

Fixing R Package Installation Errors in MacOS Big Sur, Monterey & Ventura

Yiqing Xu
Twitter @xuyiqing

Motivation

After upgrading my MacOS to Big Sur, I can no longer install many packages from tar.gz files or Github repos. I suspected it had something to do with the C++ compiler in Big Sur (I was right). I spent over 10 hours trying to fix this issue. Below is my solution, which I hope can help save your time.

The Error

The error looks something like this (even after Steps 1-3 are done) :

```
ld: framework not found CoreFoundation
collect2: error: ld returned 1 exit status
make: *** [isoband.so] Error 1
ERROR: compilation failed for package 'isoband'
* removing '/Library/Frameworks/R.framework/Versions/4.0/Resources/library/isoband'
Warning in install.packages :
  installation of package 'isoband' had non-zero exit status
```

A Step-by-Step Fix

Step 1. In App Store, install Xcode



Step 2. Install Xcode command line tools, in your terminal, after the \$ sign, type

```
$ xcode-select install
```

After this step, you can try the following two lines of codes in R to see if errors pop out:

```
> install.packages("Rcpp", type = "source")
> install.packages("RcppArmadillo", type = "source")
```

If you encounter an error like what is shown above, keep going.

Step 3. Install a GNU gfortran build (a dmg or pkg file) for your Mac from this Github repo:

<https://github.com/fxcoudert/gfortran-for-macOS/releases>

For example, I install the following one.

[gfortran 10.2 for Big Sur \(macOS 11\), for Intel processors](#)

 fxcoudert released this on Dec 19, 2020 · 1 commit to master since this release

Standalone installer of GCC 10.2, including gfortran 10.2, for macOS Big Sur (macOS 11). This is for machines with Intel processor (or ARM running under Rosetta 2).

Assets 3	
 gfortran-10.2-BigSur-Intel.dmg	104 MB
 Source code (zip)	
 Source code (tar.gz)	

Note 1: GCC (including a C++ compiler) is a part of gfortran.

Note 2: After a few hours of search, I figured that the gcc compiler inside Xcode cannot do the job – this may change in the future.

Note 3: The installer will install gfortran in this folder: `/usr/local/gfortran/`

Step 4. Set paths

Step 4.1. Now you need to make sure that your gcc compiler can be found and used by R. In your terminal, try:

```
$ gcc --version
```

```
Home-iMac:~ xyq$ gcc --version
gcc (GCC) 10.2.0
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

If you don't see something like this, it means the path is not correctly set, in which case, you need to add a line in a file called `“.profile”` or `“.bash_profile”` (if you use bash) or `“.zshrc”` (if you use zsh)-- it's normally hidden in your home folder; please google how to find it or create it.

```
export PATH="/usr/local/gfortran/bin:/Library/Frameworks/R.framework/Resources:$PATH"
```

The `“/usr/local/gfortran/bin”` part is for the compiler and the `“/Library/Frameworks/R.framework/Resources”` part is for the system to find R – for some strange reasons, my Mac cannot find where R is installed, if that's not a problem for you, simply add:

```
export PATH="/usr/local/gfortran/bin:$PATH"
```

Then check gcc version again. If this doesn't work, you can change the \$PATH variable for everybody using this computer by editing `“/etc/paths”`: in terminal, type

```
$ sudo nano /etc/paths
```

Then add `/usr/local/gfortran/bin` at the beginning of the file; and use `Ctrl+X` to save and exit.

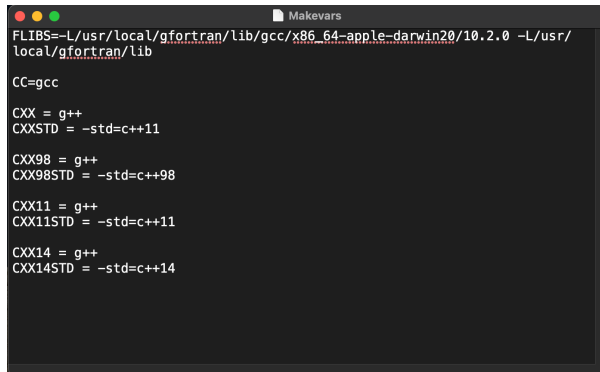
Step 4.2. Now you need to make sure that R can find the compiler. Add the following line into your `“.Renviron”` file – if it does not exist, create one by typing `“touch .Renviron”` in your terminal

```
PATH="/usr/local/gfortran/bin:${PATH}"
```

Some told me their problem is fixed after this step. Mine didn't go away until Step 5.

Step 5. Make sure R uses the right compiler.

In `.R` folder (a hidden folder), edit a file called `“Makevars”`. Mine has the following lines

A screenshot of a terminal window titled "Makevars" showing the following configuration lines:

```
FLIBS=-L/usr/local/gfortran/lib/gcc/x86_64-apple-darwin20/10.2.0 -L/usr/local/gfortran/lib
CC=gcc
CXX = g++
CXXSTD = -std=c++11
CXX98 = g++
CXX98STD = -std=c++98
CXX11 = g++
CXX11STD = -std=c++11
CXX14 = g++
CXX14STD = -std=c++14
```

```
FLIBS=-L/usr/local/gfortran/lib/gcc/x86_64-apple-darwin20/10.2.0 -L/usr/local/gfortran/lib
```

This line helps R files find the right gfortran `“bin”` (again, Step 4.2 is not sufficient in my case). Note that `“x86_64-apple-darwin20/10.2.0”` can be different for you. You need to check your folder system to find the right gcc version.

```
CC = gcc
CXX = g++
CXX98 = g++
CXX11 = g++
CXX14 = g++
CXX17 = g++
CXX20 = g++
CXXCPP = g++
FC = gfortran
F77 = gfortran
OBJC = gcc
OBJCXX = g++
```

These lines make sure R uses `gcc/g++` to compile C and C++ code, not `clang`. I find that `clang` doesn't work on my Mac with Big Sur.

Problem with Parallel Computing using the “doParallel” package

Many packages with parallel computing functionality no long work with R under MacOS BigSur. The solution is to install the “future” package. See below for a snippet of sample code. You may want to update your packages.

```
#install.packages("future")
library(future)
library(doParallel)
cl2 <- future::makeClusterPSOCK(4,verbose = F)
doParallel::registerDoParallel(cl2)
foreach(i = seq_len(2e4), .combine = 'c') %dopar% {
  sqrt(i)
}
stopCluster(cl2)
```

```
#install.packages("future")
library(future)
library(doParallel)
cl2 <- future::makeClusterPSOCK(4,verbose = F)
doParallel::registerDoParallel(cl2)
foreach(i = seq_len(2e4), .combine = 'c') %dopar% {
  sqrt(i)
}
stopCluster(cl2)
```

Hope this helps.

Reference:

<https://cloud.tencent.com/developer/article/1756392>