar only if it's at the end of line
^foobar\$	foobar only if it's the only string in line
foob.r	foobar, foobbr, fooblr and so on

The ^ metacharacter by default is only guaranteed to match at the beginning of the input string/text, the \$ metacharacter only at the end. Embedded line separators will not be matched by ^ or \$. You may, however, wish to treat a string as a multi-line buffer, such that the ^ will match after any line separator within the string, and \$ will match before any line separator. You can do this by switching the modifier /m on.

The \A and \Z are just like ^ and \$, except that they won't match multiple times when the modifier /m is used, while ^ and \$ will match at every internal line separator.

Metacharacters - Predefined Classes:

ER	Description
\w	An alphanumeric character (including _)
\W	A non alphanumeric character
\d	A numeric character
\D	A non-numeric character
\s	Any space (same as [\t\n\r\f])
\S	A non space

You may use \w, \d and \s within custom character classes.

Examples:

ER	Matches
foob\dr	foob1r, foob6r and so on but not foobar, foobbr and so on
foob[\w\s]r	foobar, foob_r, foobbr and so on but not foob1r, foob=r and so on

TRegExpr uses properties SpaceChars and WordChars to define character classes \w, \W, \s, \S, so you can easily redefine it.

Metacharacters - Word Boundaries:

ER	Matches
\b	A word boundary
\B	A non-(word boundary)

A word boundary \b is a spot between two characters that has a \w on one side of it and a \W on the other side of it (in either order), counting the imaginary characters of the beginning and end of the string as matching a \W.

Metacharacters - Iterators:

Any item of a regular expression may be followed by another type of metacharacters - iterators. Using this metacharacters you can specify number of occurrences of the previous character, metacharacter or subexpression.

ER	Matches
*	Zero or more ("greedy"), similar to {0,}
+	One or more ("greedy"), similar to {1,}
?	Zero or one ("greedy"), similar to {0,1}
{n}	Exactly n times ("greedy")
{n,}	At least n times ("greedy")
{n,m}	At least n but not more than m times ("greedy")
*?	At least n but not more than m times ("greedy")
+?	At least n but not more than m times ("greedy")
??	Zero or one ("non-greedy"), similar to {0,1}?
{n}?	Exactly n times ("non-greedy")
{n,}?	At least n times ("non-greedy")
{n,m}?	At least n but not more than m times ("non-greedy")

So, digits in curly brackets of the form {n,m}, specify the minimum number of times to match the item n and the maximum m. The form {n} is equivalent to {n,n} and matches exactly n times. The form {n,} matches n or more times. There is no limit to the size of n or m, but large numbers will chew up more memory and slow down r.e. execution.

If a curly bracket occurs in any other context, it is treated as a regular character.

Examples:

ER	Matches
foob.*r	foobar, foobalkjdflkj9r and foobr
foob.+r	foobar, foobalkjdflkj9r but not foobr
foob.?r	foobar, foobbr and foobr but not foobalkj9r
fooba{2}r	foobaar
fooba{2,}r	foobaar, foobaaar, foobaaaar, ...
fooba{2,3}r	foobaar, or foobaar but not foobaaaar

A little explanation about greediness. Greedy takes as many as possible, non-greedy takes as few as possible. For example, b+ and b* applied to string abbbbc return bbbb, b+? returns b, b*? returns empty string, b{2,3}? returns bb, b{2,3} returns bbb.

Metacharacters - Alternatives:

You can specify a series of alternatives for a pattern using | to separate them, so that fee|fie|foe will match any of fee, fie, or foe in the target string (as would f(e|i)o)e). The first alternative includes everything from the last pattern delimiter ((), [, or the beginning of the pattern) up to the first |, and the last alternative contains everything from the last | to the next pattern delimiter. For this reason, it's common practice to include alternatives in parentheses, to minimize confusion about where they start

and end.

Alternatives are tried from left to right, so the first alternative found for which the entire expression matches, is the one that is chosen. This means that alternatives are not necessarily greedy. For example: when matching `foo|foot` against `barefoot`, only the `foo` part will match, as that is the first alternative tried, and it successfully matches the target string. (This might not seem important, but it is important when you are capturing matched text using parentheses).

Also remember that `|` is interpreted as a literal within square brackets, so if you write `[fee|fie|foe]`. You're really only matching `[feio|]`.

Examples:

ER	Matches
<code>foo(bar foo)</code>	<code>foobar</code> or <code>foofoo</code>

Metacharacters - Subexpressions:

The bracketing construct `(...)` may also be used to define r.e. subexpressions (after parsing, you can find subexpression positions, lengths and actual values in MatchPos, MatchLen and Match properties of TRegExpr, and substitute it in clip strings by TRegExpr.Substitute).

Subexpressions are numbered based on the left to right order of their opening parenthesis. First subexpression has number 1 (whole r.e. match has number 0 - you can substitute it in TRegExpr. Substitute as `$0` or `$&`).

Examples:

ER	Matches
<code>(foobar){8,10}</code>	Strings which contain 8, 9 or 10 instances of the <code>foobar</code>
<code>foob([0-9] a+)r</code>	<code>foob0r</code> , <code>foob1r</code> , <code>foobar</code> , <code>foobaar</code> , <code>foobaar</code> , ...
<code>(abc(def)ghi(123))xyz</code>	<code>abcdefghijklm</code> (the only match)

Then backreferences:

ER	Description
<code>\1</code>	= <code>abcdefghijklm</code>
<code>\2</code>	= <code>def</code>
<code>\3</code>	= <code>123</code>
<code>\0</code>	= <code>abcdefghijklm</code> (the whole match)

We could find the same string using metacharacter Iterators with: `(\w{3})([d-f]{3})...(\d*)xyz` and the backreferences would have the same values. And yet the pattern would also match: `123fdd@#$4444xyz, bbbeee-abc1234567xyz, ddddddःaaaxyz.`

Metacharacters - Backreferences:

Metacharacters `\1` through `\9` are interpreted as backreferences in the *Search text* box. When used outside of the regular expression, such as in the *Replace text* field, metacharacters `$1` through `$9` are interpreted as backreferences to the last find.

`\<n>` matches previously matched subexpression `#<n>`.

Find Examples:

ER	Matches
<code>(.)\1+</code>	aaaa and cc
<code>(.+)\1+</code>	abab and 123123
<code>([""]?) (\d+) \1</code>	"13"(in double quotes), or '4' (in single quotes) or 77 (without quotes), ...

Replace Examples: Date format change from `dd.mm.yyyy` to `yyyy-mm-dd`.

This will turn the European date style `26.8.1994` or `26/8/1994` into `1994-8-26`. Search: `([0-9]{1,2}).([0-9]{1,2}).([0-9]{4})` Replace: `$3-$2-$1` Make sure you check the box for *Regular Expression*.

Regular Expression Tutorials

The following is a list of a few sites that contain tutorials on both regular expressions in general and their use in specific languages:

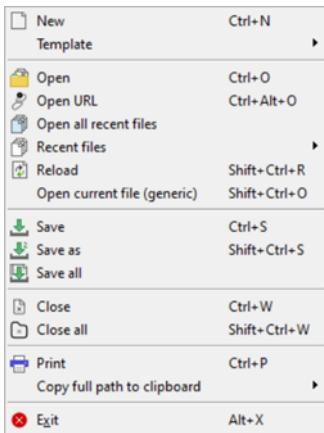
- [General Tutorial](#)
- [Regular Expressions - User guide](#)
- [Regular Expression HOWTO](#)
- [Perl Regular Expressions](#)
- [Regular Expression Basic Syntax Reference](#)

CHAPTER 4

MENU DESCRIPTION

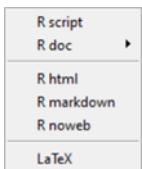
This chapter provides information about the main menu for Tinn-R.

4.1 FILE



Option	Description
New	Creates a new file
Template	<i>See options ...</i>
Open	Opens selected file as text
Open URL	Opens an URL file as text
Open all recent files	Opens all files from the Most Recently Used (MRU) file list
Recent files	Displays a Most Recently Used (MRU) file list. Selecting one of the displayed files will open that file
Reload	Reloads the current files to the last saved status
Open current file (generic)	Reopens a file opened in the Tinn-R editor in the operating system-defined application
Save	Saves the current file. If the file has not been previously saved then the 'File Save As' dialog will open first
Save as	Saves the current file with a new name
Save all	Saves all changed files. If a file has not been previously saved the 'File Save As' dialog will open first
Close	Closes the current file. If the file has not been saved you will be prompted to save it
Close all	Closes all files including projects
Print	Will open a Tinn-R dialog allowing settings and actions associated with the current file
Copy full path to clipboard	<i>See options ...</i>
Exit	Exits the application

Template



Option	Description
R script	Creates a R script template
R doc	<i>See options...</i>
R html	Creates a R html template
R markdown	Creates a R markdown template
R noweb	Creates a R noweb template
El ^p X	Creates a El ^p X template

R doc:

Function
Dataset
Empty

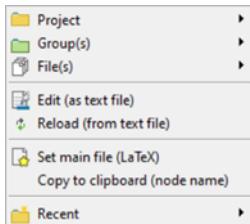
Option	Description
Function	Creates a R doc function template
Dataset	Creates a R doc dataset template
Empty	Creates a R doc empty template

Copy full path to clipboard

Unix mode/..
Windows mode ..\..\..

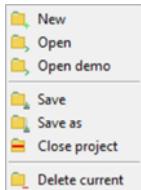
Option	Description
Unix mode/..	Copy full path of current file to clipboard in Unix mode/..
Windows mode ..\..\..	Copy full path of current file to clipboard Windows mode ..\..\..

4.2 PROJECT

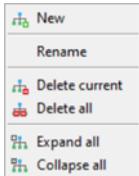


Option	Description
Project	<i>See options ...</i>
Group(s)	<i>See options ...</i>
File(s)	<i>See options ...</i>
Edit (as text file)	Opens the textual description of the project for editing
Reload (from text file)	Reloads the graphical project interface from the textual description of the project
Set main file (El ^E X)	Sets (in the current project) the main El ^E X project file
Copy to clipboard (node name)	Copy project node name to Windows clipboard
Recent	The option will display a Most Recently Used (MRU) project file list. Selecting one of the displayed files will open that file

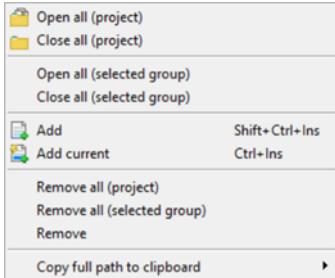
Project



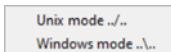
Option	Description
New	Creates new project. If you have an open unsaved project you will be prompted to save the project file
Open	Opens existing project and restores the project's state
Open demo	Opens existing demo project and restores the project's state
Save	Saves the project file
Save as	Saves the current file with a new name
Close project	This option will close any files that are in a virtual folder
Delete current	This option will delete the virtual folder of the current project

Group(s)

Option	Description
New	Creates a new group of current project
Rename	Renames a selected group of current project
Delete current	Removes the selected group from the current project
Delete all	Removes all groups from the current project
Expand all	Expands all groups of the current project in the graphical interface
Collapse all	Collapses all groups of the current project in the graphical interface

File(s)

Option	Description
Open all (project)	Opens all files of a project
Close all (project)	Closes all files of a project
Open all (selected group)	Opens all files of a selected group
Close all (selected group)	Closes all files of a selected group
Add	Opens the windows interface to select file(s) and add the selected(s) files to a selected group
Add current	Add the current file to the selected group
Shift+Ctrl+Ins	
Ctrl+Ins	
Remove all (project)	Removes all files from the project
Remove all (selected group)	Removes all files from the selected group
Remove	Removes selected file
Copy full path to clipboard	<i>See options ...</i>

Copy full path to clipboard

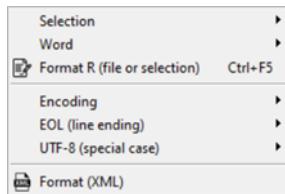
Option	Description
Unix mode ..\..	Copy full path of current file to clipboard in Unix mode ..\..
Windows mode ..\..	Copy full path of current file to clipboard Windows mode ..\..

4.3 EDIT

 Undo	Ctrl+Z
 Redo	Shift+Ctrl+Z
 Copy	Ctrl+C
 Copy selection to clipboard (RTF formatted)	
 Cut	Ctrl+X
 Paste	Ctrl+V
Select all	Ctrl+A
 Comment	Alt+C
 Uncomment first(s)	Alt+N
 Uncomment all	Alt+Z
Delete (bracket)	Ctrl+Del

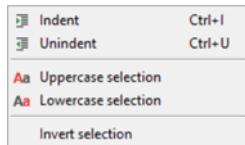
Option	Description
Undo	Undoes the last action
Redo	Re-applies any actions undone using the Undo option
Copy	Copies the selected text and places it in the Windows clipboard
Copy selection to clipboard (RTF formatted)	Exports a selection to the Windows clipboard in RTF format. Useful for documenting R code in RTF editors.
Cut	Cuts the selected text and places it in the Windows clipboard
Paste	Places any text in the Windows clipboard at position indicated by the cursor within the file
Select all	Selects the whole text contained in the file
Comment	Adds comments to selected line(s)
Uncomment first occurrence	Removes the first occurrence from a comment in the selected line(s)
Uncomment all occurrence	Removes all occurrences from a comment in the selected line(s)
Delete (bracket)	Delete the pair: current and matching bracket

4.4 FORMAT



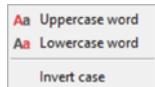
Option	Description
Selection	<i>See options ...</i>
Word	<i>See options ...</i>
Format R (file or selection)	Reformat a whole file or selection by using formatR package
Encoding	<i>See options ...</i>
EOL (line ending)	<i>See options ...</i>
UTF-8 (special case)	<i>See options ...</i>
Format (XML)	Formats XML files to be easily read by humans

Selection



Option	Description
Indent	Indents selected line(s)
Unindent	Unindents selected line(s)
Uppercase selection	Converts selected text into upper case
Lowercase selection	Converts selected text into lower case
Invert selection	Inverts the case of all selected text

Word



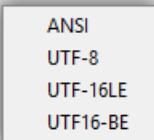
Option	Description
Uppercase word	Converts the word under the cursor to upper case
Lowercase word	Converts the word under the cursor to lower case
Invert case	Inverts the case of the word under the cursor

Encoding



Option	Description
Default (new files)	<i>See options ...</i>
Convert (current to)	<i>See options ...</i>

Default (new files):



Option	Description
ANSI	Sets encoding to ANSI
UTF-8	Sets encoding to UTF-8
UTF16-LE	Sets encoding to UTF16-LE
UTF16-BE	Sets encoding to UTF16-BE

Convert (current to):



Option	Description
ANSI	Converts current encoding to ANSI
UTF-8	Converts current encoding to UTF-8
UTF16-LE	Converts current encoding to UTF16-LE
UTF16-BE	Converts current encoding to UTF16-BE

EOL (line ending)



Option	Description
Default (new files)	<i>See options ...</i>
Convert (current to)	<i>See options ...</i>

Default (new files):

WIN (CR + LF)
MAC (CR)
UNIX (LF)

Option	Description
WIN (CR+LF)	WIN (CR+LF)
MAC (CR)	MAC (CR)
UNIX (LF)	UNIX (LF)

Convert (current to):

WIN (CR + LF)
MAC (CR)
UNIX (LF)

Option	Description
WIN (CR+LF)	Convert to WIN (CR+LF)
MAC (CR)	Convert to MAC (CR)
UNIX (LF)	Convert to UNIX (LF)

UTF-8 (special case)

Apply to open ANSI files
Always add the BOM

Option	Description
Apply to open ANSI files	ANSI files without any special characters it will be always recognized as UTF-8 encoding
Always add the BOM	When saving it will be always added the BOM

4.5 MARKS



Option	Description
Block	<i>See options ...</i>
Unmark all	Unmarks all marks of the current file

Block



Option	Description
Mark	Marks selected block: 0 to begin and 1 to end
Unmark	Unmarks any previous marked block. It is not necessary to select the marked block

4.6 INSERT



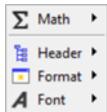
Option	Description
R (assignment)	<i>See options ...</i>
ET _E X	<i>See options ...</i>
Date / Time stamp	Inserts the current system time and date
Snippet insert (trigger context)	Replaces the trigger with the corresponding block of text (snippet)

R (assignment)



Option	Description
<-	Insert left assignment
->	Insert right assignment

Latex



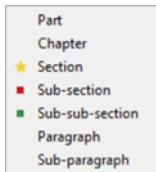
Option	Description
Math	<i>See options ...</i>
Header	<i>See options ...</i>
Format	<i>See options ...</i>
Font	<i>See options ...</i>

Math:



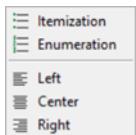
Option	Description
Dimensional	Opens a dialog box to insert a dimensional element: Array, Matrix, Tabular or Tabbing
frac{}{}	Inserts frac{}{} . If there are two selected elements, for example 1 2, it will place both elements in the correct position, i.e. frac{1}{2}
sqrt{}	Inserts sqrt{} . If an element is selected, say 9, it will place this element in the correct position, i.e. sqrt{9}
sqrt[]{}	Inserts sqrt[]{} . If there are two selected elements, for example 3 27, it will place both in the correct position, i.e. sqrt[3]{27}

Header:



Option	Description
Part	Inserts \part{} if no selection or \part{selected}
Chapter	Inserts \chapter{} if no selection or \chapter{selected}
Section	Inserts \section{} if no selection or \section{selected}
Sub-section	Inserts \subsection{} if no selection or \subsection{selected}
Sub-sub-section	Inserts \subsubsection{} if no selection or \subsubsection{selected}
Paragraph	Inserts \paragraph{} if no selection or \paragraph{selected}
Sub-paragraph	Inserts \ subparagraph{} if no selection or \ subparagraph{selected}

Format:



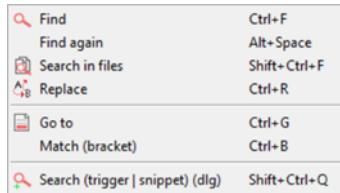
Option	Description
Itemization	Inserts itemization or itemizes a selection
Enumeration	Inserts enumeration or enumerates a selection
Left	Inserts tag to align the text on the left or to align the selection on the left
Center	Inserts tag to align the text on the center or to centralize the selection
Right	Inserts tag to align the text right or to align the selection on the right

Font:

Enphase
B Bold
I Italic
Slatend
Typewriter
T _T Small caps
Tiny
Script size
Footnote size
Small
Normal
Large
Larger
Largest
Huge
Huger

Option	Description
Enphase	Inserts <code>\emph{}</code> if no selection or <code>\emph{selected}</code>
Bold	Inserts <code>\textbf{}</code> if no selection or <code>\textbf{selected}</code>
Italic	Inserts <code>\textit{}</code> if no selection or <code>\textit{selected}</code>
Slatend	Inserts <code>\textsl{}</code> if no selection or <code>\textsl{selected}</code>
Typewriter	Inserts <code>\texttt{}</code> if no selection or <code>\texttt{selected}</code>
Small caps	Inserts <code>\textsc{}</code> if no selection or <code>\textsc{selected}</code>
Tiny	Inserts <code>\tiny{}</code> if no selection or <code>\tiny{selected}</code>
Script size	Inserts <code>\scriptsize{}</code> if no selection or <code>\scriptsize{selected}</code>
Footnote size	Inserts <code>\footnotesize{}</code> if no selection or <code>\footnotesize{selected}</code>
Small	Inserts <code>\small{}</code> if no selection or <code>\small{selected}</code>
Normal	Inserts <code>\normalsize{}</code> if no selection or <code>\normalsize{selected}</code>
Large	Inserts <code>\large{}</code> if no selection or <code>\large{selected}</code>
Larger	Inserts <code>\Large{}</code> if no selection or <code>\Large{selected}</code>
Largest	Inserts <code>\LARGE{}</code> if no selection or <code>\LARGE{selected}</code>
Huge	Inserts <code>\huge{}</code> if no selection or <code>\huge{selected}</code>
Huger	Inserts <code>\Huge{}</code> if no selection or <code>\Huge{selected}</code>

4.7 SEARCH

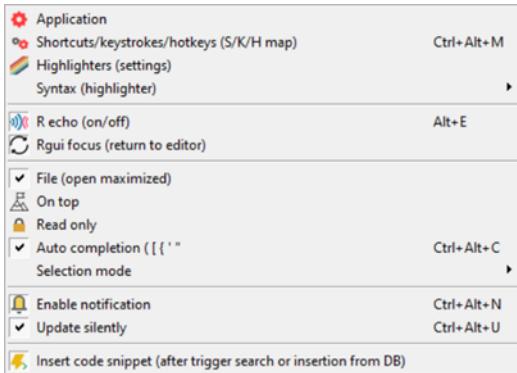


Option	Description
Find	Opens the Find dialog
Find again	Uses the previously entered search criteria to find the next occurrence, i.e., one closer to the end of the file. This option is not available if a search has not been carried out.
Search in files	Opens the Search in files dialog
Replace	Opens the Replace dialog
Go to	This option produces the dialog below and allows you to move the cursor to the specified position
Match bracket	Search for matching bracket. See details below
Search (trigger snippet) (dlg)	Open window (dlg) for efficient search of triggers and snippets. See more details ...

How to match: The cursor must be placed immediately before any of the bracket characters. When this option is called the cursor will move to the point immediately before the matching bracket.

Recognized brackets: The bracket characters are (), [] and {}.

4.8 OPTIONS



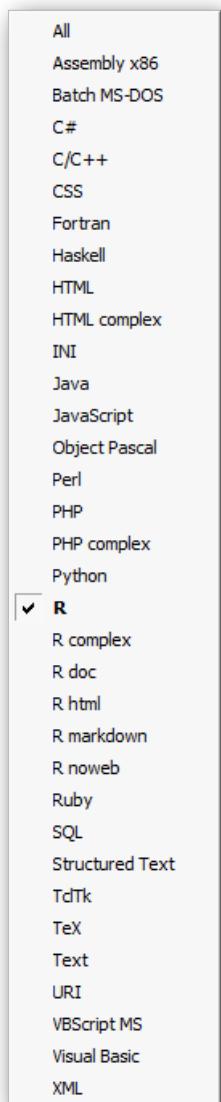
Option	Description
Application	Opens the Application options dialog
Shortcuts/keystrokes/hotkeys (SKH map)	Opens the Shortcuts customization dialog
Highlighters (settings)	Opens the Highlighters (settings) dialog
Syntax (highlighters)	<i>See options ...</i>
R echo (on/off)	Toggles R echo option (file, selection, clipboard, block, contiguous and lines to end page)
Rgui focus (return to editor)	When this option is set the focus will go back to the active editor after any Send or Control action
File (open maximized)	When this option is set all files will be opened maximized
On top	Toggles Tinn-R's ability to be the topmost window on the desktop
Read only	Toggles file read-only status. When set as read-only the file name on the file tab is among <...>
Auto completion	Toggles auto completion resource
Selection mode	<i>See options ...</i>
Enable notification	Toggles file notification resource
Update silently	Toggles update silently resource
Insert code snippet (after trigger search or insertion from DB)	If this option is not checked, after a search via Search trigger or insertion via tools/database/snippets, only the trigger is inserted. Otherwise the trigger is inserted and immediately replaced with the corresponding block of text

Syntax (highlighter)



Option	Description
Set	<i>See options ...</i>
Default (to new files)	<i>See options ...</i>

Set:



Option	Description
All	File without extension or not recognized extension
Assembly x86	x86 Assembly files
Bath MS_DOS	MS_DOS Bath files
C#	C# files
C/C++	C/C++ files
CSS	Cascading SS files
Fortran	Fortran files
Haskell	Haskell files
HTML	Hypertext Markup Language (HTML) files
HTML complex	Hypertext Markup Language (HTML) complex (HTML & JavaScript) files
INI	INI files
Java	Java files
JavaScript	JavaScript files
Object Pascal	Pascal files
Perl	Perl files
PHP	PHP files
PHP complex	PHP (HTML & JavaScript & PHP) complex files
Python	Python files
R	R files
R complex	R complex (R & URI) files
R doc	Rd files
R html	Rhtml files
R markdown	Rmd files
R noweb	R noweb (TeX & R) files
Ruby	Ruby files
SQL	SQL files
Structured Text	Structured Text files
TclTk	TclTk files
TeX	TeX files
Text	Text files
URI	Uniform Resource Identifiers (URI) files
MS VBScript	MS VBScript files
Visual Basic	Visual Basic files
XML	XML files

If necessary select manually one of the list. Tinn-R recognizes automatically the syntax based on the file extensions.

Default (to new files):



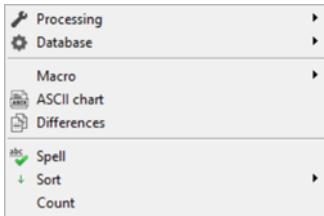
Option	Description
R	When this option is set the highlighter of all new files will be set as <i>R</i>
R complex	When this option is set the highlighter of all new files will be set as <i>R complex</i>
Text	When this option is set the highlighter of all new files will be set as <i>Text</i>

Selection mode

	Normal	Shift+Ctrl+N
	Line	Shift+Ctrl+L
	Column	Shift+Ctrl+C

Option	Description
Normal	See selection type normal ...
Line	See selection type line ...
Column	See selection type column ...

4.9 TOOLS



Option	Description
Processing	<i>See options ...</i>
Database	<i>See options ...</i>
Macro	<i>See options ...</i>
ASCII chart	Allows you to insert an active char to the active document
Differences	Opens the nice TextDiff command by Angus Johnson integrated within Tinn-R
Spell	Starts the speller (<i>see instructions ...</i>)
Sort	<i>See options ...</i>
Count	Shows the result of the count action (words, characters + spaces, character - spaces and spaces) for files or a text selection

Processing



Option	Description
Conversion	<i>See options ...</i>
\LaTeX set main (file)	Set the current file as the main file for operations associated with \LaTeX^*
Compilation (\LaTeX)	<i>See options ...</i>
Viewer	<i>See options ...</i>

*: When there are two main file indications for \LaTeX (one in the project and the other in the main interface for any file), the option from within the project takes precedence, that is, it is mandatory.

Conversion:



Option	Description
Deplate to	<i>See options ...</i>
Pandoc	<i>See options ...</i>
Txt2tags to	<i>See options ...</i>

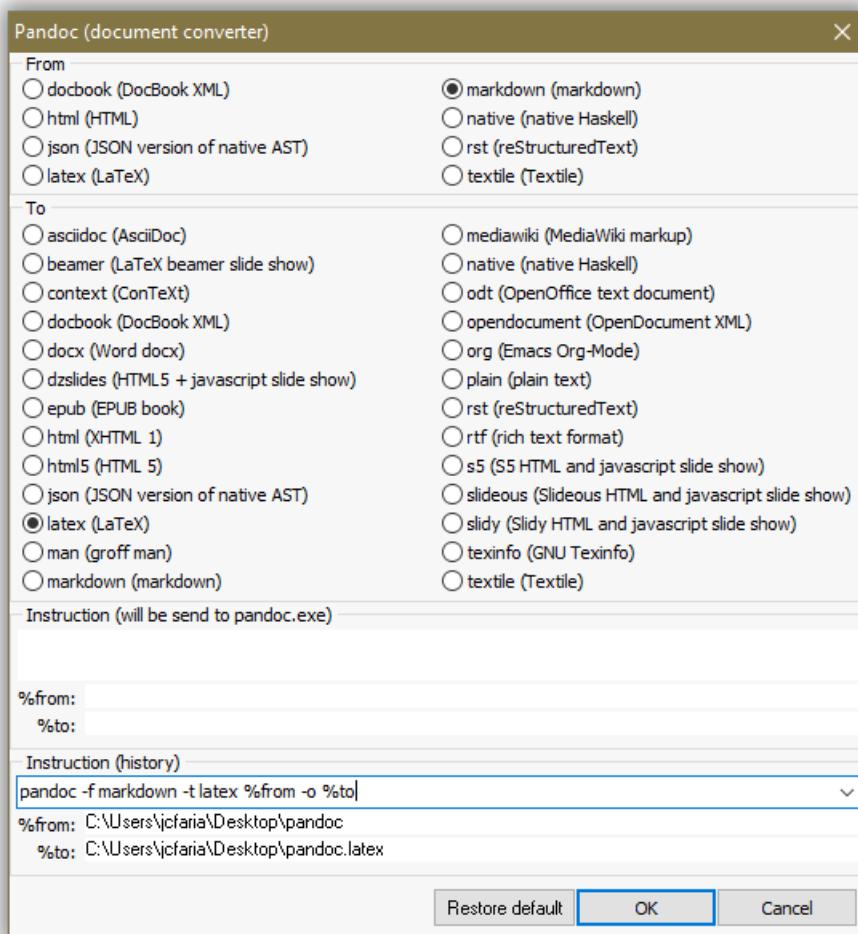
Deplate to:



Option	Description
\TeX	Converts a Deplate file to \TeX
LaTeX-dramatist	Converts a Deplate file to LaTeX-dramatist
Sweave	Converts a Deplate file to Sweave
Plain	Converts a Deplate file to Plain
HTML	Converts a Deplate file to HTML
HTML site	Converts a Deplate file to HTML site
HTML slides	Converts a Deplate file to HTML slides
XHTML 1.0 transitional (xhtml10t)	Converts a Deplate file to XHTML 1.0 transitional
XHTML 1.1 with MathML (xhtml11m)	Converts a Deplate file to XHTML 1.1 with MathML
PHP	Converts a Deplate file to PHP
Dbk-article	Converts a Deplate file to Dbk-article
Dbk-book	Converts a Deplate file to Dbk-book
Dbk-ref	Converts a Deplate file to Dbk-ref

Tip: see details ...

Pandoc



Tip: see details ...

Txt2tags to:

Latex
Sweave
Text
HTML
XHTML
Sgml
Lout
Man page
Wikipedia
Google code wiki
Doku wiki
Moinmoin
Magic point
Page maker

Option	Description
\LaTeX	Converts a Txt2tags file into \LaTeX
Sweave	Converts a Txt2tags file into Sweave
Txt	Converts a Txt2tags file into txt
HTML	Converts a Txt2tags file into HTML
XHTML	Converts a Txt2tags file into XHTML
SGML	Converts a Txt2tags file into SGML
Lout	Converts a Txt2tags file into Lout
Man page	Converts a Txt2tags file into Man page
Wikipedia	Converts a Txt2tags file into Wikipedia
Google code wiki	Converts a Txt2tags file into Google code wiki
Doku wiki	Converts a Txt2tags file into Doku wiki
Moinmoin	Converts a Txt2tags file into Moinmoin
Magic point	Converts a Txt2tags file into Magic point
Page maker	Converts a Txt2tags file into Page maker

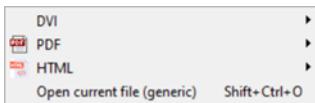
Tip: see details ...

Compilation (\LaTeX):

<input checked="" type="checkbox"/>	Always shows console minimized when compiling
<input type="checkbox"/>	DVI (single)
<input type="checkbox"/>	DVI (bibtex)
<input type="checkbox"/>	PDF (single)
	Ctrl+Alt+P
<input type="checkbox"/>	PDF (bibtex)
	Shift+Ctrl+Alt+P
<input type="checkbox"/>	Make index (makeindex)
<input type="checkbox"/>	Ctrl+Alt+I
<input type="checkbox"/>	LaTeX clear (waste)
	Shift+Ctrl+X
<input type="checkbox"/>	Set main (LaTeX)

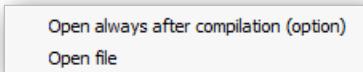
Option	Description
Always shows console minimized when compiling	When this option is set the DOS console will be minimized when compiling
DVI (single)	Compiles a \LaTeX file to DVI in single way
DVI (bibtex)	Compiles a \LaTeX file to DVI in bibtex way (three compilation)
PDF (single)	Compiles a \LaTeX file to PDF in single way
PDF (bibtex)	Compiles a \LaTeX file to PDF in bibtex way (three compilation)
Make index (makeindex)	
\LaTeX clear (waste)	
Main \LaTeX file is set to (main file name)	

Viewer:



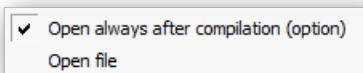
Option	Description
DVI	<i>See options ...</i>
PDF	<i>See options ...</i>
HTML	<i>See options ...</i>
Open current file (generic)	

DVI:



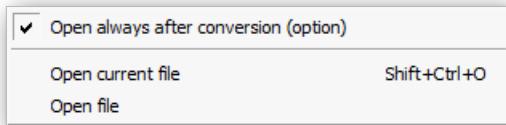
Option	Description
Open always after compilation (option)	When this option is set the DVI file will be opened by the viewer after the compilation
Open file	Shows the <i>Windows Open dialog</i> to select a DVI file to be opened by the viewer

PDF:



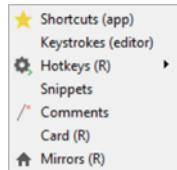
Option	Description
Open always after compilation (option)	When this option is set the PDF file will be opened by the viewer after the compilation
Open file	Shows the <i>Windows Open dialog</i> to select a DVI file to be opened by the viewer

HTML:



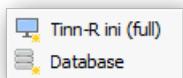
Option	Description
Open always after conversion (option)	When this option is set the HTML file will be opened by the viewer after the conversion
Open current file	Opens the current HTML file with the viewer
Open file	Shows the <i>Windows Open dialog</i> to select a HTML file to be opened by the viewer

Database

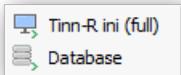


Option	Description
Shortcuts (app)	Shows Shortcuts database (XML based) dialog
keystrokes (editor)	Shows keystrokes database (XML based) dialog
Hotkeys (R)	Shows hotkeys database (XML based) dialog
Snippets	Shows Snippets database (XML based) dialog
Comments	Shows Comments database (XML based) dialog
Card (R)	Shows R card database (XML based) dialog
Mirrors (R)	Shows R mirrors database (XML based) dialog

Backup



Option	Description
System configuration	Backups Tinn-R configuration (ini files)
Database	Backups database (Cachexml, Comments.xml, Completions.xml, Rcard.xml, Rmirrors.xml and Shortcuts.xml)

Restore

Option	Description
System configuration	Restores a prior Tinn-R backup (ini files)
Database	Restores a prior database backup (Cachexml, Comments.xml, Completions.xml, Rcard.xml, Rmirrors.xml and Shortcuts.xml)

Macro

Option	Description
Record	Toggles macro recording on and off. Note that when recording a macro the button changes
Play	Plays a previous recorded macro

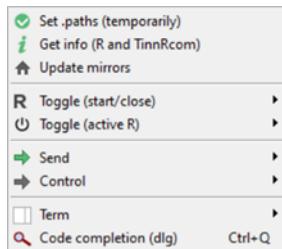
It is not possible to save/edit macros, they are temporary

Sort

Option	Description
Strings	Sorts strings
Numbers	Sorts numbers
Dates	Sorts dates

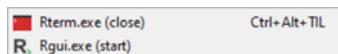
Sort works on the entire document unless some text is selected

4.10 R



Option	Description
Set .paths (temporarily)	Sets (temporarily) the necessary .paths object in R environment. This object provided by TinnRcom package. This option is useful only if the user can not, for some reason, install the TinnRcom package
Get info (R and TinnRcom)	Get information about R and the necessary TinnRcom package
Update mirrors	Updates the Rmirrors.xml file
Toggle (start/close)	<i>See options ...</i>
Toggle (active R)	<i>See options ...</i>
Send	<i>See options ...</i>
Control	<i>See options ...</i>
Term	<i>See options ...</i>
Code completion (dlg)	Opens the dialog window for efficient search of R objects. <i>See more details ...</i>

Toogle (start/close)

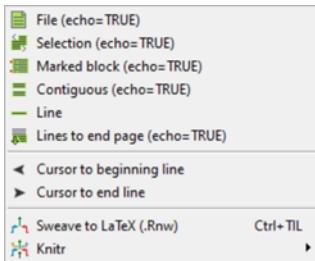


Option	Description
Rterm.exe (start/close)	Toggles Rterm.exe
Rgui.exe (start/close)	Toggles Rgui.exe

Toogle (active R)



Option	Description
Rterm.exe	(is set as active*)
Rgui.exe	

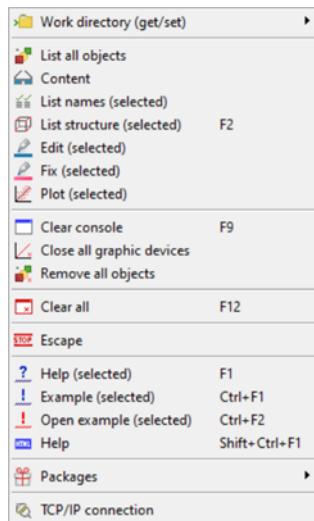
Send

Option	Description
File	Sends current file to R interpreter
Selection	Sends current selection to R interpreter
Marked block	Sends current marked block to R interpreter
Contiguous	Sends contiguous lines to R interpreter
Line	Sends current line to R interpreter echoing it
Lines to end page	Sends all visible lines to end page echoing it
Cursor to beginning line	Sends cursor position to beginning line
Cursor to end line	Sends cursor position to end line
SWeave	Sends to R interpreter SWeave('Active file') instruction
Knitr	<i>See options ...</i>

Knitr:

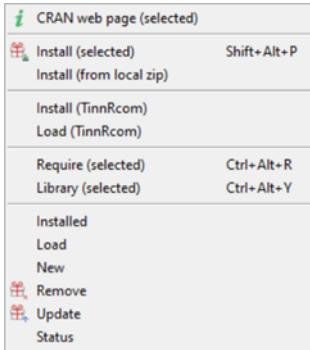
Option	Description
Knit to \LaTeX (Rnw)	Knit the *.Rnw file to \LaTeX
Knit to HTML (Rmd, Rhtml)	Knit the *.Rmd or *.Rhtml file to HTML

Control



Option	Description
Work directory (get/set)	<i>See options ...</i>
List all objects	Sends to R interpreter a <code>ls()</code> instruction
Content (selected)	Sends to R interpreter a <code>selected</code> word
List names (selected)	Sends to R interpreter a <code>names(selected)</code> instruction
List structure (selected)	Sends to R interpreter a <code>str(selected)</code> instruction
Edit (selected)	Sends to R interpreter a <code>edit(selected)</code> instruction
Fix (selected)	Sends to R interpreter a <code>fix(selected)</code> instruction
Plot (selected)	Sends to R interpreter a <code>plot(selected)</code> instruction
Clear console	Sends and executes the virtual CTRL + L (clear screen) instruction
Close all graphic devices	Sends to R interpreter a <code>graphics.off()</code> instruction
Remove all objects	Sends to R interpreter a <code>rm(list=ls())</code> instruction
Clear all	Sends to R interpreter a <code>graphics.off(); rm(list=ls())</code> CTRL + L instructions
Escape	Stops all computations in Rgui.exe
Help (selected)	Sends to R interpreter a <code>help(selected)</code> instruction
Example (selected)	Sends to R interpreter a <code>example(selected)</code> instruction
Open example (selected)	Sends to R interpreter an instruction to generate an example text file of the object selected
Help	Sends to R interpreter a <code>help.start(update=FALSE)</code> instruction
Packages	<i>See options ...</i>
TCP/IP connection	Sends to R interpreter an instruction to start: <code>startSocketServer(port=portnumber)</code> or stop: <code>stopSocketServer(port=portnumber)</code> the TCP/IP connection

Packages:



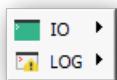
Option	Description
CRAN web page (selected)	Open the selected package page in the default browser
Install (selected)	Sends to R interpreter an <code>utils:::menuInstallPkgs()</code> instruction
Install (from local zip)	Sends to R interpreter a <code>utils:::menuInstallLocal()</code> instruction
Install (TinnRcom)	Sends to R interpreter instruction to install TinnRcom package and its dependencies. By default it is not necessary since the TinnRcom package is automatically installed
Load (TinnRcom)	Sends to R interpreter an <code>library(TinnRcom)</code> instruction. By default it is not necessary since the TinnRcom package is automatically loaded when R starts
Require (selected)	Sends to R interpreter a <code>require(selected)</code> instruction
Library (selected)	Sends to R interpreter a <code>library(selected)</code> instruction
Installed	Sends to R interpreter a <code>installed.packages()</code> instruction
Load	Sends to R interpreter a <code>local({pkg <- select.list(sort(.packages(all.available = TRUE))); if(nchar(pkg)) library(pkg, character.only=TRUE)})</code> instruction
New	Sends to R interpreter a <code>new.packages()</code> instruction
Remove	Sends to R interpreter a <code>local({pkg <- select.list(sort(.packages(all.available = TRUE))); if(nchar(pkg)) remove.packages(pkg)})</code> instruction
Update	Sends to R interpreter an <code>update.packages(ask='graphics')</code> instruction
Status	Sends to R interpreter a <code>packageStatus()</code> instruction

Term



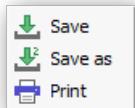
Option	Description
Term (show/hide)	Toggles (show/hide) Term interface
File	<i>See options ...</i>
Clear	<i>See options ...</i>
Focus	<i>See options ...</i>
Size	<i>See options ...</i>
LOG (size)	Minimize, divide or maximize LOG
Split	<i>See options ...</i>
Highlighter	<i>See options ...</i>
Line wrap	<i>See options ...</i>
History	<i>See options ...</i>
Workspace	<i>See options ...</i>
Font of active control (not permanent)	<i>See options ...</i>

File:



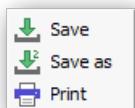
Option	Description
IO	<i>See options ...</i>
LOG	<i>See options ...</i>

File IO:



Option	Description
Save	Saves the content of the IO interface
Save as	Saves the content of the IO interface as a new file
Print	Opens the Tinn-R print dialog with the content from the IO interface

File LOG:



Option	Description
Save	Saves the content of the LOG interface
Save as	Saves as the content of the LOG interface
Print	Opens the Tinn-R print dialog with content from LOG interface

Clear:

 IO	
 LOG	Ctrl+Alt+L
 IO and LOG	Ctrl+L
 History	Alt+BkSp

Option	Description
IO	Clear IO
LOG	Clear LOG
IO and LOG	Clear IO and LOG

Focus:

 Editor	Alt+Left
 IO	Alt+Right
 LOG	Alt+PgDn

Option	Description
Editor	Places the focus inside of the <i>editor</i>
IO	Places the focus inside of the <i>IO</i>
LOG	Places the focus inside of the <i>LOG</i>

Size:

 Term (maximize)	Ctrl+F12
 Term (divide)	Ctrl+F11
 Term (minimize)	Ctrl+F10

Option	Description
Term (maximize)	Maximizes the Term interface
Term (divide)	Divides the Term interface
Term (minimize)	Minimizes the Term interface

Split:

 Horizontal split (IO and LOG in the same view)
 Vertical split (IO and LOG in the same view)
Remove split (IO and LOG in distinct view)

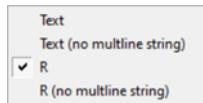
Option	Description
Horizontal split (IO and LOG in the same view)	Splits horizontally the Term interface placing <i>IO</i> and <i>LOG</i> on the same view
Vertical split (IO and LOG in the same view)	Splits vertically the Term interface placing <i>IO</i> and <i>LOG</i> on the same view
Remove split (IO and LOG in distinct view)	Removes split placing <i>IO</i> and <i>LOG</i> in distinct view

Highlighter:



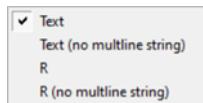
Option	Description
IO	<i>See options ...</i>
LOG	<i>See options ...</i>

Highlighter IO:



Option	Description
Text	Sets the IO highlighter to Text
Text (no multiline string)	Sets the IO highlighter to Text without string multiline support
R	Sets the IO highlighter to R
R (no multiline string)	Sets the IO highlighter to R without string multiline support

Highlighter LOG:



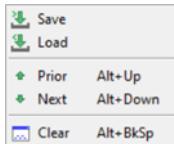
Option	Description
Text	Sets the LOG highlighter to Text
Text (no multiline string)	Sets the LOG highlighter to Text without string multiline support
R	Sets the LOG highlighter to R
R (no multiline string)	Sets the LOG highlighter to R without string multiline support

Line wrap:



Option	Description
IO	Sets line wrap to IO
LOG	Sets line wrap to LOG

History:



Option	Description
Save	Saves the history
Load	Loads the history
Prior	Prior section of the history
Next	Next section of the history
Clear	Clear the Term history

Workspace:



Option	Description
Save	Saves the workspace
Load	Loads the workspace

Font of active control (not permanent):



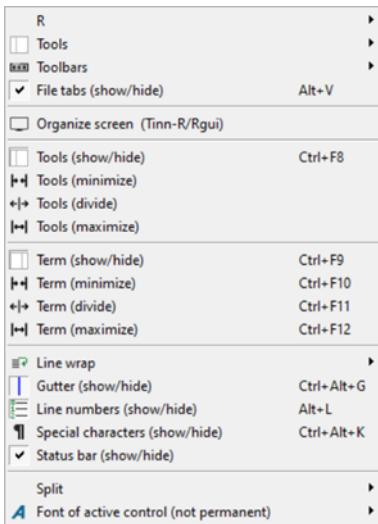
Option	Description
Increase	Increase the font size
Decrease	Decrease the font size

Work directory:



Option	Description
Get (current)	Get the work directory of the R interpreter
Set (to current file path)	Set the work directory of the R interpreter to the current file path
Set (choose)	Set the work directory of the R interpreter allowing to choose from the Windows dialog

4.11 VIEW



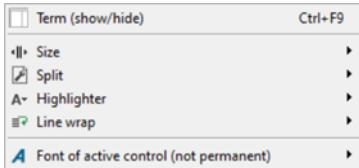
Option	Description
R	<i>See options ...</i>
Tools	<i>See options ...</i>
Toolbar	<i>See options ...</i>
File tabs (show/hide)	Toggles (show/hide) the tabs of files
Organize screen (Tinn-R/Rgui)	Organizes the screen (Tinn-R and Rgui.exe - SDI mode) according to the user set defined in the application options (Options/Application/R/Rgui). <i>See options ...</i>
Tools (show/hide)	Toggles (show/hide) Tools interface
Tools (minimize)	Minimizes the Tools interface
Tools (divide)	Divides the Tools interface
Tools (maximize)	Maximizes the Tools interface
Term (show/hide)	Toggles (show/hide) Term interface
Term (minimize)	Minimizes the Term interface
Term (divide)	Splits the Term interface
Term (maximize)	Maximizes the Term interface
Line wrap (show/hide)	<i>See options ...</i>
Gutter (show/hide)	Toggles (show/hide) gutter
Line numbers (show/hide)	Toggles (show/hide) line numbers
Special characters (show/hide)	Toggles (show/hide) special characters
Status bar (show/hide)	Toggles (show/hide) status bar
Split	<i>See options ...</i>
Font of active control (not permanent)	<i>See options ...</i>

R



Option	Description
Organize screen (Tinn-R/Rgui)	Organizes the screen (Tinn-R and Rgui.exe - SDI mode) according to the user set defined in the application options (Options/Application/R/Rgui). <i>See options ...</i>
Term	<i>See options ...</i>

Term:



Option	Description
Term (show/hide)	Toggles (show/hide) Term interface
Size	<i>See options ...</i>
Split	<i>See options ...</i>
Highlighter	<i>See options ...</i>
Line wrap	<i>See options ...</i>
Font of active control (not permanent)	<i>See options ...</i>

Size:



Option	Description
Term (maximize)	Maximizes the Term interface
Term (divide)	Splits the Term interface
Term (minimize)	Minimizes the Term interface

Split:



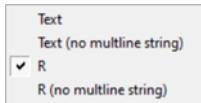
Option	Description
Horizontal split (IO and LOG in the same view)	Horizontally splits the Term interface placing <i>IO</i> and <i>LOG</i> in the same view
Vertical split (IO and LOG in the same view)	Vertically splits the Term interface placing <i>IO</i> and <i>LOG</i> in the same view
Remove split (IO and LOG in distinct view)	Removes split placing <i>IO</i> and <i>LOG</i> in distinct view

Highlighter:



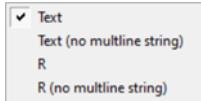
Option	Description
IO	<i>See options ...</i>
LOG	<i>See options ...</i>

Highlighter IO:



Option	Description
Text	Sets the IO highlighter to Text
Text (no multiline string)	Sets the IO highlighter to Text without string multiline support
R	Sets the IO highlighter to R
R (no multiline string)	Sets the IO highlighter to R without string multiline support

Highlighter LOG:



Option	Description
Text	Sets the LOG highlighter to Text
Text (no multiline string)	Sets the LOG highlighter to Text without string multiline support
R	Sets the LOG highlighter to R
R (no multiline string)	Sets the LOG highlighter to R without string multiline support

Line wrap:



Option	Description
IO	Sets Line wrap to IO
LOG	Sets Line wrap to LOG

Font of active control (not permanent):



Option	Description
Increase	Increase font size
Decrease	Decrease font size

Tools



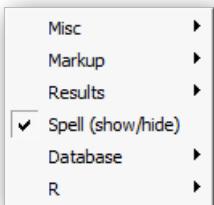
Option	Description
Tools (show/hide)	Toggles (show/hide) <i>Tools</i> interface
Size	<i>See options ...</i>
Resources	<i>See options ...</i>

Size:



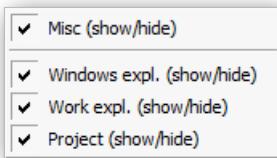
Option	Description
Tools (maximize)	Maximizes the <i>Tools</i> interface
Tools (divide)	Divides the <i>Tools</i> interface
Tools (minimize)	Minimizes the <i>Tools</i> interface

Resources:



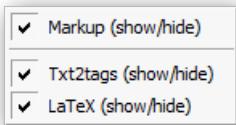
Option	Description
Misc	<i>See options ...</i>
Markup	<i>See options ...</i>
Results	<i>See options ...</i>
Spell (show/hide)	Toggles (show/hide) <i>Spell</i> tab of <i>Tools</i> interface
Database	<i>See options ...</i>
R	<i>See options ...</i>

Misc:



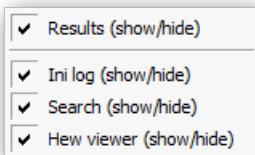
Option	Description
Misc (show/hide)	Toggles (show/hide) <i>Misc</i> tab of <i>Tools</i> interface
Windows expl. (show/hide)	Toggles (show/hide) <i>Windows expl.</i> tab of <i>Misc</i>
Work expl. (show/hide)	Toggles (show/hide) <i>Work expl.</i> tab of <i>Misc</i>
Project (show/hide)	Toggles (show/hide) <i>Project</i> tab of <i>Misc</i>

Markup:



Option	Description
Markup (show/hide)	Toggles (show/hide) <i>Markup</i> tab of <i>Tools</i> interface
Txt2tags (show/hide)	Toggles (show/hide) <i>Txt2tags</i> tab of <i>Markup</i>
TeX (show/hide)	Toggles (show/hide) <i>\TeX</i> tab of <i>Markup</i>

Results:



Option	Description
Results (show/hide)	Toggles (show/hide) <i>Results</i> tab of <i>Tools</i> interface
Ini LOG (show/hide)	Toggles (show/hide) <i>Ini LOG</i> tab of <i>Results</i>
Search (show/hide)	Toggles (show/hide) <i>Search</i> tab of <i>Results</i>
Hex viewer (show/hide)	Toggles (show/hide) <i>Hex viewer</i> tab of <i>Results</i>

Database:



Option	Description
Database (show/hide)	Toggles (show/hide) <i>Database</i> tab of <i>Tools</i> interface
Shortcuts (show/hide)	Toggles (show/hide) <i>Shortcuts</i> tab of <i>Database</i>
Keystrokes (show/hide)	Toggles (show/hide) <i>Keystrokes</i> tab of <i>Database</i>
Hotkeys (show/hide)	Toggles (show/hide) <i>Hotkeys</i> tab of <i>Database</i>
Snippets (show/hide)	Toggles (show/hide) <i>Snippets</i> tab of <i>Database</i>
Comments (show/hide)	Toggles (show/hide) <i>Comments</i> tab of <i>Database</i>

R:



Option	Description
R (show/hide)	Toggles (show/hide) <i>R</i> tab of <i>Tools</i> interface
Explorer (show/hide)	Toggles (show/hide) <i>Explorer</i> tab of <i>R</i>
Card (show/hide)	Toggles (show/hide) <i>Rcard</i> tab of <i>Database</i>
Mirrors (show/hide)	Toggles (show/hide) <i>R mirrors</i> tab of <i>Database</i>

Toolbars



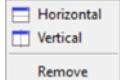
Option	Description
All bars (show/hide)	Toggles (show/hide) <i>All bars</i> tab of <i>Tools bar</i> interface
Edit (show/hide)	Toggles (show/hide) <i>Edit</i> tab of <i>Tools bar</i> interface
File (show/hide)	Toggles (show/hide) <i>File</i> tab of <i>Tools bar</i> interface
Filter (show/hide)	Toggles (show/hide) <i>Filter</i> tab of <i>Tools bar</i> interface
Format (show/hide)	Toggles (show/hide) <i>Format</i> tab of <i>Tools bar</i> interface
Macro (show/hide)	Toggles (show/hide) <i>Macro</i> tab of <i>Tools bar</i> interface
Misc (show/hide)	Toggles (show/hide) <i>Misc</i> tab of <i>Tools bar</i> interface
Processing (show/hide)	Toggles (show/hide) <i>Processing</i> tab of <i>Tools bar</i> interface
R (show/hide)	Toggles (show/hide) <i>R</i> tab of <i>Tools bar</i> interface
Search (show/hide)	Toggles (show/hide) <i>Search</i> tab of <i>Tools bar</i> interface
Spell (show/hide)	Toggles (show/hide) <i>Spell</i> tab of <i>Tools bar</i> interface
Syntax (show/hide)	Toggles (show/hide) <i>Syntax</i> tab of <i>Tools bar</i> interface
View (show/hide)	Toggles (show/hide) <i>View</i> tab of <i>Tools bar</i> interface

Line wrap



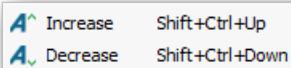
Option	Description
Editor (show/hide)	Toggles (show/hide) <i>Editor</i> line wrap
Term/IO (show/hide)	Toggles (show/hide) <i>Term/IO</i> line wrap
Term/LOG (show/hide)	Toggles (show/hide) <i>Term/LOG</i> line wrap

Split



Option	Description
Horizontal	Horizontally splits the editor
Vertical	Vertically splits the editor
Remove	Removes split

Font of active control (not permanent)



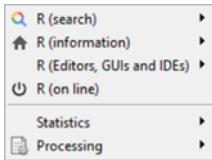
Option	Description
Increase	Increases font size
Decrease	Decreases font size

4.12 WINDOW



Option	Description
Tile vertically	Shows two views of the same file tiled vertically, to the left and right. Each can be scrolled independently
Tile horizontally	Shows two views of the same file tiled horizontally, one above the other. Each can be scrolled independently
Minimize all	Minimizes all windows (editor)
Cascade	The windows cascade from the upper left to the lower right of the workspace
Arrange icons	Windows are tiled horizontally, but the active document comes on top. You may also drag your document tabs to the order you prefer and then tile them horizontally
Files opened	If many files are opened, a dialog will be open to select a file

4.13 WEB



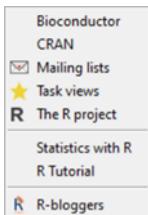
Option	Description
R (search)	<i>See options ...</i>
R (information)	<i>See options ...</i>
R (Editors, GUIs and IDEs)	<i>See options ...</i>
R (online)	Opens URL rdrr.io
Statistics	<i>See options ...</i>
Processing	<i>See options ...</i>

R (search)



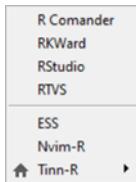
Option	Description
Google (selected)	Opens URL Google and lists the results associated with the word under the cursor or selected text
Search (R project)	Opens URL Search R project
R-help (mailing list information)	Opens URL r-help
Rseek	Opens URL Rseek

R (information)



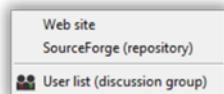
Option	Description
Bioconductor	Opens URL Bioconductor project
CRAN	Opens URL The Comprehensive R Archive Network
Mailing lists	Opens URL Mailing Lists
Task views	Opens URL CRAN Task Views
The R project	Opens URL The R Project for Statistical Computing
Statistical with R	Opens URL Statistical with R
R tutorial	Opens URL R tutorial
R-bloggers	Opens URL R-bloggers

R (Editors, Gui's and IDEs)



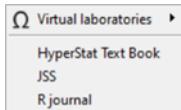
Option	Description
R Comander	Opens URL The R Commander: A Basic-Statistics GUI for R
RKWard	Opens URL RKWard
RStudio	Opens URL RStudio
ESS	Opens URL Emacs Speaks Statistics (ESS)
Nvim-R	Opens URL Nvim-R: Plugin to work with R
Tinn-R	<i>See options ...</i>

Tinn-R



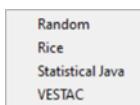
Option	Description
Web site	Opens URL Web page of Tinn-R project
SourceForge (repository)	Opens URL Sourceforge.net Tinn-R
User list (discussion group)	Opens URL Groups Google Tinn-R

Statistics



Option	Description
Virtual laboratories	<i>See options ...</i>
HyperStat Text Book	Opens URL HyperStat Text Book
JSS	Opens URL Journal of Statistical Software
R Journal	Opens URL R Journal

Virtual laboratories:



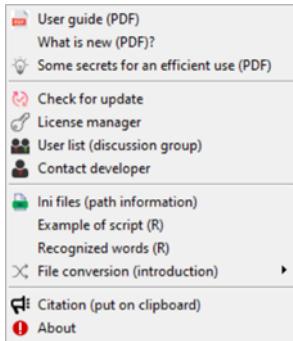
Option	Description
Random	Opens URL Probability, Mathematical Statistics, Stochastic Processes
Rice	Opens URL Rice Virtual Lab in Statistics
Statistical Java	Opens URL Statistical Java
VESTAC	Opens URL Java Applets for Visualization of Statistical Concepts

Processing



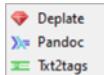
Option	Description
Deplate	Opens URL Sourceforge.net Deplate
MiKTeX	Opens URL MiKTeX project page
Pandoc	Opens URL Pandoc (a universal document converter)
TeX Live	Opens URL TeX Live project page
Txt2tags	Opens URL Txt2tags ONE source, MULTI targets

4.14 HELP



Option	Description
User guide (PDF)	Opens the User guide with the PDF viewer default
What is new (PDF)?	Opens the User guide with the PDF viewer default at <i>What is new?</i>
Some secrets for an efficient use (PDF)	Opens the User guide with the PDF viewer default at <i>Some secrets for an efficient use</i>
Check for update	Opens an updater dialog
Licence manager	Opens Licence manager dialog
Contact developer	Opens URL Tinn-R Editor - GUI for R Language and Environment contact
Ini files (path information)	Displays a single dialog with the path information of ini files for Tinn-R
Example of script (R)	Opens the file <i>Tinn-r_example of script.r</i>
Recognized words (R)	Opens the file <i>Tinn-R_recognized words.r</i>
File conversion (introduction)	<i>See options ...</i>
Citation (put on clipboard)	Places a text containing the Tinn-R citation in the clipboard
About	Opens the dialog About

File conversion (introduction)



Option	Description
Deplate	Opens the file <i>deplate_intro.t2t</i>
Pandoc	Opens the file <i>pandoc.markdown</i>
Txt2tags	Opens the file <i>txt2tags_intro.t2t</i>

CHAPTER 5

SOME SECRETS FOR AN EFFICIENT USE

The idea behind this chapter is that it has contributions from multiple users. If you find something that is not important but that you would like to include here, please submit it to the [project leader](#).

5.1 INTRODUCTION

Let us assume that you are a basic user of R and you do not understand the intricacies of computer languages and that you want to write and run an R-script. The software providing an interface between R and the user is called a *user interface*. If this interface also has graphical capabilities, it is called a graphical user interface or *GUI*.

R running under Windows and Mac have a graphical interface (Rgui) that allows you to submit your commands and see the respective results, but that interface is a bit limited.

Many attempts have been made to provide R with a user friendly graphical interface running on Windows. One of the most successful ones is Tinn-R, which is arguably the most used "GUI" among R users on Windows.

A large number of Tinn-R users do not know how to improve its performance and their productivity, so this chapter is an attempt to help you to make the most out of its features.

Let us start, then. After downloading Tinn-R from its main *Web page* or *Sourceforge*, you should install and open the software.

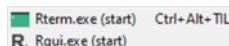
5.2 AFTER INSTALLATION

At the top of the screen there is a Main menu (1) and two toolbars: the top toolbar is the *main toolbar* (2) and the bottom task bar is the *R toolbar* (3).



If your version is the same or above 3.0.1.0, Tinn-R does not require any special configuration. That is, the program is ready to be used. One important thing to be done before using it: set a R mirror as close as possible to where you work. For that, first click on CTRL + F8. This opens the Tools window, then click on R/Mirrors. Select the R mirror and push the button that shows an hourglass in the taskbar. The chosen repository will be the new default for all actions dependent repository (install packages, upgrade packages, etc).

The second step is connecting Tinn-R with R. Look at the *R toolbar*. Almost at the right end you will see two icons together: one is the R symbol and the second is like a green television screen:



The first connects Tinn-R to R Console (32 or 64 bit), also called Rgui . exe, the second with Rterm . exe. With Tinn-R the Rgui.exe should be used only when you need heavy and intensive processing; otherwise, you should always use Rterm.exe, which is a lot more much friendly, having many editorial features of Tinn-R editor. However it consumes more computer resources. Click on that little screen as soon as the connection with R is made. That screen will become red and Term will appear on a window by itself. You can move that window across Tinn-R main window and dock it either at the left, right or bottom side.

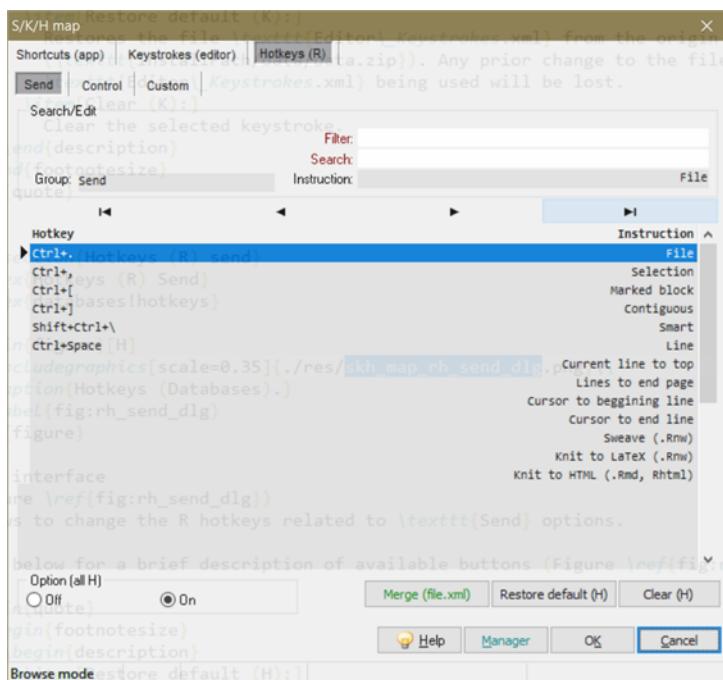
The best location will depend on the size of your computer screen. To do so, just put the mouse on the blue strip at the top of Term window, click the left button of the mouse and move. To dock it at either side just pull it closer and closer to the chosen side and then, bingo, you will see how it becomes when docked. We like to use Tinn-R with two monitors: the editor docked in one and the Rterm.exe (or Rgui.exe) interface in other. It is a very comfortable and productive arrangement. You should follow the same procedure to dock the Tools window.

Since R computer language is an interpreted one, each command given

gets its answer right away, therefore the most used command in Tinn-R is the send line which sends the command line to be interpreted by R. You will see the answer to the command at Term or Rgui.exe window. This command appears at the bottom icon bar, the sixth box, the one with just one line.



You just have to click there and the line is sent to R. Even though this is a nice way to do it, there is a faster way to send a line to R. First, click on Options/Shortcuts/Keystrokes/Hotkeys (map) at the Main menu and then on Hotkeys (operational system). On the open window click on the line send line to select it. Then go to the Manager button, click there and then press (for example, Ctrl + \), then click on the bottom bar the button OK and then set all to on.



Now, whenever you press **Ctrl + ** the line you are on is sent to R, it is much faster than clicking with the mouse at the appropriate button and

has the advantage that you can send the line to R wherever is the focus of your present work in your computer.

You can also turn on and off the hotkeys by clicking the hotkeys on the status bar at the bottom of the Tinn-R screen.

Another important feature on the status bar is the `smNormal` (`s=selection`, `m=mode`) box. This allows you to select a portion of a file. Selecting it will change your options to `smLine` or `smColumn`. The latter is helpful since it allows you to select columns within a file without having to carry the whole line with it. Give it a try.



The first icons at the R task bar are related to different ways of sending instructions: The whole file, selected parts, the clipboard content, block marked, contiguous lines, single line, current line to bottom of page, parts of a line, Sweave and Knitr. Those are helpful when you are dealing with long scripts, and may very well enhance your programming efficiency. Almost all of them have the option to send lines straight to R and with `(echo=TRUE)` option.



5.3 SHORTCUTS THAT CAN MAKE YOU LOSE THE VISIBILITY OF A RESOURCE

CTRL + ALT + V: Toggles the visibility for two main bars (Editor and R)

CTRL + F8 : Toggles the visibility for \texttt{Tools} window

CTRL + F9 : Toggles the visibility for \texttt{Term} window

If a shortcut is accidentally typed, just toggle it and the related resource will be visible again.

5.4 I'M PANICKING: WHAT DO I DO?

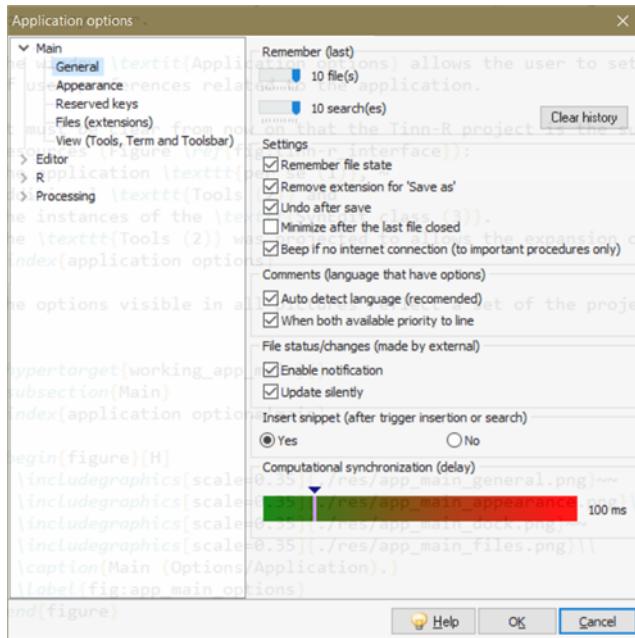
First: **don't panic (it's not good for your health)!**

- **Do not uninstall and reinstall Tinn-R or R unnecessarily.** The origin of the problem (although rare) can be mainly the folder where Tinn-R has installed its ini files (it does not use the windows registry). For some reason, it was corrupted or damaged. The first step is to know where Tinn-R stores these files: *Help/Ini files* (path information) will do the job for you. After that, close Tinn-R, rename (or remove) this folder (it will be recreated after Tinn-R is restarted). If this folder is not visible in your computer *see useful links here*.
- **Feel free to write to the *project leader*.** If you submit a bug report, please provide as much detail as possible. This includes indicating the Tinn-R version, your operating system (Windows XP, Windows 7, etc) and language (English, French, Portuguese). If the bug is related to an interface with R, indicate which version of R you are using, as well as whether you are running Rterm.exe or Rgui.exe. You should also add the content of the *Tools/Results/Ini log* interface since this will help us to address the issue promptly.

5.5 APPLICATION OPTIONS

General

At the main menu click on Options/Application. It will open the window Options/Application/Main/General.

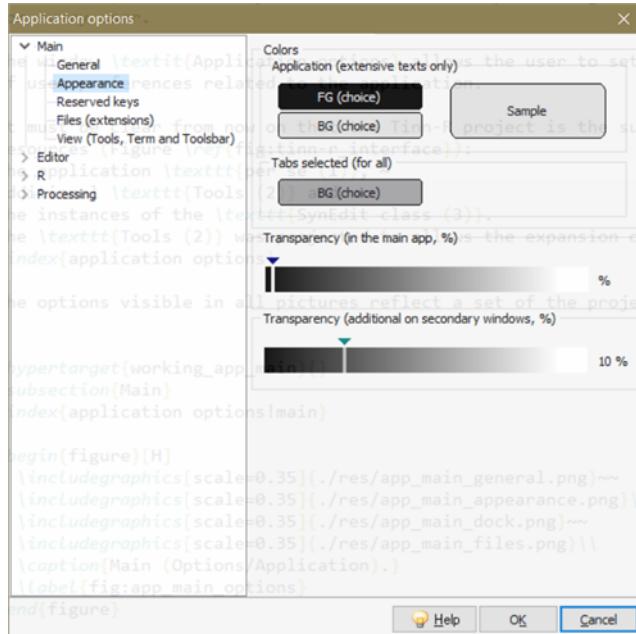


One important option here is the *Computational synchronization (delay)*. Several processes are dependent on synchronization between applications (R, converters, compilers). The optimal value of the delay is determined by the following characteristics: user habits, hardware and software available. The ideal value is unique to the various possible combinations of those three characteristics. Try to reduce to the minimum value (50 ms) and test it: if something does not work, increase it gradually and keep testing until getting to the optimal value. The default value (100 ms) may not be optimal for all users.

Appearance

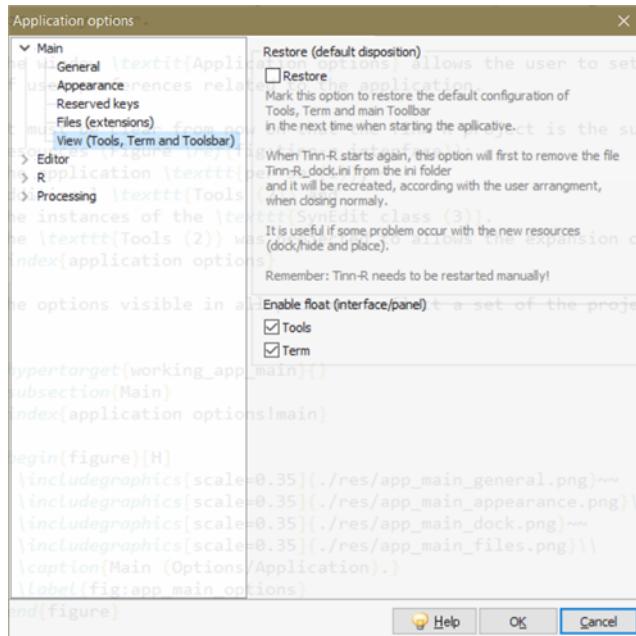
At the Options window click on the tab Main/Appearance. You can choose colors for each character's foreground (FG) and background (BG) color. The color palette will open and you can then choose the appropriate color. For people working extensive periods of time in front of a computer mon-

itor, dark (or pale) colors with a low level of radiation are recommended for background, obtaining a contrast with characters.



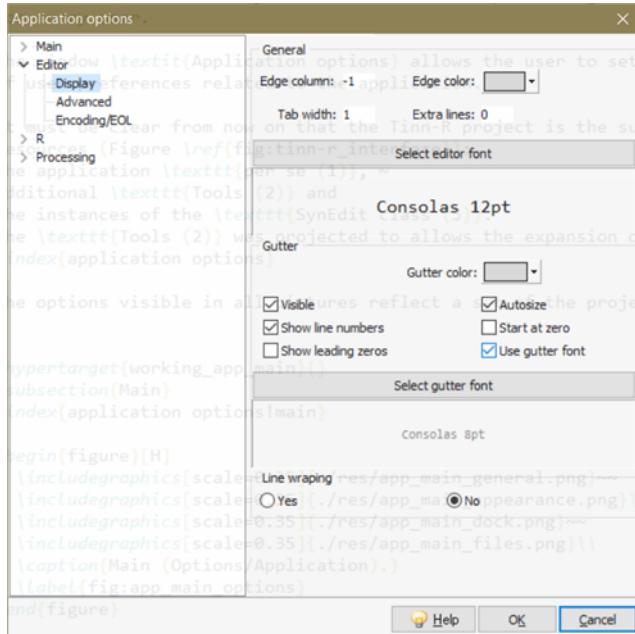
View

The Views/Dock contains a button called `Restore default`. When clicked, every time you start Tinn-R the Term and Tools windows will be at the default position. You usually do not want to mark it as you will lose your customization. It is useful if any problem occurs with the resources (dock-/hide and place).



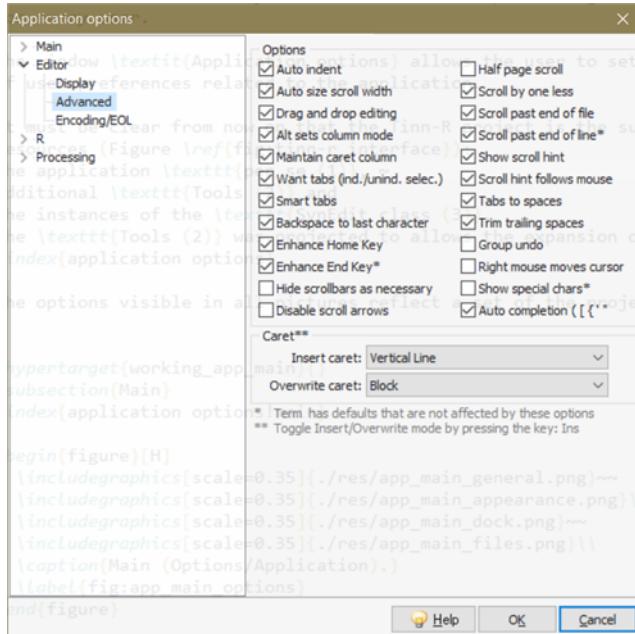
Editor options

The first window showing up is the Display. When first glancing at the editor window (where you type your scripts) you may have noticed a thin vertical line which is located at exactly 80 characters (default) from the beginning of each line. This is called the edge line. This line is very helpful to have a standard width for your texts, mainly when you increase or reduce the font size (CTRL + SHIFT + UP/DOWN) as you increase the font the edge line keeps on moving to the right, so that you can adapt the edge column to your preferred font size.



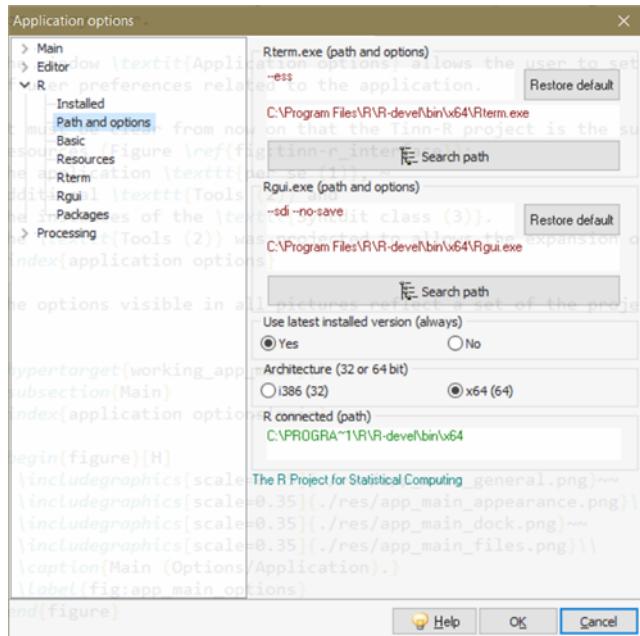
You may choose the font and the font size by clicking on font (Consolas 11pt is a nice option). Gutter is the space at extreme left of the editor, out of the text window, where you may number the lines of your script, just click at your choice.

For the other choice, Advanced options *see Editor options*.

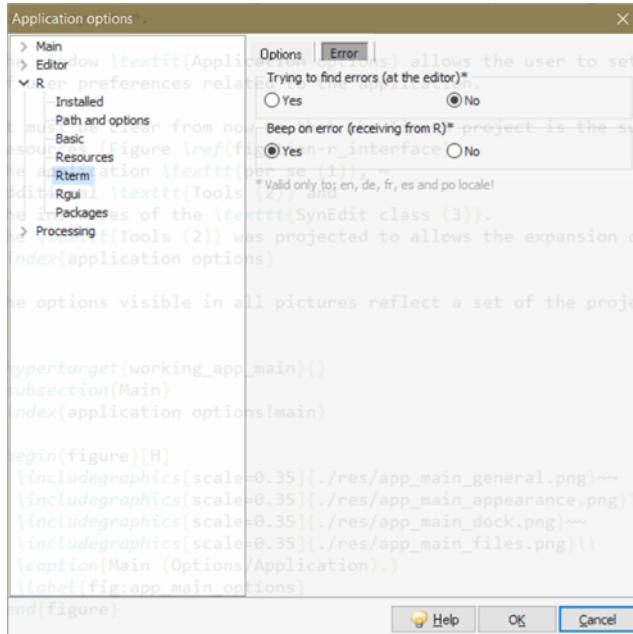


R

The first tab (Path) shows the paths to **Rterm.exe** or **Rgui.exe**. You can also choose whether Tinn-R gets the latest installed version of R or the version you would like to use. If you have Windows 7 64 bit choose that option located at the bottom of the window.

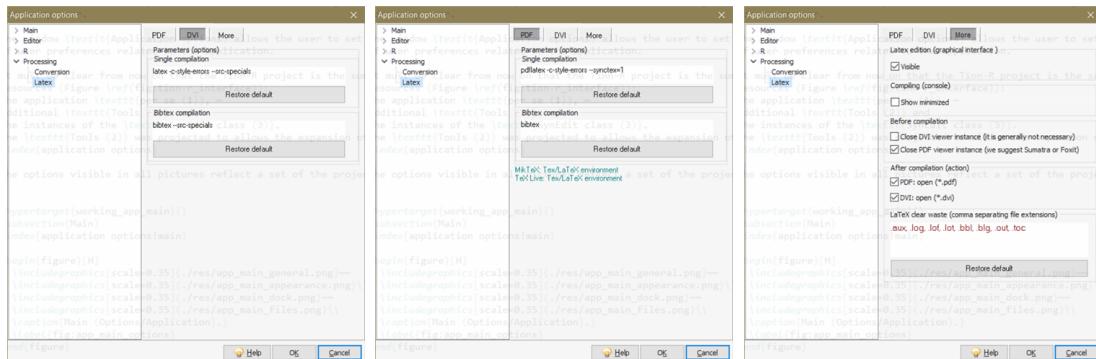


The next important tab is Term. The button Trying to find errors (at the editor), yes means that errors in R syntax are searched within the editor to find where the error shows up the first time within the script. You will usually have to keep on clicking the F3 key until you find the error position.



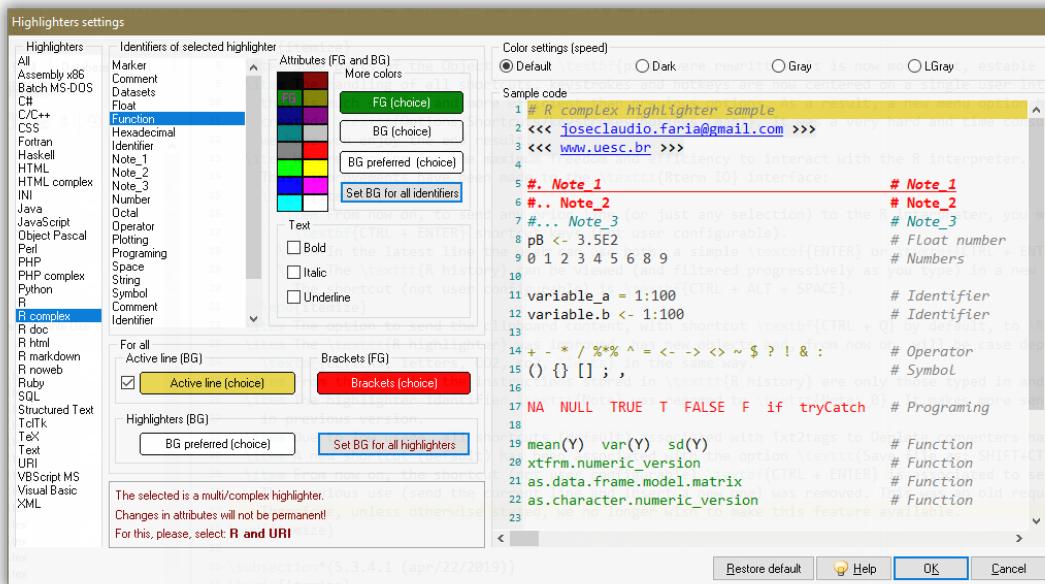
Processing

The tab **Processing** allows you to set preferences related to processing, namely text conversion and compilation. For example, click on **Latex/PDF** at **Viewer** (we suggest to use Foxit Reader or Sumatra). That, as it will be shown later, will enable you to compile L^AT_EX texts and open them in **.pdf** using Tinn-R. Do not forget to install the Miktex on your computer. If you prefer DVI instead of PDF click on the **DVI** button.



5.6 HIGHLIGHTERS (SETTINGS)

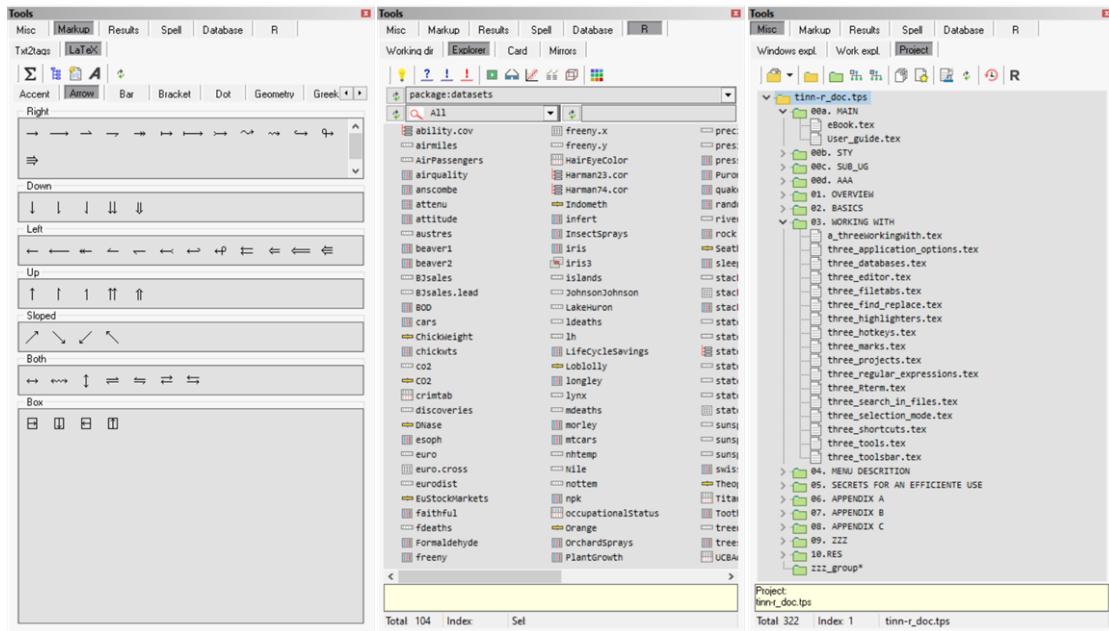
Highlighters settings is a window that allows you to customize the text you are working with in relation to appearance. Usually, you will be working with R, R doc, R noweb, Tex or Text files. The type of file is marked automatically at the left column of the window Highlighters. Within the Identifiers section of the selected highlighter you may choose the foreground (FG) color and the background color (BG) of each identifier. Since the best configuration of BG is the same for every identifier it is possible to set BG for all at once; use a dark color for BG.



The option Options/Syntax(highlighter) is used to set the highlighter of the type of file you would like, provided it is not the same as the one you are currently working on.

5.7 TOOLS

This interface was designed to allow the expansion of resources. The visibility of the window Tools can be changed using CTRL + F8. This interface contains many useful features that will make your life easier.



Project

Project allows you to organize different types of files in a hierarchical manner.

Even though your project can be found on the main menu bar, it is easier to work with it using the tools window. Press CTRL + F8 and the tools window will open. Click Misc/Project.

Beginning from the left, the first icon opens up a window where you can search for project files you want to work on. This file has an extension .tps. Otherwise, go to the next icon to open a new project.

When you click on that project a window will open, allowing you to choose the folder where the project will be stored.

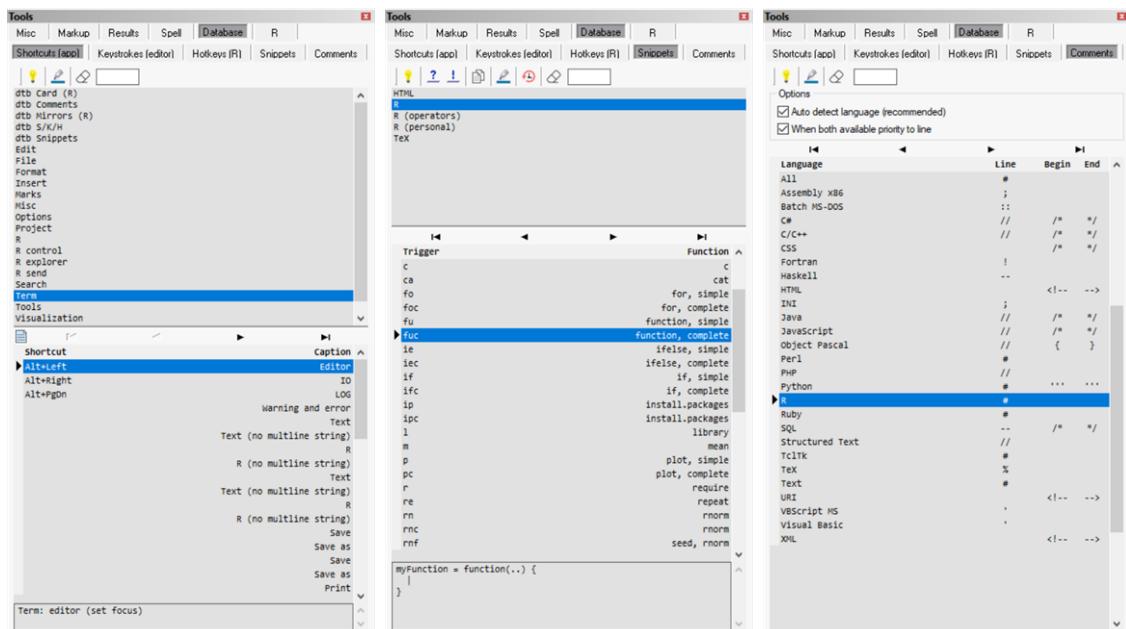
The tree is composed by groups connected at the node level. Groups will contain files with something in common. For example, if you are writing a package, one group might contain all data files, another the demo files, another the R files, and so on. The third box contains three icons. The first allows you to create new groups and also rename the deleted groups. The fourth box (sixth icon) allows you to open a project or its parts.

The next icon allows you to open a text file with the path corresponding to all files in the project. This functionality is useful when you move the whole project to another computer. The path on the other computer may be different from your computer, and so you have to change the corresponding paths. Save it and start working on the same project on that computer. You may use the search and replace options to perform that change in paths.

Database

Shortcuts

You then go to Tools/Database as you click on the box, a small window will open with five tabs. The first is Shortcuts. There is a long list of commands, some with a shortcut already configured as a default while other shortcuts can be configured by the user as needed. You may configure the shortcuts that you are used to. However, if they are already in use by the Windows operating system, they won't work. The use of shortcuts can improve the efficiency of your work. You can also have more than one shortcut table.



Snippets

Snippets will help you speed up the process of writing **anything from any language**. It allows you to personalize its database so that every function, script or text that you frequently use can be automatically inserted in your script through the click of a button or trigger.

Let us start by using snippets with the functions which are already in the default database. First, click on Tools/Database/Snippets. A window will open showing all the functions saved in the database at the bottom of the window. Now click on the fourth icon Snippets: edit. Another window will open, the group to which the function belongs being marked under Group, the function being marked under Function and the trigger under Trigger.

Now open a new file (CTRL + N) and write rn which is the trigger for the function rnorm and then press CTRL + J or click on Insert/Snippets at the main menu.

Now, suppose that you have a section of a script which you use very often when writing scripts. For example, imagine the following script: x <- rnorm(50); y <- rmnorm(50) which generates two pseudo-random normal vectors for x- and y-coordinates.

You then select and copy it to the clipboard, click again on the main menu at Tools/Database/Snippets and at the bottom of window click New. Now, at the top in Group type the name of the group, in this case let the name be Examples, it means that every other example could be saved under this group. In Function type the name of this specific function, for example, norm, and then type the trigger of this function you are just creating, say, nm and in complete paste what you copied. Finally click on Save at the bottom of the window. Now go to your text type nm and press CTRL + J, that part of the script will appear on your text. Did you get the idea?

Comments

The *Comments* resource is very simple and allows high level of user customization.

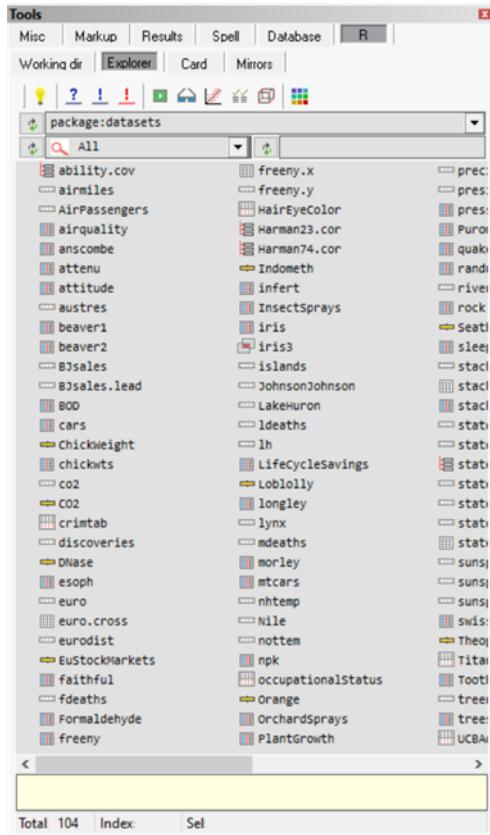
From version 3.0.1.0 Tinn-R automatically recognizes the language of the file on focus. Further, inside the file - if it is a syntax a multi-highlighter (complex syntax) - which language of the line where the cursor (or selection) is found.

This identification is done automatically if (and only if) the option (*x*) *Auto detect language (recomended)* is checked. Otherwise the user is forcing the application to use the comments of the selected language (indicator arrow).

Selected code snippets involving more than one language will not be commented/uncommented and a warning message is issued. That is, you must select only the snippet of a single language.

R

Explorer



This interface has its own pop-up menu, toolbar and three combo box. The pop-up menu and toolbar contain the most common actions related to an object explorer.

The button *R explorer: refresh environment* sends an instruction to R environment requesting the list of all loaded packages in the current session. The result is shown inside a graphical classified list. When one of these is selected, the graphical list (and structure) of the objects are shown.

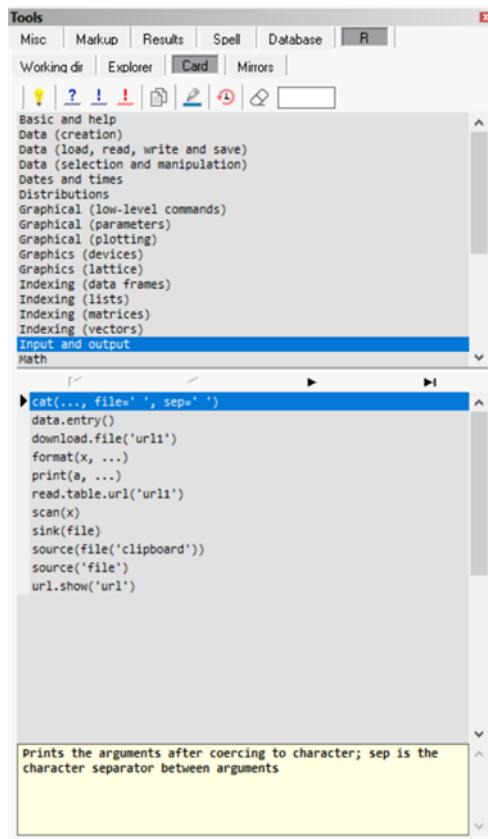
There are two options of filter: type of objects and any sequence of characters associated with the names of the objects.

It is possible to remove visible objects of the user workspace (.GlobalEnv) using the key *Delete*. To do this, select an object and type *Delete*.

A double click in any selected object will add its name to the editor. If the object is dragged to the editor interface, the textual description of the object is always shown in a new file. It is useful to know the sources of

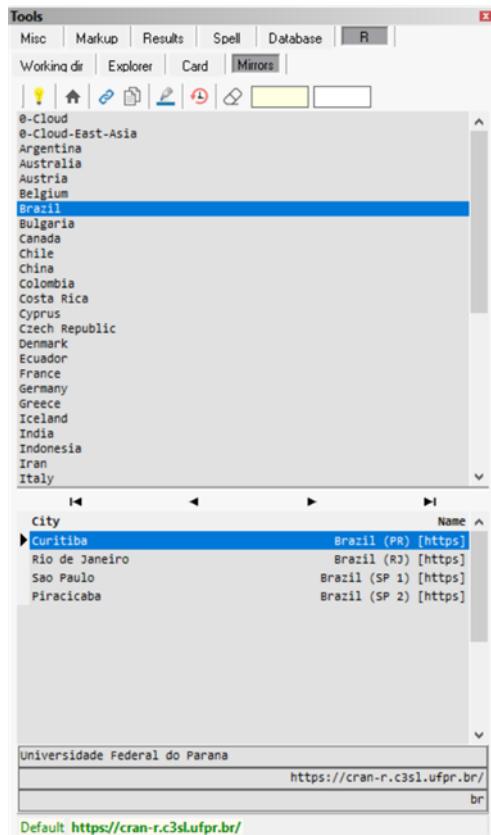
functions and to see data objects (vectors, frames, list, etc).

Card



The *card* is a database of distributions, mathematical functions, graphical and statistical functions used in R. The main goal of this card is to serve as a means of consultation for the most frequently used commands. A double mouse click will insert your selection in the text you are editing.

Mirrors



The *mirrors* is an interface that allows the user to manage the repositories (or mirrors) of R. You should always choose a repository physically closest to where you are, so that, the Web communication tends to be faster and more efficient.

The default mirror is the University [Wien](#) (Austria). Consider that this is the central mirror of CRAN.

The reasons for the Tinn-R always set a repository (assets) are two:

- To prevent R keeps asking which repository you want to use in each session;
- Workaround of intermittency (only related to `Rterm.exe`) not always showing the dialog for selecting the repository. That is, sometimes the dialog is displayed and not others. The cause of this intermittency is still unknown.

One important thing to be done is to set a R mirror as close as possible to where you work. For that, first click on CTRL + F8. This opens the Tools window, then click on R/Mirrors. Select the R mirror and push the button that shows an hourglass in the taskbar. The chosen repository will be the new default for all actions dependent repository (install packages, upgrade packages, etc).

R repositories are little changed with the passage of time. Thus, the database `Rmirros.xml` not need to be updated frequently. However, there are options that make the update whenever the user deems necessary:

- Button whose icon is a house on the taskbar Tools/R/Mirrors;
- Menu R/Update mirrors.

5.8 BARS

Tinn-R has three bars. The top task bar is related to the application and editor, the second task bar is focused on R, and the bottom bar displays the program status.



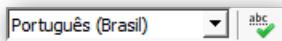
The top task bar (2) can have its location modified by dragging and dropping with your mouse. Right click anywhere on top task bar (2) to see the pop-up menu where you can choose which box of the bar you would like to show or hide. Every box of that bar has a vertical line at its left side. Right click at that location and you can move the bar up and down.

The R task bar (3) can be moved wherever you wish. Right click the vertical line at the left side of the bar and move it across either the editor or Term. Right click anywhere within the R task bar and choose between Send to R and Control R to have them displayed or hidden.

The status bar at the bottom of the editor window, shows the current line you are working on, the total number of lines, and the current column at the first box. It also shows the modes which can be either normal or read only. You are not allowed to make any changes in the read only mode. There is also information about selection mode. It allows you to change the way you select parts of the current working file. See [After installation](#).

5.9 SPELL CHECKER

See the steps to be followed in order to install a dictionary. After installing the dictionary, Tinn-R will automatically load it upon starting a new session. To perform a spell check press CTRL + F8 to open the tools window, and then click on Spell. Go to the editor task bar where it says ABC. The arrow to the left allows you to choose the language in which the text is written. Click on ABC. All corrections are automatically added to the text. After finishing this process, focus again on the editor and press CTRL + S to save all corrections.



5.10 SPLIT

Splitting the editor window is a very useful feature whenever you are working with long and complex programs. Go to the main bar, click on view/split. You may choose to split the window either horizontally or vertically. You will get the same text on two different and independent windows. Any change made in one window will be added automatically to the other and so to the text itself.

5.11 EDITOR: HOW TO?

Here we describe some tips.

Macros: how to use?

Tinn-R provides a useful macro option to help you with repetitive actions.



Click on Tools/Macro at the main menu bar. Then you will see two icons: Record F7 and Play F8. Those icons may also be found on the editor task bar. Here is an example: write a vector 1:20 and send it to R. In the Term window you will see the numbers from 1 to 20 in a single line. Copy and paste that to the editor's window.

Suppose that you want to write a comma after every number, maintaining a one space distance after the comma. Place the cursor just before the number one and then press F7. A + sign will appear on the Record icon at the editor taskbar, meaning that you are beginning to record action you take on the screen. Press the right arrow to move across the number 1 and type a comma (,), delete one space and move the cursor one space to the right with the right arrow and press again F7 to stop the recording process. The little + sign will disappear. Now just press F8 and see what happens. When you reach the number ten it won't work because now the numbers consists of two digits and the spacings will not fit. After doing that for the number nine the cursor will stop in the middle of the number ten, go back one step to the left with the left arrow, and press space bar so that there will be one blank space between the comma and the number ten. Now press again F7 to go across the number ten with the right arrow, type the comma (,) and go one space to the right with the right arrow so the cursor is now at the left side of the number eleven, press again F7, and then F8 until the end. Try now to write those numbers in a vertical line across the left margin without the commas using the macros again .

Marks: how to use?

One very useful navigation tool is the bookmark. To define the bookmark, use CTRL + SHIFT + [0-9] (a key from 0 to 9) on the line you have chosen. Then, to go to the corresponding bookmark just use CTRL + [0-9]. A visual indicator appears in the left margin, just before the line number, of

that line. Suppose you are working on a long script and need to constantly return to a specified line. You will see how handy it is. To undo it, just use again **CTRL + SHIFT + [0-9]** using the same number, on the same line. You may also mark a whole block. To do that, first select the block and then click on Marks at the main menu and click on mark. As in the above paragraph the number 0 will appear at the beginning of the block and the number 1 at the end. You can modify the box and send it to R to be processed.

Editor, Tools and Term: how to arrange?

CTRL + F8 toggles the visibility of window Tools and **CTRL + F9** toggles the visibility of window Term. Now press **CTRL + F10**, **CTRL + F11** or **CTRL + F12** and see what happens.

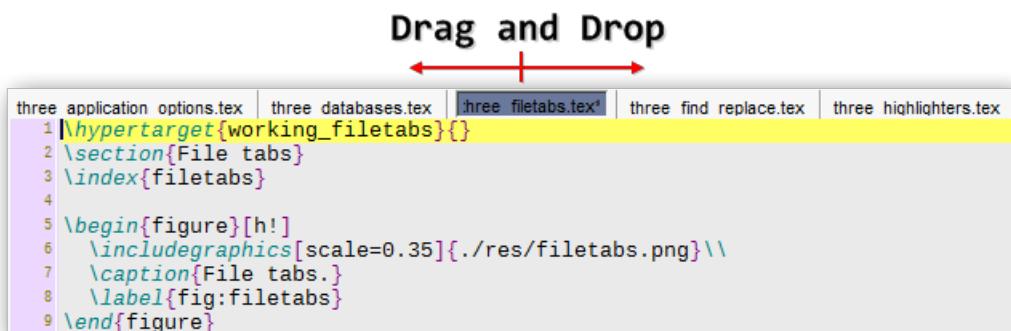
The use of two monitors is recommended in case of intensive use. Placing R (Rgui.exe or Rterm.exe) on a second monitor makes working with data analysis and development tasks very comfortable.

Active page using the keyboard: how to change?

CTRL + TAB	: Change sequentially the active page to the right (requires more than one)
CTRL + SHIFT + TAB	: Change sequentially the active page to the left (requires more than one)

Tab order: how to change?

Drag and drop the tab to the left or right.



Auto completion: how to use?

In principle it just opens and closes, with just one click, parentheses, square brackets, brackets and quotation marks, i.e.: ([{ ' ". You just click, say, the open parentheses and the editor automatically completes the closing parentheses and after that puts the cursor in the middle of the two.

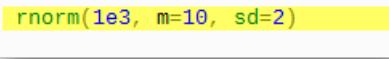
However, it is not only this. If already exists a previous text doing just that might be a bit laborious, since you have to delete one of the two elements (the subsequent to the cursor), to move the cursor to the end of the text and then to close the element (parentheses, square brackets, brackets or quotation marks). There is an alternative for this: select the previous text and type the open element, i.e: ([{ ' ". It is all!

One of the most important feature of this resource is the following: you just write the expression you want to complete, i.e., ([{ ' " then you select the whole expression and press the element key. *The two elements will automatically be placed at the beginning and the end of the selection and the entire selection will be preserved.*

Important note: This can be done in both, the editor as well as in the Term window.

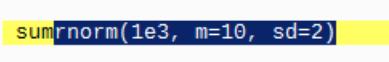
A single and illustrative example:

- A previous text:



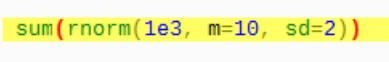
```
rnorm(1e3, m=10, sd=2)
```

- type `sum` and after to select the text:



```
sumrnorm(1e3, m=10, sd=2)
```

- type the open parentheses:



```
sum(rnorm(1e3, m=10, sd=2))
```

A more complex and illustrative example:

- A previous text:

```
Learning completion
```

- select it:

```
Learning completion
```

- type the quotation mark:

```
"Learning completion"
```

- type cat:

```
cat "Learning completion"
```

- select the text:

```
cat "Learning completion"
```

- type the open parentheses:

```
cat ("Learning completion")
```

APPENDIX A

WHAT IS NEW?

This topic provides information about what is new in Tinn-R.

Year	Versions released	Cumulative sum
2024	01	238
2023	05	237
2022	02	231
2021	15	229
2020	05	214
2019	04	209
2018	06	205
2016	01	198
2015	05	197
2014	08	192
2013	10	184
2012	03	174
2010	10	171
2009	16	161
2008	14	145
2007	26	131
2006	64	105
2005	25	041
2004	16	016

A.1 VERSIONS RELEASED IN 2024 (01)

9.0.1.1 (*Mar/31/2024*)

Bug(s) fixed

- Possible failures in sending lines, when many lines - one by one - are sent to the Term interface (with the "|+" option checked) has been fixed. In other words, sending contiguous lines sequentially to the Term interface has been improved and is now more consistent. But you shouldn't abuse this feature either. Do not continually rest your finger on the shortcut key or hotkey defined for this action to send large quantities of lines, as there are better options for this purpose, with lower computational cost.
- A bug associated with the Tools/R/Mirrors/Mirrors(R) update button has been fixed. The issues that emerged with the new version of R (4.2.0 - UTF-8).
- Bug in changing the position of tabs of the Main Interface, Options, Tools and Term has been fixed. From now on, all of them must only be in the top position.
- Fixed some Term interface bugs with the knitr package. The bugs arose due to changes in the sending patterns of R messages under the package, mainly if the progress bar option is checked (quiet=FALSE, which is the default). I hope the authors don't make drastic changes to the parsing progress messages to invalidate the regular expressions made to match the Term interface to the messages in the Knitr package.
- Bug associated to code completion has been fixed.
- Format XML, recognizes: .xml and .xsd.
- Bug in filtering in the Tools/Database/Hotkeys interface has been fixed.

News

- Development was quite intense and uninterrupted throughout 2023. Only beta (or test) versions were released for a limited user base. The changes to the beta versions that occurred throughout the year were documented - collectively - only in this version.
- The user guide has been updated and improved.

- This is the latest compiled version of the project developed and compiled in the Delphi 2007 IDE. The project is in an advanced stage of migration to Embarcadero's new Delphi 11.3 (Alexandria) Community platform.

Interface

- A new tab was created in Tools/R called `Working dir`. The objective is to allow a graphical visualization of the active R folder. It also allows storing favorites and indicating the work folder to R with several interesting options.
- OpenURL is a new option (in the File menu and taskbar) allows opening remote files in the editor.
- New options in the Options/Application interface make it possible to discover the program's reserved shortcuts. They cannot be changed by the user. There are few, but very important to know.
- Some buttons (somewhat redundant) were removed from the Term interface. It has become visually simpler.
- The Tools interface has been substantially improved.
- A new interface button Tools/Windows expl. and Tools/Work expl. allow physical deletion of displayed files.
- A small window was added in the Tools/R explorer interface that informs the description (.Rd) of each selected object.
- New (and useful) functions in the button (and corresponding menu) R control: packages.
- The indentation of the Term/I0 interface has been improved. It was easier to read the submitted code due to the preservation of the script's original indentation.
- The R Explorer interface was improved.
- The S/K/H manager interface received updates and improvements.
- Change (hopefully for the better) in the alignment of texts in the Tools interface.
- Tools/Completion has been replaced by Tools/Snippets. The Completion.xml table is now named Snippets.xml. Users who have made (and saved) an extensive dataset of completion (before) (snippets from now on) will be able to notice a new button [Merge (file .xml)] that has been added in the new Snippets dialog window.

When a new dataset (or any update to the source) is made, the user's dataset (`Completion.xml` or `Snippet.xml`, respectively) is saved in the **bkp** folder which is located in the Tinn-R startup folders (`Help/Ini files (path information)`). The new button allows the user's old base (completion or snippets) to be merged with the current one distributed with the new versions and eventually modified at the source. Thanks to Ivan B. Allaman for the helpful suggestion.

- Replacement and addition of many icons has been made.
- Filters can be applied to virtually all database-related features in the Tools interface:
 - Database: Shortcuts (app), Keystrokes (editor), Hotkeys (R), Snippets, Comments, Card and Mirrors.
- Possibility of controlling the transparency of main and secondary windows independently.
- The Options interface has been improved.
- New resizing options in the Term/LOG interface.
- The Web menu item received updates and improvements.
- New options in the popup menu associated with projects allow sending files Rnoweb (Rnw, Rmd and RHtml) (via Sweave and knit) directly to be interpreted, without being open.

Code

- Many efforts have been made in improving, standardizing and optimizing the source code.
- Tip and Snippets received updates and improvements.
- Dynamic updating of R mirrors has been improved.

Packages

- The `TinnRcom` package has been improved and has new functions. These functions are associated with searching for information in the documentation (.Rd) of objects.
- Updated dependencies of package `TinnRcom`.

New features

- Built a (new) search interface for Snippets triggers: thanks to Ivan B. Allaman for the suggestion.

- A (new) search interface was built with information linked to the documentation (.Rd) of the objects.
- Implemented search and replace in the Term/I0 and Term/L0G interface.
- The Snippets (and respective search interface) are also functional in the Term interface.

Database

- Practically all database tables were updated.

Changes to shortcuts

- Shortcuts (app)
 - Ctrl + L: In the Term interface clears the I0 and L0G windows
 - Alt + Space: Find again
- The default hotkeys for R have been changed.

A.2 VERSIONS RELEASED IN 2023 (05)

9.0.0.1 (2023)

Development was quite intense and uninterrupted throughout 2023. Only beta (or test) versions were released for a limited user base. The changes to the beta versions that occurred throughout the year were documented - collectively - only in the first version released (to everyone) in 2024.

A.3 VERSIONS RELEASED IN 2022 (02)

8.2.2.1 (*jul/06/2022*)

- The Web menu item received updates and improvements.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.2.1.1 (*jul/02/2022*)

- Bug(s) fixed:
 - The issues that emerged with the new version of R (4.2.0 - UTF-8) have been fixed. Because the Tinn-R project supports both old and new versions of R, **from now on Rgui.exe must be launched from within Tinn-R so that it is properly identified at startup.**
 - A bug associated to MRO, the Rterm interface and the `delete_checkpoint()` function of the package `checkpoint` was fixed.
- Increase (CTRL + Mouse Reel UP) and decrease (CTRL + Mouse Reel DOWN) of editor font and Rterm via CTRL + Mouse Reel.
- Almost all of the interpreter's control options were enabled when there is no open file. In this case searches for words or selections will be done in the instance of SynEdit Rterm/IO. This allows greater freedom and interactivity of the Rterm interface.
- Returning focus from other applications to Tinn-R (Editor or Rterm) has been improved.
- Some default R Hotkeys have been changed and some new ones have been defined.
- In the project interface double clicking opens only individual files, it has been disabled for groups and entire project. For entire project and groups the drag option remains functional.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

A.4 VERSIONS RELEASED IN 2021 (15)

8.1.3.5 (*jun/22/2021*)

- Bug(s) fixed:
 - A bug associated to the completion resource with content longer than one line (eg: `ma`, `iec`, etc) has been fixed. This bug was accidentally introduced in previous version 8.1.3.4.
 - A bug associated to the interface `Tools/R/Explorer` and the package `dplyr` was fixed.
 - A bug associated to error message `baseenv()` from R flavors MRO and R-devel was fixed. This bug was associated to the package `svMisc`. Thanks to SciViews Team.
- The toolbar of the `Tools/Project` interface has been redesigned.
- The `Tools/R/Explorer` interface has the default option, from now on, not to display all objects. That is, hidden objects from the environment (`.object`) are not shown.
- A new updated version (1.0.22) of the `TinnRcom` package is being distributed with this version of the project.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.1.3.4 (*jun/01/2021*)

- The project's old web page (<https://nbcgib.uesc.br/tinnr/en/>) has been permanently removed.
- The demo project files were updated.
- The files in the `utils` folder have been updated. As a result, the versions of packages that the `TinnRcom` package depends on are checked at R startup and will be updated whenever necessary.
- Balthazar Mattos Farnese started working on the project as a contributor responsible for the project's web page and email server. Welcome Balthazar.
- Some improvements and refinements have been made in the previous version 8.1.3.3.

- Automatic indentation (after carriage return) during editing has been considerably improved in this version; it was hard work. If you notice any inconsistencies, please prepare a small sample file and send detailed information to be corrected.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.1.3.3 (mai/10/2021)

- Improvements were made to the Tools/Markup/LaTeX interface.
- Improvements have been made to the demo project.
- Improvements have been made to the recently implemented features that enable you to mark a ~~TEX~~ file (.tex) as the main (root), both in the main interface and in the projects.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.1.3.2 (mai/09/2021)

- Bug(s) fixed:
 - Bugs associated with some templates were fixed.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.1.3.1 (mai/08/2021)

- Bug(s) fixed:
 - A bug associated with automatic updating (menu/Help/Check for update) was fixed. The origin of the bug was the exchange of the NBCGIB project server for Digital Ocean and the differentiated security standards.

- A bug associated with the position of the code completion (or tips) window when two monitors are used has been fixed.
- All dialogs for choosing folders (Updater, R choose dir, etc.) have been improved due to the use of the most modern Windows API.
- The elements of the Tools/LaTeX interface have been updated and improved. The `LaTeX_symbols.tex` file was added to the demo project and the corresponding `LaTeX_symbols.pdf` file was added to the `doc` folder where the application was installed.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.1.2.10 (mai/03/2021)

- A new option has been added that enables the user **to sets the main L^AT_EX file**. Thus, he can be editing in any file that, when compiling (generate the PDF or DVI), the instruction will be sent correctly for the compiler. This new feature speeds up the process of creating files (PDF and DVI) as the user does not need to navigate to the main file each time to submit to the compilation. The feature is functional for all options associated with compilation (.pdf, .dvi, makeindex and bibitex). This resource is volatile with each session and project, that is, it is not stored with the project information.
- When there are two main file indications for L^AT_EX (one in the **project** and the other in the **main interface** for any file), this option, from within the project, takes precedence, that is, it is mandatory.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.1.2.9 (mai/02/2021)

- Tinn-R is enabled (for many years) to do reverse search in DVI and PDF. However, only the reverse search in DVI was documented, as it had been used in large projects (books) written in L^AT_EX. Recently working on a book directly from L^AT_EX to PDF, the feature was tested and documented in the user guide. The feature is fully functional

as long as the correct directive is used: **pdflatex -c-style-errors –synctex=1**.

- A new project-related option has been added that enables the user **to sets the main L^AT_EX project file**. Thus, he can be editing in any project file that, when compiling (generate the PDF or DVI), the instruction will be sent correctly for the compiler. This new feature speeds up the process of creating files (PDF and DVI) as the user does not need to navigate to the main file each time to submit the project to the compilation. For projects with many files, this feature (already present in more specialized L^AT_EX editors) makes work much easier. The feature is functional for all options associated with compilation (.pdf, .dvi, makeindex and bibitex). This resource is volatile with each session and project, that is, it is not stored with the project information.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.1.2.8 (apr/28/2021)

- Bug(s) fixed:
 - Version 8.1.2.7, due to the detection of two serious bugs, accidentally inserted in the project's source code, has been removed from all project repositories. The bugs have been fixed.
It is strongly recommended not to use version 8.1.2.7!
- Some improvements have been made to some program windows.
- Automatic indentation (after carriage return) during editing has been considerably improved in this version; it was hard work. If you notice any inconsistencies, please prepare a small sample file and send detailed information to be corrected.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.1.2.7 (apr/26/2021) - *Removed from all project repositories*

- Some improvements have been made to some program windows.
- Some default database were updated (`Rmirrors.xml`, `Completion.xml`).
- Trying to circumvent the technological delay of the Brazilian economy related to the difficulties of transferring values from abroad to Brazil, the project now receives donations via cryptocurrencies: **Bitcoin** and **Bitcoin Cash**. In the main menu Help/About/Donation, the accounts for receiving donations (transfers) can be copied (CTRL + C) and pasted (CTRL + V) at the destination.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.1.2.6 (apr/23/2021)

- The main menu main menu: R/Control/Packages/Install has been changed, it now installs the selected package in the editor (or Rterm interface).
- A new button has been created on the Rterm interface task bar that enables the user to send or resend any line or selection. This feature (using CTRL + ENTER) has been available for a long time on the Rterm interface, but not in the form of a button. We believe that the button makes it easier to use for new users. Rterm: send or resend (any line or selection) CTRL + ENTER.
- Some adjustments were made to the license request mechanism.
- Some improvements have been made to some program windows.
- The license verification and validation procedure is still disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.1.2.5 (apr/21/2021)

- Bug(s) fixed:
 - Bugs related to the assignment of some Hotkeys (R), which required restarting Tinn-R to function, have been fixed.

- From that version on, the Tinn-R program has version control. The Tinn-R project remains free to use for the educational sector at any level. For these, request a free license (renewed annually). See (`main menu: Help/About/Licensing`) and (`main menu: Help/License manager`). The license verification and validation procedure is disabled in this version. So, if you want, you can request a license to help improve the mechanism, as it will be being tested on more diverse hardware and software platforms than those used by testers. In the future version, the validation mechanism will be enabled.

8.1.2.4 (apr/18/2021) - Restrict to testers

- New options to get and set the R's working directory.
- Improvement of the mechanism for locating particular (simple and complex) words in the R environment. The new mechanism is heavily based on regular expressions, so unexpected things can happen in unforeseen cases as well as in long comments on lines that contain the desired word (associated with a data function). If something unexpected happens: select the desired word and report the error to the developers.

8.1.2.3 (mar/26/2021) - Restrict to testers

- Bug(s) fixed:
 - Check for update procedure. The bug was due to the change of the project server and security rules of the communication protocol.

8.1.2.2 (mar/25/2021) - Restrict to testers

- Some improvements and refinements have been made in the previous version 8.1.2.1.

8.1.2.1 (jan/01/2021) - Restrict to testers

- Bug(s) fixed:
 - Black streaks on the main interface when the application is closed maximized. Not yet it was possible to find the source of the bug, so to work around the problem, if it is maximized when closed, the program will automatically leave the maximized logic state. Therefore, when reopened, the main window will be displayed in the stable size and position before being maximized and closed.

- Tools/Markup/LaTeX: the taskbar buttons were missing their options. The bug has been fixed.
- Implementation of version control. The Tinn-R project remains free to use for the educational sector at any level. For these, request a free license (renewed annually). See (`main menu: Help/About/Licensing`) and (`main menu: Help/License manager`).

A.5 VERSIONS RELEASED IN 2020 (05)

6.1.1.6 (*mai/09/2020*)

- Bug(s) fixed:
 - All dependencies of the TinnRcom package that are distributed with the Tinn-R setup (formatR, svMisc and svSocket), as well as the TinnRcom package itself, have been rebuilt with the R-devel version (4.1). The startup issues that arose in the current version 4.0 of R have been fixed.
 - A bug associated with the package `semPlot` was fixed. Thanks to Frank for pointing it out.
- Filter of Tools/R/Mirros from now on will be applied by Host.

6.1.1.5 (*jan/06/2020*)

- Some improvements and refinements have been made in the previous version 6.1.1.4.

6.1.1.4 (*jan/02/2020*)

- Many improvements and refinements have been made in the previous version 6.1.1.3.

6.1.1.3 (*jan/02/2020*)

- Some parts of the Object Pascal **pipe** were rewritten. It is now faster, stable and robust.
- The handling of all shortcuts, keystrokes and hotkeys are now centered on a single user interface that is simpler and more efficient than previous options. As result, a new menu option has been created: Options/Shortcuts/keystrokes/hotkeys (map). It was a very hard and time consuming work, we hope you enjoy the final results.
- We are looking for the maximum freedom and efficiency to interact with the R interpreter. Thus, improvements have been made to the **Rterm** IO interface:
 - From now on, to send any prior line (or just any selection) to the R interpreter, you must press the **CTRL + ENTER** shortcut keys (not configurable).
 - In the latest line the user can use: a simple **ENTER** or **CTRL + ENTER** to send the current line.

- The R history can be visualized (and filtered progressively as you type) in a new SynEdit completion window. The shortcut is **CTRL + ALT + SPACE** (not configurable).
- From now on, the shortcut **CTRL + ENTER** (not configurable) is also associated to send line (or selection) from the Editor to the R interpreter. The previous use (send the current line and insert a new one) was removed. This was an old request from users but we believe it is no longer used much. Therefore, unless otherwise stated, we don't wish to make this feature available anymore.
- The option to send the clipboard content, with shortcut **CTRL + Q** by default, to R interpreter was removed.
- The R highlighter was improved, it has new objects and from now on, will be case sensitive. That is, it will highlight, for example, **LETTERS, letters, CO2, co2, ...**, recognized and different R objects, in the same group with the same attributes.
- From this version, the instructions stored in R history are only those typed in and submitted directly into Rterm IO interface.
- The highlighter identifier Note was renamed to Note_0. It makes more sense, since Note_1 and Note_2 were created in previous version.
- Due to low usage, all shortcuts (default) associated with Txt2tags to Deplate converters have been removed.
- A new shortcut (default) has been associated with the option Save file as: SHIFT+CTRL+S.
- The interface **Highlighters** setting was improved. From now on, the program will be distributed with at least 4 fast coloring sets: Default, Dark, Gray and LGray. Additionally, the interface allows you to restore all default values with a new option.

A.6 VERSIONS RELEASED IN 2019 (04)

5.3.4.1 (*apr/22/2019*)

- Two new identifiers have been added to highlighters **R** and **Text**: **Note_1** and **Note_2**. They are useful to be used as differentiated marks (now in three levels) in scripts and texts.

5.3.3.1 (*fev/25/2019*)

- Bug(s) fixed:
 - A bug associated with the procedure **Help/Check for update** and the prior 5.3.2.1 (fev/12/2019) portable version was fixed.

5.3.2.1 (*fev/12/2019*)

- The recognition of complex R object names for **print**, **plot**, **names** **structure**, **edit**, **fix**, **help**, **example** and **open example** were improved. For example, for all cases:
 - **iris**
 - **iris[1]**
 - **iris[c(1, 2)]**
 - **iris3(1, 1, 3)**
 - **iris\$Sepal.Length**
 - **iris['Sepal.Length']**
 - **iris[["Sepal.Length"]]**

that it will be correctly recognized.

- The **Replace** procedure was improved. From hereafter, it will remember, for all files, the same word list used for replacement.
- The **Tinn-R web site** was updated, the new URL is: <https://tinn-r.org/en/>.

5.3.1.1 (*jan/06/2019*)

- The status bar was redesigned.
- The option **R send: line** has a new feature that allows the main editor to send sequential instructions while the interpreter returns the signal +(plus) after each incomplete instruction. This option can be fast and easily switched in the status bar.

- A new option to send instructions to R interpreter is being implemented: `R send: smart`. The goal is to simplify it for beginner users to send complete instruction blocks when the cursor is located in a complex context: out of the first line of functions, instructions with several contiguous lines and similar situations. The work is underway, but not yet fully done. Because of that, this feature is disabled. If the user try to use/enable it, he/she will get the message: *This feature is still in development.*
- The interface Hotkeys (operational system) was simplified and improved. Additionally, the number of customizable user actions increased to 20. On demand, this number can be easily increased. Due to these changes, the previous hotkeys will be lost, recreating them from the new GUI is necessary.
- Excepting the source files of Project Description (`Tinn_R.dpr` and `Tinn-R_portable.dpr`), all source code files were unified. Therefore, from now on it became easier to compile and distribute the setup and portable `Tinn-R_X.X.X_X_setup.exe` and `Tinn-R_X.X.X_X_portable.zip` versions, respectively. The custom support in PortableApps standards was completely removed.
- This version allows using some special characters (for example: `.aa`, `-bb`, `_cc`, `dd`, `?ee!ff` and `@gg`) as trigger to data completion. All triggers of the new HTML group begin with '`<`' character. It will give more flexibility to create completion to other languages.
- The mechanism of completion was enhanced a bit. From now on it will be more user-friendly regarding to the indentation of the text in all supported languages.
- Corrections were made in the popup menu interfaces `Rterm` `I0` and `LOG`.
- The option to donate to the project has changed. Hereafter, it will be a bank account in the name of the project leader. **Please, do not use the PayPal account anymore.**

A.7 VERSIONS RELEASED IN 2018 (06)

5.2.2.0 (*oct/04/2018*)

- Parts of the source code were enhanced.
- Some window and dialogs were reworked.
- The recognition of complex R object names for `search` and `search in files` were improved.
- Some keys from `INI file` were renamed to be more reliable:
 - [ROptions]→[R Options]
 - [Options TextDiff]→[Diff Options]
 - [PandocHistory]→[Pandoc History]
 - [PandocHistoryFrom]→[Pandoc History From]
 - [PandocHistoryTo]→[Pandoc History To]
 - [SearchTextHistory]→[Search Text History]
 - [SearchDirHistory]→[Search Dir History]
 - [SearchFileMaskHistory]→[Search File Mask History]
 - [ExplorerFavorites]→[Explorer Favorites]
 - [FileMRU]→[File MRU]
 - [ProjectMRU]→[Project MRU]

Therefore, some user preferences can be lost (when applicable, it will use the default values). Sorry for the inconvenience.

5.2.1.0 (*sep/28/2018*)

- The main menu `Web/R(search)` were reworked.
- The recognition of complex R object names for `send` `instructions` were improved.

5.1.6.0 (*sep/26/2018*)

- The `Application options` interface was reworked.

5.1.5.0 (*sep/20/2018*)

- Bug(s) fixed:
 - A bug from the versions 5.1.3.0 and 5.1.4.0 (*sep/20/2018*) related to the demo project was fixed.

5.1.4.0 (*sep/20/2018*)

- Bug(s) fixed:
 - A bug from version 5.1.3.0 (*sep/20/2018*) related to R scripts `utils/infor.R` and `install.R` was fixed.

5.1.3.0 (*sep/20/2018*)

- Bug(s) fixed:
 - A bug related to sending instructions when `Rgui.exe` was started out of Tinn-R was fixed. Thanks to Allan Stewart-Oaten for pointing it out.
- A new button was added to the Tools/R/Card interface to open an example of the selected item in the editor.
- New resources were added to let Tinn-R access R remotely, via PuTTY. The way of working is the same as using `Rgui.exe` locally, but there are a few limitations.
- The interface Options/Application/R were reworked, and new options were added.
- The searching algorithm from the installers of R in the System Registry were remade. The problems issued by the non standardization after the installation of Microsoft R (MRAN), which were hindering the correct identification of the installed versions, were re-worked.

A.8 VERSIONS RELEASED IN 2016 (01)

5.1.2.0 (*oct/15/2016*)

- Bug(s) fixed:
 - Two major bugs introduced in the pre-release version 4.2.6.0 have been fixed.
 - A bug related to Help, Example and Open example when all files were closed was fixed.
 - A bug related to Rterm interface (IO) and the package debug was fixed. Thanks to Ivan B. Allaman for pointing it out.
 - A bug related to the custom color storage was fixed.
 - A bug related to the Options/Syntax (highlighter)/Default (to new files) was fixed.
 - A bug related to the Options/Syntax (highlighter) was fixed. Thanks to Manuela Huso for pointing it out.
 - Some issues about the completion followed by F3 to find the next occurrence of | were corrected.
 - Issues about the installation of TinnRcom package (and its dependencies) were corrected.
- A lot of PRE-RELEASE versions of the project have been released, not restricted to testers. Thanks for testing and making suggestions.
- The installation of TinnRcom package (and its dependencies) was improved. From now on, the sources (.tar.gz) and the binaries (.zip) of the packages formatR, svMisc, svSocket and TinnRcom will be released within the Tinn-R setup. After the installation, TinnRcom package will be placed at packages folder. Thus all of them will be installed from the local .zip.
- The TinnRcom package was upgraded to version 1.0.20.
- The menu options has a new option: R echo (on/off). This grants the user the choice to echo (or not) some options of send file, selection, clipboard, block marked, contiguous lines a line to end of page. A related button was added to R task bar. This made these options simpler and user-friendly. The default shortcut to toggle this option is ALT + E
- The procedure to open a remote file from an URL was improved and covers https protocol.

- If the user's choice in Options/Application/R/Patch (R) is **No** to Use the latest installed version (always) option, at startup, Tinn-R will search all letters of the system drives for the fully informed path of R. This is very useful in the portable flavours due to letters changes in different computers.
- The visibility of all Page Control and Tab Sheets caption was improved.
- R will always start with two options:
 - `options(pkgType='binary')`
 - `options(install.packages.check.source='no')`
- If the user do not have a personal library to manage the packages, from this version on, Tinn-R will create a folder named `x.y`, related to major and minor R version in the `C:/Users/User/Documents/R/win-library/`. This library will be used as default to package manager. The user can change the default library at Options/Application/R/- Packages (R) in the Library trees (`.libPaths()`).
- The default files `Shortcuts.xml` and `Mirrors.xml` were updated.
- The default shortcuts related to focus on Rterm interface (Editor, IO and LOG) were changed due to conflict with Windows 10.
- When using Rterm (except for the Send Fileoption), all options related to more than one line will be added to the R history.
- The source code related to identification of library trees `.libPaths()` was improved.
- The R explorer interface related to the identification of environments from objects was improved.
- The resources related to update `mirrors` in runtime were improved.
- The resources related to knitr and Sweave were improved and new options are in Options/Application/R/Packages/Knitr.
- The argument `quiet=TRUE` it will be added to all knit procedures. So, if you want more control, or for big documents, (for while) the use of the knit with `Rgui.exe` instead of `Rterm.exe` is suggested.
- The menu Insert has a new option: R (assignment). This option makes flexible to insert <- and -> assignment. As it is associated with shortcuts, the user can customize both assignments.
- The navigator of all database was replaced.

- The `I0` prompt of the `Rterm` interface is more user friendly. For this `setOptions/Application/Editor/Advanced/Scroll pas end of line option.`
- A new item was add to the main menu: `Tools/Processing/Viewer/Open current file (generic).`
- The menu `Web` was heavily reworked and has new options.
- The windows `Options/Application` were improved.
- A new resource allowing the user to `open/learn/edit` the example script of any R objet, from the `editor`, `Rterm/I0`, `Rterm/LOG` or window `Tools/R/explorer` was added.
- The `tip` and `data completion` were deeply improved. `CTRL + SPACE` is the single trigger for both and the distinction is made in the context of the call. Additionally this feature now find the object, the package and the pattern in complex multiline contexts.
- The shortcuts `CTRL + TAB` and `SHIFT + CTRL + TAB`, not user configurable, can now be used in all interfaces `Editor`, `Rterm` and `Help` to sequentially switch the pages.
- This version is full compatible with `Microsoft R Open - MRO`.
- The folder `utils` was deeply restructured an instruction `unlockBinding("last.warning", baseenv())` was added to the new file `info.R` to workaround a small bug between `svSocket` and `MRO`. Thanks to `Marc Laurencelle` for pointing it out.
- The identifier `Note` was brought back to the highlighter of R family, and due to this utility, it was also added to Text highlighter. Thanks to `Ari` for pointing it out.
- Improvements related to identification of `strings` were made to `R family`, `All` and `Text` highlighters. So, they do not identify the shortened forms (he's, you'd, you've, etc.) as string.
- New options related to `Rterm`: `I0` and `LOG` highlighters.
- The usability of `Rterm` interface is now more robust, stable and user friendly.
- A new resource `Check for update` was added to the main menu `Help`.
- The development team would like to publicly thank `Marc Laurencelle` for the contributions and suggestions to Tinn-R project.

- Tinn-R Team has two new members: Philiphe A. Kramer and Swami de P. Lima, welcome!

A.9 VERSIONS RELEASED IN 2015 (05)

4.0.3.5 (*jun/22/2015*)

- Due to two portable versions (simple and compatible with Apps) some adjustments were made in the application source code and the project structure.

4.0.3.4 (*jun/11/2015*)

- Bug(s) fixed:
 - A bug related to `Rterm` interface (`IO` and `LOG`) and to the packages `car` and `rms` was fixed. The bug occurs because the pattern of messages on the pipe changes when both packages are loaded. Thanks to Frank for pointing it out.
 - A bug related to `Update R mirrors` and to the countries Germany, Spain and Sweden was fixed.
- The versions 4.0.3.1, 4.0.3.2 and 4.0.3.3 were restricted to testers: thanks for the tests and suggestions.
- From this version on, Tinn-R will be released in three flavors: Tinn-R, Tinn-R Portable and compatible with PortableApps platform.
- Some files of the folder `utils` were updated to meet the Tinn-R Portable project needs. Therefore, the folder and paths of the R variable `.trPaths` were changed to TEMP environment variable.
- The `Help` menu was slightly changed.
- The `TinnRcom` package was updated to version 1.0.18.

4.0.2.1 (*Apr/29/2015*)

- Bug(s) fixed:
 - A bug related to the recent release (R 3.2.0) and to the installation of the necessary `TinnRcom` package was fixed. Thanks to Duncan Murdoch for the support.
 - The file `Rinstall.R` located at the folder `utils`, where Tinn-R is installed, was updated to suppress the download of two packages (`Hmisc` and `R2Html`), which are no longer necessary to the `TinnRcom` package.
- The `TinnRcom` package was updated to version 1.0.17.

4.0.2.0 (Apr/22/2015)

- Features Added:

- Drag and drop from R explorer to Editor.
- An experimental Hex viewer ([ATBinHex](#)) was added to Tools/Results/Hex viewer. A bug is related to the UTF-8 BOM encoding file: the selection of (Char == Hex), only in the first line, is not correctly associating the character with its corresponding Hex. It has its own pop-up menu with many (and useful) options.
- New resources to autocompletion related to: ([{ ' " were added. If there is a selection, the new feature will respect that.
- A new option was added to Options/Application/R/Rterm/Options (Rterm) enabling the user to choose if the setwidth will be sent automatically when the panel width or font size is changed.
- A new option was added: Project/Open demo. The objective is to show the novice user what is a project and its utility.
- The main menu Options has three new options:
 - * Auto completion ([{ ' " CTRL + ALT + C
 - * Enable Notication CTRL + ALT + N
 - * Update silently CTRL + ALT + U

- Bug(s) fixed:

- A bug related to Sumatra (PDF viewer) and User guide (PDF), location of topics, was fixed. If you have it installed, please upgrade to 3.1.1 or higher ([SumatraPDF](#)).
- Rterm is now entirely updated after some changes in Options/Application or Options/Highlighters (settings).
- A bug related to line wrap has been fixed: thanks to Frank for pointing it out.
- A bug related to character recognition of Tools/Database/R/Mirrors was fixed.
- A bug related to Rterm/LOG text highlighter was fixed.
- A bug related to Page control files and hints of files were fixed.
- Some bugs were fixed and it is more user friendly when the editor is in split mode.
- Some bugs related to Hide/Show resources of Tools panel were fixed.

- The pre-release versions 4.0.0.0 to 4.0.1.5 were restricted to testers: thanks for the tests and suggestions.
- Philipe Silva Farias began to work in the project as co-author.
- An experimental file notification resource has been added to the project. The options are at Options/Application/General.
- Prior backup (full or database) will no longer be compatible from this version on.
- A set of new icons was created specially to the project: thanks to Philipe Silva Farias for the hard work.
- The visual identity of the project changed: thanks to Carolina Sartório Faria for the work.
- Some improvements were made in the Options/Application and this interface has new options now.
- The usability of Rterm interface was improved.
- The URI highlighter has now a new identifier named Space.
- The default shortcut to alternate the focus between Editor, IO and LOG was changed.
- The automatic recognition of hardware architecture (32 or 64 bit) was improved.
- The project Jedi used in the Tinn-R project was updated to the latest version. Some bugs related to Tools and Rterm panels and Auto Hide option were corrected.
- The list of recognized words of R highlighter family related to plotting has been updated. Thanks to Berry Boessenkool for pointing it out.
- The order of the objects was changed in Tools/R/Explorer. It has now a more natural order and hierarchy.
- The available font family was restricted to True Type and Fixed-PitchOnly.
- It is now possible to open any URL from the Windows Open Dialog: File/Open.
- Tinn-R Editor - GUI for R Language and Environment is a project under GPL and distributed as freeware. Since creating and maintaining the project involve many costs, donations are welcomed!

A.10 VERSIONS RELEASED IN 2014 (08)

3.0.3.7 (Mar/03/2014)

- Restrict release.
- Some improvements in the main interface and the User guide.

3.0.3.6 (Feb/28/2014)

- The conversion options using Deplate or Txt2tags depends of the file extension. For Deplate are recognized: .dp, .dpt, .dplt, .deplate and .txt. ForTxt2tags are recognized: .t2, .t2t, .txt2tags. and .txt.

3.0.3.5 (Feb/10/2014)

- *Highlighters settings* window was deeply reworked.
- *Print preview* window was reworked.

3.0.3.4 (Feb/08/2014)

- Bug(s) fixed:
 - R highlighter: expressions like the below. Thanks to Arnold for pointing it out.

```
gsub("/", "\\\\", text)
```
 - Comment/Uncomment: the automatic detection of the language and chunks (regions) for multiple (or complex) languages (like: R noweb, R doc, HTML complex, PHP complex, etc).
- The Comment, Uncomment first and Uncomment all procedures were improved.

3.0.3.3 (Feb/05/2014)

- Parts of the source code were enhanced.
- The control over the previous focus related to various options (Options, Help, View, etc.) has been improved.
- The *IPC (Inter Process Communication) communicating Rterm and Tinn-R* was re-optimised. Now it its approximately **4x faster than the prior version.** \o/\o/\o/
- The *User guide* has been revised and improved.

3.0.3.2 (Jan/30/2014)

- Bug(s) fixed:
 - Advanced options of the editor: Options/Application/Editor/Advanced/Want tabs. Now when tabbing (if there is a selection) <TAB> and <SHIFT><TAB> act really as block indent, unindent. It works only inside of the more important instances of SynEdit class: Editor and Rterm/LOG.
Within Rterm/I0 <TAB> has another function: to complete.
- Some options of the interface Options/Application/Editor/Advanced are now more understandable. Thanks to Berry Boessenkool for the suggestions!
- The menu Help has a new option: What is new?

3.0.3.1 (Jan/29/2014)

- Bug(s) fixed:
 - The installer of version 3.0.3.0 related to the file data.zip. The file data.zip was corrupted. The installer of version 3.0.3.0 has been deleted from all servers and we advise users to not redistribute this version. Thanks to Mark A. for pointing it out!
- The STOP button is working again for Rgui.exe (but is not active for Rterm.exe yet).

3.0.3.0 (Jan/28/2014)

- Bug(s) fixed:
 - Pop-up menu of: Tools/Database/Completion, Tools/R/Explorer and Tools/R/Card.
- Parts of the source code were enhanced.
- The *IPC (Inter Process Communication) communicating Rterm and Tinn-R* was optimised. Now it is approximately **8x faster and also more accurate**. This really was an old dream!
- The *User guide* has been revised.
- The menu Tools/Processing/Compilation (Latex) has a new option: Make index (makeindex). The default shortcut is *CTRL + ALT + I*.

- The Rterm support to the *function debug* and the *package debug* was a bit enhanced. The necessary instruction (below) is automatically sent to the R interpreter from this version.

```
options(debug.catfile = 'stdout')
```

3.0.2.8 (Jan/22/2014)

- Bug(s) fixed:
 - R highlighter: *quote* argument of *read.table* function.
 - Pop-up menu of Tools/R/Mirrors.
- TinnRcom package was upgraded to version (1.0-15). The package will be automatically updated.
- Tools/R/Mirrors:
 - It has a new status bar showing the default repository
 - It has a new button on the tool bar which enables the opening of the URL: current and default.
- The Print preview interface was a bit enhanced.
- Some icons of the Main interface and Tools panel were changed.
- From now SumatraPDF will be the default viewer of the Tinn-R User guide. If Sumatra is the default system PDF viewer it will be used. Otherwise, a compact version released jointly with Tinn-R (Install.path/sumatra/SumatraPDF.exe) will be used.

A.11 VERSIONS RELEASED IN 2013 (10)

3.0.2.7 (Dec/30/2013)

- The menu `View` was a bit improved.
- The *User guide* has been revised.

3.0.2.6 (Dec/28/2013)

- **Tools:** the interface related to `Card` and `Mirrors` is now in a more logic place: `R`.
- The menu `View` was a bit improved. The more used options related to `Rgui`, `Tools` and `Rterm` were duplicated, but can be easily found.
- The menu `Help` was a bit enhanced and has a new option: `User list (discussion group)`.
- The *User guide* has been revised.

3.0.2.5 (Dec/03/2013)

- Bug(s) fixed:
 - Double click on `Tools/Database/Comments`.
 - Double click on `Tools/Database/R mirros`.
 - The content of `Help/Citation` (put on clipboard).
 - Active line (BG) option (`Show/Hide`) and `Rterm`.
- The default transparency was set to 0%.
- Some small errors were fixed in the User guide.
- The version 3.0.2.3 and 3.0.2.4 were restricted to testers: many thanks to all testers, mainly to Michal Sacharewicz for his suggestions and tests related to *R path identification!*

3.0.2.2 (Nov/20/2013)

- Bug(s) fixed:
 - An intermitence related to `Sweave` and `Knitr`.
- The *User guide* has been revised.
- The main menu `File` has a new option: `Template`. This submenu has template to: `R script`, `R doc (Function, Dataset and Empty)`, `R html`, `R markdown` and `R noweb`.

3.0.2.1 (Nov/19/2013)

- Bug(s) fixed:
 - Edit/Comment.
 - Recognition of latest version and update of all five database: Shortcuts.xml, Completion.xml, Comments.xml, Rcard.xml and Rmirrors.xml.
 - Options/Application when the user choice Cancel.
- The support to Knitr package was improved and has more options and resources:
 - A new button: Knit to L^AT_EX (.Rnw).
 - A new button: Knit to HTML (.Rmd, Rhtml).
 - Two new multi/complex highlighter: R_markdown and R_html were added.
 - From now Tinn-R will always open/update the output (*.tex, *.md and/or *.html) file(s) after interpretation.
 - If the option Tools/Processing/Viewer/HTML is marked, the *.html output file will be opened on the default browser.
- Parts of the source code related to Sweave were enhanced. From now Tinn-R will always open/update the .tex file after interpretation.

3.0.1.10 (Nov/15/2013)

- Bug(s) fixed:
 - Rterm/LOG/Gutter.
- Parts of the source code related to R_path identification were enhanced.
- Parts of the source code related to Code_completion were enhanced.

3.0.1.9 (Nov/07/2013)

- The URL of the project was changed to: <https://tinn-r.org/en/>
- The Rterm interface now accepts the option eoScrollPastEol (Allows the cursor to go past the last character into the white space at the end of a line).

3.0.1.8 (Nov/06/2013)

- Bug(s) fixed:
 - Options/Highlighters (settings) not remembering when more than one option (Bold, Italic and Underline) is marked.
 - The visibility of the Toolbar Edit.
- The *User guide* has been revised. From this version it will be distributed only in PDF format and written entirely in \LaTeX .
- This version went through a long period of development and testing. This version brings several improvements to the project and is the best version ever released to the users. We hope you enjoy it!
- The versions 3.0.1.0, 3.0.1.1, 3.0.1.2, 3.0.1.3, 3.0.1.4, 3.0.1.5, 3.0.1.6 and 3.0.1.7 were restricted to RC (Release Candidate) testers: many thanks to all testers, mainly to Jakson A. Aquino (main author of the nice Vim-R-plugin), for his good suggestions!
- The *SynEdit* component used in Tinn-R project was updated to the latest version. Unlike previous versions, all necessary changes and adjustments in the sources of *SynEdit* were made separately from the component, that is in units following the project source code of Tinn-R. This change is intended to facilitate the collective development of the Tinn-R sources.
- The sources of the project were deeply improved.
- Support to *UNICODE* was added (this was a very hard work).
- The highlighters: *R*, *TeX*, *All* and *Text* were recreated to *UNICODE* and all have new resources.
- The *Sweave* highlighter was renamed to *R noweb*. It is a multi highlighter: *R + TeX*.
- From this version Tinn-R Editor/GUI needs a new version of the TinnR package: **TinnRcom** distributed with the Tinn-R install/setup program.
- The *RegEx PCRE* was implemented in the project. We have now more flexibility to deal with strings in the sources of the project.
- The window *About* was a bit enhanced.
- The support to *DDE protocol* was removed from the project, it is a bit old and hard to maintain.

- The database for *call tip* was removed, due to constant updates of covered packages, it is hard to maintain.
- Two new database were added: `Rmirrors.xml` and `Comments.xml`. The first allows the user to manage the R mirrors and the second, the comments for all supported languages by Tinn-R.
- The `Comment`, `Uncomment first` and `Uncomment all` procedures were improved and from now are in the main menu `Edit`. Now it detects automatically the language and chunks (regions) for multiple (or complex) languages (like: `R noweb`, `R doc`, `HTML complex`, etc). The comments default are all now in the database `Comments.xml` (user configured). The user can force the use of an specific comment by unchecking the option `Tools/Database/Comments: (x) Auto detect language` (recommended).
- The unique communication protocol with R is now the TCP/IP under the necessary `svSocket` package. There is a new option controlling whether R is connected and whether R is running automatically or not. By default it is set to True.
- The `R explorer` is almost fully based on temporary files, but the request is made silently (if the package `svSoket` is loaded and the communication (TCP/IP) is active). The reception of the information from R under TCP/IP was removed for this purpose. For massive exchanging information temporary files are faster, more accurate and stable.
- Parts of the source code, related to call tip and data completion were enhanced: it is now faster and more accurated (mainly related to the OOP).
- `Options/Application` window and related options were deeply upgraded. It is important to note that the editor options are now inside this interface.
- The menu `Options/Editor` was removed.
- The menu `View` has a new option: `Gutter (show/hide)` with the shortcuts `CTRL + ALT + G` associated to it. It will affect all instances of `SynEdit` class: `Editor`, `Rterm` and `LOG`.
- The menu `Edit` and `Format` were a bit improved.
- An option menu `Encoding` was added to the main menu.
- The menu `Help` was a bit simplified.

- The main menu R was a bit improved and has three new options: Set .trPaths (temporarily), Get info (R and TinnRcom) and Update mirrors.
- The button R controls: packages has new options: Install and Load TinnRcom package.
- The status bar was new panels showing the encoding and line endings of the files.
- The Organize screen procedure (related to screen arrangement of Tinn-R and Rgui.exe) from now runs twice: since just once is not enough for all flavors of the OS Windows.
- Inside Tools/R explorer(Tools window) now CTRL + C copy the R object name selected to the clipboard. This can be very useful.

2.4.1.7 (Mai/08/2013)

- Bug(s) fixed:
 - Copy button of Tools/Database (Rtip and Completion).
 - R send: line (with empty files).
- Options/Application/R/Rterm/Options (Rterm) has a new option: Workspace image (close without ask for save). Thanks to Roland E. Joss for the suggestion.
- Some buttons of R toolbar (print, plot, list names, list structure, edit and fix) are now enabled only with at least one open file.
- The main menu Web/R Editors/GUIs and IDEs/Tinn-R was a bit enhanced. Tinn-R project now has its own [Web page](#).

2.4.1.6 (Mai/06/2013)

- Bug(s) fixed:
 - A bug, noticed after the release of the R 3.0.0, related to the Options/Application/R/Path (R) and recognition of the latest version has been fixed.
- The user guide was revised: many thanks to Jakson A. Aquino (the main author of Vim-R-plugin).

A.12 VERSIONS RELEASED IN 2012 (03)

2.4.1.5 (Dez/06/2012)

- Bug(s) fixed:
 - A bug related to the Options/Application/Rgui/Recognition (Rgui) accidentally added in the prior compilation (pre-release 2.4.1.2) was fixed.
- The versions 2.4.0.2, 2.4.1.0, 2.4.1.1, 2.4.1.2, 2.4.1.3 and 2.4.1.4 were restricted to pre-release testers: Many thanks to all testers and for the suggestions!
- Basic support to [Pandoc](#) (a universal document converter) was added.
- The main menu Help/Main/File conversion (introduction) was updated. Now it also contains an example of pandoc.markdown file.
- The User guide was thoroughly updated.
- The file Rconfigure_default.r was updated.
- The conversion tools (Deplate, Pandoc and Txt2tags) are currently enabled to all file extensions (including empty extension).
- The main menu Web was updated and has new options.
- The message of any problem related to R/Configure/Permanent (Rprofile.site) was enhanced.

2.4.0.1 (Nov/07/2012)

- Bug(s) fixed:
 - A bug related to CTRL + ENTER was fixed. Many thanks to Leandro Marino for pointing it out!
 - A "bug" (*the Microsoft eventually makes some drastic changes*) related to the Windows 7 (64 bit) and the path detection of R (32-bit and 64-bit) was fixed.
- The internal shortcut CTRL + ENTER doesn't break the current line at the cursor position any more, i.e. it preserves the entire line content and starts a new one. Many thanks to Leandro Marino for pointing it out!
- The versions 2.3.7.4 and 2.4.0.0 were restricted to pre-release testers: many thanks to all testers and for the suggestions!

- The graphical interface was updated with some improvements, mainly the Application options:
 - A hierarchical tree view replaced the classical tabs approach. It is better for large options like now.
 - The options related to path of DVI and PDF viewer was removed. The default system for .dvi and .pdf will be used.
 - The option to close a prior instance of the viewer (DVI and PDF) before compilation is now independent. It gives more user control.
- A new toolbar Format was added to the main toolbar.
- A new resource allowing reformat R code (selection or whole file) using formatR package was added. The icon resource was placed in the Format toolbar, so that from this version on, the formatR package will be necessarily together with the already traditional TinnR and svSocket packages.
- Due to the new resource related to the reformat code the variable .trPaths was changed. As a result, it will be necessary to run again the R/Configure/Permanent (Rprofile.site). In this case, **do not forget to remove any prior script generated by Tinn-R in the Rprofile.site file.**
- The R/Hotkeys interface was thoroughly reworked. Now it has two tabs, Default and Custom:
 - Default: contains the already traditional instructions of Tinn-R;
 - Custom: **allows the user to customize any instructions** to be send to R interpreter (thanks to Philemon Lenherr for the suggestion). The instructions must be as follows:
 - * Simple: search(). The R interpreter will receive > search();
 - * Replace word or small selection: View(%s, title='View of iris dataset'). If the editor cursor is over the word iris or it is selected, the R interpreter will receive > View(iris, title='View of iris dataset')
 - * Replace whole file: source(%f, echo=TRUE, verbose=TRUE). The R interpreter will receive > source(.trPaths[4], echo=TRUE, verbose=TRUE). All rules related to send file are preserved.
- Sorry, due to thoroughly changes made to the R/Hotkeys, all hotkeys configured prior to this version will be lost. It will be necessary to reconfigure them all.

- The chapter `Some secrets for an efficient use` is being revised by Ricardo Pietrobon and soon will be completed. Many thanks for his hard work!

2.3.7.3 (Set/23/2012)

- Bug(s) fixed:
 - Not really a bug, but a correction related to prior versions which do not recognize the `Options/Application/R/Rgui/Recognition/Type` is set to `Whole` related to R Console (32-bit).
- This version was compiled with **Code Gear 2007 running under Windows 7**. Previously it was compiled under Windows **Vista** or **XP**. We noticed that some boring stuff were automatically corrected by simply changing the operating system. That is, there were bugs caused by older operating systems.
- Some default startup values were changed. We hope it is now better for novices.
- Basic support to R package Knitr was added.
- Minor improvements in the graphical interface.
- The user guide has a new chapter: `Some secrets for an efficient use`.

A.13 VERSIONS RELEASED IN 2010 (10)

2.3.7.1 (Nov/24/2010)

- Bug(s) fixed:
 - A bug, accidentally introduced in version 2.3.7.0, which averted opening a file by Enter or Double click, a recognized Tinn-R file from Windows environment, was fixed.

2.3.7.0 (Nov/22/2010)

- Bug(s) fixed:
 - A bug related to the intermittent loss of connection (or apparent freeze) occurring on Rgui.exe was fixed.
- The versions 2.3.6.4, 2.3.6.5, 2.3.6.6 and 2.3.6.7 restricted to pre-release testers.
- The Application options interface was a bit changed:
 - The Application options/R/Rterm was split in two tabs: Error and Options. The tab Error has a new option: Trying to find code errors (at the editor)*. It enables the user to set Tinn-R in order to find code errors at the editor when sending instructions to Rterm. It may happen that the error will not be found at the right place, for example the error might be the same word appearing in a comment which comes before the actual along the code. In that case the user should use the shortcut F3 (Find again). The word will appear selected, than just press OK until finding the right error. The first search done internally by Tinn-R has *Case sensitive* and *Whole word only* as default, but, this is not passed to the search interface, therefore the user should just select them if convenient. If the error has number among letters *Whole word only* is not a good option.
- This version is fully compatible with Windows 7 and R 2.12.0.
- The component XPmenu was removed from the project. Windows XP users might find the Tinn-R appearance less attractive, but the applicative is now more stable. As soon as possible, the project will get a better option for skins.
- Parts of the source code were optimized.

2.3.6.3 (Nov/14/2010)

- Bug(s) fixed:
 - A small bug (introduced in version 2.3.6.0) related to interactive use of Rterm interface, when R returns only one value, was fixed.
- Parts of the source code were optimized.

2.3.6.2 (Nov/13/2010)

- Bug(s) fixed:
 - A bug (introduced in version 2.3.6.1) related to automatic recognition (and setting) of the paths of Rterm and Rgui from R version 2.12.0 was fixed.
- Parts of the source code were optimized.

2.3.6.1 (Nov/12/2010)

- Bug(s) fixed:
 - Permanent configuration of the file Rprofile.site from the menu option: R/Configure/Permanent (Rprofile.site).
- Parts of the source code were optimized.

2.3.6.0 (Nov/10/2010)

- Bug(s) fixed:
 - Paths of Rterm and Rgui from R version 2.12.0 were fixed.
- The Application options/R/Path has a new option: Architecture (bit). It enables the user to set R in order to run Rterm.exe or Rgui.exe in either 32 or 64 bit. They run independently one from the other. If the above dialog option Use latest version (always) is set to No, the dialog box Architecture (bit) becomes disabled. In other words, the path must be manually set. In this case, the user preference (if valid) will not be changed when Tinn-R is starting. If the dialog Use latest version (always) is set to Yes, the path of the latest R version (32 or 64 bit) will be found automatically. In case the user's computer is 64 bit and it has the latest R version installed with both options (32 or 64 bit) the latest one has the preference.

- Rterm interface now recognizes the single occurrence of a string pattern when using TAB in the prompt. The general behavior is not as friendly as in Rgui or in Linux console, but it is working.
- Parts of the source code were optimized.

2.3.5.2 (Apr/11/2010)

- The User guide was revised by Ricardo Pietrobon in an attempt to improve its flow and style. This is a work in progress, and so we should be improving it over time!

2.3.5.1 (Mar/28/2010)

- Version restricted to developers.
- The User guide was revised by Ricardo Pietrobon to improve its style and readability. So far, we have partial revisions implemented to chapters Overview, Basics, Working with, Menu description and parts of What is new.
- The Application options/R/Basic has a new option: Smart (all send). When this option is set to Yes, all single line commands will not be sent through source(...), but instead as Send line.

2.3.5.0 (Mar/04/2010)

- Bug(s) fixed:
 - Small corrections in the *Application options* interface.
- Parts of the source code were optimized.
- The Rterm interface has now a simple toolbar including the more usual options.
- The User guide is being revised by Ricardo Pietrobon and soon will be more readable and intelligible. Up to now only chapters Overview and Basics have been worked out. Many thanks for his hard work!

2.3.4.4 (Jan/10/2010)

- Bug(s) fixed:
 - A bug related to the visibility of the buttons *Send contiguous* was fixed.

A.14 VERSIONS RELEASED IN 2009 (16)

2.3.4.3 (Dec/21/2009)

- Parts of the source code were optimized.

2.3.4.2 (Dec/18/2009)

- Bug(s) fixed:

– A bug related to the function `debug` (package `base`) in the *Rterm* interface was fixed. It was generating an exception Access violation at address NUMBER in module 'Tinn-R.exe'. Read of address FFFFFFFF. This bug started in version 2.3.4.0 (Dec/06/2009).

2.3.4.1 (Dec/13/2009)

- Bug(s) fixed:
 - A bug related to `Search in Files` and its links in the *Tools* panel was fixed. It was not going to the correct line in the file. This bug was related to the implementation of the resource *Remember file state* in version 2.3.2.5 (Nov/03/2009).
- The user can now hide/show the button `Editor: current line to top` in the *R toolbar*.

2.3.4.0 (Dec/06/2009)

- Bug(s) fixed:
 - A bug related to the package `sem` by John Fox (functions: `read.moments` and `specify.model`) which generated an error when submitting line by line within *Rterm* interface was fixed. Many thanks to Frank for pointing it out!
- Parts of the source code were optimized.
- The menu *R/Rterm/Clear* and the pop-up menu *Rterm* (IO and LOG), both related to *Clear* option, were changed. This change allows more specific control over *IO* and *LOG* of *Rterm* interface.
- Two new options enable the user to send contiguous lines of script to the R interpreter.
- The small script generated by Tinn-R in the file `Rprofile.site` was changed:

- The parameter `dep=TRUE` was removed from the line

```
install.packages(necessary[!installed], dep=TRUE)
```

since the `Hmisc` package, which enables Tinn-R to export R objects to TeX format, has several dependencies. This change will speed up the basic R configuration.

```
## Check necessary packages
necessary <- c('TinnR', 'svSocket')
installed <- necessary %in% installed.packages()[, 'Package']
if (length(necessary[!installed]) >=1)
  install.packages(necessary[!installed])
```

- Under the *Rterm* interface the graphical menu should enable the user to choose the repository for a current session. However, this is inconsistent, sometimes showing this option and sometimes not. Therefore, the best option is to set the preferred repository from the file `Rprofile.site`.

2.3.3.1 (Nov/10/2009)

- The menu *Tools/Utils* was removed from the executable: it is restricted only to developers and we forgot to make it not visible.

2.3.3.0 (Nov/09/2009)

- Parts of the source code were optimized.
- The menu *View* was enhanced with a new option *Word wrap*. It allows the user to control the *Editor*, *Rterm/IO* and *Rterm/LOG* word wrap. Word wrap is a feature of most text editors, word processors, and web browsers, of breaking lines between and not within words, except when a single word is longer than a line.

2.3.2.6 (Nov/07/2009)

- Bug(s) fixed:
 - A bug related to *Rterm interface* and `Auto hide` option was fixed.
 - The automatic WordWrap option of Rterm interface is now off.

2.3.2.5 (Nov/03/2009)

- Bug(s) fixed:

- A bug related to R identification, if *Options/Application/R/-Path/Use latest installed version (always)* option is Yes was fixed. This bug was detected only after the R 2.10.0.
- Parts of the source code were optimized.
- Tinn-R now has the *Remember file state* setting option. The file states are: all marks (0..9), the position of the cursor and top line of the file. It was implemented using a XML database (Cache.xml). It can be found at *Options/Application/Main/General/Remember file state*.
- The option Send *Marked block* was enhanced. Now it works as below:
 - **The file has no marks:** the option will not be available (gray);
 - **The file has one or more marks and the cursor is above the first mark, or below the last:** all text (above or below this mark) will be sent, according to the cursor's position (above or below the mark);
 - **The cursor is between any two adjacent marks:** all text between those two marks will be sent.
- Under *Rterm* interface it is possible to use the TAB as follow: > ba (followed by TAB). This procedure will send to R interpreter the instruction apropos('^bla', case.insensitive=FALSE) and it will not be visible. R will returns a character vector giving the names of all objects in the search list matching ba. For example:

```
> ba (followed by TAB)
[1] "backsolve"           "backSpline"          "bacteria"
[4] "balanceMethodsList" "ballocation"         "bandwidth.kernel"
[7] "bandwidth.nrd"       "barplot"            "barplot.default"
[10] "bartlett.test"     "base...POSIXt"      "base.+.POSIXt"
[13] "base.difftime"     "base.help"          "base.library"
[16] "base.loadhistory"   "base.lockEnvironment" "base.rbind.data.frame"
[19] "base.save.image"    "base.savehistory"   "baseenv"
[22] "basehaz"             "basename"
```

> ba

- The family *rmControls* of components was removed from the project.

2.3.2.3 (Ago/06/2009)

- The *Application options interface* was enhanced.
- Parts of the source code were optimized.

2.3.2.2 (Jul/20/2009)

- Bug(s) fixed:
 - A bug related with *Hotkeys* (*operational system*) when changing the status (Active not Active).

2.3.2.1 (Jul/19/2009)

- Bug(s) fixed:
 - An undesired and potentially danger option in the menu *Tools/Utils*, used only in the development, was removed.

2.3.2.0 (Jul/18/2009)

- Bug(s) fixed:
 - All bugs related to database pointed out by users.

2.3.1.0 (Jul/15/2009)

- Bug(s) fixed:
 - All bugs related to database pointed out by users.

2.3.0.0 (Jul/10/2009)

- Bug(s) fixed:
 - The error message when typing CTRL + TAB inside the *Rterm* interface whenever it was not split.
 - It now remembers the position of the *Tabs files* (top or bottom) when starting.
 - It now properly organizes the *Tabs files* in relation to other toolbars whenever the user uses the show/hide option in the toolbars.
 - The intermittency of completion resources.
 - The *Auto completion* and *Data completion* now recognizes split by a dot as a complete word: for example `my.function()` and `my.data$`.
 - Under Windows Vista the option *R/Configure/Permanent (Rprofile.site)* now checks if the user has administrative privileges to change the content of the file `Rprofile.site`, before inserting the necessary script. If the user receives an error message, it is necessary to manually change the security properties to enable full control of the folder `etc` where R is installed.

- The *Completion* resource migrated to XML, it is more flexible and easy to use. Now it is located in the the menu *Tools/Database*.
- *R card* and *R tip* are now located in a more convenient place: *Tool/- Database*.
- The *R tip* resource was updated.
- Menu *Format/Block* was removed and all associated resources were relocated to a more logic place: *Format/selection*.
- The *User guide* was expanded/enhanced in various topics.
- The *Application options interface* was enhanced.
- A new option in the *Application options* allows more specific recognition of Rgui. Now it is possible to avoid any windows caption with the word **Console** to be recognized as a R instance.
- The interface *Tinn-R hotkeys* was fully reworked and it is now more simple and efficient.
- Parts of the source code were optimized.
- A new resource was added to the *R send: Clipboard*. It enables the user to send the content of the clipboard easily to Rterm.
- Sorry, due to bugs the highlighters *Deplate* and *Txt2tags* were removed from the project. New ones will be made in the future.

2.2.0.2 (Feb/09/2009)

- Bug(s) fixed:
 - Using TinnR package version 1.0.1 or 1.0.2 Tinn-R did starts when R starts with the instructions below in the Rprofile.site:

```
# uncomment the line below if you want Tinn-R starts always \RR{} starts
options(IDE='C:/Tinn-R/bin/Tinn-R.exe')
```

The origin of this bug was the change of the packages svIDE (and others no longer necessary) to TinnR package. One function that should do the job was not present (TinnR package version 1.0.1 or 1.0.2). In the new version of the TinnR package (1.0.3), this function `trStartIDE` was added. The option `R/Configure/Permanent (Rpfile.site)` will generate a new script:

```

##=====
## Tinn-R: necessary packages and functions
## Tinn-R: >= 2.2.0.2 with TinnR package >= 1.0.3
##=====
## Set the URL of the preferred repository, below some examples:
options(repos='http://cran.at.r-project.org/')      # Austria/Wien
#options(repos='http://cran.r.c3sl.ufpr.br/')        # Brazil/PR
#options(repos='http://cran.fiocruz.br/')            # Brazil/RJ
#options(repos='http://www.vps.fmvz.usp.br/CRAN/')  # Brazil/SP
#options(repos='http://brieger.esalq.usp.br/CRAN/') # Brazil/SP

library(utils)

## Check necessary packages
necessary <- c('TinnR', 'svSocket')

installed <- necessary %in% installed.packages()[, 'Package']
if (length(necessary[!installed]) >=1)
  install.packages(necessary[!installed])

## Load packages
library(TinnR)
library(svSocket)

## Uncomment the two lines below if you want Tinn-R to always start R at start-up
## (Observation: check the path of Tinn-R.exe)
#options(IDE='C:/Tinn-R/bin/Tinn-R.exe')
#trStartIDE()

## Option
options(use.DDE=T)

## Start DDE
trDDEInstall()

## Short paths
.trPaths <- paste(paste(Sys.getenv('APPDATA'),
                           '\\\\Tinn-R\\\\tmp\\\\',
                           sep=''),
                    c('',
                      'search.txt',
                      'objects.txt',
                      'file.r',
                      'selection.r',
                      'block.r',
                      'lines.r'),
                    sep='')

```

If you uncomment this part:

```

## Uncomment the two lines below if you want Tinn-R starts always R starts
## (Observation: check the path of Tinn-R.exe)
options(IDE='C:/Tinn-R/bin/Tinn-R.exe')
trStartIDE()

```

The new TinnR package will do the job!

2.2.0.1 (Feb/05/2009)

- Bug(s) fixed:
 - L^AT_EX font *Enphase* was fixed. It was inserting `\textbf{}` instead of `\emph{}`.
 - Save and load workspace is now enabled only if Rterm is running.
 - A bug associated with the *Color preferences interface* related to *Txt2tags* and *Deplate* syntax, not allowing to change the background color of the root element in the correct way was fixed.
- All prior documentation of the Tinn-R project was updated. Some parts were expanded and others were excluded, new ones were generated and, finally, all were joined in the new *User guide*. This *What is new* is now part of this user guide. The source files (written in *Txt2tags*) of this User guide are available in the folder `doc/english/user_guide` where Tinn-R is installed. This way, we hope that the user will be able to help us in making it better, day by day, by sending us any useful contribution.
- The *R toolbar* can now be docked at left, top, right and bottom side of the main interface. Some issues related to *Rterm* and *Tools* interface when dragging the R tools bar, have not completely been solved yet. In order to fix any problem, hide and show again the *Rterm* or *Tools* interface.
- The match brackets resource (default shortcut is **CTRL + B**) was added also to the Rterm interface (*IO* and *LOG*).
- The *Tools* interface was a bit reworked and the menu *Views* was changed to accommodate the changes.
- The menus *View* and *Help* were a bit reworked.
- The send and control R resources were extended to all instances of the *synEdit* class. In other words, if you put the cursor in any word, or select any text in the *Editor* (split or not), *IO* or *LOG* (docked or not, in the same or distinct monitors) interface and, after this, select an action (print content, plot, etc) it will be executed.

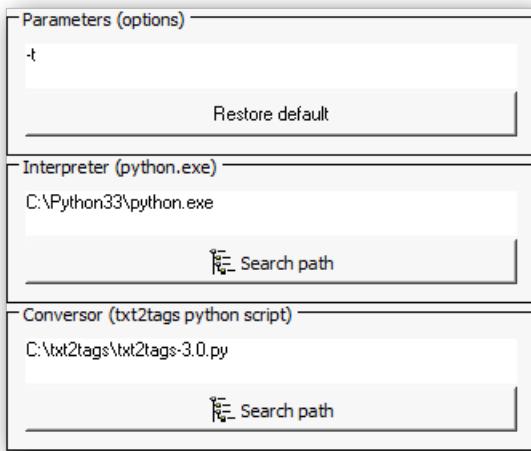
A.15 VERSIONS RELEASED IN 2008 (14)

2.1.1.6 (Nov/17/2008)

- Bug(s) fixed:
 - All fonts of *Tools/Markup* interface are now visible.
 - The *Application options* was a bit reworked.
- Due to focus changes behavior in the latest Tinn-R version (2.1.1.5), the focus control button (options toolbar) will be not enabled if Rterm is running. In this case, the focus will be always returned to the interface (editor or Rterm) that has the prior focus before all send/control options.

2.1.1.5 (Nov/12/2008)

- Bug(s) fixed:
 - Related to *Inverse DVI Search*;
 - Focus control working with dual monitor.
- The latest [Windows installer](#) version is 2.3. Therefore, it was necessary to adapt Tinn-R to work with the interpreter Phyton for Windows (python.exe) and the python script to make the conversion (txt2tags). The current version of Txt2tags is 2.5.
- To install and configure the necessary resources follow these simple steps:
 1. Download and install the interpreter [Python](#);
 2. Download and unzip [Txt2tags](#) inside any folder anywhere in your computer;
 3. Inside of Tinn-R, go to 'Options/Application/Processing/Txt2tags' and inform the necessary parameters (-t is the default) and the paths of interpreter (python.exe) and the converter (txt2tags python script);
 4. It is enough to use these nice tools in the Tinn-R project.



2.1.1.4 (Nov/10/2008)

- Bug(s) fixed:
 - Closing the applicative if *Rterm* is running (random bug).
 - *Options: return focus after sending to R*:
 - * If in editor, the focus shifts back after sending to R;
 - * If in *Rterm* (or *Rgui*), the focus does not jump back to editor when you press return: ENTER (prompt line), or CTRL + ENTER (any prior line).
- The *Options/Application/Main* menu has a new Dock page. It enables the user to fix any possible problem related to visualization (*dock/hide*) and layout of the Tools and *Rterm* interfaces.
- The resources of *Rterm* interface were extended:
 - All resources available enabling the control of R are now also available to *IO* and *LOG* interfaces: print content, plot, list names, etc;
 - To all options enabling the control of R, THE FOCUS WILL BE MAINTAINED IN THE CURRENT WORK INTERFACE Editor or IO, DISREGARDING the Options: return focus to editor after send/control R (toggle).
- *Rterm* interface and debug package:
 - Changes were made to the debug package (1.0.2) on the message system (*stdout* and *stderr*). The default option is not compatible with *Rterm* interface implementation.

- The best way to make it compatible again is to add the option below in the Rprofile.site file:

```
options(debug.catfile = 'stdout')
```

2.1.1.3 (Nov/04/2008)

- Bug(s) fixed:
 - *Application options/Appearance* when the user select *Cancel*.
 - *R control: packages (both: Load and Remove)* and *Rterm interface*.
 - *Rterm interface under package debug* of Mark V. Bravington when the user type `qqq()` to quit the debugger.
- The interface *Go to line number* was a bit reworked.

2.1.1.2 (Oct/20/2008)

- The appearance of the main menu and all pop-up menus were improved.
- Now Tinn-R is able to perform `inverse DVI` search. It is only necessary to set the path of the binary of Tinn-R and the parameter for file and line in your DVI previewer. For example, using the YAP of Miktex, the necessary configuration will be (assuming the default Tinn-R path):

```
C:\Tinn-R\bin\Tinn-R.exe "%f;%l"
```

Be sure that there is not space between the parameters `%f`(related to file) and `%l`(related to line).

Inside Tinn-R (*Options/Application/Processing/Latex/DVI*) it is necessary to add the compilation Miktex parameter: `latex -c-style-errors -src-specials`. Tinn-R can do it automatically with the option *Restore default*. Now, it is:

```
latex -c-style-errors --src-specials  
and  
bibtex --src-specials
```

2.1.1.1 (Oct/15/2008)

- Bug(s) fixed:
 - Bugs of the version 2.X.X.X (prior of this version) related to the *R/Configure/Temporary (current session)* and *R/Configure/Permanent (Rprofile.site)* were fixed.
 - A bug related to the *Rterm* interface with incomplete instructions ('+') was fixed. An undesirable carriage return after the '+ CR', the origin of the bug, was eliminated.
 - A bug related to the status (show/hide) of the toolbars *Spell* and *Search* when closing and starting the application was fixed.
- Version restricted to pre-release testers.
- The way Tinn-R creates the variable `.trPaths` was changed. Now it is automatic. It was posted on [Tinn-R forum](#) by KeithJ (keith_jewell) - 2008-09-15 16:00. It is very useful in laboratories where most users have their own account but use the same computer. Now it is not necessary to adapt the file `Rprofile.site` for each user and R session any more. Many thanks to keith Jewell for the nice suggestion.

```
.trPaths <- paste(paste(Sys.getenv('APPDATA'), '\\\\Tinn-R\\\\tmp\\\\', sep=''),  
c(' ', 'search.txt', 'objects.txt', 'file.r', 'selection.r', 'block.r', 'lines.r'), sep='')
```

- The option *Save/Load* workspace to the *Rterm* interface was added. Many thanks to Maria Conceição for the suggestion.
- The *Tools* and *Rterm* interfaces were reworked and now both are dockable. It makes the interface flexible and user customizable. Many thanks to Thomas Petzoldt for the suggestion.
- A new option was added to the *Options: max.deparse.length (echo=TRUE)*. It is used if echo is TRUE and gives the maximal number of characters output for the deparse of a single expression.
- The weblinks links of the menu *Web* were all updated.
- In order to attend to the request of some users, the menu (and related pop-up menu) *Edit* go back to the project interface with a new implementation. Due to the large frequency of the use of class *SynEdit* (editor, editor split, IO, LOG, etc), the action will be applied in the active (selected) instance of the class *SynEdit*.
- The *font* of the *Rterm* interface is now updated after changes to *Options/Editor/Display/Font*.

2.0.0.7 (Sep/04/2008)

- Bug(s) fixed:
 - A bug related with the package *Rterm* interface was fixed.

2.0.0.6 (Aug/21/2008)

- Bug(s) fixed:
 - A bug related to the package *Matrix* was fixed. Many thanks to Frank for point it out!
- The *Rterm* interface can be used to connect with R as a remote server. Sorry, it is already running, but it was not finished yet!

2.0.0.5 (Aug/17/2008) - pre-release

- Bug(s) fixed:
 - A bug related to the function *debug(base)* was fixed. Many thanks to Steven for pointing it out!

2.0.0.4 (Aug/12/2008) - pre-release

- Small corrections and adaptations suggested by testers. Many thanks to Steven.
- The R history now stores up to 100 instructions.

2.0.0.3 (Aug/05/2008) - pre-release

- Version 2.0.0.2 (Jul/23/2008) was released to some selected users for tests; it was the lastest version compiled under the IDE Delphi 7 of Borland.
- This version, and from now on, all versions will be compiled under the IDE Code Gear 2007, the latest of Borland.

2.0.0.2 (Jul/23/2008) - pre-release

- Small corrections suggested by pre-release testers. Many thanks to John, Steven and Frank for the tests and suggestions.
- This version, unlike 2.0.0.1, is fully compatible with the very useful package *debug* (of Mark Bravington). It remains an unsolved problem to submit, when debugging, an external script with incomplete line, like the one below:

```
plot(rnorm(1e2), main = 'It will cause error!')
```

2.0.0.1 (Jul/22/2008) - pre-release

- Version restricted to pre-release testers.
- The structure of the *ini* and the *routine of initialization* were deeply reworked:
 - The old backups will not be compatible anymore with this one and future versions;
 - When used for the first time, this version will try to recognize almost all prior user preferences.
- A new folder, *bkp*, was added to the ini files structure. It contains all backups of all prior user preferences not (or partially) compatible with the new versions. The control of all default resources are now individual, i.e: *custom*, *data*, *latex*, *shortcuts* and *syntax*.
- The *R explorer* functionality was reworked:
 - The functions (*trObjSearch* and *trObjList*) were reworked. These functions are distributed with the package *svMisc* (>= 0.9.40) of Philippe Grosjean;
 - To make the R interface cleaner, a new variable *.trPath* is used as additional parameter to the two functions above. It will avoid a long string like
`path="C:/Documents and settings/User/Application Data/Tinn-R/Tmp/..."`
 to Windows XP or
`"C:/Users/jcfaria/AppData/Roaming/Tinn-R/tmp/..."`
 on Windows Vista;
 - The R object *.trPath* is a *list* described below:

```
.trPath <- list()
Tmp = 'C:/Users/jcfaria/AppData/Roaming/Tinn-R/tmp/',
Search = 'C:/Users/jcfaria/AppData/Roaming/Tinn-R/tmp/search.txt',
Objects = 'C:/Users/jcfaria/AppData/Roaming/Tinn-R/tmp/objects.txt',
File = 'C:/Users/jcfaria/AppData/Roaming/Tinn-R/tmp/file.r',
Selection = 'C:/Users/jcfaria/AppData/Roaming/Tinn-R/tmp/selection.r',
Block = 'C:/Users/jcfaria/AppData/Roaming/Tinn-R/tmp/block.r',
Lines = 'C:/Users/jcfaria/AppData/Roaming/Tinn-R/tmp/lines.r'
)
```

- The best way to set the variable *.trPath* inside of R is using the options: *R/- Configure/Temporary* or *R/Configure/Permanent*, both can be found in the main menu of Tinn-R *R/Configure*;
- The options to send *File*, *Selection*, *Block* and *Lines to end page*, were reworked:

- The objective was to maintain the R interface cleaner, thus avoiding long strings;
 - All above are also dependent on the variable .trPath;
 - As already noted, the best way to set the variable .trPath inside R is using the options: R/ Configure/ Temporary or R/ Configure/Permanent. Both can be found in the main menu of Tinn-R R/Configure
- The source code of Tinn-R was deeply reworked. The aim was to maintain the prior user knowledge, the stability, and the structural simplicity, but add more flexibility and power to the GUI, mainly associated with the new Rterm interface. The *IO* and *LOG* interfaces are instances of the class synEdit:
 - **IO:** the aim was to add flexibility and power, i.e., joining the power of synEdit (editor) and the functionality of a common console. All prior user knowledge of the resources associated with the editor were preserved: marks, shortcuts, syntax, match brackets, tips, code completion, data completion, etc;
 - **LOG:** has two basic objectives:
 1. To make the *IO* interface cleaner;
 2. To avoid synchronization difficulties with the inter process communication (IPC) called *pipe* in use.
 - The shortcuts and the pop-up menu makes it easy to change among the interfaces: *Editor*, *IO* and *LOG*:
 1. The common Windows shortcut CTRL + TAB changes the active page (IO-LOG).
 2. Any prior line can be sent another time by just placing the cursor anywhere on and typing: CTRL + ENTER;
 - The last line of the *IO* interface (the prompt) has special features:
 1. It has some restrictions for edition and navigation;
 2. ALT+DOWN and ALT+UP are the shortcuts (prior/later) for command history. Both are continuous, cyclic and have a 50 lines limit (we think it is sufficient, but if necessary it can be increased to 100 or more).
 - When more than one recognized instance of R is running the priority order is:
 1. Rterm;
 2. Rgui;

- 3. Rserver (remote);
- The use of the clipboard as IPC was removed for all. It was an old request of users to avoid conflict with other application, mainly with the Open Office suit.
- The communication with `Rterm` is more specific and efficient than with `Rgui`. Therefore, it will receive more attention from the developers from now on. The interface with `Rterm` is not finished yet, but it is running nicely...
 - If some problem happens: press ENTER;
 - If it is not solved, type anything and press ENTER;
 - If it still not solved, sorry: close and restart the `Rterm` instance;
 - Remember: it is not finished yet!
- The TCP/IP resources now are used only with two objectives:
 1. Local: to make the `Rterm` and `Rgui` console cleaner;
 2. Remote: for all: send, control and output.
- A new interface `Work expl.` was added to the `Tools` interface. It will always show the folder and files of the latest file opened. It is simpler and has complementary resources to the `textitWindows expl.` interface.
- The menu `Edit` and associated pop-up menu were removed, but, all the functionality was preserved under shortcuts. To set the preferences see *Options/Editor/Keystrokes*.
- The `Tools` interface was a bit reworked and has new resources.
- The main menu was a bit reworked.
- The interface of the application options was a bit reworked.
- Due to new resources (`IO` interface related with `Rterm`) the default shortcuts to view `Line numbers` and `Special characters` were changed to `CTRL + ALT + L` and `CTRL + ALT + K` respectively. Now `CTRL + L` clears the `IO` and `LOG` interface, like `Rgui`.
- The main component `synEdit` was updated to the latest stable version and minor bugs were fixed.
- The descriptions of the `synEdit` editor options is:

Option	Description
Auto indent	Will indent the caret on new lines with the same amount of leading white space as the preceding line
Auto size scroll width	Automatically resizes the MaxScrollWidth property when inserting text
Drag and drop editing	Allows you to select a block of text and drag it within the document to another location
Alt sets column mode	Holding down the <ALT> key will put the selection mode into columnar format
Maintain caret column	When moving through lines w/o cursor past EOL, keeps the X position of the cursor
Want tabs	When active <TAB> and <SHIFT> <TAB> act as block indent, unindent when text is selected
Smart tabs	When tabbing, the cursor will go to the next non-white space character of the previous line
Smart tab delete	Similar to Smart Tabs, but when you delete characters
Enhance home key	Enhances HOME key positioning, similar to visual studio
Enhance end Key	Enhances END key positioning, similar to JDeveloper
Hide scrollbars as necessary	If enabled, then the scrollbars will only show when necessary. If you have Scroll-PasteEOL, then the horizontal bar will always be there (it uses MaxLength instead)
Disable scroll arrows	Disables the scroll bar arrow buttons when you can't scroll in that direction any more
Half page scroll	When scrolling with page-up and page-down commands, only scroll a half page at a time
Scroll by one less	Forces scrolling to be one less
Scroll past end of file	Allows the cursor to go past the end of file marker
Scroll past end of line	Allows the cursor to go past the last character into the white space at the end of a line
Show scroll hint	Shows a hint of the visible line numbers when scrolling vertically
Scroll hint follows mouse	The scroll hint follows the mouse when scrolling vertically
Tabs to spaces	Converts a tab character to a specified number of space characters
Trim trailing spaces	Spaces at the end of lines will be trimmed and not saved
Group undo	When undoing/redoing actions, handle all continuous changes of the same kind in one call instead undoing/redoing each command separately
Right mouse moves cursor	When clicking with the right mouse for a pop-up menu, move the cursor to that location
Show special chars	Shows special characters

A.16 VERSIONS RELEASED IN 2007 (26)

1.19.4.7 (Dec/23/2007)

- *Application option* was a bit improved. Now, it is possible to set Tinn-R to close previous instance of the viewer (PDF or DVI) in the *Processing/Latex/More* options. Makes it easier to compile under \LaTeX , as Miktex will not gives the message: I can't write the file ... in the DOS prompt. Note that any instance of the viewer (if opened) with the caption 'FileBeeingCompiled'.pdf will be closed.

1.19.4.6 (Dec/17/2007)

- A very simple splash screen was added to application starting.
- The functionality of *Latex/Accents* was improved. Now if you have a selection in the editor it will be filled inside the brackets '{}', otherwise, the simple accent will be inserted.

1.19.4.5 (Dec/14/2007)

- Bug(s) fixed:
 - A couple of bugs related to the prior version 1.19.4.4 (Dec/13/2007) were fixed.
- Parts of the source code were optimized.

1.19.4.4 (Dec/13/2007)

- The appearance of the \LaTeX symbols now have a better layout, alphabeticalilly ordered. The structure of ini file, related with to it, was also changed.
- This version enables the user to set two comment(s) character(s): the main (used for all syntax) and to \LaTeX . The options were added to the main menu *Format/Block*: it is possible to *comment all*, *uncomment first* and *uncomment all*. The *Application options* was a bit reworked to allow the user to set the preferred \LaTeX comment (% , %% , %%% , etc).
- Some default shortcuts were changed. The *ini file related* will be updated to reflect these changes and a backup of the old resource file will be made at first use of this version.

1.19.4.3 (Dec/12/2007)

- Bug(s) fixed:
 - A bug related with the new resource *Count* in latest version 1.19.4.2 (Dec/11/2007) was fixed.
- The tool *Markup/Latex* was improved.
- \LaTeX Itemization and Enumeration procedures were a bit improved.
- The color of the matched brackets are now user configurable. Set it at *Options/ Syntax and colors/ Preferences/ Brackets(FG)*.

1.19.4.2 (Dec/11/2007)

- The tool *Markup/Latex* was improved by adding a toolbar.
- Parts of the source code were optimized.
- The *Application options* interface was a bit reworked.

1.19.4.1 (Dec/01/2007)

- Bug(s) fixed:
 - A couple of bugs related to the prior version 1.19.4.0 (Nov/29/2007) were fixed.
- The structure of the *Tinn.ini* file and the *routine of initialization* were a bit reworked. So, the old backups will not be compatible anymore with that one and future versions. This version will recognize the basic old system configurations, but not all preferences. Sorry for the inconvenience.

1.19.4.0 (Nov/29/2007)

- The *Main menu* was a bit reworked and some options have now a better logic place.
- The tool *Markup* was improved by adding a graphical interface to \LaTeX symbols:
 - This is the first approach to a functional \LaTeX symbol classification:

Functional LaTeX Symbols Classification Criteria (according with the KISS principle):

1. Empty
2. Natural order (ex: Greek: alpha, beta...; Solar system: Sun, Mercury...)

3. Usability
4. Structural simplicity
5. Number of straight lines
6. Number of curved lines
7. Number of sloped lines
8. Clock wise rotation

Authors: José Cláudio Faria and Jorge Alexandre Wiendl

Date : 11/27/2007 00:55:20

KISS : Keep it simple, stupid

- We would like it if advanced L^AT_EX users could help us with this classification;
 - The graphical interface is based on the text files organization of the folder *latex*, located at *Ini files*;
 - Once the order of the symbols (number controlled) in each folder is changed, after restarting Tinn-R, it will reflect this order and maintain all functionality. Therefore, it is very flexible;
 - It is recommended that the numeration criteria used in the classification (ex: 001_mysymbol1.gif, 002_mysymbol2.gif) be maintained;
 - With the exception of png, many image formats can be used;
 - Please, don't remove or rename any of the folders or their names;
 - It was a hard task, so, we don't consider the job finished yet: all users will be welcome with their contributions.
- The main menu has a new option *Insert/Latex*. It enables the user to make *Numerical elements: Array, Matrix, Table and Tabbling*.

1.19.3.1 (Nov/15/2007)

- Bug(s) fixed:
 - A couple of bugs related with *ini file* were fixed.

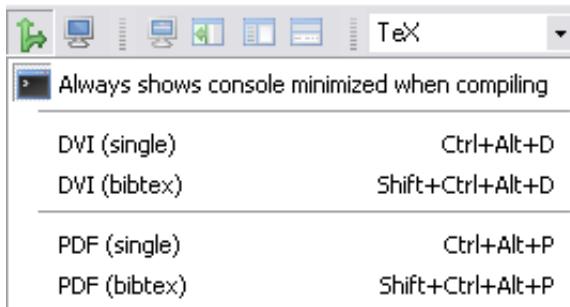
1.19.3.0 (Nov/12/2007)

- Bug(s) fixed:
 - A couple of bugs related to the *Project* interface were fixed and its interface was enhanced.
 - A bug related to the *Most Recent Used (MRU)* files was fixed.
- All prior Tinn-R version of the release 1.19.3.X were considered pre-released (restricted to beta testers only). Many thanks to all testers.

- Good news for Sweave users: debug, i.e, send line by line a Sweave script, was enhanced. Now, Tinn-R will search automatically only inside of the chunks and will disregard all \LaTeX texts. You must start this resource inside of any chunk.
- The structure of the *Tinn.ini* file was a bit reworked. So, the old backups will not be compatible anymore with that one and future versions. This version will recognize the basic old system configurations, but not all preferences. Sorry for the inconvenience.
- The default *ini* file for main shortcuts was changed from *shortcuts.bin* to *shortcuts.txt*. Therefore, any old preferences will be lost. It will be necessary to define all the personal preferences again. Sorry for the inconvenience.
- The main component *SynEdit* was updated to the latest stable version (2.0.6).
- The *Search* results have now the same appearance of the *Text* highlighter.
- In *Tools* the interface *Tags* was renamed to *Markup*. The next resource (we are working in it) will be related with \LaTeX .
- Parts of the source code were optimized.
- The *Application options* interface was a bit reworked.
- The pop-up menu of *Project* interface was a bit enhanced.
- The *status bar* functionality was a bit enhanced. Now you can click inside a specific panel to change the options.
- The way as *Tinn-R* closes the *Rgui* was enhanced.
- The way the brackets are highlight (parent matching) was changed. Now it will contrast with the color with the *Active line highlighted*; see in (Options/Main/Application).
- The *Results* interface was a bit reworked and has now new resources: *Ini log* and a new button enabling full expansion. It is a good idea to check the content of the *Ini log* if any thing is wrong.

1.19.2.3 (Mar/05/2007)

- Parts of the source code were optimized.
- This version enables you to compile \LaTeX files with **bibtex** option:

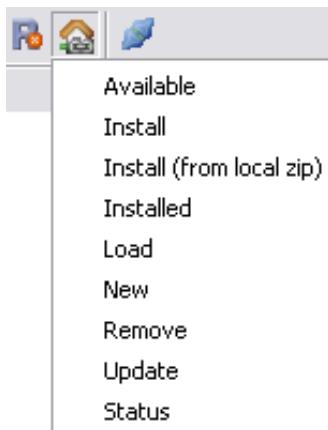


1.19.2.2 (Feb/22/2007)

- Parts of the source code were optimized.

1.19.2.1 (Feb/19/2007)

- Parts of the source code were optimized.
- The first approach to a easy R package manager was implemented:



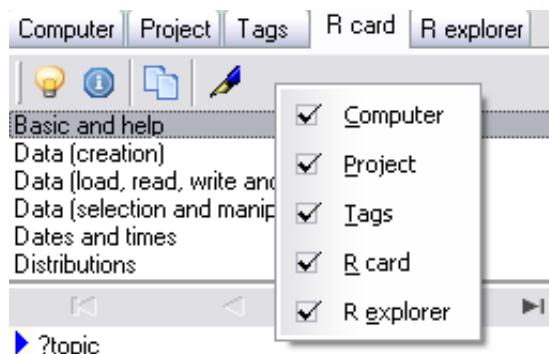
- To avoid R flicker, if TCP/IP connection is active and the *R output* is visible, the *Clear console (F9)* and *Clear all (F12)* will not work on the R console. We will be searching for the best solution.

1.19.2.0 (Feb/16/2007)

- Bug(s) fixed:
 - A bug related to the prior version and *R toolbar* was fixed.

1.19.1.13 (Feb/13/2007)

- Bug(s) fixed:
 - A bug that occurred when organizing the *main toolbar*, in the first use, was fixed.
- Parts of the source code were optimized.
- An experimental graphical interface for tags was added to the program. It was made by adding a new tab named *Tags* on the panel *Tools* and it will aid the user to write Txt2tags, TeX, HTML and others from Tinn-R in graphical mode. In the current version only Txt2tags interface was implemented.
- The structure of the *Tinn.ini* file was a bit reworked. So, the old backups will not be compatible anymore with this and future versions. This version will recognize the basic old system configurations, but not all preferences. Sorry for the inconvenience.
- The main menu *View/Tools* was a bit reworked and has new resources.
- The *Tools* window has now a new pop-up. It enables you to choose the visible pages:



1.19.1.12 (Feb/11/2007)

- *Search and Replace* interface was a bit reworked.

1.19.1.11 (Feb/07/2007)

- Bug(s) fixed:
 - A bug related to the prior version and *R toolbar* (flicker) was fixed.

1.19.1.10 (Feb/04/2007)

- Parts of the source code were optimized.
- The Tinn-R card was updated.
- The *Application options* interface was a bit reworked and the *R resource visibles* was enhanced. Now if this option is not marked, all R resources will not be visible.
- A new option was added to the main menu *View: R resources visible*.

1.19.1.9 (Jan/25/2007)

- Bug(s) fixed:
 - A bug related to *R toolbar* was fixed.
- Parts of the source code were optimized.

1.19.1.8 (Jan/24/2007)

- Parts of the source code were optimized.

1.19.1.7 (Jan/23/2007)

- Bug(s) fixed:
 - Bugs related to selection in 'smLine' mode, were fixed. Many thanks to SynEdit team.
- Parts of the source code were optimized.
- The main component *SynEdit* was updated to the latest version.

1.19.1.6 (Jan/21/2007)

- Bug(s) fixed:
 - A bug related to the project interface was fixed.
 - Small bugs, related to the editor in split mode (vertical or horizontal), were fixed.
- Parts of the source code were optimized.

1.19.1.5 (Jan/09/2007)

- Parts of the source code were optimized.
- Small adjustments were made to the program interface.

1.19.1.4 (Jan/05/2007)

- Parts of the source code were optimized.
- The use of clipboard between R and Tinn-R was enhanced. For example, instructions such as the ones below are now possible:

```
a <- round(rnorm(1000), 2)
b <- round(rnorm(1000), 2)
c <- round(rnorm(1000), 2)
d <- data.frame(a, b, c)
write.table(d, file="clipboard")
```

Tinn-R is frozen until R releases the clipboard. Thanks to Igor Kojanov for pointing out the problem and the solution's direction.

- In the first use of the program, default options were changed.
- The *R/Database* interface was reworked:
 - A new option enabling the user to restore the original database was added.
 - After changes it is necessary to type *Save*. Otherwise, all changes will be lost when closing the interface.
 - The button *Save and close* was removed.

1.19.1.3 (Jan/04/2007)

- Parts of the source code were optimized.
- Menu *R* was a bit reworked.
- Icons were changed.
- In the first use of the program, default options were changed.
- The *R/Database* interface was a bit reworked. It now has a new option enabling the user to restore the original database.

1.19.1.2 (Jan/01/2007)

- The menu *R* was a bit reworked and now has a new option: *Customize*. It enables the user to customize the files *Rcompletion.r* and *Rconfigure.r*. Before that, administrator privileges were necessary to personalize these very useful files.
- The folder where Tinn-R stores its internal files has now a new sub-folder named *custom* that stores the files *Rcompletion.r* and *Rconfigure.r*. It will be also included in backups of the system configuration.

A.17 VERSIONS RELEASED IN 2006 (64)

1.19.1.1 (Dec/27/2006)

- Bug(s) fixed:
 - A bug related to version '1.19.1.0' and 'midas.dll' was fixed: the content of this dll was compiled inside the binary code to avoid its redistribution.

1.19.1.0 (Dec/26/2006)

- Parts of the source code were optimized.
- The DBase database (RCard and RTip) was upgraded to the native XML engine provided by Borland. It is very small, stable and fast.
 - Therefore, do not restore the database from old versions to the new ones.
 - It will be easier to integrate R and Tinn-R in the future.

1.19.0.18 (Dec/13/2006)

- The *Print* resource was a bit reworked: it will always close the print dialogue after pressing the *Print* option.
- The *Application options* was a bit reworked.
- Two non-configurable shortcuts were added:
 1. **CTRL + (: insert (or replace) ()**
 2. **CTRL +): insert (or replace) ()**
- A new folder named *custom* was added to the program folders. It contains two customizable files: **Rcompletion.r** and **Rconfigure.r**:
 1. **Rcompletion.r**: allows the user to customize the new *auto completion* resource (described below);
 2. **Rconfigure.r**: allows the user to customize the R configuration.
- The first approach of *auto completion* was implemented and the default keystroke is **CTRL + J**, but the user can customize it in the *Editor* options: *Options/Main/Editor/Keystrokes*. Use it, after any valid word, by pressing the keystroke. For example:

```
if<CTRL + J> to obtain:  
if (| < )  
  
ifc<CTRL + J> to obtain:  
if (| < ) {  
  
}  
  
fo<CTRL + J> to obtain:  
for (i in 1:i|)  
  
foc<CTRL + J> to obtain:  
for (i in 1:|) {  
  
}  
  
sw<CTRL + J> to obtain:  
switch(|,  
a = ' ',  
b = ' ',  
)  
  
wh<CTRL + J> to obtain:  
i = 0  
while (i < |) {  
  
    i = i + 1  
}  
  
eq<CTRL + J> to obtain:  
\begin{equation}\label{eq_01}  
|\br/>\end{equation}
```

Observations:

1. Only two letters were used to define simple structures (for example: *fo* = *for*, *fu* = *textitfunction*);
2. Considering the above, letter c was added to complete or more complex structures (for example: *foc*, *fuc*);
3. If there are no conflict among the names in the *Rcompletion file*, the first letter is enough. For example: *s* = *swc*. If there are conflicts, it will be necessary to add more letters to complete the differentiation;
4. The | symbol is used to define where the cursor will first stop after the auto completion;

5. To better understand it, have a look at the file `Rcompletion.r` within the folder *custom* where the program was installed and do your own tests.

1.19.0.17 (Dec/10/2006)

- Bug(s) fixed:
 - When both options *Organize screen* and *Return focus* were marked, it triggered a bug, which has been fixed.

1.19.0.16 (Dec/08/2006)

- The *Main menu* was reworked and several options now have a better logic place.

1.19.0.15 (Dec/07/2006)

- The *Main menu* was reworked and several options now have a better logic place.

1.19.0.14 (Dec/06/2006)

- The interfaces *Search text* and *Replace text* were reworked.
- Parts of the source code were optimized.
- It is now possible to increase and decrease (non-permanent) the font size of any individual file easily:

```
<SHIFT + CTRL + +> (plus (+) on the numeric keypad): INCREASE (upper limit is 50)  
<SHIFT + CTRL + -> (minus (-) on the numeric keypad): DECREASE (lower limit is 1)
```

1.19.0.13 (Dec/04/2006)

- All images of the documentation were updated to the latest version.

1.19.0.12 (Dec/04/2006)

- Parts of the source code were optimized.
- The way as the opened files are closed was enhanced. Now, if any file was changed and the user chooses *Cancel*, the current procedure will be canceled too. For example, closing: projects, all right, all left, etc.

1.19.0.11 (Nov/30/2006)

- The *Application options* was a bit reworked. Now the user can set the initialization parameters of Rgui. Set it at *Application options/R/General/Rgui (Parameters and path)*.

1.19.0.10 (Nov/30/2006)

- The *Project* interface was a bit enhanced.
- Parts of the source code were optimized.

1.19.0.9 (Nov/29/2006)

- Bug(s) fixed:
 - A bug related to the *Project* and associated with the prior version of Tinn-R (1.19.0.8) was fixed.

1.19.0.8 (Nov/28/2006)

- The pop-up menus related to the *Project*, and to the *Editor*, were a bit reworked with respect to *Put full path in the clipboard* option.
- Parts of the source code were optimized.
- *Search text* and *Replace text* interface were reworked.

1.19.0.7 (Nov/27/2006)

- If the variable *rGuiPreferred* is empty (as in the first use) in the *ini file*, Tinn-R will check in the Windows registry at *HKLM\R-core\R* to find the path where Rgui.exe was installed. So, if the user want to use Tinn-R with R console, it is no longer necessary to set it in the *Application options*.

1.19.0.6 (Nov/23/2006)

- The command *Send line* to R was enhanced. Now, whenever the user set it not to send comment lines, and the current line is either commented or empty, Tinn-R will search below towards the end of file, for the first not commented or empty line and will send it automatically. Also, it will always search below towards the end of the file, for the first not commented or empty line. So, the user's work when sending line by line will be reduced.

1.19.0.5 (Nov/21/2006)

- The *Application options* was a bit reworked. Now the user can decide whether the position of the Tinn-R and Rgui windows will be organized automatically or not, when Rgui is running. Set it at *Application options/R/Organize*.

1.19.0.4 (Nov/18/2006)

- Bug(s) fixed:
 - A bug associated with the use of CTRL + to insert content of the tip with 'R complex' and another was fixed.

1.19.0.3 (Nov/17/2006)

- Now Tinn-R has four multi-highlighters: *HTML complex*, *PHP complex*, *R complex* and *Sweave*:

```

1. HTML complex = HTML & JavaScript
2. PHP complex = HTML & JavaScript & PHP
3. R complex = R & URI ('<<<' begin URI '>>>' end URI)
4. Sweave = TeX & R ('>>=' begin R '@' end R)

```

URI : Uniform Resource Identifiers.

R: complex: The main syntax is R, '<<<' and '>>>' the tags enable the user to insert a block of URI syntax.

Sweave: The main syntax is TeX, '>>=' and '@' the tags enable the user to insert a block of R syntax.

The SynHighlighterURI unit implements an URI syntax highlighter for SynEdit.

Recognition of URIs is based on the information provided in the document "Uniform Resource Identifiers (URI): Generic Syntax" of "The Internet Society", that can be found at <http://www.ietf.org/rfc/rfc2396.txt>.

Also interesting is <http://www.freesoft.org/CIE/RFC/1738/33.htm> which describes general URL syntax and major protocols.

These protocols are recognized:

```
-----
http://
https://
ftp://
mailto:
news: or news://
nntp://
telnet://
gopher://
prospero://
wais://
```

As well as commonly used shorthands:

someone@somewhere.org
www.host.org

Example of R complex:

```
# <<< http://www.r-project.org/ >>>
# <<< joseclaudio.faria@terra.com.br >>>
# <<< phgrosjean@sciviews.org >>>

mean(rnorm(100))
```

or

```
# <<< http://www.r-project.org/
#      joseclaudio.faria@terra.com.br
#      phgrosjean@sciviews.org >>>

mean(rnorm(100))
```

Therefore, by pressing CTRL and right button of the mouse on any valid link, it is possible to follow the link (browser, client of email, etc).

- Menu *Options* was a bit reworked to enable the user to set *R* or *R complex* as the default syntax.

1.19.0.2 (Nov/09/2006)

- The option *Automatically configure R* (*Rprofile.site*) was enhanced. Now the user can now decide whether the changes will be temporary (on the current session only) or permanent (editing automatically the *Rprofile.site* file). For both, if necessary, the user can edit (customize) the file *Rconfigure.r* in the folder *doc/English* where Tinn-R is installed. So we think this new resource is now more powerful and flexible.
- The *Application options* was reworked. Now it is possible to set the position of the Tinn-R window (top, bottom, left or right with respect to the *Rgui*). Left or right positions are recommended only for very large screens. Set it at *Application options/R/Organize*.

1.19.0.1 (Nov/07/2006)

- One new option `Automatically configure R (Rprofile.site)` was added to the *R* main menu. It will enable the user to automatically configure the `Rprofile.site` file with all packages which are necessary to work efficiently. If any of those are not found, *R* will download and install them when starting.

1.19.0.0 (Nov/06/2006)

- Bug(s) fixed:
 - Bugs when opening files in Tinn-R by double clicking from *Windows Explorer* were fixed. In other words, the interaction with *Windows Explorer* was enhanced.
 - A bug associated with SHIFT+ESPACE was fixed.
- Tinn-R has now a new logo/image identity which reflects its natural evolution.
- The sources codes were deeply remade and optimized.

1.18.7.3 (Oct/30/2006)

- Bug(s) fixed:
 - A small bug was fixed: from now on, whenever the user starts *R* inside Tinn-R, it will open automatically in the SDI mode, otherwise, in the default mode defined in the `Rconsole` configuration file. Also, if any file happens to be open, and `active`, the `R Working Directory` will be set to its path. Otherwise, it will be set to the default path (`../bin`) where *R* was installed.
 - Bugs with TCP/IP protocol (associated with the prior version 1.18.7.2) were fixed: sorry for this. By the way, we would like to remember the users there are still some unsolved problems with *R output* under this protocol. For example, the codes below:

```
f <- function(x)
x/10 + 1
```

or, i.e, any instruction with more than one single line,

```
a <- 0
for (i in 1:20) {
  a = i
  if(a <= 5 ) {
```

```
    cat('a = ', a, '(<= 5)'); cat('\n')
    next
}
if(a == 18) {
    cat('a = ', a, '(= 18)'); cat('\n')
    break
}
}
```

will always generate errors. We've been working hard in order to solve it, but it is not easy! Moreover, consider that this protocol's main objective is to be used with R explorer interface and any other use is still experimental.

1.18.7.2 (Oct/30/2006)

- Bug(s) fixed:
 - A small bug, when closing the *Rgui* and the choice of *Cancel*, was fixed.
- The *Application options* was reworked a little bit. The options enabling the user to set Tinn-R to always show *Tips* and *Completion* were removed. We think it was an annoyance. Now, the triggers can be used to do that.
- One new function was added both in the *R toolbar* and the *R* menu: *Toogle start/close*. It will enable the user to start/close the preferred user *Rgui*.
- From now on, whenever the user starts R inside Tinn-R, it will open automatically in the *SDI mode*, otherwise, in the default mode defined in the *Rconsole* configuration file. Also, if there is any opened file, the *R Working Directory* will be set to its path. Otherwise, it will be set to the default path (*../bin*) where R was installed.
- Several parts of the source code were optimized.
- The performance of the communication with the *R server* (under all protocols) was enhanced. Now the user can try to set the delay for computational synchronization (*Options/Main/Application/R/R server*) to small values (50, 80 or 100 ms, in harmony with the hardware). We recommend testing different delays until a good performance is obtained.
- Icons of *Controlling R* were remade.

1.18.7.1 (Oct/26/2006)

- Many cosmetic changes were made to the program's interface and documentation.

1.18.7.0 (Oct/25/2006)

- All the HTML documentation of Tinn-R was easily remade; we became very tired of the *blue colour*: `txt2tags` is an exceptional tool!
- From now on (Enio Jelihovschi - a new Tinn-R member) is responsible for the production and quality of all Tinn-R documentation. Welcome Enio, we wish you have a good work on board.
- One new function was added both to the R toolbar and the R menu: *TCP/IP toggle connection*.

1.18.6.13 (Oct/24/2006)

- Parts of the source code were optimized.
- *Tools/Sort* was enhanced: it now enables you to sort by *Strings*, *Numbers* and *Dates*.
- Popup menu associated with tabs was a bit reworked.
- The speller now will advise whenever it finishes the corrections.

1.18.6.12 (Oct/22/2006)

- Several parts of the source code were optimized.
- Popup menu associated with the *Tabs* was enhanced. It now additionally enables you to close all files located at the right or left from the active tab.
- Main menu *File* was a bit remade.
- *Popup menu editor* was a bit remade.

1.18.6.11 (Oct/20/2006)

- Bug(s) fixed:
 - A small bug associated with main menu, *Tools* was fixed.

1.18.6.10 (Oct/20/2006)

- Several parts of the source code were optimized.
- R explorer performance was enhanced. Now you can try to set the delay for computational synchronization (*Options/Main/Application/R/R server*) to small values (50, 80 or 100 ms, in harmony with your hardware). We recommend testing different delays until you obtain a good performance.
- We also recommend you to use the resources of *R explorer* under TCP/IP protocol (it makes the Rgui console cleaner). By the way, there are still some unsolved problems with *R output* under this protocol.
- Window *About* was remade.

1.18.6.9 (Oct/17/2006)

- Bug(s) fixed:
 - A small bug associated with CTRL + W shortcut closing two files instead of just one, was fixed.
- The *Application options* was somehow reworked. When closing it, the screen will not be organized. Therefore it will be necessary to toggle it on the main menu *View/Organise screen* or in the toolbar *Tools*.
- The options *Send selection* will be enabled only if any selection was done.

1.18.6.8 (Oct/10/2006)

- Bug(s) fixed:
 - A small bug associated with *Status bar* was fixed.
- The *Application options* was reworked. It is now possible to set the position of R (top or bottom) on *Organize screen* resource by using *Application options/R/General/settings*.
- Clipboard actions were enhanced.

1.18.6.7 (Oct/08/2006)

- A new option *Organize screen* was added to the menu *View* and the button associated to it was added to the bar *Tools*. This has been, up to now, an old user's request, and it enables you to put, in a fast way, the R Console on the top of the screen and the Tinn-R on the bottom.
- *Tools/Sort* was enhanced: now it maintains the file in the original *Top line* position.
- The way project files (*.tps) are opened was changed: the textual file (by default) will not be opened anymore. The user, now, can easily edit the file in the *Project* interface.

1.18.6.6 (Oct/06/2006)

- A new option *Sort* was added to menu *Tools*. This enables the user to sort an entire file (any selection mode) or selection (for while only for *smNormal* and *smColumn*).
- *Popup menu editor* was reworked.
- Icons of *R explorer* were changed.

1.18.6.5 (Oct/03/2006)

- Bug(s) fixed:
 - A bug related to *Spell*, that always broke down when finding < or <- symbol, was fixed. The origin of this bug was a component conflict (hard coded) among the HTML brackets, declared as below:
 - A bug associated to *Project/Close entire project* was fixed.

```
const
  OpenBracket: array[THTMLBracket] of PChar=('<', '<!--', '<%');
  CloseBracket: array[THTMLBracket] of PChar=('>', '-->', '%>');
```

and the R assign symbols:

```
<-
<<-
->
->>
```

Therefore, if any file has HTML syntax (with any of those tags above) all text among the declared brackets is free of speller. We believe that will work nicely from now on.

1.18.6.4 (Oct/01/2006)

- The *Shortcuts customization* interface was reworked.

1.18.6.3 (Sep/29/2006)

- The *Shortcuts customization* interface was remade, we think it is now more user friendly.
- There is a new button in the interface above, that enables the user to restore the main default Tinn-R shortcuts.

1.18.6.2 (Sep/28/2006)

- Bug(s) fixed:
 - A small bug associated with the buttons icons of *Tools/R explorer* related with version 1.18.6.1 was fixed.

1.18.6.1 (Sep/28/2006)

- Version 1.18.6.0 was updated, due to some problems detected in file *default.bin*. Accordingly, we strongly recommended that, before using this new version, the user manually removes that file. In order to help the user in finding out where Tinn-R stores all the ini files, a new option was added to the main menu *Tools* named *Ini files*. The *default.bin* file is located in the sub-folder *shortcuts*. Therefore, go to the folder and delete it. Sorry for the inconvenience.

1.18.6.0 (Sep/26/2006)

- Several parts of the source code were optimized.
- Menu and pop-up menu options were remade and some icons replaced/changed.
- A new option *Shortcuts customization* was added to the menu *Options*. This will enable the user to manage (edit, load and save) all actions of Tinn-R interface. The latest in use will be always reloaded when restarting the program.

1.18.5.12 (Sep/15/2006)

- Bug(s) fixed:
 - A few small bugs pointed out by users were fixed.

1.18.5.12 (Sep/13/2006)

- Bug(s) fixed:
 - A small bug, which has been happening whenever the user opens the first file located in the floppy drive and access this floppy in subsequent files, was fixed.

1.18.5.11 (Sep/12/2006)

- Bug(s) fixed:
 - A small bug, related to versions 1.18.5.9 and 1.18.5.10, associated with menu *File/Save as*, was fixed.

1.18.5.10 (Sep/11/2006)

- The command *Send line to R* was enhanced. Now, whenever the user sets it not to send comment lines, and the current is either commented or empty, Tinn-R will search below to the end of file, for the first not commented or empty line.

1.18.5.9 (Sep/10/2006)

- Bug(s) fixed:
 - A small bug associated with dragging files in project interface was fixed. Before, whenever you changed files among groups it would not be associated with changes.
- One new function was added both to the R toolbar and the R menu: *Set work directory (current file path)*.
- The icons associated with *Computer* interface (*Add* and *Remove* favorite folder) were replaced.
- Tinn-R now recognizes all valid extensions for any syntax when saving files. For example, the new valid R extensions are: *.r, *.Rhistory, *.q, *.s and *.ssc; so, whenever you type:
 - **Myfile** - it will be saved as **Myfile.r** (the first extension is the default);
 - **Myfile.R** - it will be saved as **Myfile.R** (R is a valid extension and Tinn-R is now case insensitive for that purpose);
 - **Myfile.Rhistory** - it will be saved as **Myfile.Rhistory** (Rhishistory is a valid extension);

- `Myfile.help` - it will be saved as `Myfile.help.r` (`help` it is not a valid extension and `*.r` is the default);
 - PS: in order to save any file with any extension, you must choose *All* syntax.
- Whenever the *tab file* has the focus, a new associated pop-up menu is available with the following options: *Close*, *Close others* and *Close all*.
 - The way Tinn-R opens and closes a project, was changed. Now, any `*.tps` (from any source) will also be opened in the *Project* interface and all files of the project will be closed only and only if you choose the option *Close entire project*.
 - The *Project* interface was isomehow enhanced. Now, the user can type **DELETE** to delete either groups or files, **INSERT** to add files to selected groups and **<CTRL + INSERT>** to add the current file to the selected group.

1.18.5.8 (Ago/03/2006)

- Tinn-R version 1.18.5.7 was pre-released (restricted to developers only).
- The conflict between the *Editor option/Alt sets column mode* and the possible user option *Format/Selection mode/Column* was fixed.

1.18.5.7 (Ago/01/2006)

- A first approximation to build a speller in Tinn-R was made.
- To install this new resource do the following:
 1. Close Tinn-R;
 2. Download the dictionaries
<http://www.luziuschneider.com/Speller/English/index.htm> that you want and install it from the installer (for example IS-
pEnFrGe.exe).
- It is very simple, works nice, has power and all resources are open-source.
- For a while this component did not recognize the main SynEdit component used for edition. Consequently, it was necessary to do the spelling in two steps, making the correction in a new page (*Spell*) on *Results* interface.

- Whenever it finds any selection, the correction will be made only for it, as well as for subsequent updates (or for subsequent update).
- The basic edition resources of the *Spell* are available in a pop-up menu.
- In the meanwhile, please consider this new resource as still experimental and under development.

1.18.5.6 (Jun/24/2006)

- The highlighters *All* and *Text* were remade.

1.18.5.5 (May/29/2006)

- *Save* and *Save as* were remade. Thank you John for the good suggestion.
- The user can change the selection mode by clicking the mouse anywhere in the *Main status bar*. The order of the changes is sensitive to:
 - Left - corresponds to the menu *Format/Selection mode bottom to top order*;
 - Right - correspond to top to bottom order.
- Tinn-R now recognizes the RGUI JGR. In the meanwhile, please consider it as still experimental.

1.18.5.4 (May/28/2006)

- Bug(s) fixed:
 - Small bugs (detected in version 1.18.5.3) were fixed.

1.18.5.3 (May/27/2006)

- Bug(s) fixed:
 - Small bugs (detected in prior versions) were fixed:
 - * Open PDF files;
 - * Appearance of the panel *Results*;
 - * Icons association from the pop-up menu *RCard*.
- Small changes were made to the icons.
- A single click mouse in the editor gutter now selects the entire line.

- Window *About* was reworked.
- Window *Credits* (previously called *Information*) was updated to reflect the lastest changes.
- The echo of the instructions sent to R interpreter (visible in *R output* window) is now optional in Tinn-R. In order to set it up, you can mark/unmark the option *R output echo on* in *Options/Main/Application/R/General/Settings*. If it is marked the R output window will show the instructions (not for all options of *send to R*) and the R output. Please, remember that all these resources are still experimental and not too interactive.

1.18.5.2 (May/22/2006)

- Bug(s) fixed:
 - Small icon changes and bugs corrections.

1.18.5.1 (May/20/2006)

- Bug(s) fixed:
 - A small bug related with *reload* was fixed. Many thanks to Michael Prager for pointing this out to us.
 - A couple of small bugs (pointed out by users) were fixed.
- Tinn-R versions 1.18.4.6 and 1.18.5.0 were considered pre-released (restricted to developers only).
- All icons were changed. Thanks to Philippe for the hard work of its selection and organization. We hope you like this new look.
- All syntax are now alphabetically ordered.
- Whenever the user chooses *Save* or *Save as* Tinn-R will try to recognize the active syntax and the related file extension.
- The images of the files *ReadMe.html*, *LizesMoi.html*, *LeiaMe.html* and *LeaMe.html* were changed so that they reflect the latest changes.
- Several parts of the source code were optimized.
- *Editor options* interface was a bit reworked.
- The *Search in files* and *R output* interface were reworked.
- Menu *R* was a bit reworked.

- Now, the user can choose which R resources, related to *Send to R* and *Controlling R*, will be visible. To do that, two options are available: *Application options/R/R menu and toolbar* and a pop-up menu associated to R toolbar.

1.18.4.5 (Apr/07/2006)

- The recommendation that, under Windows XP, the user should configure the appearance of windows and buttons to *Classic style of Windows (not for XP style)* is no longer used. In other words, the old conflict among the XPMenu component that Tinn-R uses and Windows the XP skins was fixed.

1.18.4.4 (Apr/02/2006)

- The menu *Help* was reworked and new options were added to it. The basic idea is to show how the HTML files are created by using the tool converter *txt2tags* and, the most important, it now enables the users to help us with the constructions and corrections of these files. So, sorry for any language mistake, but my native language (JosÃ© ClÃ¡udio Faria) is not English and I don't want to annoy anyone, anymore, with English corrections. If possible, please help us, because these documents could be useful to many other users.
- Tool *Computer* has now two new buttons that enables the user to add and remove favorite folders.
- Application options interface was reworked. Now it enables user to set how the DOS console will be shown for compilation.
- As a consequence, the *Tools* menu and *Processing* toolbar also have the related options.
- For both file compilation ('.tex' to '.dvi') or ('.tex' to '.pdf') whenever the *Open always after compilation (option)* is marked, Tinn-R will freeze while waiting for the MikTeX compilation to open the compiled file, in the other way it will not. In another words, whenever the *Open always after compilation (option)* isn't checked Tinn-R will not freeze and the user can continue normally his job.
- Tinn-R now is closed with //Yap (Yet Another Previewer)// as DVI viewer, because it is the default released with [MikTeX](#) distribution, and it is a very nice and fast DVI viewer. Tinn-R will open only a single instance of the Yap, but allowing as many files as the user whishes.

- If the user uses the function `utils::setWindowTitle(paste("-", getwd()))` in Rprofile.site, Tinn-R still recognizes Rgui in SDI mode (it was not the case in previous versions between 1.18.XX and 1.18.4.3).

1.18.4.3 (Mar/28/2006)

- Tinn-R versions 1.18.3.1, 1.18.3.2, 1.18.4.0, 1.18.4.1 and 1.18.4.2 were considered pre-released (restricted to developers only).
- The structure of the Tinn-R ini files was changed again:

```
..\Tinn-R           : ReadMe.txt file;
..\Tinn-R\data     : dbRcard.dbf, dbRcard.dbt, dbRCard.ndx, dbRtip.dbf,
dbRtip.dbt, dbRTip.ndx and ReadMe.txt files;
..\Tinn-R\ini       : Shortcuts.tinn, Tinn.ini and ReadMe.txt
..\Tinn-R\ini\colors : customColors.ini and ReadMe.txt
..\Tinn-R\ini\syntax : C#.ini, C++.ini, ... , XML.ini and ReadMe.txt;
..\Tinn-R\ini\syntax bkp : temporary syntax file and ReadMe.txt;
..\Tinn-R\syntax    : deplate.xml, txt2tags.xml and ReadMe.txt.
..\Tinn-R\temp       : temporary files.
```

So, the old backups will not be compatible anymore with this and future versions. This version will recognize the basic old system configurations, but not all syntax preferences. Sorry for the inconvenience, but, it was necessary. From now on, Tinn-R will make real system (all ini files) backup of your settings and preferences.

- Two useful tools for file conversion were added: `deplate` and `txt2tags`. For deplate, the extension `.dlpt` was proposed, and for `txt2tags` Tinn-R recognizes `.t2t`. The basic highlighters, based on XML, were proposed for both. We hope users can help us in the development. Also, the interface for syntax colors preferences for both are different from the already known Tinn-R interface. So, please consider that it is working nicely, but that it will be temporary.
- Tinn-R now enables you to compile `LATEX` files with `MiKTeX` and view the DVI and PDF results (Yap and Acrobat) and also to see HTML files in your preferred browser starting from Tinn-R.
- Application options interface was reworked. It now enables you to set the basic preferences for file conversion, file compilation and viewers.
- As a consequence the `Tools` menu and toolbar have new options to file conversion, compilations and file view.
- All help files were changed from `'.txt'` to `'.html'` using the new `txt2tags` tool inside Tinn-R: we hope you like the new resources.

- Several parts of the source code were optimized.
- Menu *Web* was reworked.
- Menu *Help* was reworked.

1.18.3.0 (Mar/16/2006)

- The structure of the Tinn-R ini files was changed:

```
.. \Tinn-R           : ReadMe.txt file;
.. \Tinn-R\data      : dbRcard.dbf, dbRcard.dbt, dbRCard.ndx, dbRtip.dbf,
dbRtip.dbt, dbRTip.ndx and ReadMe.txt files;
.. \Tinn-R\ini        : Shortcuts.tinn, Tinn.ini and ReadMe.txt
.. \Tinn-R\ini\colors   : customColors.ini and ReadMe.txt
.. \Tinn-R\ini\syntax    : C#.ini, C++.ini, ... , XML.ini and ReadMe.txt;
.. \Tinn-R\ini\syntax bkp : temporary syntax file and ReadMe.txt;
.. \Tinn-R\temp         : temporary files.
```

Therefore, old backups will not be compatible with this and next versions anymore. This version will recognizes the basic old system configurations, but not the syntax preferences. Sorry for the inconvenience, but they were necessary. From now on, Tinn-R will make real system (all ini files) backup of your settings and preferences.

- *Syntax color preferences* interface was reworked and it has new options.
- Tinn-R has three multi-highlighters: SWeave, HTMLcomplex and PHPcomplex:
 1. SWeave = TeX & R
 2. HTML complex = HTML & JavaScript
 3. PHP complex = HTML & JavaScript & PHP

These highlighters have no priorities when you set the syntax color preferences. Thus, if you change the color preferences of any of these multi-highlighters (SWeave, HTML complex and PHP complex) these settings will be valid only in the current Tinn-R session and will not be saved when Tinn-R is closed. So, if you want to make permanent changes, set the preferences from all simple highlighters (R, TeX, HTML, JavaScript and PHP).

- Menu *Option/Syntax* was reworked.
- A lot of the source code was optimized.
- All Tinn-R color dialogs were reworked and they automatically save and recover all preferred colors.

1.18.2.1 (Mar/08/2006)

- Some aspects of *Syntax color preferences* interface were optimized.
- *Replace text* and *Search text* interface now recognize if more than one text line was selected and set the option *Selected text only* automatically.

1.18.2.0 (Mar/06/2006)

- Now Tinn-R has three multi-highlighters: Sweave, HTMLcomplex and PHPcomplex:
 1. Sweave = TeX & R
 2. HTMLcomplex = HTML & JavaScript
 3. PHPcomplex = HTML & JavaScript & PHP

These highlighters have no priorities when you set the syntax color preferences. Thus, if you change color preferences of any of these simple highlighters (Sweave, HTMLcomplex or PHPcomplex) these settings will be valid only in the current Tinn-R session and will not be saved when Tinn-R is closed. So, if you want to make permanent changes, set the preferences of all these simple highlighters (TeX, R, HTML, JavaScript and PHP) from the multi-highlighters (Sweave, HTMLcomplex and PHPcomplex) respectively.

- Several parts of the source code were optimized.
- *Tools/Project* interface was reworked; now it has new resources.

1.18.1.10 (Feb/18/2006)

- Bug(s) fixed:
 - A small bug with Sweave highlighter related to color preferences was fixed. In fact, Sweave highlighter is a multi-highlighter made from prior R and TeX highlighters. So, if you set the color of the background to all TeX elements (comment, space, etc) with the same color (gray for example), you can get the Sweave syntax with two background colors: one for TeX (gray) and another to R (white).
 - A small bug with *View/Line numbers* was fixed.

1.18.1.9 (Feb/18/2006)

- Bug(s) fixed:
 - A small bug with menu *Edit/Undo* and *Edit/Redo* was fixed. Many thanks to users for pointing this out.

- A small bug with menu *Edit/Undo* and *Edit/Redo* was fixed.
Many thanks to users for pointing this out.
- Sweave syntax highlighter was added to Tinn-R with .rnw and .snw extensions. For Tinn-R the delimiters are:

```
>>= start a R block
@ close it.
```

- Sweave provides a flexible framework for mixing text and S code for automatic document generation. A single source file contains both documentation text and S code, which are then woven into a final document containing:

1. The documentation text together with
2. The S code and/or
3. The output of the code (text, graphs) by running the S code through an S engine like R.

Hence, the full power of L^AT_EX (for high-quality typesetting) and S (for data analysis) can be used simultaneously. For more information see ?Sweave from R.

- All combo-box related with files and extensions were reworked and now they are simpler.

1.18.1.8 (Feb/12/2006)

- Bug(s) fixed:
 - Small bug with *Copy* of pop-up menu *R card* was fixed.
 - A couple of small bugs (pointed out by users) were fixed.
- Main menu *Help* and *Web* were reworked.
- Tinn-R and SciViews-R GUI performance was enhanced.
- Small correction with *R explorer* interface related to enabled/disabled options.
- Two new options were added to menus *R* and *R toolbar*: *Edit* and *Fix*. These options enable you to edit or to fix R objects (if they exist in the R environment).

1.18.1.7 (Jan/14/2006)

- Small correction with *R card* and *R explorer* interface related to enabled/disabled options.

1.18.1.6 (Jan/10/2006)

- Bug(s) fixed:
 - Small bugs pointed out by users were fixed. Many thanks users!
- A new option *GUI Wiki* was added to menu *Web/R-information on line*. This Wiki was designed mainly to deal with R beginners problems. Although we would like to emphasize the use R GUIs (Graphical User Interfaces), this Wiki is not restricted to those GUIs: one can also deal with command-line approaches. Thus, the main idea is to have material contributed by both beginners, and by more advanced R users, that will help novices or casual R users.
- In menus *R* and *R toolbar* the position of the buttons was a little changed for a more logical arrangement; additionally, some icons were changed and a new and very useful option was added: *Example*.
- Print/preview, R explorer and R card Interfaces were reworked.
- Under TCP/IP some functions to *Controlling R* were reworked, which are now more user friendly.
- Horizontal and vertical splits were deeply reworked. Now they are more user friendly. In consequence, the main menu *View* was changed.
- The pop-up menu of the open files was enhanced and two new options were added: *Close all* and *Close others*. The *Close others* was also added to the Main menu *File*. The first enables you to close all files and the second, all files except the active (current) one. Please note that both will be enabled only if two or more files are open.
- A lot of procedures and functions were optimized.

1.18.1.5 (Jan/01/2006)

- A Tinn-R card, *Help/Tinn-R card (PDF)*, was added to the project. Many thanks to Suresh Krishna.
- New interface/options for printing, which now enables you to print preview with new resources. In consequence, menu *File/Print preview* was removed.

A.18 VERSIONS RELEASED IN 2005 (25)

1.18.1.4 (Dec/24/2005)

- The extensions ‘*.s’ and ‘*.ssc’ were added to R syntax.
- New interface/options for printing, which now enables you to choose the file’s printing range: All, Selection and Pages.

1.18.1.3 (Dec/17/2005)

- The changes (since Tinn-R 1.17.2.6) have been considered stabilized.
- A new option *R syntax as default* was added to the main options and menu *Options*. It enables you to choose if a new file and file without extension will be opened for edition with *R* or *General* syntax as default.

1.18.1.2 beta (Dec/11/2005)

- Tinn-R version 1.18.1.1 BETA was considered pre-released (restricted to beta testers only). Many thanks to all testers.
- The pop-up menu of the *R explorer* was enhanced and new options were added.
- You can now drag and drop objects from the *R explorer* to editor.
- Many procedures and functions were optimized.

1.18.1.1 beta (Dec/03/2005)

- Bug(s) fixed:
 - A small bug with the install program (related to desktop shortcut creation) was fixed. The old shortcut was pointing to *Tinn-R.exe.manifest* file and not to *Tinn-R.exe*.
- An experimental R explorer was added to the program. This was accomplished by adding a new tab named *R explorer* to the panel *Tools*; it enables you to see R objects from Tinn-R. Meanwhile, this explorer must be updated by the user because it is not automatically done by the R server. It was based on the SciViews R explorer, but it is not exactly the same.
- The speed of the procedures/updates with the panel *Tools: Computer* and *Project*, were enhanced.

- A new option *Send to R* was added to the pop-up menu of *Project*. It enables you to send the whole project, individual groups or individual files to the server without opening it for edition.
- A new menu option was added to the *R* menu: *Server: connections and tests*. This menu will open a new window of the program that allows you to connect and test R as server using TCP/IP and DDE protocol.
 - The window *Application options* was somehow reworked to reflect the new features of the program (specially TCP/IP protocol to communicate with R server). This resource is still experimental and, under tests, it shows instability. So, if any problem occurs, disconnect the server *Server: connections and tests* and all new resources (using another option - DDE and/or temporary files) will work. In the future the use of TCP/IP protocol by Tinn-R will be enhanced.

- An experimental code completion was added. For a basic use:

1. Set the options on *Application options*;
2. Like *Tip*, if you check *Always shown*, after the data object type and '\$' related completion will be shown;
3. Select the desired completion and press ENTER to make it work.

A tip: my preferred shortcuts are: ``CTRL + D`` for tip after //(/// and ``CTRL + SHIFT + D`` after '\$' for code completion.

- A new menu option was added to the *View* menu: *R output*. This menu will open a new window on the bottom of the program that allows you to receive the output of the R as server using only TCP/IP protocol. It is still experimental.
- The communication with R under DDE protocol was enhanced: it is now faster and more stable.
- Tool *Computer* now allows you to open (with double click or drag and drop) any type of files, and no only the ones recognized.
- Two new syntax highlighters were added to Tinn-R:
 1. Fortran: useful to R developers;
 2. Visual basic: still lives.
- A new pop-up menu was added to the list view of Tools *Computer* that enables you to refresh and to choose the view style.
- Many procedures and functions were optimized.

1.17.2.6 (Oct/27/2005)

- Bug(s) fixed:
 - A small bug related to *On top* was checked and Tinn-R sometimes minimizes and others times it doesn't (hides in the task-bar), was fixed. Many thanks to Mihai Nica for pointing that out.

1.17.2.5 (Oct/25/2005)

- The memory use increases (16 k every second) related to some prior versions was fixed. Many thanks to users for pointing this out.
- The SynEdit.pas unit component was updated to the latest version 1.428 2005/10/07 21:16:10. So, gutter width was always calculated using editor font was fixed.
- Shortcut for *Format/Block/Mark* was changed from CTRL + ALT + B to CTRL + ALT + S.

1.17.2.4 (Oct/08/2005)

- Bug(s) fixed:
 - A small bug with *Options/Syntax/Color preferences* related to *Cancel* option was fixed.
- The *TeX highlighter* was a little reworked: now it matches brackets. Thanks Sheldon Kelly for suggest it.
- The *Restore* procedure was a little changed: it now checks if any file or project was changed (and not saved) prior to restart the program.
- The database was a little reworked. So, if you had any old Tinn-R version installed, we recommend that you restore the data from this version. To do this, select *Tools/Restore/Database* from the main menu and choose the *data* folder where Tinn-R was just installed, as a prior backup.
- Main program icon was a little changed (the R is now darker).

1.17.2.3 (Oct/03/2005)

- Bug(s) fixed:
 - A small bug with database (related with version 1.17.2.2 only) was fixed.

1.17.2.2 (Oct/02/2005)

- Bug(s) fixed:

- A bug with messages below:

"Severe problem! The program cannot be initiated. Please, contact the developers."

"Access violation at address 'hexa number' in module 'Tinn-R'. Read address 'hexa number'."

It was now fixed. This bug was related to more than one file were selected from the explorer to be opened and Tinn-R was not running.

So, if Tinn-R is not running (if and only if), avoid to make it because it will open one instance of Tinn-R and one file (the first in alphabetic order) only. BTW, if Tinn-R is running, you can select simultaneously any desired number of files.

- A small bug when you did a global replace in Tinn-R, i.e., Search/Replace/OK/Yes to all, and get the following error message: Assertion failure (D:\...\SynEdit.pas, line 1460), was fixed: thanks John!

- Tinn-R versions 1.17.1.2 and 1.17.2.1 were considered pre-released (restricted to beta testers only). Many thanks to all testers.
- The picture, when opening the program, doesn't appear anymore.
- Main program icon was changed: I hope you like the new one. BTW, I got problems because some parts of the Windows showed the old icon and others the new. I spent a lot of time to find the original problem: ShellIconCache.

SO, IF YOU GET INCORRECT ICON DISPLAYED ON WINDOWS, AFTER INSTALLING THE NEW VERSION OF TINN-R, PLEASE, READ THE INSTRUCTIONS BELOW:

For acceleration of the show of the icons Windows stores images in the ICON CACHE (ShellIconCache) a hidden icon cache file in your windows directory.

Sometimes the icon of the object changes, but Windows shows the old icon instead of the former one. To solve this problem we suggest to use the program IconChanger (shareware available at <http://www.shelllabs.com/>).

If you have just installed Tinn-R with new icon but Windows has not changed the image, the guide advises you to choose REBUILD ICON CACHE and if it will not help then choose REMOVE ICON CACHE.

If you have chosen REBUILD the icon cache will start reconstructing at once. If you have chosen REMOVE, you will see the warning message. Choose YES, and then you should restart your computer.

- If Tinn-R is closed when maximized, when restarted, it will open maximized but it will remember the last position when not maximized.
- A new shortcut CTRL + ENTER was added. When pressed, Tinn-R sends the current line (entire) to R interpreter and adds a line break at cursor position.
- A new toolbar with Undo and Redo was added.
- A new button was added to the *Misc toolbar* that allows you to choose how Tinn-R will behave after sending anything to R interpreter, if checked, the focus will return to Tinn-R.
- The function *Clear all* was updated and it is now faster. If any problem occur with different machines and OS, please, tell us.
- The menu *Web* was a little reworked and a new option is available: *Web/Tinn-R* with links to home page (sciViews), sourceforge server and check for update (from sciViews server).
- A new menu option *Help/Tinn-R citation* was added. It enables you to put the Tinn-R citation in the clipboard.
- After restoring the configurations file, Tinn-R now can close and restart itself.

1.17.1.1 (Aug/28/2005)

- Bug(s) fixed:
 - A small bug with recent files was fixed.
- Tinn-R versions 1.16.1.9 beta and 1.16.1.10 beta were considered pre-released (restricted to beta testers only): thanks to all testers.
- The SynEdit component was updated to the latest version (v2.0.1 beta).
- The windows *Application options* and *Editor options* were a little reworked.
- The main menu *File* was a little reworked.

1.16.1.8 (Aug/07/2005)

- Bug(s) fixed:
 - A small bug with restore database was fixed.

1.16.1.7 (Aug/05/2005)

- Tinn-R version 1.16.1.6 beta was considered pre-released (restricted to beta testers only): thanks to all testers.
- The *R highlighter* was a little reworked: identifiers with the minus signal '-' (ex: codes-deprecated) will not be recognized, any more.
- Tinn-R team has a new member working on the source code: welcome Huashan Chen.
- Tinn-R project (source and bin) is now also available from [SourceForge](#). Thanks Huashan.
- The incompatibility of Tinn-R with WinNT 4.0/SP6 has been solved. Thanks Uwe.
- In *R toolbar* the position of the buttons was a little changed for a more logical arrangement.
- After installation, the *Tinn-R/data* folder contains data.zip (102 KB). When starting Tinn-R, the files it contains are automatically unpacked, and for each user four files are generated: dbRcard.dbf, dbRcard.dbt, dbRtip.dbf and dbRtip.dbt plus two (dbRcard.ndx and dbRtip.ndx) index files. This way, each user has an independent database.
- Also, now each user has independent configurations files.
- Considering the two topics above, Tinn-R is now multiuser. So, for instance, you can have an administrator account with full access to all files for installing programs and maintaining the machine. But for everyday work, you can log in as `main user` or `normal user` (with restricted access) and run Tinn-R without problems in those restricted environments. This feature is very useful to educational purposes in statistical laboratories.
- Only three folders are generated by Tinn-R for each user:
 - Windows XP:
 - * *Drive:\Documents and settings\UserName\Application data\Tinn-R\ini* with two files *Tinn.ini* and *Shortcuts.tinn*;
 - * *Drive:\Documents and settings\UserName\Application data\Tinn-R\data* with six database files (see above);
 - * *Drive:\Documents and settings\UserName\Application data\Tinn-R\temp* for essential temp files (will be automatically removed by Tinn-R after each session).
 - Windows NT:
 - * *Driver:\Winnt\Profiles\UserName\Application data\Tinn-R\ini*;

- * *Driver:\Winnt\Profiles\UserName\Application data\Tinn-R\data;*
- * *Driver:\Winnt\Profiles\UserName\Application data\Tinn-R\temp.*

- The database component (TDBF) was updated to the latest version.
- Menu *Tools* was reworked a little and new options were added to backup/restore your personal database and configuration files.

1.16.1.5 (Jul/15/2005)

- Bug(s) fixed:
 - A small bug with project (*Save* and *Save as*), related to automatic extension '.tps', was fixed.
- Tinn-R now works with R in MDI mode if the device graphic is maximized with no more limitations of the number of device graphic, as pointed in the Tinn-R 1.15.1.7 (07 Apr 2005).
- If you use the function `utils:::setWindowTitle(paste("- ",getwd()))` in Rprofile, Tinn-R still recognizes Rgui (was not the case in previous versions).
- A new menu option (and respective button and hotkey) was added to the main menu: *R/Controlling R/List structure of selected variable* that allows you to list the structure of any R object/variable.
- A new menu (and respective pop-up menu) was added to the main menu: *File/Put full path in clipboard*. This option allows you to put the full path of the file in the clipboard with two options: Unix mode (../..) or Windows (..\..) (useful to get full path of data files).
- In *R toolbar* the position of the buttons were a little changed for a more logical arrangement.
- The menu *Format/Block* was a little reworked.

1.16.1.4 (May/29/2005)

- Bug(s) fixed:
 - Small bug with *Send marked block* was corrected.
 - Small bug with *Search* and *Search in files* were corrected.
- A new button was added to the *Misc toolbar* that allow you unmark all marks.
- Menu *Format* was reworked a little bit.

1.16.1.3 beta (May/22/2005)

- Tinn-R versions 1.15.1.8, 1.15.1.9, 1.15.1.10, 1.15.1.11, 1.16.1.1 and 1.16.1.2 were considered pre-released (restricted to beta testers only).
- The Tinn-R installer now proposes to associate Tinn-R with '.Rd' files (the R help source code).
- The menu *Format* was changed. There are now two new options:
 1. *Format/Block/Mark*;
 2. *Format/Block/Unmark*.

The first menu allows you to select a given block of code (line start and line end) to send it at once to the R interpreter and the last menu entry allows you to clear an existing marked block. Bookmark 0 is used to mark the beginning of a marked block and bookmark 1 is used to mark its end. *Unmark* remove them only if they were marked with the *Block/Mark* tool. Otherwise, they are treated as simple bookmarks. Note that, whatever the way you defined them, you can always change their position as usual. There is a distinct marked block defined for each open file, and the tools are enabled only if a block is marked for the current active document.

- As a consequence the *R* menu and toolbar have new tools to send the marked block at once *R/Send to R/Marked block (source)* or line by line *R/Send to R/Marked block* to the R interpreter.
- Another new tool was also added to send the current selected text as a source file to R (and not line by line, as usual).
- The *Main toolbar* was a little reworked to support the new mark block option.
- The accuracy of all functions (send to, or, control R) interacting with R was enhanced.
- The stability of the database, respect to duplication of keys, was enhanced.
- The option *Spanish* was added to the menu *Help/From this version*. It contains:
 1. Lea me.html;
 2. FAQ.txt;
 3. Palabras reconocidas.r.

The translation was made by Jairo Cugliari: thanks Jairo, very much!

- A R card was added to the program. This was accomplished by adding a new tab named *R card* to the panel *Tools*. The R card was based on two R card already published: R/Rpad Reference Card by Tom Short and R reference card by Jonathan Baron.
 1. It was made using a user-expandable database (DBase);
 2. The component used is named TDBF, it is free (open source), does not use BDE and it is not necessary to have the database server. The DB engine code is compiled right into the Tinn-R executable. TDBF is a native data access component for Delphi, BCB, Kylix and FreePascal. More information can be found [here](#).
- As a consequence, the *R* menu was a little reworked. You have now two options for the database:
 1. *R/Database/Tip* for tip management;
 2. *R/Database/Card* for R card management.
- A freeware resource provided by Angus Johnson was added to the program. It is a very useful Generic Diff Format (GDIFF) named TextDiff. The GDIFF format can be used to efficiently describe the differences between two arbitrary files or folders. The format does not make any assumptions about the type or contents of the files, and thus can be used to describe the differences between text files as well as binary files. The work was made adapting the sources code of the demo project of the component to the Tinn-R project: thanks Angus, very much!
- As a consequence, the *Tools* menu has a new tool: *Tools/Differences*.
- Tinn-R is now more flexible if the screen is split: the functions *Send to R* and *Controlling R* works for both (top or below, left or right) frames.
- The *Ascii chart* is now more flexible.
- The menu *Format/Selection mode* was reworked, new and useful shortcuts were added (see in *Main/Editor/Keystrokes*): thanks Zoltan Butt.
- A new field (third field from left) was added to the status bar with three possible values corresponding to the current selection state of the editor:
 1. smNormal : selection in normal mode;
 2. smLine : selection in line mode;

- 3. smColumn : selection in column mode.
- The files *Tinn-R_Read me.html*, *Tinn-R_Leia me.html* (in Portuguese) and *Tinn-R_Lisez moi.html* (in French) in the \doc subdirectory of Tinn-R were updated.
- Seven new syntax highlighters were added to Tinn-R:
 1. Rd files (based on the existing TeX highlighter);
 2. Tcl/Tk;
 3. Ruby;
 4. TeX;
 5. Python;
 6. Bat;
 7. HP48.
- The picture, when opening the program, was changed. Tinn-R program is searching for a new logo/image identity. This "tinny" (two n intentionally), but colorful bird, is a good symbol for the "tinny" Tinn-R, but rich in nice features regarding the edition of R code!
- The program user interface was reworked a little bit.
- Two hotkeys were changed:

```
<ALT+LEFT> : Tools align left;
<ALT+RIGHT> : Tools align Right.
```

- The automatic extensions for all *Save* action was improved. Now, if you select *All files (*.*)* you can save the file with any desired extension or even with no extension at all. Thanks Posta Giovanni.
- The structure of the *Tip database* was improved. The tip is not limited to 254 characters anymore, because it is now of *memo* type.

1.15.1.7 (Apr/07/2005)

- Bug(s) fixed:
 - Minor bug with *View/Toolbars/Macro* was corrected.
- Tinn-R now works with R in MDI mode if the device graphic is maximized:
 1. You can work with 1..11 device graphic *ACTIVE* maximized;
 2. You can work with 1..10 device graphic *inactive* maximized.

The caption of \RR{} can be:
RGui - [R Graphics: Device 1..11 (ACTIVE)] or
RGui - [R Graphics: Device 1..10 (inactive)]

- The state of the Caps Lock key (keyboard) doesn't influence any more the *Send to R* and *Controlling R* functions.
- *Tools/Project* interface was improved with the replacement of the combo-box by a new dropdown button.
- A new toolbar named *Tools* with two buttons (*Toggle tools visible* and *Align tools right/left*) was added to the main toolbar, in consequence the pop-up *Tools* was removed.
- *Project/Recent* is now checking for changes in current project.
- The *View/Tabs* icon was changed.
- The delay for DDE communication with R (for call tips) is now user-selectable in the *Options/Main/Application* dialog box (you need to load svIDE package from SciViews bundle to use this feature). If Tinn-R seems to freeze and you got no call tip, just increase the delay.
- Menu *Web* and *Help* were a little reworked and new information in French and Portuguese were added.

1.15.1.6 beta (Mar/19/2005)

- Open files in Tinn-R with *Tools panel* was reworked:
 1. A file must have no extension or one that Tinn-R recognizes;
 2. If the file is not opened there, you can open a copy by double clicking or dragging it into the main form;
 3. With file already opened, you can now open a copy by double clicking or dragging it in the editor area in Tinn-R;
 4. You can drag a project (all files of the project will be opened), a group (all files of the group will be opened) or an individual file.
- A new pop-up menu is available for *Tools*. This menu is related with visibility and position of the *Tools panel*.
- The menu *Options* and *View* were reworked.

1.15.1.5 beta (Mar/17/2005)

- *Project* interface was a little reworked and two new options were added: *Expand all* and *Collapse all* groups.

1.15.1.4 beta pre-release (Mar/12/2005)

- *Application options* was reworked and an option *Delay for synchronization* was added. Because DDE (Dynamic Data Exchange) through tcltk and svIDE packages consumes a certain time - that is variable between different computers - this option allows the user to customize the Tinn-R delay with the R as server with call-tip. So, it is very important to adjust this delay until getting high functionality and performance.
- *Project* interface was reworked:
 1. Only the file names are shown (no more full paths);
 2. All nodes of the project can be dragged:
 - a) If you drag a project all file will be opened;
 - b) If you drag a group all files of this group will be opened;
 - c) if you drag a single file it will be opened.
- The *Search results* interface was reworked and two old bugs (all versions of Tinn-R that I don't know) were fixed.
- In *Tools panel/Computer* a double click on the Tinn-R project files (*.tps) will open the project interface. On the other hand, if you drag this file, the context of the project file will be opened for manual edition.

1.15.1.3 beta pre-release (Mar/06/2005)

- Changes to projects are monitored now.
- To open files in Tinn-R with *Tools panel*:
 1. A file must have no extension or one that Tinn-R recognizes;
 2. If the file is not opened there, you still can drag it into the main form;
 3. If any file is already opened, you can open a copy by double clicking or dragging it to the page control or to the main menu.
- The R highlight dictionary was updated and about 270 new functions were added to be compatible with R parameters completion proposal (RPCP) database.

1.15.1.2 beta pre-release (Mar/05/2005)

- Bug(s) fixed:
 - Small (and old) bug with painted symbols '(', '[', '{' inside the gutter was corrected: thanks Marco!
- A new folder named *res* with a single file (Tinn-R_img.bmp) was added to the Tinn-R program.
- The Tinn-R binary is about 350 Kb smaller than version 1.15.1.1.
- The *About* box was reworked.
- The project combo box was reworked. It is not possible to drag it any more.
- *RegEx* was reworked: it is now possible to paste clipboard and use carriage return in it.
- It is now possible to place the *Tools* panel at right or left of the main window.
- Two new hotkeys were added (not user-configurable):

```
<CTRL + ALT + LEFT> : Tools align left;  
<CTRL + ALT + RIGHT> : Tools align Right.
```

- It is possible to dock the R toolbar only at the opposite side of the *Tools* panel, not on the same side.

1.15.1.1 beta pre-release (Mar/01/2005)

- A picture was added when opening the program. Thanks Carolina (my daughter) for basic art creation.
- A new file was added to the project: Tinn-R_shortcuts.txt. It is possible to load the file from *Help/Tinn-R/Shortcuts*.
- *Open dialog* for Open file and *Add file* to project now allow the selection of multiple files.
- The database was reworked and a new information (location) was added.
- Tab order of the search/replace interface was reworked.

- Due to the growth of the code source, it is not possible any more to maintain Tinn-R code synchronized with the original Tinn project. So, Tinn-R is now a new open source project. The version numbers is thus changed. The new convention is: AA.BB.CC.DD S/B

```
That is, AA : major version  
BB : minor version  
CC : release  
DD : build  
more : beta or pre-release version
```

- The status bar was reworked.
- A new menu *Web* was added.
- The *About* dialog box was reworked.
- The menu *Project* was completely reworked.
- New panel named *Tools* (with two tabs: *Computer* and *Project*) was added. This feature will grow up to enhance the interaction with R interpreter (progressive inclusion of SciViews-R dock Window feature like the object explorer).
- A new hotkey (not user-configurable) was added: CTRL + / (divide on the numeric keypad) show/hide the *Tools* panel.
- Second version of the R parameters completion proposal (call-tip). It now works with a user-expandable database (DBase) and with a R server (DDE - Dynamic Data Exchange through tcltk and svIDE packages). That is, this version communicates with R to get R function arguments directly if a function is not defined in the database. We thus have: database (priority 1) and R server (priority 2). To use the R server, the package svIDE (SciViews bundle) must be loaded in the current R session.

0.0.9.4 r1.15 beta 2 (Feb/11/2005)

- Bug(s) fixed:
 - Small bug with CTRL + Y (delete entire line), CTRL + T (delete word right from cursor) and CTRL + SHIFT + Y (delete words from cursor to the end of line) that did not enable the save options, was corrected.
- The position of the cursor is now preserved when using CTRL + * (multiply on the numeric keypad) to insert/replace text with parameters of the current R function.

- Send line now works when the cursor is on the end the line.

0.0.9.3 r1.15 beta 1 (Feb/11/2005)

- The problem with Tinn-R 0.0.9.3 r1.15 beta 1 (released 30 Jan 2005) that stopped working due to expiration of trial period for Delphi was solved.
- The first version with the R parameters completion proposal (Call-tip):
 1. It was made with a user-expandable database (DBase), that is, this version does not communicate with R to guess R function arguments;
 2. The component used is named TDBF, it is free (open source), does not use BDE and it is not necessary to have the database server. The DB engine code is compiled right into the Tinn-R executable. TDBF is a native data access component for Delphi, BCB, Kylix and FreePascal. More information can be found [here](#).
- A new hotkey (not user-configurable): CTRL + * (multiply on the numeric keypad) inserts/replaces text with parameters of the active function (a call-tip must be visible).
- The *Application options* was reworked. It is now possible to set the basic application options for R Call-tip.

0.0.9.3 r1.14 (Feb/11/2005)

- Bug(s) fixed:
 - Some minor bugs were corrected and the buttons appearance was a little modified.
 - A bug with menu *File/Recent files* not showing at startup was fixed.
 - A bug which added the file extension twice was fixed. This bug was related to full name including the extension, like *myfile.r*, where Tinn-R recorded it as *myfile.rr*.
 - A bug with save was fixed.
 - A bug with Pascal files extensions was fixed.
 - A bug with shortcut of the menu *Edit>Select all* was fixed.
- The problem with Tinn-R 0.0.9.3 r1.14 (released 30 Jan 2005) that stopped working due to expiration of trial period for Delphi was solved.

- New interface/options for printing, it remembers latest choices/preferences.
- *About* dialog box was reworked.
- The program user interface was a little reworked.
- Most of the interactive R tools do no require any more that files be saved with a valid S language extension (.r or .q).
- The R highlight dictionary was updated:
 1. About five hundred words were added;
 2. Fixed a bug with '_' in keywords like *.decode_package_version*.
- This version was compiled with Delphi 7 (Delphi 5 was used for previous versions).
- With Delphi 7, the default dialogs were re-established. So, psvDialogs is not used any more.
- Menu *Format/Block/Uncomment/Firsts occurrence* was renamed to Uncomment First Occurrence.
- Add menu *Help/Tinn-R/Changes on Line*.

A.19 VERSIONS RELEASED IN 2004 (16)

0.0.9.3 r1.14 beta 9 (Dec/04/2004)

- Bug(s) fixed:
 - Fixed a bug that caused R GUI window to shrink or expand to a certain size, whenever code was submitted to Rgui in MDI mode: thanks Steven!
 - Small bug with the function SEND FILE fixed when the file is set READ ONLY.
- For a while, it is strongly recommended to set the BACKGROUND COLOR of ALL SUPPORTED LANGUAGES to WHITE.
- The defaults Delphi dialogs were replaced by a new, free and good suite of components called psvDialogs written by Serhiy Perevoznyk. Thanks Serhiy! So, the Dialogs are now more pretty.
- Add menu *Format/Block/Uncomment first occurrence*.
- Search/Replace was improved with the solution proposed for the 0.0.9.0 r1.09 issue. The user can now work in all windows (even horizontally or vertically splitted): thanks the Tinn Team.
- The user can now work in all window (even horizontally or vertically splitted) for comment/uncomment blocks.
- The syntax selector on file load was reworked.
- Added *C# highlighter* instead of that mocked up Java one I had been using like a fool.

0.0.9.2 r1.14 beta 8 (Oct/2004)

- Menu *R>Show and Hide R Toolbar* was eliminated. This option is changed to *View\Toolbars\R* with others toolbars. It is a more logic place for this option.
- Menu *Editor: line below and top* was replaced by *Editor: Current line to top*. This improves work alternate with *Send: Line to end page* and *Editor: Current line to top*.
- The send line code was improved.
- The component SK donated by MJT Net was replaced by the snd-Keys32.pas written by Ken Henderson and adapted for use in Tinn-R. This change impacts all send and controlling R function.

0.0.9.1 r1.14 beta 7 (Oct/2004)

- Bug(s) fixed:
 - Minor bug with menu *R/Controlling R* (enabled/disabled) was corrected.
- Tinn and Tinn-R can now run together.

0.0.9.1 r1.14 beta 6 (Sep/2004)

- Bug(s) fixed:
 - Minor bugs with general appearance (LikeXP, unlike XP) were corrected.
- Marco de Groot (Matic Software), old Tinn developer, is the new co-author of the Tinn-R: welcome Marco!
- Backup and restore were reworked: now the user can choose the directory.
- Tinn-R is now more user-friendly to run with R: better R open/save dialogs.
- Save file interface was reworked: R files (*.r; *.q) is now the primary option.
- Open file interface was reworked: the three major file extensions are now the first one available:
 1. R files (*.r; *.q);
 2. Text Files (*.txt);
 3. All files (* *).
- Add menu *Help/FAQ*.
- Delays in large files were corrected: thank you Marco de Groot.
- *About* was reworked: mails of all authors are now included.

0.0.9.0 r1.14 beta 5 (Sep/2004)

- Bug(s) fixed:
 - A small bug in resizing the gutter (respect to autoSize and bookmarks) was corrected.
 - A bug with bracket colors was corrected: it eliminated the bracket background when the active line highlighting is set to off.

- General stability was improved.
- Appearance was relooked.

0.0.9.0 r1.14 beta 4 (Aug/2004)

- A tooltip was missing in the *search in files* toolbar button.
- New tools to save/restore configuration files for Tinn-R are now added.

0.0.9.0 r1.14 (Aug/2004)

- Bug(s) fixed:
 - A bug when R and a graph device are maximized and the latter had the focus is now fixed (nothing happened in this case when sending data from Tinn-R to R).

Note: Tinn-R 0.0.9.0 r1.13 is now frozen and goes through a wider beta test
Tinn-R 0.0.9.0 r1.14 and over are considered development versions; they may possibly show some instabilities during development.

- Specific R hotkeys are now automatically activated/deactivated.
- Hotkeys are now permanently set.
- For everyday work, use Tinn-R 0.0.9.0 r1.13

0.0.9.0 r1.13 (Jul/2004)

- Two new functions were added both to the R toolbar and the R menu: *Print content of selected variable* and *List names of selected variables*. Just experiment with these tools to figure out what they do (for instance, enter *iris* in a .R file edited in Tinn-R and try these two tools while this word is selected).
- The icon for *Help on selected word* was changed, and the *Send file* icon in the R toolbar now has a transparent background.
- The R highlight dictionary was updated (about a hundred words were added).

0.0.9.0 r1.12 (Jun/2004)

- Bracket matching is reworked back to a more classical presentation: match brackets are now presented in bold and in a contrasting color (i.e., red if symbols are in a different color, or blue if symbols are colored in red).

- The prior content of the clipboard is now preserved in all *Send to R* functions.
- Now the R console is completely cleaned when you use *Clear all* (previously, commands to clear the console were left at the command line).

0.0.9.0 r1.11 (Jun/2004)

- Bug(s) fixed:
 - Minor bug with match bracket was corrected.
- On the menu *Options/Application options* it is now possible to indicate if R resources are visible or not (switch between plain Tinn and Tinn-R).
- The menu *Tinn to R* was renamed *R*.
- Menu *Tools* was reworked. Tools/Initiate or Close preferred Rgui are now included in the *R* menu.
- *Copy Formatted* was added to the pop-up menu.
- Behavior of the R Toolbar was improved.
- Automatic match brackets was reworked: current bracket is shown surrounded by a red rectangle and previous match brackets are shown surrounded by a background symbol color of the specific highlighter (user defined). Thus, if the active line is highlighted (*Options/Application options*) and the cursor is moved throughout a line of text with some brackets, it is now possible to identify which brackets have already been matched during the cursor move.

0.0.9.0 r1.10 (May/2004)

- Bug(s) fixed:
 - Bugs with Search/Search in files/[x] Search in directories were corrected.
- The Window of the results of the Search/Search in files can now be moved.
- Project interface was reformulated and pop-up menu is now available.

0.0.9.0 r1.09 (Mar/2004)

- Bug(s) fixed:
 - Minor bugs with ControlBar (on resize for a lesser size that the ControlBar, close application and start new) and View/Toolbar (checked/unchecked) menu was corrected.
 - A minor bug (Save/Save as) with associated extensions was corrected.
 - A minor bug with hotkeys was corrected.
 - A minor bugs with project was corrected.
 - A minor bug with tollbar of macro was corrected.
- Comment/Uncomment block was reformulated. This now work like Tinn-R 0.0.9.0.
- Two new hotkeys (not user configurable) were added:

```
<CTRL + ADD> key (numeric keypad)
will insert/replace text with the attribution symbol '>';
<CTRL + SUBTRACT> key (numeric keypad)
will insert/replace text with the attribution symbol '<-'. 
```

- Search/Replace was improved. It is possible now to decide if the occurrence will be, or not, replaced.

Temporarily, and until a better solution is found, to work properly, if the window is horizontally split the Search/Replace must be made at the bottom (main) window, because if it's at the top window (secondary) it doesn't question the confirmation to each substitution of the text occurrence. If the window is vertically split, the Search/Replace must be made at the left window (the right window does not question the confirmation to each substitution of the occurrence). For both, any key will synchronize the split windows.

- Now Tinn-R can initiate (Tolls/Initiate ...) and close (Tolls/Close ...) preferred RGui. For the first, the path of preferred RGui must be previously set (Options/Application options).
- A new Tinn to R option was implemented: *Help selected word*.
- Interface (appearance and icons) was reformulated.
- Tinn-R hotkeys' interface was reformulated. Now it is possible to choose the desired hotkeys to main actions.

- On the menu *Options/Application options* it is possible to choose the appearance to Tinn-R like WinXP.
- It is possible to get the positions of Tinn-R toolbars (preserved) when application exits, and restore the next time it is started.
- On the (Edit/Copy formatted) it is possible to copy the native format and export it in RTF, HTML and TeX format.
- Improvements were made to the highlighting.
- Now it is possible to open *Projects* files with one mouse click.

0.0.8.9 r1.08 (Mar/2004)

- Bug(s) fixed:
 - Small bug with Comment/Uncomment block was corrected.
- Now Tinn-R also runs with R, SciViews-R and S-Plus (the last still not finalized but it is running if the window Commands or S-PLUS - [Commands] have the focus).

0.0.8.9 r1.07 (Feb/2004)

- Specific HotKeys (F1..F11) were implemented for all Tinn to R functions. If and while activated, they overlap all the HotKeys of the Windows and any application running.
- Tinn-R now generates the two files (Tinn.ini and Shortcuts.ini) on C:\Documents and Settings\UserName\Application Data (or Dados de aplicativos)\Tinn\.
- Upgrade Tinn 0.0.8.8 to 0.0.8.9 as [changelog](#).
- A few tweaks were made in brackets highlighting, as per Marco's suggestions.
- jcfcaria suggested a little fix to the File Search window. Russ put it in.
- jcfcaria suggested a change to the RegEx filter interface to check, to make sure the filter won't be blank before running it. Russ put that in.
- Marco suggested a few changes to the open procedure. Now it checks if the file is changed and allows the user to open the file or not.
- fLineWdith has been commented out.

- Marco also suggested a few changes to the Search In Files code. Russ put that in, too.
- Russ pulled out the bitmaps on a few buttons in a few dialogs as per Marco's suggestion to keep things unified.

0.0.8.8 r1.06 (Feb/2004)

- Bug(s) fixed:

- Bug = Assertion failure

(c:\arquivos de programas\borland\componentes\synedit\source\SynEditWordWrap.pas,
line149)

Why and when?

if you create a new file and type:

<ENTER>
and press <UP> -> bug!

This was corrected: thank you Marco Groot!

0.0.8.8 r1.05 (Feb/2004)

- Comment/uncomment block was implemented: default symbol used for the comment/uncomment must be set in Options/Application options.

Comment block : Format/Comment block <ALT+C>.
Uncomment block : Format/Uncomment block <ALT+Z>.

- Option of Format/Column was eliminated: it is better to use this options (when necessary) keeping ALT pressed. This options are set in Options/Editor Options/(x)Alt sets column modes.
- ShortCuts to Print and Play macro were set the same = CTRL + P:

F7 : new shortcut to Record Macro
F8 : new shortcut to Play Macro.

APPENDIX B

R MANUALS ON CRAN

The R core team creates several **manuals** for working with R¹.

The platform dependent versions of these manuals are part of the respective R installations. They can be downloaded as PDF files from the URL given above, or directly browsed as HTML.

<http://cran.r-project.org/manuals.html>

The following manuals are available:

- An Introduction to R is based on the former “Notes on R”, gives an introduction to the language and how to use R for doing statistical analysis and graphics.
- A draft of The R language definition documents the language per se. That is, the objects that it works on, and the details of the expression evaluation process, which are useful to know when programming R functions.
- Writing R Extensions covers how to create your own packages, write R help files, and the foreign language (C, C++, Fortran, ...) interfaces.
- R Data Import/Export describes the import and export facilities available either in R itself or via packages which are available from CRAN.
- R Installation and Administration.
- R Internals: a guide to the internal structures of R and coding standards for the core team working on R itself.
- The R Reference Index: contains all help files of the R standard and recommended packages in printable form, (approx. 3000 pages).

¹The manuals are created on Debian Linux and may differ from the manuals for Mac or Windows on platform-specific pages, but most parts will be identical for all platforms.

The latex or texinfo sources of the latest version of these documents are contained in every R source distribution. Take a look at the subdirectory doc/manual of the extracted archive.

The HTML versions of the manuals are also part of most R installations. They are accessible by using the function `help.start()`.

APPENDIX C

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others. For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it. For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users. Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free. The precise terms and conditions for copying, distribution and modification follow.

Terms and Conditions

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a

computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding

Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such

circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered

work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code

saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have

received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you

grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY

YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

End of Terms and Conditions

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <textyear> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lGPL.html>.

INDEX

- .paths?, 37
- About, 59
 - Acknowledgments, 59
 - Project, 59
- about, 194
- App shortcuts, 23
 - alphabetical list, 32
 - ALT Keys, 32
 - call tip, 31
 - code completion, 31
 - compilation, 26
 - conversion, 26
 - CTRL keys, 33
 - DEL key, 34
 - edit, 28
 - edition, 25
 - END key, 34
 - file menu, 27
 - format, 28
 - function + 1:12, 34
 - function keys, 25
 - HOME key, 34
 - insert, 28
 - main menu, 27
 - marks, 25, 28
 - navigation, 24
 - not user configurable, 35
 - options, 29
 - project, 26, 27
 - R explorer, 31
 - R interface, 23
 - R menu, 30
- R script edition, 26
- reserved, 35
- search, 29
- search/replace and go, 24
- selection, 26
- SHIFT keys, 35
- tools, 29
- view, 31
- vizualization, 24
- application options, 52
 - appearance, 54
 - editor, 54
 - editor advanced, 55
 - editor display, 54
 - editor encoding, 56
 - editor EOL, 56
 - general, 53
 - main, 52
 - processing, 57
 - processing conversion, 57
 - processing LaTeX, 57
 - R, 57
- ASCII, 61
- ASCII chart, 61
- bookmarks, 44
- bug report, 36
- Card (R), 63
- Code completion, 64
- color preferences, 47
- column mode, 44
- Comments, 66
- communication, 49

configuration, 10
copy and paste highlighted R code,
 46
Ctrl+Shift+0, 49

database, 90
 card (R), 63, 98
 comments, 66, 97
 hotkeys, 71
 hotkeys control (R), 94
 hotkeys custom (R), 95
 hotkeys send (R), 93
 keystrokes, 92
 Keystrokes (Editor), 73
 mirrors (R), 75, 99
 shortcuts, 90
 Shortcuts (Application), 86
 snippets, 88, 96
dataset
 card (R), 63, 98
 comments, 66, 97
 Hotkeys (Control), 71
 Hotkeys (Custom), 71
 Hotkeys (Send), 71
 hotkeys control (R), 94
 hotkeys custom (R), 95
 hotkeys send (R), 93
 keystrokes, 92
 Keystrokes (Editor), 73
 mirrors (R), 75, 99
 shortcuts, 90
 Shortcuts (Application), 86
 snippets, 88, 96
debugging, 15
default editor, 40
defining hotkeys, 44
Deplate, 18
Differences, 67
discussion group, 194
DVI
 inverse search, 15

edit menu, 149
editor
 keystrokes, 73
 snippets, 88
Emacs, 41
encoding, 151
 ANSI, 151
 UTF-8, 151
 UTF16-BE, 151
 UTF16-LE, 151
EOL, 153
 MAC (CR), 153
 UNIX (LF), 153
 WIN (CR+LF), 153
Export
 RTF formatted, 149
export syntax highlighted R code,
 46

FAQ, 36
 .paths?, 37
 bookmarks, 44
 bug report, 36
 color preferences, 47
 column mode, 44
 comunication, 49
 copy and paste, 46
 copy and paste highlighted
 R code, 46
 Ctrl+Shift+0, 49
 default editor, 40
 defining hotkeys, 44
 Emacs, 41
 export syntax highlighted R
 code, 46
 feedback, 36
 find errors in my script, 48
 focus control, 46
 freeze, 49
 graphical debugger, 45
 gutter, 45
 hotkeys, 43
 icon on Windows, 46
 install, 36
 knit, 49
 knitr, 49

latest version, 36
marked blocks, 48
marks, 49
reuse my preferences, 38
save my preferences, 38
shortcut for cycling, 44
source code, 38
starting GUI, 39
step-by-step, 45
suggestions, 36
Tinn-R and Rgui, 49
Tinn-R and Rterm, 49
Tinn-R installation, 36
tools panel, 45
triggers, 41
What is Tinn-R?, 36
WinEdt, 41
feedback, 36
file menu, 144
 copy to clipboard, 146, 148
 template, 144
filetabs, 131
Find, 68
find, 79
find errors in my script, 48
focus control, 46
format menu, 150
 encoding, 151
 eol, 151
 selection, 150
 UTF-8, 153
 word, 150
freeze, 49

getwd(), 178
graphical debugger, 45
gutter, 45

help menu, 194
 main file conversion, 194
Highlighters, 69
 Editors colors, 69
 multi-highlighters, 69
Hotkeys
 Control, 71
 Custom, 71
 Send, 71
hotkeys, 43
Hotkeys control (R), 94
Hotkeys custom (R), 95
Hotkeys send (R), 93

icon on Windows, 46
ini file, 194
insert menu, 155
 LaTeX, 155
 LaTeX font, 158
 LaTeX format, 157
 LaTeX header, 157
 LaTeX math, 155
 R, 155
install, 36
installation, 10
 sourceforge, 43
inverse DVI search, 15
inverse PDF search, 17
IO Interface, 103

Keystrokes (Editor), 73
Keystrokes (editor), 92
keywords
 all, 24–28, 30–35
 as, 27, 35
 browser, 26, 29, 35
 by, 26
 C, 25, 26, 28, 29, 32, 33, 35
 choose, 30
 comment, 25, 28, 32
 D, 27, 30, 32, 35
 default, 26, 29, 35
 environment, 31
 example, 23, 30
 file, 23, 26–30, 34, 35
 help, 23, 35
 I, 25–29, 32, 35
 index, 26, 29
 mode, 26, 35
 next, 30

numeric, 25, 26, 28, 33, 35
objects, 25, 30, 34
off, 23, 29, 32
on, 23, 25, 29, 32, 34
optimize, 24, 30, 31
or, 23, 25, 26, 28, 31, 33–35
page, 24, 35
path, 30, 35
R, 23–25, 27–32, 34, 35
replace, 25, 33
seq, 23, 24, 35
set, 23, 26, 30–32, 35
show, 24, 30–32
single, 26, 29
start, 30
structure, 25, 30, 34
system, 26, 29, 35
text, 24, 29, 35
window, 23, 32

Knitr, 172

latest version, 36

LaTeX, 127

- main file, 164
- main LaTeX project file, 127
- set the main file, 164
- sets the main LaTeX project file, 127

License manager, 74

LOG Interface, 105

Main dialogs, 58

- About, 59
- ASCII chart, 61
- Card (R), 63
- Check for update, 89
- Code completion, 64
- Comments, 66
- Differences, 67
- Find, 68
- Highlighters, 69
- Keystrokes (Editor), 73
- License manager, 74
- Mirrors, 75

Open URL, 76

Pandoc, 77

Print, 78

Print preview, 79

R code completion, 64

S/K/H manager, 87

Search in files, 82

Search snippet, 84

Shortcuts (Application), 86

Tinn-R updater, 89

marked blocks, 48

marks, 49, 134

marks menu, 154

- block, 154

markup, 108

metacharacters, 138

- alternatives, 140
- backreferences, 141, 142
- find examples, 142
- iterators, 139
- line separators, 138
- predefined classes, 138
- subexpressions, 141
- word boundaries, 139

Miktex, 15

Mirrors, 75

Open URL, 76

options menu, 160

- default, 162
- syntax, 160
- syntax set, 161

overview, 1

- bug reports, 7
- quick start, 2
- what do you get?, 4
- what is Tinn-R?, 3
- why Tinn-R?, 4

Pandoc, 20, 77

PDF

- inverse search, 17

Print, 78

Print preview, 79

- processing, 207
 - DVI, 207
 - PDF, 207
- project, 107
 - LaTeX, 127
 - main LaTeX project file, 127
 - sets the main LaTeX project file, 127
- project menu, 147
 - file, 148
 - group, 148
 - project, 147
- projects, 124
- Python, 21
- R
 - Code completion, 64
 - card, 98, 121
 - explorer, 120
 - mirror, 99, 215
 - mirrors, 121, 215
 - set mirror, 215
 - Working dir, 120
- R explorer, 120
- R menu, 171
 - control, 173
 - control packages, 173
 - send, 172
 - Term, 174
 - Term clear, 176
 - Term file, 175
 - Term file IO, 175
 - Term file LOG, 175
 - Term focus, 176
 - Term fontsize, 178
 - Term highlighter, 177
 - Term highlighter IO, 177
 - Term highlighter LOG, 177
 - Term history, 178
 - Term line wrap, 177
 - Term size, 176
 - Term split, 176
 - Term workspace, 178
 - Toogle (active R)), 171
- Toogle (start/close)), 171
- work directory, 178
- recognized words, 194
- regular expressions, 136
 - character classes, 137
 - escape sequences, 136
 - simple matches, 136
- replace, 79, 80
- reuse my preferences, 38
- Rgui, 14
- Rterm, 14
 - interface, 101
- RTF formatted, 149
- Ruby, 18
- R functions
 - help.start, 336
- R packages
 - Hmisc, 267
- S/K/H manager, 87
- save my preferences, 38
- search
 - case sensitivity, 82
 - inverse DVI , 15
 - inverse PDF, 17
 - regular expressions, 82
 - results, 83
- Search in files, 82
- search menu, 159
- secrets, 195
 - active page, 221
 - after installation, 197
 - appearance, 202
 - arrange, 221
 - auto completion, 221
 - bars, 217
 - card (R), 214
 - colors preference, 208
 - database, 210
 - shortcuts, 210
 - snippets, 210
 - dock, 203
 - editor options, 204
 - Editor: how to?, 220

general, 202
highlighters preference, 208
I'm panicking, 201
introduction, 196
macros, 220
marks, 220
project, 209
R, 206, 212
 explorer, 212
spell checker, 218
split, 219
tab order, 221
tools, 209
views, 203
visibility, 200
selection mode, 132
 column, 133
 line, 132
 normal, 132
selection mode menu, 162
setwd(), 178
shortcut for cycling, 44
Shortcuts (Application), 86
Shortcuts (application), 90
Snippet, 84
Snippets, 117
snippets, 88
source code, 38
speller, 15
starting GUI, 39
step-by-step, 45
suggestions, 36
Sumatra, 17
SynEdit, 102

template
 LaTeX, 144
 R doc, 146
 R html, 144
 R markdown, 144
 R noweb, 144
 R script, 144

Term
 interface, 101

IO, 103
LOG, 105
Tinn-R and Rgui, 49
Tinn-R and Rterm, 49
Tinn-R installation, 36
Tinn-R updater, 89
tools, 106
 backup, 169
 database, 169
 interface, 106
 macro, 170
 markup, 108
 markup L^AT_EX, 108
 markup txt2tags, 108
 misc, 106
 project, 107
 restore, 170
 results, 109
 results inilog, 109
 sort, 170
 Windows expl., 106
 Work expl., 107

tools interface
 card (R), 121
 Comments, 117
 database, 114
 hex, 111
 Hotkeys, 116
 Keystrokes (editor), 115
 mirrors (R), 121
 R, 119
 R explorer, 120
 search, 110
 shortcuts, 115
 snippets, 116
 spell, 113
 Working dir, 120

tools menu, 164
 compilation, 167
 conversion deplate, 165
 processing, 164, 165
 processing Txt2tags, 167
 processing viewer, 168
 processing viewer dvi, 168

- processing viewer html, 169
- processing viewer pdf, 168
- tools panel, 45
- toolbar, 129
 - disposition, 130
 - showhide, 130
- triggers, 41
- Txt2tags, 21
- Updater, 89
- user guide, 194
- user list, 194
- view menu, 180
 - Line wrap, 188
 - fontsize, 188
 - R, 180
 - split, 188
 - Term, 182
 - Term highlighter, 183
 - Term highlighter IO, 183
 - Term highlighter LOG, 183
 - Term line wrap, 183
 - Term size, 182
 - Term split, 182
 - toolbars, 187
 - tools, 184
 - tools resources, 184
 - tools resources database, 187
 - tools resources markup, 186
 - tools resources misc, 186
 - tools resources R, 187
 - tools resources results, 186
 - tools size, 184
- web menu, 190
 - processing, 193
 - R Editors, 191
 - R GUIs, 191
 - R IDEs, 191
 - R information, 190
 - R search, 190
 - statistics, 191
 - statistics virtual labs, 191
- Tinn-R, 191
- What is New?, 225
- 2004, 328
- 2005, 312
- 2006, 290
- 2007, 282
- 2008, 273
- 2009, 266
- 2010, 263
- 2012, 260
- 2013, 255
- 2014, 252
- 2015, 249
- 2016, 245
- 2018, 243
- 2019, 241
- 2020, 239
- 2021, 232
- 2022, 231
- 2023, 230
- 2024, 226
- What is Tinn-R?, 36
- windows menu, 189
- WinEdt, 41
- Work directory, 178

ABOUT THE AUTHOR AND COLLABORATORS

AUTHOR:

Faria, José Cláudio teaches statistics at Universidade Estadual de Santa Cruz - UESC
(Santa Cruz State University), Bahia/Brazil.
He is head of the Laboratory of Computational Statistics - [LEC](#).

COLLABORATORS:

Grosjean, Philippe teaches biostatistics and aquatic ecology at Mons University, UMONS, Belgium.
He is head of the Laboratory of Numerical Ecology of Aquatic Systems at UMONS.

Jelihovschi, Enio Galinkin teaches statistics at Universidade Estadual de Santa Cruz - UESC
(Santa Cruz State University), Bahia/Brazil.

Pietrobon, Ricardo is Associate Vice Chair and Associate Professor, Department of Surgery, Research on Research Group, Duke University Medical Center.

Farias, Philipe Silva is a Computer Science student at Universidade Estadual de Santa Cruz - UESC
(Santa Cruz State University), Bahia/Brazil.

Kramer, Philiphe Alexandre Ribeiro is graduated in Computer Science at Universidade Estadual de Santa Cruz - UESC
(Santa Cruz State University), Bahia/Brazil.

Farnese, Balthazar Mattos is an independent WEB developer hired to maintain the [Tinn-R Website](#).
Ilhéus, Bahia/Brazil.

