24, 25

Oliver Sailer
Fachbereich Statistik, Universität Dortmund, Germany
sailer@statistik.uni-dortmund.de

R Help Desk

Make 'R CMD' Work under Windows – an Example

Uwe Ligges and Duncan Murdoch

Introduction

During the last couple of months there have been a number of questions on both mailing lists 'R-help' and 'R-devel' on how to install R packages from source under Windows. Even though the procedure is documented in detail in the "R Installation and Administration" (R Development Core Team, 2005a) manual, people still seem to be uncertain about some details. This is perhaps partly caused by some confusing online material published by others who tried to help, but there is also the fact that requirements change over time, and the fact that the procedure is quite different from what most Windows users are used to doing.

To install packages from source, you will need to use the 'R CMD' commands at a command shell (which Windows users sometimes call a "DOS prompt", even though DOS is long gone from modern versions of Windows). A well set up environment is required. For recent versions of R, all you need to do is described in the "R Installation and Administration" manual (R Development Core Team, 2005a), Appendix "The Windows toolset". It is immensely important to follow the instructions given therein exactly. While some variations will work, you will cause yourself a lot of grief if you happen on one that does not.

We will give *an example* and describe what we did on a Windows XP machine to get a working environment. We give the URLs of downloads in the footnotes. These URLs work at the time of this writing (November, 2005), but may change over time. Check the Rtools web page[1] for updates. All the software will be installed into the directory 'c:\devel', but the user can choose arbitrary directories. It is highly recommended to avoid special characters and blanks in the directory names, though.

First, we have installed a recent (binary) version of R from a local CRAN mirror[2] into 'c:\devel\R-2.2.0' containing the executable in its 'bin' directory. Of course, it is also possible to install/compile R from sources itself, but this is a bit more complicated than just installing packages. Now follow the sections of the mentioned Appendix:

1. "The command line tools" are downloaded[3] and unzipped so that the executables are located in 'c:\devel\tools\bin'.

2. "Perl" is downloaded[4] and installed so that the executables are located in 'c:\devel\perl\bin'.

3. The current release of the "The MinGW compilers" is downloaded[5] and installed so that the executables are located in 'c:\devel\MinGW\bin'.

   Alternatively, one may download the recent versions of the components separately from http://www.mingw.org: 'gcc-core-VERSION.tar.gz', 'gcc-g++-VERSION.tar.gz', 'gcc-g77-VERSION.tar.gz', 'binutils-VERSION.tar.gz', 'mingw-runtime-VERSION.tar.gz', and 'w32api-VERSION.tar.gz' (replace 'VERSION' with the current version numbers) and unpack them manually after the rest of the setup. More details below.

4. "The Microsoft HTML Help Workshop" is downloaded[6] and installed to 'c:\devel\HtmlHelp\'.

5. "LaTeX": We choose the 'MiKTeX' distribution[7] and install it to 'c:\devel\texmf\' and 'c:\devel\localtexmf\', respectively. Therefore, the executables are in 'c:\devel\texmf\miktex\bin'.

The paths of 'Perl' and 'MiKTeX' are set automatically by their installers. However, you need to put the other tools on your path yourself. You can do this in the Windows Control Panel under "System | Advanced | Environment variables", but this will affect all programs on your system, and it may end up being modified when you install other software in the future. A better way is to create a small file

[1] http://www.murdoch-sutherland.com/Rtools
[2] A list of CRAN mirrors is given at http://CRAN.R-Project.org/mirrors.html.
[3] http://www.murdoch-sutherland.com/Rtools/tools.zip
[4] http://www.activestate.com/Products/ActivePerl/Download.html
[5] 'MinGW-5.0.0.exe' from http://sourceforge.net/projects/mingw/
[6] http://www.microsoft.com/office/ork/xp/appndx/appa06.htm
[7] http://www.miktex.org/setup.html

called ‘c:\devel\Rpaths.bat’ containing the single line (without any blanks and only using ‘;’ as the separator):

```
PATH=.;c:\devel\tools\bin;c:\devel\MinGW\bin;
c:\devel\R-2.2.0\bin;c:\devel\HtmlHelp;
c:\devel\Perl\bin;c:\devel\texmf\miktex\bin;
AllTheOtherStuff
```

where ‘`AllTheOtherStuff`’ stands for all the other stuff that was already set in your path. Most users can simply replace ‘`AllTheOtherStuff`’ by ‘`%PATH%`’, but do not include any ‘cygwin’ path. If you have ‘cygwin’ in your path before, you can also replace ‘`AllTheOtherStuff`’ by a minimal configuration that is typically ‘`%SystemRoot%\system32;%SystemRoot%`’.

You also need to add an environment variable ‘`TMPDIR`’ pointing to an existing directory to be used for temporary files as described in Section “Installing packages” of R Development Core Team (2005a). This can also go into ‘Rpath.bat’; a sample version of this file is available for download.[8].

To make sure the path is set each time you open a command shell, right click on the icon for your shell, choose “Properties | Shortcut” and change the “Target” to

```
C:\WINDOWS\system32\cmd.exe
/k c:\devel\Rpath.bat
```

(again, all on one line). Finally, edit the file ‘c:\devel\R-2.2.0\src\gnuwin32\MkRules’ and specify the directory of “The Microsoft HTML Help Workshop” at the appropriate place: `HHWDIR=c:/devel/HtmlHelp`. (Be careful here: you need to use an editor that leaves TAB characters unchanged. This file, like all of the R sources, uses Unix-style line terminators. This means that Notepad is *not* a good editor to use.)

After the steps above (which only need to be done once for each version of R), installation of a package is straightforward. For example, obtain the source (e.g. ‘abind_1.1-0.tar.gz’ [9]), and place it in ‘c:\devel’. Open the command shell, change to this directory, and type `R CMD INSTALL abind_1.1-0.tar.gz`.

Some more information and examples on package management can be found in previous Newsletters (Ligges, 2003; Ripley, 2005).

To extract the source and look into the package’s structure, type `tar -xfz abind_1.1-0.tar.gz`. This will create a subdirectory named ‘abind’ containing all of the source for the package. (You would use the same command to extract each of the MinGW components, if you had a problem above with ‘MinGW-5.0.0.exe’. Do this in the ‘c:\devel\MinGW’ directory.) By extracting the sources we have prepared an example directory (‘abind’) that can be used to (re-)build the package’s source tarball and install the package in the same way as you would do for a self written package as described in the following paragraph.

To develop your own package (e.g. ‘mypackage’), create a subdirectory ‘c:\devel\mypackage’, and follow the instructions in the “Writing R Extensions” (R Development Core Team, 2005b) manual. The function `package.skeleton()` is especially helpful when getting started. Once you have all the files in place, go into ‘c:\devel’ in the command shell and type `R CMD build mypackage`. If that works, the next step is to try to install the package as described above: `R CMD INSTALL mypackage_version.tar.gz`. Finally, try `R CMD check mypackage_version.tar.gz` for extensive tests of your code and documentation. Installing or checking directly on the source tree of a package may pollute it (e.g. with object files, ‘.chm’ files etc.), hence it is recommended to work on the already *built* package.

Bibliography

Ligges, U. (2003): R Help Desk: Package Management. *R News*, 3 (3), 37–39. ISSN 1609-3631. URL http://CRAN.R-project.org/doc/Rnews/. 28

R Development Core Team (2005a): R Installation and Administration. ISBN 3-900051-02-X. R Foundation for Statistical Computing, Vienna, Austria. 27, 28

R Development Core Team (2005b): Writing R Extensions. ISBN 3-900051-11-9. R Foundation for Statistical Computing, Vienna, Austria. 28

Ripley, B.D. (2005): Packages and their Management in R 2.1.0. *R News*, 5 (1), 8–11. ISSN 1609-3631. URL http://CRAN.R-project.org/doc/Rnews/. 28

Uwe Ligges
Fachbereich Statistik, Universität Dortmund, Germany
ligges@statistik.uni-dortmund.de
Duncan Murdoch
Statistical and Actuarial Sciences, University of Western Ontario, Canada
murdoch@stats.uwo.ca

[8] http://www.murdoch-sutherland.com/Rtools/Rpath.bat

[9] http://cran.r-project.org/src/contrib/Descriptions/abind.html