

Using Geany With R on GNU/Linux

François Tonneau
Universidade Federal do Pará

December 1, 2014

These notes will help you to customize Geany for R programming on a system with GNU/Linux.

1 Rationale

Geany's system-wide configuration file for the R language, `filetypes.r`, provides support for syntax highlighting and code navigation. It is also possible to send code to R running in Geany's terminal widget. However, this leads easily to mistakes if the virtual terminal runs the system shell instead of R (see http://wiki.geany.org/howtos/using_geany_with_r), and the limited size of the terminal widget can be problematic on computers with a small monitor.

Accordingly, we will use a different approach: we will run R in a terminal window independent from Geany, and we will send code to this window only if it actually runs R. We will also use a tag file generated from R to add function-name autocompletion and calltips to Geany. The result will be a minimal IDE for the R language with components that are as lightweight as possible.

2 Prerequisites

- I assume that you have Geany and R installed on a GNU/Linux system.
- Your terminal emulator should accept command-line arguments so as to run R in a new window and set this window title. The terminal emulator should also interact well with the clipboard, accepting `ctrl+shift+v` as a control sequence for pasting. Most modern terminal emulators (except those of the drop-down type) will do. Examples of terminal emulators that should work are *evilvte*, *gnome-terminal*, *lxterminal*, *roxterm*, *sakura*, *terminator*, and *xfce4-terminal*.
- We will also need two lightweight utilities: *xsel*, to write R commands into the clipboard, and *xdotool*, to find the window in which R is running and paste commands into it. If not already on your system, install both utilities from your repositories.

3 Provided files

Three text files accompany these notes:

- `filetypes.r`, a filetype definition file for the R language that defines R-specific commands for Geany's Build Menu and a context-action command to call R help from Geany
- `geany_tags.R`, an R script that will generate a tag file for Geany from the function names in your R packages
- `base.r.tags`, an example of tag file generated by `geany_tags.R`

4 Sending code from Geany to R

To be able to send code from Geany to R, you will need to run the latter in a window titled **Running R**. You could do this manually, but it is better to let a shell script do the work for you. Thus, with `gnome-terminal` as a terminal emulator, you could write the following script:

```
#!/bin/sh
gnome-terminal --title="Running R" --execute R
```

Other terminal emulators accept a similar syntax; see their man pages for details. If you use `lxterminal`, for example, your script could read as:

```
#!/bin/sh
lxterminal --title="Running R" --command R
```

And with terminator:

```
#!/bin/sh
terminator --title "Running R" --command=R
```

Regardless of which terminal you use, save your shell script in a directory of easy access (such as `/home/username/bin`), make it executable, and associate a desktop icon or keyboard shortcut to it to be able to open R quickly whenever you need so.

Now copy the accompanying `filetypes.r` to Geany's user directory for filetype definition files, normally located at `/home/username/.config/geany/filedefs` (create this directory if needed). This `filetypes.r` file will add to the system-wide definition file for R a context action command to call R help from Geany and build-menu commands to source, paste, and script code into R. The file manages to do this by calling `xsel` and `xdotool` as copy/paste utilities from Geany to the window titled **Running R**. After restarting Geany, you will be able to:

- source any R file into R by pressing **F8** (Build > Source)
- select any block of lines and send it to R by pressing **F9** (Build > Paste)

- run your file with RScript by pressing F5 (Build > Script)
- obtain R help on any term by right-clicking on this term and pressing **x** (context action command).

Note 1: Because the F5/Build Script command works by calling the RScript program, it will run independently of R being open. The other commands, however, do require R to be running in a terminal window with the right title. (If this is not the case, Geany will just tell you that the command failed.)

Note 2: The script you use to run R may also specify the window geometry of your choice. With terminator, for example, preceding `--command=R` by the `-m` flag will run R in a maximized window. Again, read the man page of your terminal emulator to find out which options it accepts.

Note 3: Some terminal emulators like *xterm* or *stterm* accept `shift+Insert` instead of `ctrl+shift+v` for pasting. If you want to use these terminals with their default keybindings, open the `filetypes.r` configuration file with a text editor and simply replace `ctrl+shift+v` by `shift+Insert`.

5 Autocompletion and calltips

To add function-name autocompletion and calltips to Geany, you will need a tag file specific to the R language. You can use the provided `base.r.tags` file, or alternatively, generate it for yourself by opening R and sourcing the accompanying `geany.tags.R` script. This R script gathers the function names present in your R packages, generates the tag file, `base.r.tags`, and saves it in your working directory. Either way, copy `base.r.tags` to Geany's user directory for tag files, normally located at `/home/username/.config/geany/tags` (create this directory if needed). After restarting Geany, autocompletion for function names and calltips will be available whenever you open an R file.

Note 1: For autocompletion to function properly, make sure that Geany's Autocomplete Symbols option is checked in Edit > Preferences > Editor > Completions. Recommended parameters are 4 characters to type for autocompletion, and 100 as a maximum number of name suggestions per tag.

Note 2: Geany does not allow autocompletion on names that include periods (R has plenty of them!) As a workaround, in the tag file all periods have been replaced by the string, `_dot_`. The function name, `read.table`, for example, has been replaced by `read_dot_table`. Autocompletion and calltips will work as usual, but keep in mind that after writing your function call you should replace any `_dot_` in it by an actual period if you want your R code to run correctly. This can be done manually or, better, through a custom command. Go to Edit > Format > Send Selection to > Set Custom Commands and add a new command with

```
sed s/_dot_/./g
```

as action. By going to Edit > Preferences > Keybindings, you can then assign a keyboard shortcut to the action you just defined (say, Custom Command 1). The upshot: you will be able to substitute a period for any `_dot_` string in a function name by keeping your cursor on the line in which the name appears and pressing your custom action shortcut.

Note 3: The provided R script for generating a Geany tag file is an adaptation of a script Yihui Xie wrote for Notepad++:

<https://gist.github.com/yihui/2143971>

Acknowledgments

I thank Jordan Sissel (<http://www.semicomplete.com>) for his help with *xdotool* and the users/developers on the Geany mailing list for their feedback.