



# How do I change the default library path for R packages

Asked 9 years, 9 months ago   Modified 1 year ago   Viewed 230k times    Part of [R Language](#) Collective

 **76**  I have attempted to install R and R studio on the local drive on my work computer as opposed to the organization network folder because anything that runs through the network is really slow. When installing, the destination path shows that it's my local C:drive. However, when I install a new package, the default path shown is my network drive and there is no option to change:

```
.libPaths()  
[1] "\\The library/path/I/don't/want"  
[2] "C:/Program Files/R/R-3.2.1/library"
```

I'm running windows 7 professional. How can I remove library path [1] and make path [2] my primary for all base packages and all new packages that I install?



Share   Improve this question

Follow

edited Oct 15, 2018 at 14:00



[duckmayr](#)

**17k**   3   39   62

asked Jul 29, 2015 at 18:04






[user3594490](#)

**1,999**   2   20   27

## 8 Answers

Sorted by: Highest score (default) 

 **81**  Windows 7/10/11: If your C:\Program Files (or wherever R is installed) is blocked for writing, as mine is, then you'll get frustrated editing Rprofile.site (as I did). As specified in the accepted answer, I updated R\_LIBS\_USER and it worked. However, even after reading the fine manual several times and extensive searching, it took me several hours to do this. In the spirit of saving someone else time...

 Let's assume you want your packages to reside in C:\R\Library :



1. Create the folder C:\R\Library . Next I need to add this folder to the R\_LIBS\_USER path:
2. Click Start --> Control Panel --> User Accounts --> Change my environmental variables
3. The Environmental Variables window pops up. If you see R\_LIBS\_USER , highlight it and click Edit . Otherwise click New . Both actions open a window with fields for Variable and Value .
4. In my case, R\_LIBS\_USER was already there, and value was a path to my desktop. I added to the path the folder that I created, separated by semicolon.  
C:\R\Library;C:\Users\Eric.Krantz\Desktop\R stuff\Packages .

(NOTE: In the last step, I could have removed the path to the Desktop location and simply left `C:\R\Library` ).

Share Improve this answer Follow

edited Mar 6, 2024 at 14:43

answered Aug 20, 2016 at 16:34



Eric Krantz

2,259 1 17 30

3 Thank you for the step by step, appears to work for me in Windows 10 as well. – Aaron Robeson May 20, 2020 at 14:12

This worked for me but I'm getting an error with some packages (tidyverse, sf, AER so far): 'The procedure entry point EXTPTR\_PTR could not be located in the dynamic link library.' Anyone get this? – winitheju Aug 27, 2020 at 20:30

@winitheju : see the note from sandJohnson at [github.com/r-lib/rlang/issues/416](https://github.com/r-lib/rlang/issues/416) – Eric Krantz Aug 28, 2020 at 13:53

Works for Windows 11 as well. In the search bar, type "Enviro" and "Edit environment variables for your account" comes up. That puts you at Step 3 above. – Eric Krantz Nov 9, 2023 at 21:36

Please use `R_LIBS_USER= C:\R\Library\%v` so the day you upgrade R you don't have packages compiled for a previous R version in your path. – zeehio Nov 9, 2024 at 19:43



See `help(Startup)` and `help(.libPaths)` as you have several possibilities where this may have gotten set. Among them are

68



- setting `R_LIBS_USER`
- assigning `.libPaths()` in `.Rprofile` OR `Rprofile.site`



and more.



In this particular case you need to go backwards and unset wherever `\\\\\\The library/path/I/don't/want` is set.



To otherwise ignore it you need to override it use explicitly i.e. via

```
library("somePackage", lib.loc=.libPaths()[-1])
```

when loading a package.

Share Improve this answer Follow

edited Jul 29, 2015 at 18:13

answered Jul 29, 2015 at 18:07



Dirk is no longer here

369k 59 666 741

9 Thanks so much, I just set `.libPaths("C:/Program Files/R/R-3.2.1/library")` and now that's the only path that shows. – user3594490 Jul 29, 2015 at 20:11

1 Interesting. I thought this *added* rather than overriding? – Dirk is no longer here Jul 29, 2015 at 20:35

And it sure *adds* to an an existing `.libPaths()` on my Unix box. – Dirk is no longer here Jul 29, 2015 at 20:40

In the help file it seemed to indicate to use the command to set the library path. Similar to setting the working directory. I could be wrong but that appears to be how it functioned for me. The first network path no longer appeared. The problem is when installing the base package I have to do it off network or it by default saves the program on my C:Drive as well as a copy on our network. I can't prevent the base library from being saved on the network, only any additional packages I install. – [user3594490](#)  
Jul 29, 2015 at 22:46

- 
- 1 @DirkEddelbuettel I believe this function *doesn't* add to the existing `.libPaths()`. It takes the specified path (or vector of), appends the system lib path if it's not included, and sets. Any other paths that might have previously been added, including user-specific stuff from the initialization process, would need to be explicitly retained using `.libPaths(c(mydir1,mydir2,.libPaths()))` or some such. – [user6702291](#) Dec 7, 2016 at 16:35
- 



Facing the very same problem (avoiding the default path in a network) I came up to this solution with the hints given in other answers.

26



The solution is editing the `Rprofile` file to overwrite the variable `R_LIBS_USER` which by default points to the home directory.



Here the steps:



1. Create the target destination folder for the libraries, e.g., `~\target`.
2. Find the `Rprofile` file. In my case it was at `C:\Program Files\R\R-3.3.3\library\base\R\Rprofile`.
3. Edit the file and change the definition the variable `R_LIBS_USER`. In my case, I replaced the this line `file.path(Sys.getenv("R_USER"), "R",` with `file.path("~\target", "R",`.

The documentation that support this solution is [here](#)

Original file with:

```
if(!nzchar(Sys.getenv("R_LIBS_USER")))  
  Sys.setenv(R_LIBS_USER=  
    file.path(Sys.getenv("R_USER"), "R",  
              "win-library",  
              paste(R.version$major,  
                    sub("\\.*$", "", R.version$minor),  
                    sep=".")  
  ))
```

Modified file:

```
if(!nzchar(Sys.getenv("R_LIBS_USER")))  
  Sys.setenv(R_LIBS_USER=  
    file.path("~\target", "R",  
              "win-library",  
              paste(R.version$major,  
                    sub("\\.*$", "", R.version$minor),  
                    sep=".")  
  ))
```



- 5 I recently used the procedure on a machin with Win 7 and "~\target" did not work, I had to use "~/target". I have never understood properly why and when the use of "/" or "\" matters. – Marco Jun 13, 2017 at 22:14



## Windows 10 on a Network

7



Having your packages stored on the network drive can slow down the performance of R / R Studio considerably, and you spend a lot of time waiting for the libraries to load/install, due to the bottlenecks of having to retrieve and push data over the server back to your local host. See the following for instructions on how to create an .Rprofile on your local machine:

1. Create a directory called C:\Users\xxxxxx\Documents\R\3.4 (or whatever R version you are using, and where you will store your local R packages- your directory location may be different than mine)
2. On R Console, type `Sys.getenv("HOME")` to get your home directory (this is where your .Rprofile will be stored and R will always check there for packages- and this is on the network if packages are stored there)
3. Create a file called `.Rprofile` and place it in `:\YOUR\HOME\DIRECTORY\ON_NETWORK` (the directory you get after typing `Sys.getenv("HOME")` in R Console)
4. File contents of `.Rprofile` should be like this:

```
#search 2 places for packages- install new packages to first directory- load built-in packages from the second (this is from your base R package- will be different for some)
```

```
.libPaths(c("C:\Users\xxxxxx\Documents\R\3.4", "C:/Program Files/Microsoft/R Client/R_SERVER/library"))
```

```
message("*** Setting libPath to local hard drive ***")
```

```
#insert a sleep command at line 12 of the unpackPkgZip function. So, just after the package is unzipped.
```

```
trace(utils:::unpackPkgZip, quote(Sys.sleep(2)), at=12L, print=TRUE)
```

```
message("*** Add 2 second delay when installing packages, to accommodate virus scanner for R 3.4 (fixed in R 3.5+)***")
```

```
# fix problem with tcltk for sqldf package:
```

```
https://github.com/ggrothendieck/sqldf#problem-involvlng-tcltk
```

```
options(gsubfn.engine = "R")
```

```
message("*** Successfully loaded .Rprofile ***")
```

5. Restart R Studio and verify that you see that the messages above are displayed.

Now you can enjoy faster performance of your application on local host, vs. storing the packages on the network and slowing everything down.

Share Improve this answer Follow

edited Nov 25, 2020 at 16:17

answered Sep 10, 2020 at 22:37



Carlo Carandang

225 2 8

Also, if you are having problems loading packages, you may need to increase the time further to accommodate the virus scanner for R 3.4 (fixed in R 3.5+): `trace(utils:::unpackPkgZip, quote(Sys.sleep(20)), at=12L, print=TRUE)` `message("*** Add 20 second delay when installing packages, to accommodate virus scanner for R 3.4 (fixed in R 3.5+)***")`  
– Carlo Carandang Mar 24, 2021 at 16:05



6



I was struggling for a while with this as my work computer (with Windows 10) created the default user library on a network drive, which would slow down R and RStudio to an unusable state.

In case this helps someone, this is the easiest way I found, without requiring admin rights:

- make sure the directory you want to install your packages into exists. If you want to respect the convention, use: `C:\Users\username\R\win-library\rversion` (for example, something like: `C:\Users\janebloggs\R\win-library\3.6`)
- create a `.Renviron` file in your home directory (which might be on the network drive?), and in it, write one single line that defines the `R_LIBS_USER` variable to be your custom path:

```
R_LIBS_USER=C:\Users\janebloggs\R\win-library\3.6
```

(feel free to add comments too, with lines starting with `#`)

If a `.Renviron` file exists, R will read it at startup and use the variables as they are defined in there, before running the code in the `.Rprofile`. You can read about it in `help(Startup)`.

Now it should be persistent between sessions!

Share Improve this answer Follow

answered Mar 25, 2020 at 11:39




stragu

1,321 12 16

1 This is the only solution that worked for me on a couple of different computers (my own and work) - can this also be the selected answer? – ChrisP Jun 3, 2020 at 16:20

1 @ChrisP : There can't be several "ticked" answers, no, although the person who asked the questions can change their mind. The currently selected answer does (vaguely) cover my answer though, by

mentioning that the `R_LIBS_USER` variable can be defined by hand. It's just a shame it is not highlighted as the most straight-forward of the persistent solutions. – [stragu](#) Jun 5, 2020 at 5:06 



3



I think I tried all of the above and it didn't work for me. This worked, though:

1. In home directory, make a file called ".Renviron"
2. In that file, write:

```
.libPaths(new = "/my/path/to/Libs")
```

3. Save and restart R if you had it open

Share Improve this answer Follow

answered May 6, 2021 at 22:49



[BGranato](#)

169 1 11

[stackoverflow.com/questions/40788645/...](https://stackoverflow.com/questions/40788645/...) more details about .Renviron – [Lefty](#) Nov 17, 2022 at 9:18



2



After a couple of hours of trying to solve the issue in several ways, some of which are described here, for me (on Win 10) the option of creating a Renviron file worked, but a little different from what was written here above.

The task is to change the value of the variable `R_LIBS_USER`. To do this two steps needed:

1. Create the file named Renviron (without dot) in the folder `\Program\etc\` (Program is the directory where R is installed--for example, for me it was `C:\Program Files\R\R-4.0.0\etc`)
2. Insert a line in Renviron with new path: `R_LIBS_USER = "C:/R/Library"`

After that, reboot R and use `.libPaths()` to confirm the default directory changed.

Share Improve this answer Follow

edited Mar 12, 2021 at 13:58

answered Jun 3, 2020 at 12:40



[coip](#)

1,520 20 34



[Tor](#)

21 1

For me the problem was that I couldn't create or find an .Renviron file on my PC even after changing File Explorer settings to show hidden files. The solution was to use the `versions` package to install an older version of the `usethis` package and then use `usethis::edit_r_environ()` to edit the .Renviron file by adding the following line: `R_LIBS_USER = "C:/target/directory"`, where the target directory is wherever you want to map `.libPaths()` to instead of the network drive. – [coip](#) Mar 11, 2021 at 17:37

- 1 If you have multiple versions of R on your computer--which normally store packages in different subfolders--modifying .Renviron will direct all versions to the same folder, which can create compatibility problems with various packages and their dependencies. The workaround is to create multiple subfolders for each version (e.g. `"C:/R/Library/3.5"` and `C:/R/Library/4.0"` and then use `%v` at the end of your `R_LIBS_USER` statement: `R_LIBS_USER = C:/R/Library/%v`. That way, whichever version of R you open will load up packages from the appropriate subfolder. – [coip](#) Mar 15, 2021 at 23:10

- 1 @coip Two quick notes of thanks and elaboration: (1) for those who are not handy with environment text, the %v can go at any point in the string, e.g C:/R/%v/Library, though using the default structure is probably better; and (2) the %v is also very elegant because if you have packages saved to a default directory different than the version of R, this can cause problems in operation, because by definition, any 'base' packages are being pulled from the version of R, not your default, and this may cause conflicts in operations. Using the %v allows you always to be harmonious in your packages  
– [glenninboston](#) Oct 29, 2021 at 14:24 ✎



For linux (I'm using Fedora 38 XFCE), in the `~/.bash_profile` file I added:

0

```
# R
export R_PROFILE_USER="${HOME}/.Rprofile"
export R_LIBS_USER="${HOME}/path/wanted/as/R/library"
```



Afterwards, I created the `~/.Rprofile` file with this content:



```
.libPaths("~/path/wanted/as/R/library")
```

Pay attention to the absence of the `${HOME}` environment variable (replaced by `tilde`, `~`, expansion) in the path indicated for `.libPaths()` into `~/.Rprofile` file, because otherwise it will not work.

Share Improve this answer Follow

edited Apr 13, 2024 at 19:43

answered Apr 13, 2024 at 19:30



[Riccardo Volpe](#)

1,633 1 17 33



**Highly active question.** Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.

## Start asking to get answers

Find the answer to your question by asking.

[Ask question](#)

## Explore related questions

r

See similar questions with these tags.