



*DSC 2003 Working Papers
(Draft Versions)*

<http://www.ci.tuwien.ac.at/Conferences/DSC-2003/>

R–WinEdt

Uwe Ligges*

Abstract

In the first part of this paper the status quo of R-WinEdt, a plug-in for the shareware editor WinEdt for Windows, is described. This description includes the interface between R and WinEdt, convenient editing features like syntax highlighting, and the installation procedure. The second part is aimed at the future of R-WinEdt. Is it desirable to improve the plug-in? Possible improvements related to the interface, installation procedure, and integration of Sweave functionality are discussed.

1 Introduction

R-WinEdt¹ is a plug-in for WinEdt², a shareware editor for the operating system Windows³. It provides an “interface” to R (Ihaka and Gentleman, 1996), a statistical programming language and environment. Additionally, a highlighting scheme, which illuminates the syntactical structure of the R code, replaces WinEdt’s defaults. Shortcuts, menus, and toolbar buttons are included to provide easy access to common operations for the user’s convenience. The existing functionality will be presented in Section 3, just after some comments why WinEdt is the editor of my choice in Section 2.

Section 4 is aimed at the future of R-WinEdt. Its “interface” to R will be discussed, as well as possible changes and improvements. I received quite a lot questions related to the installation procedure of R-WinEdt, therefore I would like to improve it and discuss some possible solutions. Further on, it will be discussed

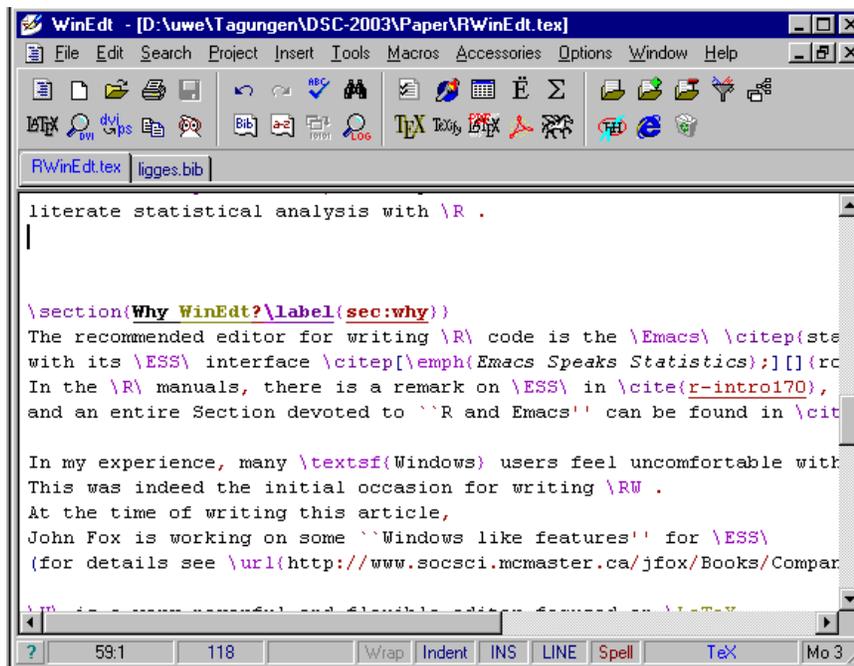
*SFB475, Fachbereich Statistik, Universität Dortmund, Germany,

ligges@statistik.uni-dortmund.de

¹<http://CRAN.R-Project.org/contrib/extra/winedt/>

²<http://www.WinEdt.com/>

³Windows is a trademark of the Microsoft Corporation.

Figure 1: WinEdt in its “regular” \LaTeX mode.

whether it is desirable to have full support of Sweave (Leisch, 2002), which provides convenient state of the art literate statistical analysis with R.

2 Why WinEdt?

The recommended editor for writing R code is the Emacs (Stallmann, 1999) with its ESS interface (*Emacs Speaks Statistics*; Rossini et al., 2001). In the R manuals, there is a remark on ESS in Venables et al. (2003), and an entire Section devoted to “R and Emacs” can be found in Hornik (2003). In my experience, many Windows users feel uncomfortable⁴ with the Emacs. This was indeed the initial occasion for writing R-WinEdt.

WinEdt is a very powerful and flexible editor focused on \LaTeX , and it is widely known among \LaTeX users under Windows (see Figure 1 for a screenshot). It is easy to define Task Bars, Menus, and quite sophisticated highlighting schemes in WinEdt. The editor also provides an extended Macro language that includes, e.g., capabilities for interaction with the operating system. There are convenient ways to automate tasks in WinEdt using this macro language (the Linux user typically

⁴ At the time of writing this article, John Fox is working on some “Windows like features” for ESS. For details see <http://www.socsci.mcmaster.ca/jfox/Books/Companion/ESS/>.

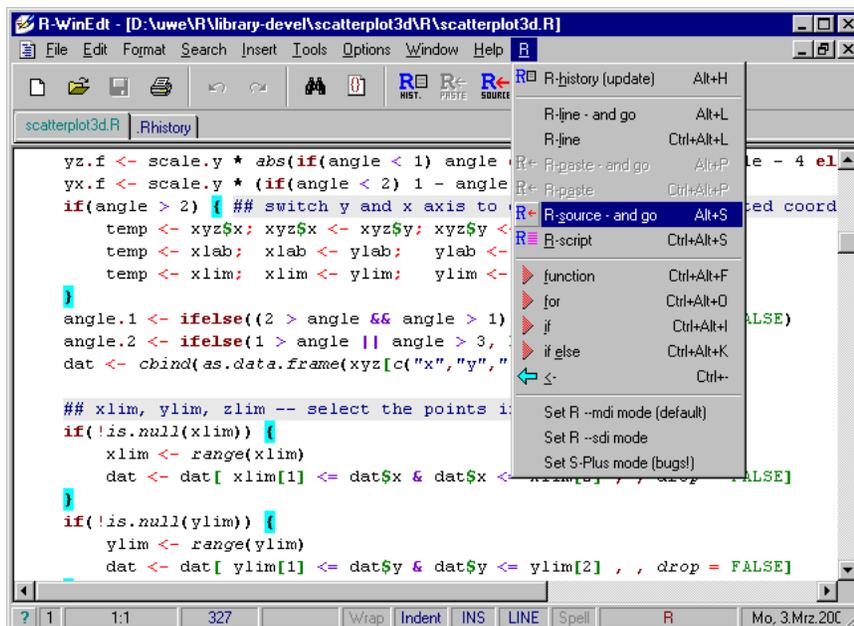


Figure 2: WinEdt running with the R-WinEdt plug-in

uses a Makefile, instead). For details I refer to the WinEdt manuals.

Because of all these features, it was easy to implement the simple plug-in in its current version. Unfortunately, WinEdt is a commercial (affordable) shareware editor, hence I have no access to its sources. So it is necessary to convince WinEdt's author, Aleksander Simonic⁵, in order to get desirable new features into the editor.

3 Status Quo

The most easiest way to get to know the current state of R-WinEdt is to try it out and play around. Nevertheless, in this Section R-WinEdt's status quo and some corresponding technicalities will be described. See Figure 2 for a screenshot of R-WinEdt in action.

The recent version (1.4-1, at the time of writing) of R-WinEdt works with WinEdt 5.2 and later on any 32-bit version of Windows and requires R-1.2.0 or later. Of course, any version numbers are subject to change.

For installation instructions, examples, and known bugs see R-WinEdt's (edited) ReadMe.txt file in the Appendix.

⁵alex@winedt.com

3.1 Interface

“Interface” is a Section title which implies more than what has been implemented. On the one hand, unfortunately, WinEdt does not support DCOM⁶, but uses DDE⁷ instead. I was not able to convince WinEdt’s author to implement DCOM support in WinEdt. On the other hand, there is a DCOM server (Neuwirth and Baier, 2001) for R available, but no DDE interface.

What I did instead is making use of some features of WinEdt’s macro language in the following manner. When any information has to be send to R, it is constructed by the macro language. Then it is copied to the Windows clipboard and pasted into the R Console window of RGui using the following macro, which implements the awkward main part of the interface:

```
SetFocus("RGui"); // set focus to R
PostMessage("RGui", $0104, $12, $20000001); // press ALT
Wait(150); // RGui needs some time
PostMessage("RGui", $0105, $12, $20000001); // release ALT
PostMessage("RGui", $0102, 87, 1); // Char: W -> Win. Menu
PostMessage("RGui", $0102, 49, 1); // Char: 1 -> Window1
// = R Console
Wait(150); // RGui needs some time
PostMessage("RGui", $0104, $12, $20000001); // press ALT
PostMessage("RGui", $0105, $12, $20000001); // release ALT
PostMessage("RGui", $0102, 69, 1); // Char: E -> Edit Menu
PostMessage("RGui", $0102, 80, 1); // Char: P -> Paste
```

So this main part can be described as follows. After the focus of programs is switched from WinEdt to R (and both applications have been started, of course), key presses are simulated in a manner like one would use the menu without any mouse:

1. Press **Alt**.
2. RGui (and Windows) needs some time to open the menu, (time depends of the operating system, it takes most time on Windows XP).
3. Release **Alt**.
4. Press **W** – opens the menu called **Windows**.
5. Press **1** – selects “window 1” which is identical to the R Console; now we are sure to have focussed the R Console (instead of, e.g., an R Graphics window).
6. Again, some time is needed to switch to the R Console window.
7. Press and release **Alt** to open the menu again.

⁶DCOM: *Distributed Component Object Model*

⁷DDE: *Dynamic Data Exchange*, basic technology of Microsoft’s OLE (*Object Linking and Embedding*).

8. Press **E** to open the **E**dit menu, and
9. press **P** to **P**aste the contents of the clipboard into the R Console window.

This macro code is written for working with RGui running in `--mdi` mode, a version working with RGui's `--sdi` mode is somewhat simpler:

```
SetFocus("R Console");
Wait(150);
PostMessage("R Console", $0104, $12, $20000001);
Wait(150);
PostMessage("R Console", $0105, $12, $20000001);
PostMessage("R Console", $0102, 69, 1);
PostMessage("R Console", $0102, 80, 1);
```

Obviously, the “interface” is an awkward one, but it is extremely simple with just 11 (7) lines of macro code. There is one “bug” in this macro: If RGui is minimized, it is not possible to interact with it. For a discussion whether it is desirable to improve the interface see Section 4.

The question arises, what kind of information can be pasted into the R session. Two examples are given, the first one shows a macro that pastes a selected region of R code from R-WinEdt (both shortcuts and icons can be set up to invoke macros):

```
CMD("Copy"); // copy selected text into a buffer
ReplaceInString("1", ">", 0, 1, 1, 9); // add a newline at the end
CopyToClipboard('%!9', 1); // copy buffer into the clipboard
LetReg(9, ""); // clear the buffer
Exe("%b\send2R.edt") // execute the main interface
End;
```

The second example shows a macro that saves the current file and `source()`s it into R:

```
CMD("Save"); // save the current file
GetChildName(9); // save the ...\path\filename into buffer1
DosToUnix("%!9", 8); // translate "\" to "/" in buffer1
// copy a "newline" in buffer2:
ReplaceInString("1", ">", 0, 1, 1, 9);
// copy 'source("buffer1")buffer2' into the clipboard:
CopyToClipboard('source("%!8")%!9', 0);
LetReg(9, ""); // clean up buffers 1 and 2
LetReg(8, "");
Exe("%b\send2R.edt") // execute the main interface
End;
```

For details on WinEdt's macro language I refer to its manual and the two versions of “The WinEdt Hacker's Guide” by [Alexander and Marquardt \(1998\)](#), and [Alexander \(1999\)](#).

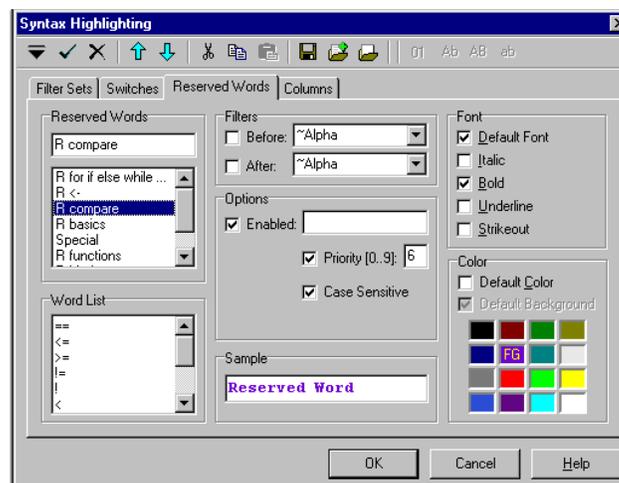


Figure 3: R-WinEdt – configuring the highlighting scheme

A complete list of all available macros is given below. The “template” macros insert templates into the current document of WinEdt. One can jump from one missing entry (marked with “.” in the examples below) to the other in order to fill in by pressing **Ctrl+Space**. For each of the marked (+) items, there are two macros. One macro submits information to R and goes back to WinEdt at once (e.g. in order to submit several discontinued regions of code), whereas the other one is intended to change the focus permanently to R after submission:

- + Copy and paste a selected region of code.
- + Copy and paste the current line.
- + Save and `source()` the whole file.
 - Save the R history in file `.Rhistory` and open it in R-WinEdt.
 - Insert function template: `function(..){ .. }`.
 - Insert loop template: `for(..){ .. }`.
 - Insert if template: `if(..){ .. }`.
 - Insert ifelse template: `if(..){ .. } else{ .. }`.

3.2 Highlighting

Setting up highlighting schemes is easy in WinEdt, and corresponding rules are highly configurable (cf. Figure 3). The very complete R-WinEdt highlighting scheme distinguishes between the following main groups of “highlighting-classes” (given in descending order):

- Comments: # and all stuff thereafter.
- Assignment operators: <-, <<-, =, ->, and “;”.
- Control structure keywords: if, else, ifelse, while, for, break, repeat, next.
- Logical operators: ==, <=, >=, !=, !, <, >, &, &&, |, ||.
- function, library, require
- Special values: NULL, NA, NaN, Inf, F, T, FALSE, TRUE, and the functions .Internal and .C.
- Matrix operators: %%, %*%, %/%, %in%, %O%, %x%.
- Contents within quotes and double quotes.
- Extraction of list and slot elements: \$, @.
- Parenthesis, brackets and braces: (), [], {}.
- A list of known R functions, currently those implemented in all shipped and recommended packages.
- Content within brackets in library(content), require(content).

3.3 Auto-Completion

A “Command Completion Wizard” by Holger Danielsson⁸ for WinEdt is available at <http://www.WinEdt.org/Plugins/complete.php>. The wizard is quite easy to use, but not yet integrated into R-WinEdt, mainly because I do not know anything about its licensing.

4 The Future

Aiming at the future of R-WinEdt: Is it desirable to improve the plug-in? From my point of view it should be redesigned in order to get Sweave support (Section 4.2), and to allow a convenient installation procedure (Section 4.3). For the computer scientist’s point of view the interface needs to be redesigned as well, but not for me being mainly an “user”.

⁸ dani@fbg.schwerte.de, he seems to be disappeared

4.1 Interface

In Section 3.1 the awkward recent “interface”, which works surprisingly stable except for the “RGui is minimized” bug, has been described. Three ways of improved interfaces can be considered:

- Convincing WinEdt’s author to integrate DCOM support.
- Writing a DDE interface for R.
- Using socket connections.

For me, the current interface works stable enough, so I will not make much effort. Benefits of possible improvements have to be compared to all the work that is required to implement these improvements. Nevertheless, contributions will be highly appreciated!

4.2 Sweave

Since WinEdt is focused on L^AT_EX, it seems to be natural to combine this L^AT_EX functionality with the capabilities of the R-WinEdt plug-in. Literate statistical analysis (for a discussion related to ESS see Rossini, 2001) with R would benefit from combining those capabilities. More specifically, full support of Sweave⁹ by Leisch (2002) is desirable for convenient state of the art literate statistical analysis with R:

R-WinEdt was initially written on a Pentium I machine (133MHz, 64MB RAM, Windows NT 4.0), where speed and memory efficiency was still an issue. Unfortunately, it was not possible to combine both R and L^AT_EX functionality without getting a big penalty in WinEdt’s starting times and memory consumption. Today, that is no longer a problem to worry about, so a redesigned plug-in could easily provide a combined interface.

The only issue to think about is the configuration of an adequate highlighting scheme for combined documents such as Sweave files. Thus highlighting must be sensitive to code chunks in a way. And, of course, an installation should not corrupt users’ adaptations.

4.3 Installation

I received quite a lot questions related to the installation procedure of R-WinEdt, which indeed is uncommon in a way, because there is just a simple file `install.bat` that has to be invoked in the exactly correct directory. Handling this installation procedure seems to be too complicated for some users. Therefore I would like to improve it and discuss possible solutions; obvious ways are:

- The “R way”: Installation as / by an R package.
- The “WinEdt way”: A redesigned plug-in might be installed by WinEdt-macros.

⁹Sweave combines typesetting with L^AT_EX and statistical data analysis with R into integrated statistical documents.

- The “Windows way”: A Setup Wizard like Inno Setup¹⁰ which is also used for the binary version of R.
- Any combination of the above mentioned strategies.

My current preference is a combination of the first two ways. The plug-in would be shipped in form of an R package. A customizable installation procedure – maybe using some simple R integrated GUI elements like a setup wizard – would have the power to

- locate the WinEdt installation
- adjust the R environment (e.g. `.Rprofile`),
- create shortcuts on the desktop and in the start menu,
- add or modify some settings of the Windows registry, and
- start WinEdt to execute installation macros (probably modified in the first step by R) for the plug-in’s integration into WinEdt.

Duncan Temple Lang’s SWinRegistry package of “The Omega Project for Statistical Computing”¹¹ will be helpful to perform required tasks related to the Windows registry.

5 Conclusion

The recent version of R-WinEdt provides the “Windows way” of editing R code – convenient, flexible, and powerful – but it is not quite as powerful as ESS for the Emacs. I think it desirable to improve the plug-in in two main topics. It should be redesigned in order to implement Sweave support, and to allow a convenient installation procedure, whereas the implementation of a better interface is of less priority.

I would like to invite the reader to submit contributions and other suggestions to any of these topics.

Appendix

Readme.txt

Using WinEdt as an editor for R *** R-WinEdt 1.4-1 *** 12.01.2003

Overview:

=====

Using WinEdt as an editor for R (only Rgui under Windows).

¹⁰<http://www.jrsoftware.org/>

¹¹<http://www.Omegahat.org/>

Features:

=====

- Syntax-Highlighting (and highlighted printing)
(function index of all "recommended" packages, including:
base boot class cluster ctest eda foreign grid KernSmooth
lattice lqs MASS methods mgcv modreg mva nlme nls nnet rpart
spatial splines stepfun survival tcltk tools ts
last update: 13.09.2002, R-1.6.0 (beta)).
- Many R functions at the same time in the same editor.
- Buttons / Shortcuts for simple access (one click) to:
 - Save document and source(.) into R.
 - Submit script to R (WinEdt as scripting window)
 - Paste selected code from WinEdt directly to R (e.g. from history)
or just evaluate the current line
 - Open (and update) .Rhistory (so you can reuse your last commands).
- Templates, e.g.: for(_ in _){_}
- Faster starting than with standard configuration (of WinEdt).
- Features of WinEdt (e.g. Delimiter Check, advanced searching,
bookmarks, macros, ...).
- Independent of another WinEdt running with "LaTeX-configuration".
- Works now with RGui running in MDI or SDI style and also with
S-PLUS (not supported!).

Archive:

=====

R-WinEdt.zip, contains:

- install.bat *simple* installation script.
- R.ini modified WinEdt initialization-file.
- R.img, R.btn modified WinEdt GUI files.
- R*.bmp Buttons for R-WinEdt's GUI.
- R*.edt WinEdt macros for R interactions.

Installation:

=====

- Install R (rw1020 or later)
- Install WinEdt 5 (V. 5.2, Build: 20001213 or later)
Note: WinEdt is Shareware (<http://www.winedt.com>).
- Unzip the archive into directory\winedt\plugins\r-winedt
(e.g. "c:\program files\winedt team\winedt\plugins\r-winedt")
The directory does not exist, probably you have to create it manually.

- In your Windows-Explorer double-click on "install.bat".
- If you are running RGui in multiple windows style (MDI), the default mode should be unchanged. For RGui in single window style (SDI) or if you want to use R-WinEdt with S-PLUS, you can change the R-WinEdt mode permanently(!) using the Menu, e.g.:
R -> Set R --sdi mode
- To invoke R-WinEdt from startmenu or desktop create a shortcut to WinEdt as follows:
"c:\program files\winedt team\winedt\winedt" -C="R-WinEdt" -e=r.ini
- In R use something like (for example in your .Rprofile):
options(editor="\c:/program files/winedt team/winedt/winedt\
+ -c=\"R-WinEdt-edit\" -e=r.ini -V")
options(pager="\c:/program files/winedt team/winedt/winedt\
+ -C=\"R-WinEdt\" -e=r.ini -V")

Used WinEdt parameters (Remark: There is a difference between -c and -C !):

- c="name": New instance of WinEdt called "name" will be started.
- C="name": If an instance "name" of WinEdt is already running, it will be used. So you can run WinEdt as LaTeX and R editor at the same time.
- e="name": Using "name" as initialization-file.
- V : Running in "virgin"-mode

Examples:

=====

```
##### Recommended procedure:
## Open WinEdt with something like:
" c:\program files\winedt team\winedt\winedt" -C="R-WinEdt" -e=r.ini
## i.e. ideally the shortcut you have already created.
##
## Then create a new document and write an R function.
## Click on the symbol "R source" or press ALT+S.
## You will be asked to specify a filename (e.g. "R-prog1.R").
## If RGui is running, it will be focussed and source(.) will be called.

#####
options(pager="\c:/program files/winedt team/winedt/winedt\  
+ -C=\"R-WinEdt\" -e=r.ini -V")
file.show(".Rhistory")
## Mark some lines, click the "R paste" button or use Alt+P as shortcut:
## Marked lines will be executed in RGui.

#####
## The following way to edit function is possible, but *not* recommended:
options(editor="\c:/program files/winedt team/winedt/winedt\  
+ -c=\"R-WinEdt-edit\" -e=r.ini -V")
my.legend <- legend
fix(my.legend)
```

Known bugs:

=====

- If RGui is minimized, it is impossible to send commands from WinEdt to RGui. Workaround: Don't minimize RGui! ;-)
- Works in S-PLUS mode only, if the command window is opened as "window1".
- Doesn't work with all versions of S-PLUS (tested with version 4.5).

References

- Alexander, J. (1999): *The WinEdt Hacker's Guide*. URL <http://www.WinEdt.org/>.
- Alexander, J. and Marquardt, C. (1998): *The WinEdt Hacker's Guide*. URL <http://www.WinEdt.com/>.
- Hornik, K. (2003): *The R FAQ*, Version 1.7-0. R-Project. URL <http://CRAN.R-project.org/manuals.html>. ISBN 3-901167-51-X.
- Hornik, K. and Leisch, F. (eds.) (2001): *Proceedings of the 2nd International Workshop on Distributed Statistical Computing, March 15-17*. Vienna: Technische Universität Wien. ISSN 1609-395X. URL <http://www.ci.tuwien.ac.at/Conferences/DSC-2001/Proceedings/>.
- Ihaka, R. and Gentleman, R. (1996): R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, 5 (3), 299-314.
- Leisch, F. (2002): *Sweave User Manual*. Institut für Statistik und Wahrscheinlichkeitstheorie, Technische Universität Wien, Vienna, Austria. URL <http://www.ci.tuwien.ac.at/~leisch/Sweave>. R Version 1.6.0.
- Neuwirth, E. and Baier, T. (2001): Embedding R in Standard Software, and the other way round. In: *Hornik and Leisch (2001)*.
- Rossini, A. (2001): Literate Statistical Analysis. In: *Hornik and Leisch (2001)*.
- Rossini, A., Mächler, M., Hornik, K., Heiberger, R. M., and Sparapani, R. (2001): Emacs Speaks Statistics: A universal Interface for Statistical Analysis. *Technical Report 164*, Department of Biostatistics, University of Washington. URL <http://software.biostat.washington.edu/statsoft/ess/ess-techrep.pdf>.
- Stallmann, R. M. (1999): *The Emacs Editor*. Boston. URL <http://www.gnu.org>. Version 20.7.
- Venables, W. N., Smith, D. M., and the R Development Core Team (2003): *An Introduction to R*, Version 1.7.0. R-Project. URL <http://CRAN.R-project.org/manuals.html>. ISBN 3-901167-55-2.