

OpenMP on macOS with Xcode tools

Warning! Everything described on this page is strictly experimental and not officially supported by CRAN, R-core or R Foundation. It may break at any time. The information is provided in the hope of being useful to some tech-savvy people. It is not intended for the regular R user.

For those impatient, skip to [how to enable OpenMP in packages](#).

OpenMP support in Xcode

Apple has explicitly disabled OpenMP support in compilers that they ship in Xcode:

```
$ clang -c omp.c -fopenmp
clang: error: unsupported option '-fopenmp'
```

even though `clang` had OpenMP support for quite a long time now (great thanks to the folks at Intel providing their library as open source!). In fact, the clang compiler in Xcode can generate all the necessary code for OpenMP. It can be tricked into performing its designed function by using `-Xclang -fopenmp` flags.

The unfortunate part about this is that Apple is not shipping the necessary `libomp.dylib` run-time library needed for OpenMP support. To make things worse, the version of the library you need depends on the clang version used, which Apple obfuscates so that it's non-trivial to reverse-engineer it. Fortunately, some clever folks were able to find the [traces in Apple's released source](#) so we can build the binaries that correspond to the `clang` version used. It is sometimes possible to use a more recent version of the runtime than the version of Apple clang.

OpenMP run-time downloads

The following are links to `libomp` OpenMP run-time built from official LLVM release sources using Xcode compilers. They are signed and support macOS 10.13 (High Sierra) and higher. All tar-balls contain the system tree `usr/local/lib` and `usr/local/include` so the recommended installation is:

```
curl -O https://mac.r-project.org/openmp/openmp-12.0.1-darwin20-Release.tar.gz
sudo tar fxvz openmp-12.0.1-darwin20-Release.tar.gz -C /
```

The contained set of files is the same in all tar balls:

```
usr/local/lib/libomp.dylib
usr/local/include/ompt.h
usr/local/include/omp.h
usr/local/include/omp-tools.h
```

so you can simply remove those to uninstall. Note that any package you compile against `libomp.dylib` will need that run-time so you have to ship it with your package or have users install it. You can verify the signature in the library via `codetool` (see below).

Build	Download	SHA1 checksum
LLVM 13.0.0 (built on macOS 12, but works on macOS 11)	openmp-13.0.0-darwin21-Release.tar.gz (Release) openmp-13.0.0-darwin21-Debug.tar.gz (Debug)	47af4cb0d1f3554969f2ec9dee450d728ea30024 f6f8f1f49c02d5ec0729b56ddc7eaf51e7f04714
LLVM 12.0.1 Xcode 13+ (Apple clang 1300.x)	openmp-12.0.1-darwin20-Release.tar.gz (Release) openmp-12.0.1-darwin20-Debug.tar.gz (Debug)	4fab53ccc420ab882119256470af15c210d19e5e 58b4323e7933e12cba5c2996b4c9ef27567c41d9
LLVM 11.0.1 (+M1 patch) Xcode 12.5 (Apple clang 1205.x)	openmp-11.0.1-darwin20-Release.tar.gz (Release) openmp-11.0.1-darwin20-Debug.tar.gz (Debug)	0dcd19042f01c4f552914e2cf7a53186de397aa1 65e83ea667c72bbe44fea699776564d2f03a080f
(All binaries above include both arm64 and x86_64 and require macOS 11 or higher. Binaries below require macOS 10.13 or higher and are Intel-only)		
LLVM 10.0.0 Xcode 12.0-12.4 (Apple clang 1200.x)	openmp-10.0.0-darwin17-Release.tar.gz (Release) openmp-10.0.0-darwin17-Debug.tar.gz (Debug)	9bf16a64ab747528c5de7005a1ea1a9e318b3cf0 d4508d3f0c2952c3f984393b088e0b4beab33b58
LLVM 9.0.1 Xcode 11.4-11.7 (Apple clang 1103.x)	openmp-9.0.1-darwin17-Release.tar.gz (Release) openmp-9.0.1-darwin17-Debug.tar.gz (Debug)	e5bd8501a3f957b4babe27b0a266d4fa15dbc23f c4c8491631504fb060f7c25ec14324d02d617d5b
LLVM 8.0.1 Xcode 11.0-11.3.1 (Apple clang 1100.x)	openmp-8.0.1-darwin17-Release.tar.gz (Release) openmp-8.0.1-darwin17-Debug.tar.gz (Debug)	e4612bfc1bf520bf22844f7db764cadb7577c28 d6c83918b28405d43950d4b864ca8d1687eed4d1
LLVM 7.1.0 Xcode 10.2-10.3 (Apple clang 1001.x)	openmp-7.1.0-darwin17-Release.tar.gz (Release) openmp-7.1.0-darwin17-Debug.tar.gz (Debug)	6891ff6f83f2ed83eed42160de819b50cf643cd 34456adde62b9a1047f906e1d7f54990a1c15a34

How to enable OpenMP in packages

- Download the `libomp` run-time corresponding to the Xcode version you use from the links above
- add `-Xclang -fopenmp` to `CPPFLAGS`, add `-lomp` to `LIBS`

How you do the latter depends on the package, but if the package does not set these environment variables itself, you can try

```
PKG_CPPFLAGS='-Xclang -fopenmp' PKG_LIBS=-lomp R CMD INSTALL myPackage
```

If that doesn't work, please consult the package's documentation, and liaise with its maintainer. It is also possible to add those flags globally by adding the following to `~/R/Makevars`:

```
CPPFLAGS += -Xclang -fopenmp
LDFLAGS += -lomp
```

but be very careful when doing this, always check your `~/R/Makevars` whenever you upgrade R, macOS or Xcode.

Side notes

It may be possible in principle to build static version of the run-time. That can be done via `-DLIBOMP_ENABLE_SHARED=OFF`, but has not been tested. There is a potential for chaos when more OMP run-times get loaded into one process. Another interesting test would be to try `-DLIBOMP_FORTRAN_MODULES=ON` and see which versions are compatible with the GNU Fortran used in R. We are intentionally removing the `gomp` symlink from the binary so that `lomp` and `gomp` don't conflict.

License and sources

All sources were obtained directly from the [LLVM releases](#) and copies are also available in the `src` directory (including the build script). See `LICENSE.txt` in the sources for the corresponding license and <https://openmp.llvm.org/> for details on the OpenMP run-time.

Verifying code signatures

You can use `codetools` to verify the signature, for example you should see:

```
$ codesign -d -vv /usr/local/lib/libomp.dylib
Executable=/usr/local/lib/libomp.dylib
Identifier=libomp
Format=Mach-O universal (x86_64 arm64)
CodeDirectory v=20400 size=5514 flags=0x0(none) hashes=167+2 location=embedded
Signature size=8927
Authority=Developer ID Application: Simon Urbanek (VZLD955F6P)
Authority=Developer ID Certification Authority
Authority=Apple Root CA
Timestamp=11/11/2021 at 1:29:04 PM
Info.plist=not bound
TeamIdentifier=VZLD955F6P
Sealed Resources=none
Internal requirements count=1 size=168
```

Acknowledgements

Thanks to John Clayden, Kevin Ushey and others who contributed to the discussion and testing.

Trademark notices

- The OpenMP name and the OpenMP logo are registered trademarks of the OpenMP Architecture Review Board.
- Intel is a trademark of Intel Corporation in the U.S. and/or other countries.
- Apple, Xcode and macOS are trademarks of Apple Inc., registered in the U.S. and other countries.

Last modified on 2021/11/11 by Simon Urbanek