

# Change R default library path using .libPaths in Rprofile.site fails to work

Asked 12 years, 2 months ago   Modified 8 months ago   Viewed 735k times    Part of [R Language](#) Collective



I am running R on Windows, not as an administrator. When I install a package, the following command doesn't work:

289



```
> install.packages("zoo")
Installing package(s) into 'C:/Program Files/R/R-2.15.2/library'
(as 'lib' is unspecified)
Warning in install.packages :
  'lib = "C:/Program Files/R/R-2.15.2/library"' is not writable
```



To install a package, I have to specify a library location:

```
install.packages("zoo", lib="C:/software/Rpackages")
```

To load a package, I also have to specify the library location:

```
library("zoo", lib.loc="C:/software/Rpackages")
```

All of this is OK, but I wanted to see if I could add `C:/software/Rpackages` to the library path somehow and thus not have to type it each time.

As I searched online, I found that one way to do this is to edit the `Rprofile.site` file and to add the line

```
.libPaths("C:/software/Rpackages")
```

However, after doing this, and starting RStudio, this is the output that I get

```
> .libPaths()
[1] "C:/Program Files/R/R-2.15.2/library" "C:/Program Files/RStudio/R/library"
```

The `.libPaths` command that I added to the `Rprofile.site` doesn't seem to have had any effect! Why is this the case? Or more importantly, how can I fix the problem so that I can install and load packages without typing in the library location?

Note: if I start RStudio the `.libPaths()` command seems to work as it is supposed to

```
.libPaths("C:/software/Rpackages")
> .libPaths()
[1] "C:/software/Rpackages"           "C:/Program Files/R/R-2.15.2/library"
```

Isn't that strange?

[r](#) [rstudio](#) [r-faq](#) [library-path](#)

Share Improve this question

edited Nov 5, 2018 at 9:51

asked Mar 2, 2013 at 3:36

Follow



[zx8754](#)

56.4k

12

126

226



[I Like to Code](#)

7,351

13

41

50

Can you expand a bit on how exactly `install.packages("zoo")` doesn't work? Error message, etc.? – [joran](#) Mar 2, 2013 at 3:45

1 @joran I added the error message when I try `install.packages("zoo")` ---but that is not my real question, which is about why `.libPaths()` in `Rprofile.site` doesn't work. – [I Like to Code](#) Mar 2, 2013 at 3:49

1 Do you have the same trouble if you use RGui instead of R Studio? – [Dason](#) Mar 2, 2013 at 7:35

7 Why does R's library paths default to folders that need admin permissions?! Surely the user's home directory would be more sensible – [geoththeory](#) Dec 4, 2015 at 15:28

[superuser.com/questions/749283/...](#) Edit `~/.Renviro R_LIBS_USER=/some/path` – [bowen.li](#) Dec 14, 2017 at 5:40

## 17 Answers

Sorted by: Highest score (default)



311



The proper solution is to set environment variable `R_LIBS_USER` to the value of the file path to your desired library folder as opposed to getting RStudio to recognize a `Rprofile.site` file.

To set environment variable `R_LIBS_USER` in Windows, go to the Control Panel (System Properties -> Advanced system properties -> Environment Variables -> User Variables) to a desired value (the path to your library folder), e.g.

Variable name: `R_LIBS_USER`  
Variable value: `C:/software/Rpackages`

If for some reason you do not have access to the control panel, you can try running `rundll32 sysdm.cpl,EditEnvironmentVariables` from the command line on Windows and add the environment variable from there.

Setting `R_LIBS_USER` will ensure that the library shows up first in `.libPaths()` regardless of starting RStudio directly or by right-clicking an file and "Open With" to start RStudio.

The Rprofile solution can work if RStudio is always started by clicking the RStudio shortcut. In this case, [setting the default working directory to the directory that houses your Rprofile](#) will be sufficient. The Rprofile solution does not work when clicking on a file to start RStudio because that changes the working directory away from the default working directory.



pollytatouin

81 1 6



Paweł Rumian

3,846 3 22 27

- 30 Definitely the proper solution. The only catch is if your user is not a admin (likely the case if you're having this problem to begin with), you have to change your environment variables via Control Panel->User Accounts->User Accounts->Change my environment variables. – David Moyer Mar 2, 2015 at 16:46
- 1 Do you (happen to) know the equivalent in Linux command line? – Diego-MX Apr 8, 2015 at 22:39
- 11 Sure, just set these as your shell variables (in bash you can edit `.bash_profile` and/or `.bashrc`). See here: [unix.stackexchange.com/questions/117467/...](http://unix.stackexchange.com/questions/117467/...) For setting these for current session only type in console `export R_LIBS_USER='directory_of_your_choice'` – Paweł Rumian Apr 9, 2015 at 10:43
- Magic solution! this worked perfect to me in RStudio with Windows7 – Darwin PC Apr 24, 2015 at 0:39
- 7 instead of setting the variable through control panel, I have modified the default value of `R_LIB_USER` in the Rprofile file. Here the solution: I came up with a similar approach, the solution is presented in as a solution to this question [stackoverflow.com/questions/31707941/...](http://stackoverflow.com/questions/31707941/...) – Marco Mar 7, 2017 at 8:52



227

I generally try to keep all of my packages in one library, but if you want to add a library why not append the new library (which must already exist in your filesystem) to the existing library path?



```
.libPaths( c( .libPaths(), "~/userLibrary" ) )  
# obviously this would need to be a valid file directory in your OS  
# min just happened to be on a Mac that day
```



Or (and this will make the userLibrary the first place to put new packages):



```
.libPaths( c( "~/userLibrary" , .libPaths() ) )
```

Then I get (at least back when I wrote this originally):

```
> .libPaths()  
[1] "/Library/Frameworks/R.framework/Versions/2.15/Resources/library"  
[2] "/Users/user_name/userLibrary"
```

The `.libPaths` function is a bit different than most other nongraphics functions. It works via side-effect. The functions `Sys.getenv` and `Sys.setenv` that report and alter the R environment variables have been split apart but `.libPaths` can either report or alter its target.

The information about the R startup process can be read at `?Startup` help page and there is RStudio material at: <https://support.rstudio.com/hc/en-us/articles/200549016-Customizing-RStudio>

In your case it appears that RStudio is not respecting the `Rprofile.site` settings or perhaps is overriding them by reading an `.Rprofile` setting from one of the RStudio defaults. It should also be mentioned that the result from this operation also appends the contents of calls to `.Library` and `.Library.site`, which is further reason why an RStudio- (or any other IDE or network installed-) hosted R might exhibit different behavior.

Since `Sys.getenv()` returns the current system environment for the R process, you can see the library and other paths with:

```
Sys.getenv()[ grep("LIB|PATH", names(Sys.getenv())) ]
```

The two that matter for storing and accessing packages are (now different on a Linux box):

```
R_LIBS_SITE           /usr/local/lib/R/site-  
library:/usr/lib/R/site-library:/usr/lib/R/library  
R_LIBS_USER           /home/david/R/x86_64-pc-linux-gnu-  
library/3.5.1/
```

Share Improve this answer Follow

edited Feb 25, 2021 at 16:24

answered Mar 2, 2013 at 4:39



IRTFM

264k

22

379

500

13 I needed to reverse the `c()` vector to ensure new libraries go to the custom folder – [geotheory](#) Feb 10, 2015 at 14:31

Excellent point. In fact the `.libPaths` -function have the same effect by default. There is also a difference between OSes in how trailing slashes are handled. (Not OK on Windows). Other options exist. See `?libPaths` – [IRTFM](#) Feb 10, 2015 at 16:20

6 The new path has to exist in the file system actually, otherwise it won't be added to the library path – [jarandaf](#) Apr 18, 2015 at 11:19

Another good point. Although the function name suggests otherwise, you can first check with `file.exists` and proceed only if the path is not missing. – [IRTFM](#) May 13, 2015 at 16:30

5 I did this and it works, but it seems that I have to change the path every time I start R. Is that the same for others? – [cyrusjan](#) May 20, 2016 at 8:45



I managed to solve the problem by placing the code in the `.Rprofile` file in the default working directory.

28



First, I found the location of the default working directory

```
> getwd()  
[1] "C:/Users/me/Documents"
```



Then I used a text editor to write a simple `.Rprofile` file with the following line in it

```
.libPaths("C:/software/Rpackages")
```

Finally, when I start R and run `.libPaths()` I get the desired output:

```
> .libPaths()  
[1] "C:/software/Rpackages"          "C:/Program Files/R/R-2.15.2/library"  
[3] "C:/Program Files/RStudio/R/library"
```

Share Improve this answer Follow

answered Mar 4, 2013 at 6:33



I Like to Code

7,351 13 41 50

This doesn't work for me (neither does the solution from BondedDust). I created a txt file and saved it as `.Rprofile` in the folder from `getwd()`. – John Apr 7, 2014 at 18:01

2 @John I believe that we are saying the same thing. – I Like to Code Apr 7, 2014 at 21:04 ✎

6 R reads `.Rprofile` from your **HOME** directory not from your **CURRENT** directory. – kgui Jun 21, 2016 at 21:29

I came up with a similar approach, the solution is presented in as a solution to this question

[stackoverflow.com/questions/31707941/...](https://stackoverflow.com/questions/31707941/...) – Marco Mar 7, 2017 at 8:49

To find the home directroy issue the r command `Sys.getenv("HOME")` in the r console. I am not sure if you get the desired output. I think the desired output is that what 42 showed in his answer

[stackoverflow.com/a/15170774/754550](https://stackoverflow.com/a/15170774/754550) . He uses `.libPaths` in a slihly different way. – miracle173 Mar 15, 2017 at 8:07



24

Edit ~/.Renviron



R\_LIBS\_USER=/some/path



Share Improve this answer Follow

answered Dec 14, 2017 at 5:45



bowen.li

449 4 6



13

I found what I think is a solution [here](#) (thank you Carl Schwarz at SFU) for adding a personal library that is permanently (you don't have to define it each session) recognized whether using R or Rstudio, and Rstudio treats it as the default on my Mac machine. I hadn't seen it laid out this explicitly on SO, so I summarized the steps they provided, for Windows and then for Mac.



For a Windows 7 OS:

1. Create a directory on the drive where you want to have your personal library, e.g. `C:\User\Rlibs` (or another that you have permissions to)

2. Search for/go to "Edit environment variable for your account" in the Windows search bar to edit control panel settings
3. Click "New..." in the middle of the "Environmental Variables" window
4. In the "New User Variable" window, type **R\_LIBS** for the "Variable name", and the path to the personal library directory you created, e.g. **C:\User\Rlibs**
5. Click OK and you should see the Variable/Value pair in the User variables window
6. Click OK again

Now when you start R (or Rstudio) and type the command `.libPaths()` you should see the personal library you created as well as the R system library.

For Mac:

1. In your "Home" or "username" directory create a folder called **Rlibs**
2. Launch the Terminal application
3. Type: `echo "R_LIBS=~/.Rlibs" > .Renvrion` Make sure the spelling and case matches.
4. Type `ls -a` to see the full list of files in the directory, which should now include **.Renvrion**
5. Verify that the **.Renvrion** file has been set properly: `more .Renvrion`

Launch R/Rstudio and type `.libPaths()` and you should see the new path to your personal library.

Share Improve this answer Follow

edited Nov 16, 2022 at 6:43



Swapneel Mehta

65 1 2 12

answered May 4, 2017 at 14:01



Whalen

141 1 3



If you do not have admin-rights, it can also be helpful to open the `Rprofile.site` -file located in `\R-3.1.0\etc` and add:

9



```
.First <- function(){
  .libPaths("your path here")
}
```



This evaluates the `.libPath()` command directly at start

Share Improve this answer Follow

answered Apr 14, 2014 at 13:46



schlusie

2,017 3 22 26



On Ubuntu, the [recommended way](#) of changing the default library path for a user, is to set the `R_LIBS_USER` variable in the `~/.Renvrion` file.

6



```
touch ~/.Renvirom
echo "R_LIBS_USER=/custom/path/in/absolute/form" >> ~/.Renvirom
```



Share Improve this answer Follow edited Nov 30, 2019 at 8:15

answered Nov 29, 2019 at 13:32



[tzabal](#)

451 5 5



just change the default folder for your R libraries in a directory with no Administrator rights, e.g.

5

```
.libPaths("C:/R/Libs")
```



Share Improve this answer Follow

answered Nov 14, 2018 at 11:12



[Sébastien Wieckowski](#)

842 15 17



I've had real trouble understanding this. gorkypl gave the correct solution above when I last re-installed my OS & Rstudio but this time round, setting my environment variable didn't resolve.

4



Uninstallled both R and Rstudio, creating directories C:\R and C:\Rstudio then reinstalled both.



Define R\_LIBS\_USER user variable to your prepered directory (as per gorkypl's answer) and restart your machine for User variable to be loaded. Open Rstudio, errors should be gone.

You can also use `sys.setenv()` to modify R\_LIBS\_USER to the path of your alternative library which is easier and does not need to restart your computer.

To see what R\_LIBS\_USER is set to: `?Sys.getenv()`

Reading `help(Startup)` is useful.

Share Improve this answer Follow edited Mar 10, 2016 at 21:24

answered Jan 7, 2015 at 16:40



Community Bot

1 1



[felixmc](#)

518 1 5 20



If your default package library has been changed after installing a new version of R or by any other means, you can append both the libraries to use all the packages with the help of the commands below. Get the existing library path :

2



```
.libPaths()
```



Now, set the existing and the old path :

```
.libPaths(c(.libPaths(), "~/yourOldPath"))
```

Hope it helps.

Share Improve this answer Follow

answered Mar 14, 2016 at 7:07



Shivong Birla

21 3



2



I had the same problem and I run into [this](#). If you want to create another location `c("C:/Users/mynewlocation")` should be working as well. As mentioned [in here](#) "You should be able to right-click on the Rstudio.exe icon, click Properties, and select an option to always run Rstudio as administrator. Be sure you use that same icon whenever you want to open Rstudio."



```
myPaths <- .libPaths() # get the paths
myPaths <- c(myPaths[2], myPaths[1]) # switch them
.libPaths(myPaths) # reassign them
```

Share Improve this answer Follow

edited Mar 12, 2022 at 10:08

answered Mar 12, 2022 at 9:49



Seyma Kalay

2,861 16 39



1



I read the readme. In that they mentioned use `.libPaths()` in command line to check which paths are there. I had 2 library paths earlier. When I used the command `.libpath("C:/Program Files/R/R-3.2.4revised/library")` where I wanted, it changed the library path. When I typed in `.libPaths()` at the command line again it showed me the correct path. Hope this helps



Share Improve this answer Follow

answered Jul 8, 2016 at 4:11



Supriya Deverkonda

11 1



1



Since most of the answers here are related to Windows & Mac OS, (and considering that I also struggled with this) I decided to post the process that helped me solve this problem on my **Arch Linux** setup.

### Step 1:



- Do a global search of your system (e.g. **ANGRYSearch**) for the term `Renviron` (which is the configuration file where the settings for the user libraries are set).
- It should return only two results at the following directory paths:

1. `/etc/R/`

2. `/usr/lib/R/etc/`

**NOTE:** The `Renviron` config files stored at 1 & 2 (above) are hot-linked to each other (which means changes made to one file will automatically be applied [ in the same form / structure ] to the other file when the file being edited is saved - [ you also need `sudo` rights for saving the file post-edit ] ).



## Step 2:

- Navigate into the 1st directory path ( /etc/R/ ) and open the `Renviron` file with your favourite text editor.
- Once inside the `Renviron` file search for the `R_LIBS_USER` tag and update the text in the curly braces section to your desired directory path.

### EXAMPLE:

... **Change From** ( *original entry* ):

```
R_LIBS_USER=${R_LIBS_USER-'~/R/x86_64-pc-linux-gnu-library/4.0'}
```

... **Change To** ( *your desired entry* ):

```
R_LIBS_USER=${R_LIBS_USER-'~/Apps/R/rUserLibs'}
```

## Step 3:

- Save the `Renviron` file you've just edited ... **DONE !!**

Share Improve this answer Follow

answered Sep 25, 2020 at 11:15



SilSur

519 1 6 24



```
getwd()
# [1] "C:/Users/....software/My R studio"
```

0



copy the above link with double inverted comma



```
.libPaths(new="C:/Users/....software/My R studio")
```



Your default path will change for installing packages

Share Improve this answer Follow

edited Nov 5, 2018 at 9:50



zx8754

56.4k 12 126 226



answered Feb 7, 2016 at 13:48

Sayed Mohammed  
Zeeshan

11



If you want to change your library path permanently (without calling `.libPath()` every time when entering in R, this works for me:

0



1. create `.Rprofile` under your home directory. (`~/Rprofile`)

2. type `.libPaths(c( .libPaths(), "your new path" ))` in `.Rprofile` file, save.

3. open R (any directory) and check, just type `.libPaths()` , you can find your library path is updated!



Share Improve this answer Follow

answered Jul 3, 2019 at 19:12



Pengyao

827 10 15



I know this is going to sound crazy, but the issue (if you're one Windows) is that you didn't launch R Studio as an administrator.

0



You'll know this is you if you click the dropdown menu to choose your install packages location and it won't let you choose a different one.



Just close R Studio. Find the app. Right click it and "Run as administrator"



You will then be able to change the installation location.

Share Improve this answer Follow

answered Aug 17, 2024 at 4:28



Jef Jeffrey Harding

Hardi 60 5



-2



I was looking into this because R was having issues installing into the default location and was instead just putting the packages into the temp folder. It turned out to be the latest update for McAfee Endpoint Security which apparently has issues with R. You can disable the threat protection while you install the packages and it will work properly.



Share Improve this answer Follow

answered Sep 18, 2017 at 23:15



Kristen Bystrom

1



**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

rstudio

r-faq

library-path

See similar questions with these tags.





