Windows Batch Files for R

G. Grothendieck

Introduction

This document describes a number of Windows batch, javascript and `.hta` files that may be used in conjunction with R. Each is self contained and independent of the others. None requires installation - just place on the Windows path. (To display the Windows path enter `path` at the Windows `cmd` line.)

`R.bat` and `Rpathset.bat` are alternatives to each other intended to facilitate the use of R without permanently modifying the Windows configuration.
`R.bat` uses heuristics to automatically locate `R`, `MiKTeX` and `Rtools`. In contrast, `Rpathset.bat` takes a simpler approach of having the user manually edit the `set` statements in it. `R.bat` does not require changes when you install a new version of R. It will automatically detect this; however, `Rpathset.bat` will require that the `set` statements be modified appropriately. `R.bat help` gives a quick overview and some examples.

`movedir.bat` and `copydir.bat` are used for moving or copying packages from one library to another such as when R is upgraded to a new version.

`el.js` runs its arguments in elevated mode (i.e. with Administrator privileges).

`clip2r.js` copies the current clipboard into a running R instance. It can be used with `vim` or other text editor.

`find-miktex.hta` displays a popup window showing where it found MiKTeX.

R.bat

Purpose

The purpose of R.bat is to facilitiate the use of R from the Windows `cmd` line by eliminating the need to make any system changes. There is no need to modify the Windows path or to set any environment variables for standard configurations of R. It will automatically locate R (and Rtools and MiKTeX if installed) and then run `R.exe`, `Rgui.exe` or other command.

Like all the other utilities here, it is a self contained no-install script with no dependencies so just place it anywhere on your Windows path.

Typical Usage

Typical usage to launch R gui is the following:

```
R gui
```

If `R.exe` were on the Windows path and before `R.bat` then it would have to be written as follows:

```
R.bat gui
```

Either of these commands runs `Rgui.exe` along with further arguments, if any. For example,

```
R gui --help
```

will run:

```
Rgui.exe --help
```

Subcommands

If the first argument is optionally one of `cd`, `cmd`, `dir`, `gui`, `help`, `path`, `R`, `script`, `show`, `SetReg`, `tools`, `touch` or the same except for upper/lower case then that argument is referred to as the subcommand.

If no subcommand is provided then the default subcommand is derived from the name of the script.

If the script is named `R.bat` then the subcommand `R` is implied. If the script has been renamed then any leading `R` is removed from the name and the remainder becomes the default subcommand. For example, if `R.bat` were renamed `Rgui.bat` then issuing `Rgui` would be the same as running `R gui` .

Other R Executables

Other executable files that come with R (`R.exe`, `Rcmd.exe`, `Rscript.exe`) can be run in a similar way:

```
R --help
R cmd --help
R script --help
```

(`RSetReg.exe` is another executable that comes with R for Windows. It will be discussed later.)

Support Subcommands

There are also some support commands:

```
R cd
R dir
R ls
R help
R show
```

`R cd` changes directory to the `R_ROOT` directory (typically `C:\Program Files\R`).

`R dir` displays the contents of that directory in chronological order - oldest first and most recent last. `R ls` is the same as `R dir`.

`R show` shows the values of the `R_` environment variables used by `R.bat` . Below is a list with typical values. These values are determined by the script heuristically (or the user can set any before running `R.bat` or `R.bat` itself can be customized by setting any of them near top of the script).

```
R_ARCH=x64
R_CMD=RShow
R_HOME=C:\Program Files\R\R-2.15.3
R_MIKTEX_PATH=\Program Files (x86)\MiKTeX 2.9\miktex\bin
R_PATH=C:\Program Files\R\R-2.15.3\bin\x64
R_REGISTRY=1
R_ROOT=C:\Program Files\R
R_TOOLS=C:\Rtools
R_TOOLS_PATH=C:\Rtools\bin;C:\Rtools\gcc-4.6.3\bin;
R_TOOLS_VERSION=3.0.0.1927
R_VER=R-2.15.3
```

`R_PATH`, `R_MIKTEX_PATH` and `R_TOOLS_PATH` are the paths to the directories holding the `R`, `MiKTeX` and `Rtools` binaries (i.e. `.exe` files).

`R_CMD` indicates the subcommand or if no subcommand specified then it is derived from the name of the script. For example if the script were renamed `Rgui.bat` then if no subcommand were specified it would default to `gui`.

`R_ROOT` is the directory holding all the R installations. `R_HOME` is the directory of the particular R installation. `R_HOME` is made up of `R_ROOT` and `R_VER` so that `R_VER` represents the directory that holds the particular R version used. `R_ARCH` is `i386` or `x64` for 32 bit or 64 bit R respectively. It can also be specified as `32` or `64` in which case it will be translated automatically.

Path Setting Subcommands

The command

```
R path
```

adds R_PATH, R_MIKTEX_PATH and R_TOOLS to the Windows path for the current cmd line session. No other cmd line sessions are affected and there are no permanent changes to the system. Once this is run the R binaries will be on the path so they can be accessed directly without R.bat .

This mode of operation has the advantage that startup will be slightly faster since the R.bat will not have to run each time that R is started. (On a 1.9 GHz Windows 8 machine R.bat show runs in 0.75 seconds.)

Note that if both R.bat and R.exe exist on the Windows path then the first on the path will be called if one uses:

```
R ...arguments...
```

thus one may wish to enter R.bat or R.exe rather than just R for clarity.

Alternately, rename R.bat to Rpath.bat in which case the command R path becomes just Rpath and R becomes unambiguous.

(An alternative to R path is the Rpathset.bat utility which will be described later.)

The command

```
R tools
```

is similar to R path except only R_TOOLS_PATH and R_MIKTEX_PATH are added to the path (but not R_PATH). This might be useful if you need to use those utilities without R.

Selecting R Version

For R installations using the standard locations and not specifying any of the R_ environment variables the registry will determine which version of R is used (assuming R_REGISTRY is not 0). If R is not found in the registry or if R_REGISTRY is 0 then the R installation in R_ROOT which has the most recent date will be used.

If we enter this at the cmd line:

```
set R_VER=R-2.14.0
```

then for the remainder of this `cmd` line session that version will be used. If one wishes to use two different R versions at once we could spawn a new `cmd` line session:

```
start
```

and then enter the same set command into the new window. Now any use of R in the original window will use the default version whereas in the new `cmd` line window it will use the specified version.

One can change the registry entry permanently to refer to a particlar version like this:

```
cmd /c set R_VER=R-2.14.0 ^& R SetReg
```

This requires Administrator privileges. If not already running as Administrator a window will pop up requesting permission to proceed.

If the registry is empty or `R_REGISTRY=0` then the default version is not determined by the registry but is determined by which R install directory is the most recent. To make a particular R install directory the most recent run the following in a `cmd` line session with Administrator privileges:

```
R dir
el cmd /c set R_VER=R-2.14.0 ^& R touch
```

The value of `R_VER` in the above line must be one of the directories listed by `R dir`. The `el.js` command used in the above code comes with these batch files and provides one way to elevate commands to have Administrator privileges.

Note that `R SetReg` and `R touch` make permanent changes to the system (namely installing or uninstalling the R key and updating the date on a particular R directory, respectively) but the other subcommands make no permanent changes.

Heuristic to Locate R

1. If `.\Rgui.exe` exists use implied `R_PATH` and skip remaining points.
2. If `.\{x64,i386}\Rgui.exe` or `.\bin\{x64,i386}\Rgui.exe` exists use implied `R_HOME`.
3. If `R_HOME` defined then derive any of `R_ROOT` and `R_VER` that are not already defined.
4. If `R_PATH` defined then derive any of `R_ROOT`, `R_HOME`, `R_VER` and `R_ARCH` that are not already defined.

5. If `R_REGISTRY=1` and R found in registry derive any of `R_HOME`, `R_ROOT` and `R_VER` that are not already defined.
6. If R_ROOT not defined try `%ProgramFiles%\R\*`, `%ProgramFiles(x86)%\R\*` and then `%SystemRoot%\R` else error.
7. If `R_VER` not defined use last directory in `cd %R_ROOT% & dir /od`.
8. if `R_ARCH` not defined try `%R_ROOT%\%R_VER%\bin\x64\Rgui.exe` and then `%R_ROOT%\%R_VER%\bin\i386\Rgui.exe`
9. If `R_ROOT`, `R_VER` and `R_ARCH` defined skip remaining points.
10. If `Rgui.exe` found on `PATH` use implied `R_PATH`.

Rpathset.bat

The command

```
Rpathset
```

adds `R_PATH`, `R_MIKTEX_PATH` and `R_TOOLS` to the Windows path for the current `cmd` line session. No other `cmd` line sessions are affected and there are no permanent changes to the system. Once this is run the R binaries will be on the path so they can be accessed directly without `R.bat` .

`Rpathset` is an alternative to

```
R path
```

but unlike `R.bat`, `Rpathset.bat` does not have any automatic heuristics. Instead, it requires that the user manually set the relevant variables in its source. Running `Rpathset.bat` then sets the path accordingly and from then on in the session one can access `Rgui.exe`, etc. on the path. Although `Rpathset.bat` involves manual editing it does have the advantage that as a consequence it is very simple – not much more than a collection of Windows batch set commands. This makes it easy to customize, less fragile to changes in the install procedures of `R` itself and is also more liklely to work on untested Windows configurations.

`Rpathset.bat` might be used like this:

```
Rpathset
Rgui
```

where `Rgui` is now directly accessing `Rgui.exe` as `Rpathset.bat` has added `R_PATH` to the Windows path.

The set statements are documented in the source of the `Rpathset.bat` file itself.

movedir.bat and copydir.bat

movedir.bat and copydir.bat move or copy the packages from one library to another. If used to transfer packages from one version of R to another it is recommended that the user run upgrade.packages() in the target. For example, assuming the default location for the user libraries:

```
cd %userprofile%\Documents\win-library
copydir 2.15\library 3.0\library

R.bat gui
... now enter update.packages() into R...
```

movedir.bat and copydir.bat will not overwrite any existing package in the target library so they are particularly safe to use. (If you do wish to overwrite such packages delete them from the target first using the Windows del command.)

el.js

el.js runs its arguments elevated (i.e. with Adminstrator privileges). For example,

```
el R touch
```

The user will be prompted to allow elevation to proceed.

clip2r.js

This program writes the clipboard into the running R session. It can be used with vim or other editor. See the source for additional instructions.

find-mixtex.hta

This program displays a window showing where MiKTeX was found. It uses the MiKTeX API. This API is not used by R.bat . Instead R.bat just looks in common places. (Using this API may be incorporated into the R.bat heuristic in the future.)

make-batchfiles-pdf.bat

This batch file creates a pdf of this documentation from the markdown file `batchfiles.md` . `pandoc` must be installed for this to run. `pandoc` can be found here. It is run without arguments:

```
make-batchfiles-pdf
```