How to Deploy Your Qt Cross-Platform Applications to macOS Operating System Using macdeployqt

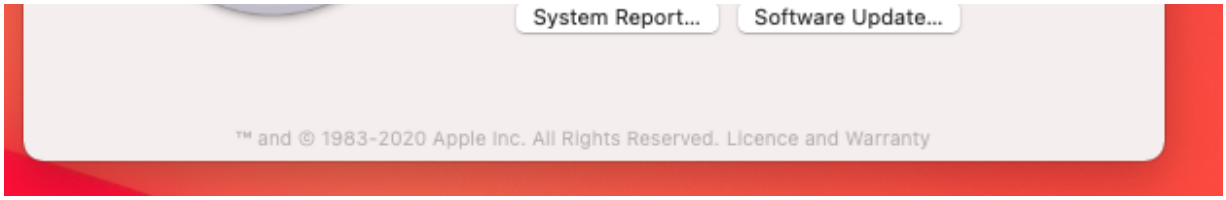
This tutorial explains the manner to deploy cross-platform projects made with the Qt framework to macOS Operating Systems using macdeployqt.

George Calin Follow

Jan 5 · 5 min read ★

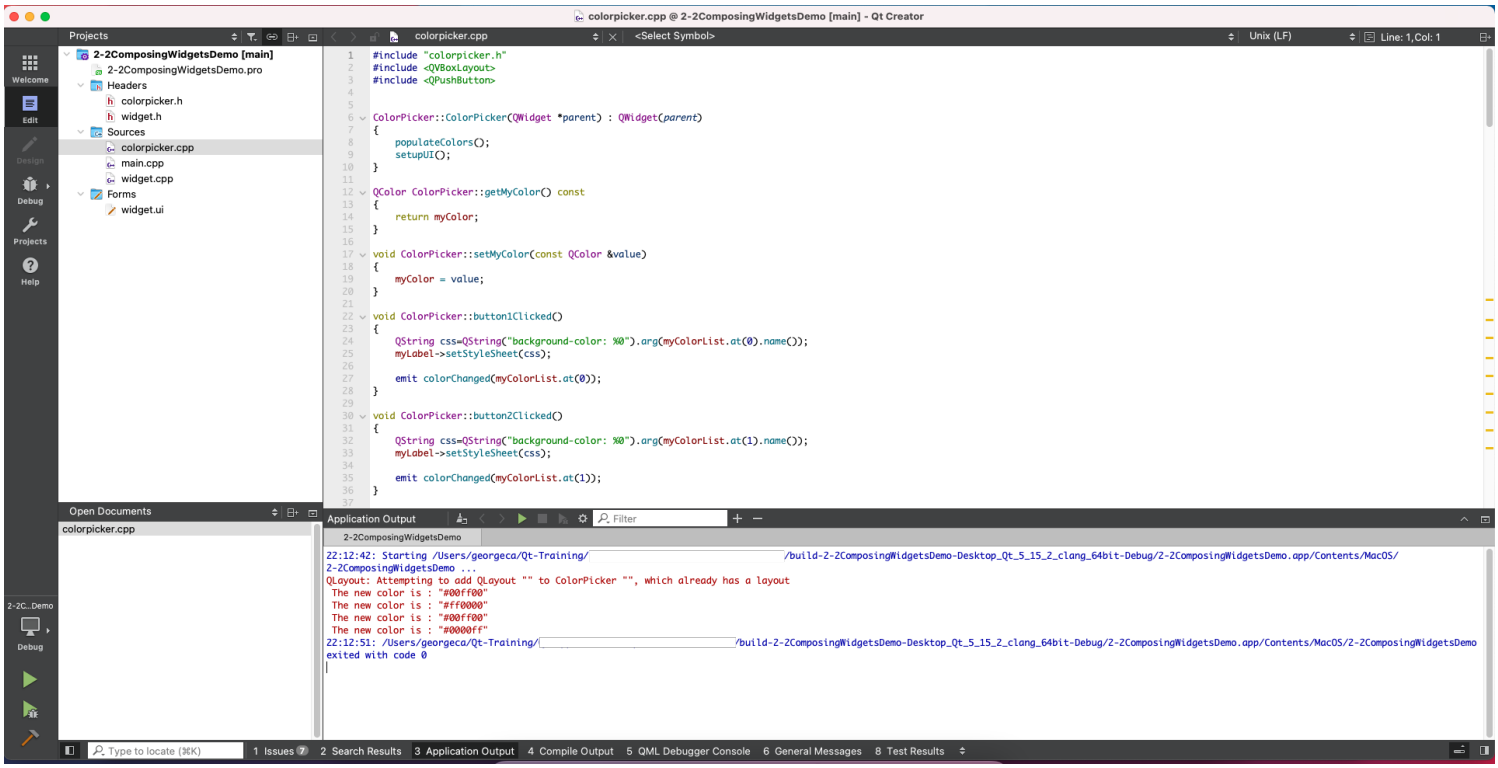
Set up the environment

- operating system information (macOS)



the example of configuration on my machine

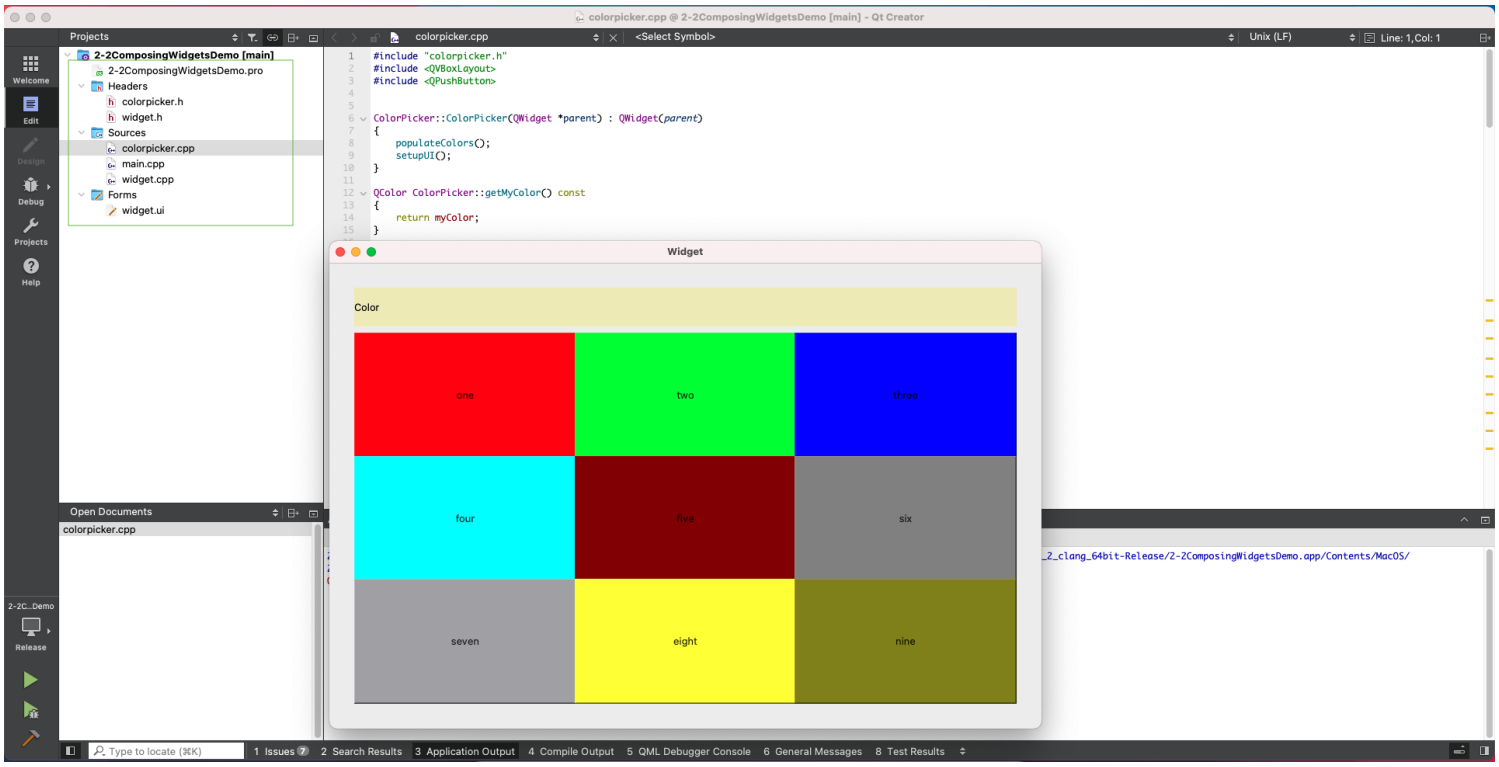
- a set-up of Qt Creator IDE (for macOS)



If you do not have it already, **Qt Creator IDE can be downloaded** and installed from :

> **Qt Creator - A Cross-platform IDE for Application Development**
>
> Qt Creator is a cross-platform integrated development environment (IDE) built for the…
>
> www.qt.io

- Qt code from an application



If you wish to get conversant with the Qt cross-platform framework, I recommend following along the courses on Udemy from the following authors:

Daniel Gakwaya | Software Engineer at Blikoon Technologies| Udemy

Daniel is a Senior Software Engineer at Blikoon Technologies. He has been writing software since…

www.udemy.com

Bryan Cairns | Computer Guru| Udemy

"Bryan has a few courses available on Udmey, specifically his course on Qt 5 Design Patterns and…

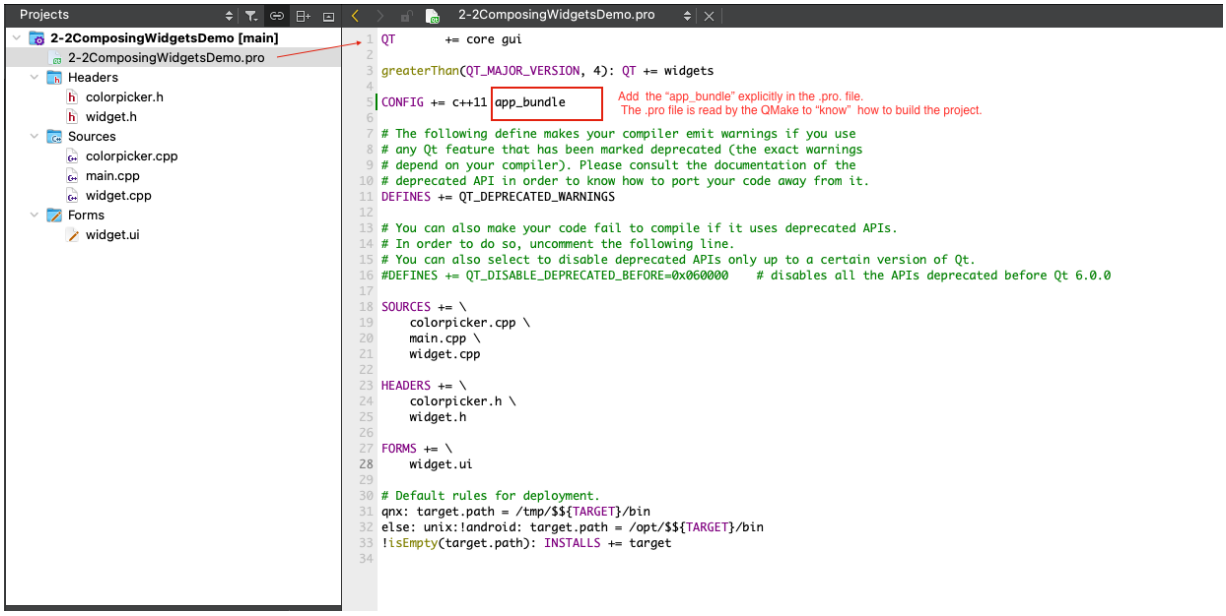
www.udemy.com

Go deploy the Qt cross-platform application using macdeployqt

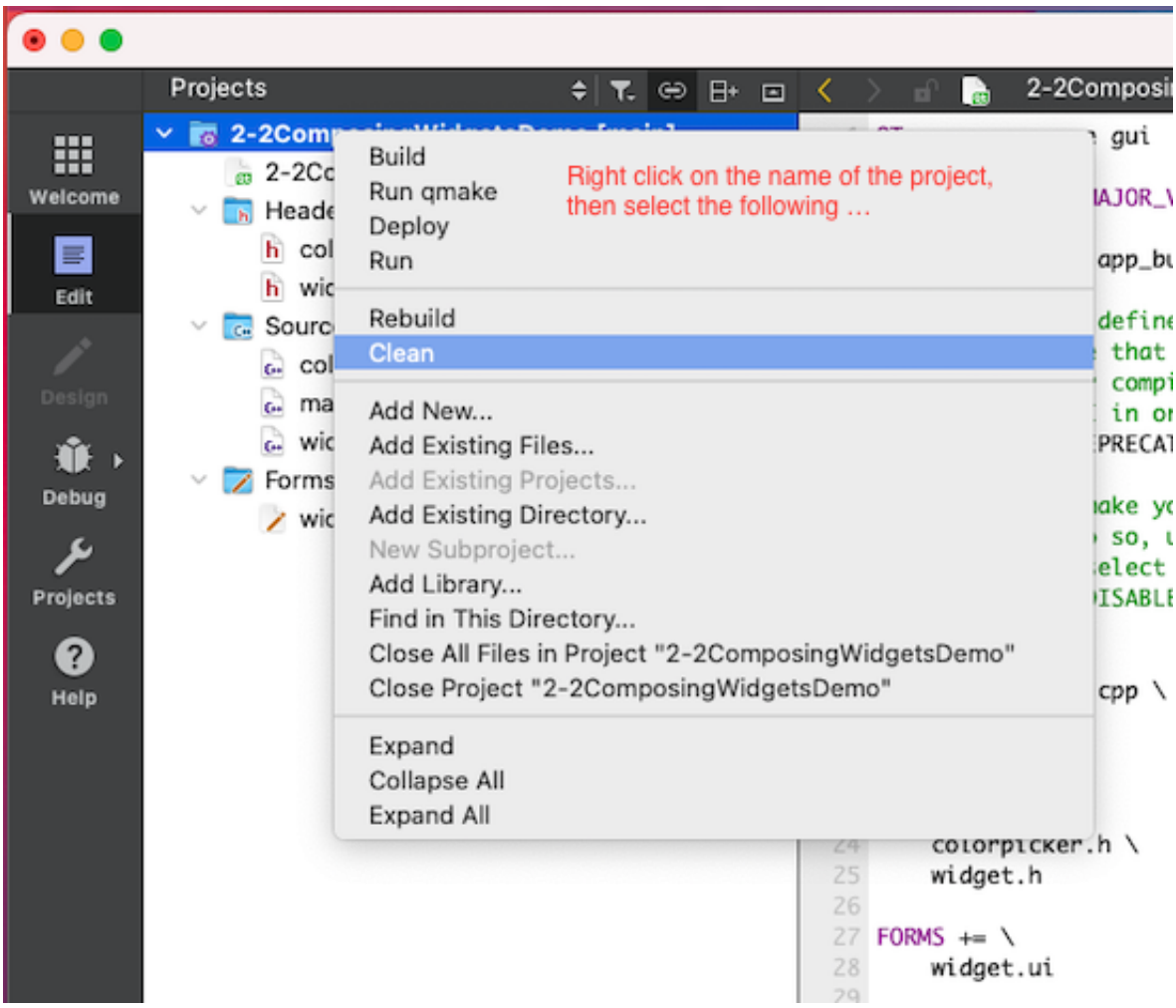
macOS uses the concept of bundling. A macOS application is actually a folder that holds a .app extension.

Step 1. Add the “app_bundle” specification in the project’s .pro file

QMake should generate automatically a bundle for the application in the build folder. Nevertheless, it is recommended to add explicitly this in the .pro file:

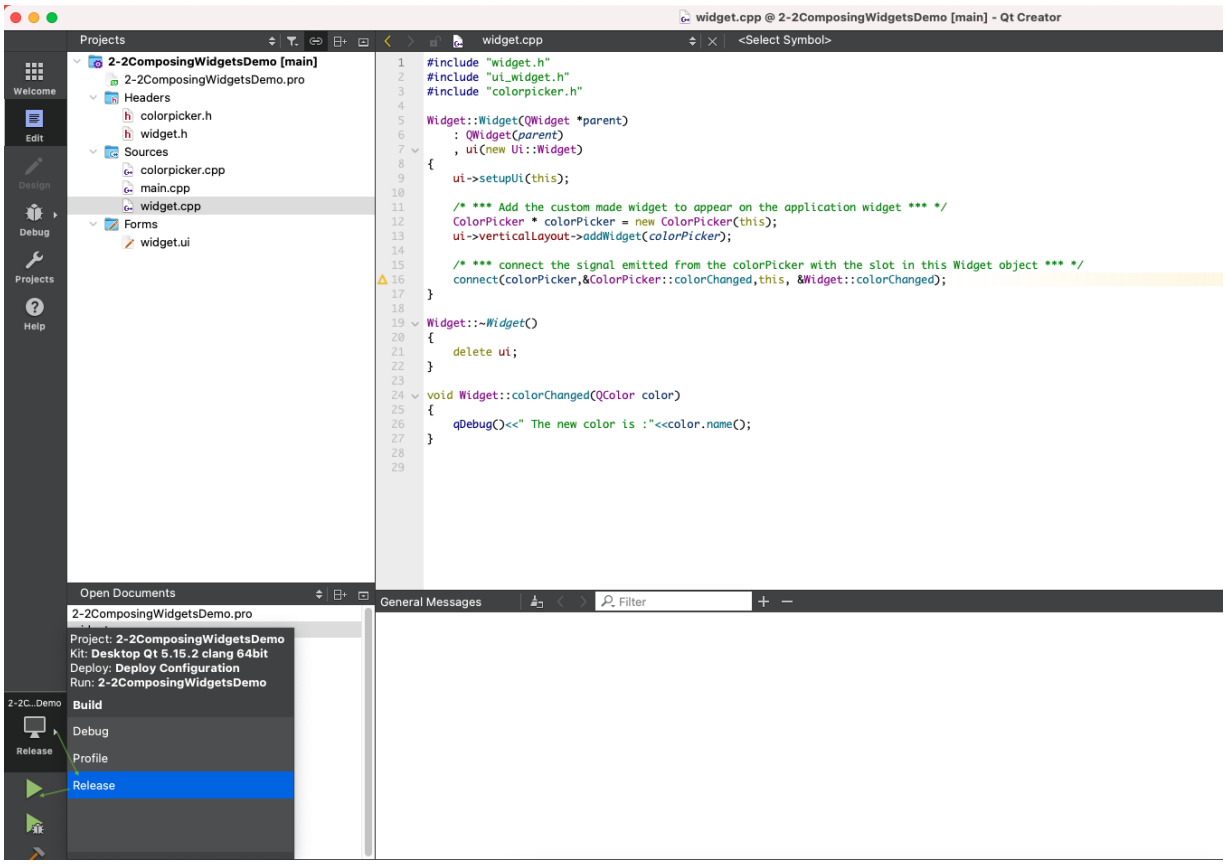


Step 2. Clean the project from the previous builds



Cleaning the project from previous builds is a good practice to avoid errors that might be related to other builds you have already done.

Step 3. Build the project using the “Release” option



Step 4. Deploy the application for usage in MacOS

4.1. Locate the ‘**macdeployqt**’ **binary** inside your Qt installation

```
georgeca@Georges-Mini ~ % ls -lh
total 0
drwx------@ 17 georgeca  staff   544B Jan  4 22:32 Desktop
drwx------@  5 georgeca  staff   160B Aug 12 21:40 Documents
drwx------+ 15 georgeca  staff   480B Jan  3 23:59 Downloads
drwx------@ 73 georgeca  staff   2.3K Jan  3 22:52 Library
drwx------+  4 georgeca  staff   128B Jul 21 18:42 Movies
drwx------+  7 georgeca  staff   224B Jan  3 22:56 Music
drwx------+  6 georgeca  staff   192B Aug 12 20:51 Pictures
drwxr-xr-x+  4 georgeca  staff   128B Jul 21 18:30 Public
drwxr-xr-x  16 georgeca  staff   512B Jan  4 22:12 Qt
drwxr-xr-x   5 georgeca  staff   160B Jan  4 21:15 Qt-Training
georgeca@Georges-Mini ~ % ls -lh Qt
total 2168
drwxrwxr-x   4 georgeca  staff   128B Jan  4 22:11 5.15.2
drwxrwxr-x   3 georgeca  staff    96B Jan  4 22:11 Docs
drwxrwxr-x   3 georgeca  staff    96B Jan  4 22:11 Examples
-rw-r--r--   1 georgeca  staff   321K Jan  4 22:12 InstallationLog.txt
drwxr-xr-x   8 georgeca  staff   256B Jan  4 22:12 Licenses
drwxrwxrwx   3 georgeca  staff    96B Jan  4 22:12 MaintenanceTool.app
-rw-r--r--   1 georgeca  staff   728K Jan  4 22:12 MaintenanceTool.dat
-rw-r--r--   1 georgeca  staff    13K Jan  4 22:12 MaintenanceTool.ini
drwxrwxr-x   3 georgeca  staff    96B Jan  4 22:11 Qt Creator.app
drwxrwxr-x   4 georgeca  staff   128B Jan  4 22:12 Tools
-rw-r--r--   1 georgeca  staff   5.0K Jan  4 22:12 components.xml
drwxrwxr-x   3 georgeca  staff    96B Jan  4 22:11 dist
drwxr-xr-x  11 georgeca  staff   352B Jan  4 22:12 installerResources
-rw-r--r--   1 georgeca  staff   362B Jan  4 22:12 network.xml
```

```
georgeca@Georges-Mini Qt % cd 5.15.2
georgeca@Georges-Mini 5.15.2 % ls -lh
total 8
drwxrwxr-x  11 georgeca  staff   352B Jan  4 22:11 clang_64
-rw-rw-r--   1 georgeca  staff   2.2K Nov 13 08:10 sha1s.txt
georgeca@Georges-Mini 5.15.2 % cd clang_64
georgeca@Georges-Mini clang_64 % ls -lh
total 0
drwxr-xr-x   56 georgeca  staff   1.8K Jan  4 22:12 bin
drwxr-xr-x    4 georgeca  staff   128B Jan  4 22:11 doc
drwxr-xr-x   18 georgeca  staff   576B Jan  4 22:11 include
drwxr-xr-x  121 georgeca  staff   3.8K Jan  4 22:11 lib
drwxr-xr-x   79 georgeca  staff   2.5K Jan  4 22:11 mkspecs
drwxr-xr-x   15 georgeca  staff   480B Jan  4 22:11 phrasebooks
drwxr-xr-x   28 georgeca  staff   896B Jan  4 22:11 plugins
drwxr-xr-x   23 georgeca  staff   736B Jan  4 22:11 qml
drwxr-xr-x  285 georgeca  staff   8.9K Jan  4 22:11 translations
georgeca@Georges-Mini clang_64 % cd bin
```

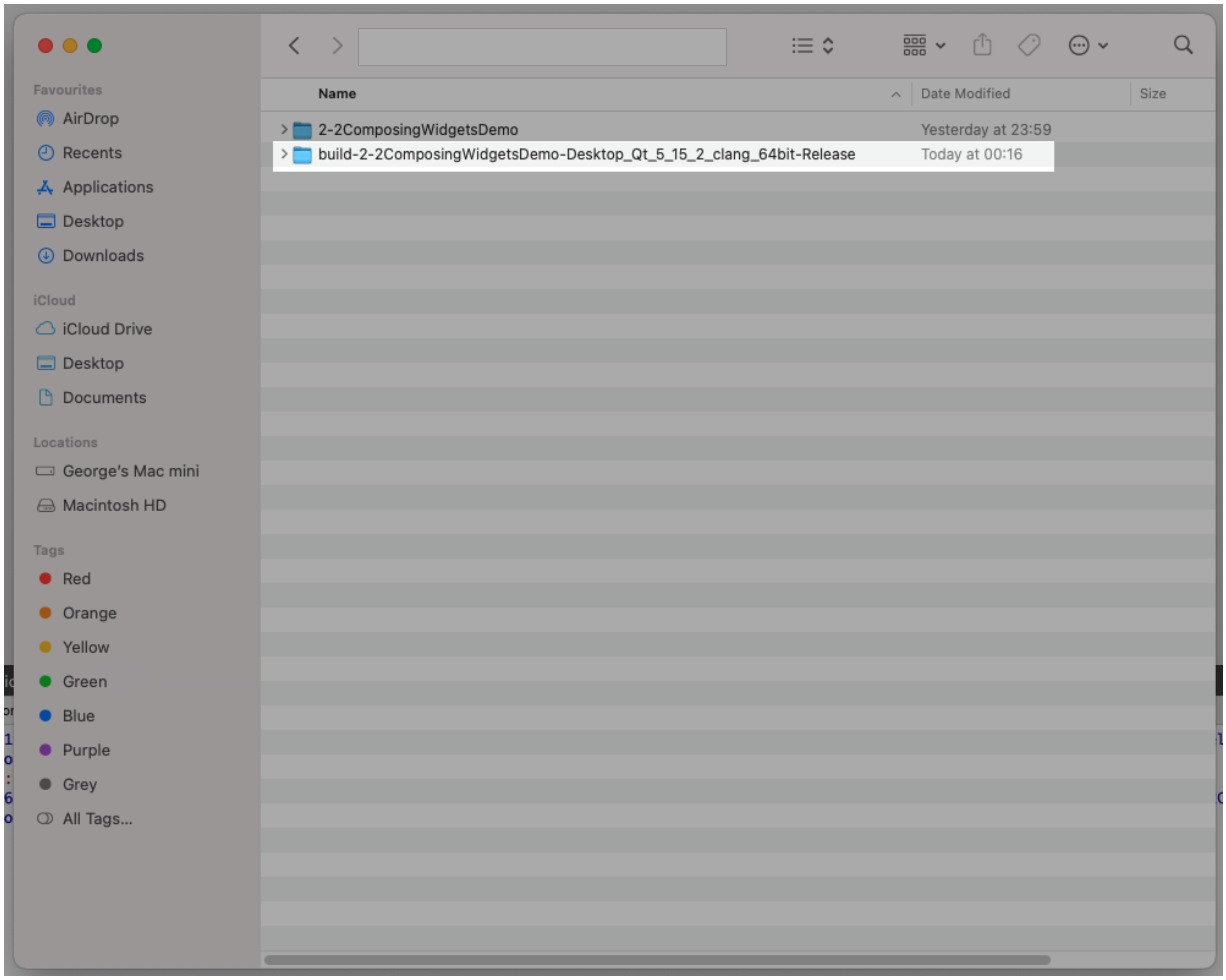
```
georgeca@Georges-Mini clang_64 % cd bin
georgeca@Georges-Mini bin % ls -lh
total 221224
drwxr-xr-x  3 georgeca  staff    96B Jan  4 22:11 Assistant.app
drwxr-xr-x  3 georgeca  staff    96B Jan  4 22:11 Designer.app
drwxr-xr-x  3 georgeca  staff    96B Jan  4 22:11 Linguist.app
-rwxr-xr-x  1 georgeca  staff    53K Nov  6 10:24 canbusutil
-rwxr-xr-x  1 georgeca  staff   6.2K Oct 27 10:02 fixqt4headers.pl
-rwxr-xr-x  1 georgeca  staff   257K Nov  6 10:56 lconvert
-rwxr-xr-x  1 georgeca  staff    94K Nov 12 19:08 licheck_mac
-rwxr-xr-x  1 georgeca  staff   322K Nov  6 10:56 lprodump
-rwxr-xr-x  1 georgeca  staff   289K Nov  6 10:56 lrelease
-rwxr-xr-x  1 georgeca  staff    31K Nov  6 10:55 lrelease-pro
-rwxr-xr-x  1 georgeca  staff   792K Nov  6 10:56 lupdate
-rwxr-xr-x  1 georgeca  staff    31K Nov  6 10:56 lupdate-pro
-rwxr-xr-x  1 georgeca  staff   207K Nov  6 10:50 macchangeqt
-rwxr-xr-x  1 georgeca  staff   228K Nov  6 10:50 macdeployqt
-rwxr-xr-x  1 georgeca  staff   983K Nov  6 07:17 moc
drwxr-xr-x  3 georgeca  staff    96B Jan  4 22:11 pixeltool.app
-rwxr-xr-x  1 georgeca  staff    13K Nov  6 10:49 qcollectiongenerator
-rwxr-xr-x  1 georgeca  staff    67K Nov  6 10:50 qdbus
-rwxr-xr-x  1 georgeca  staff   278K Nov  6 07:19 qdbuscpp2xml
drwxr-xr-x  3 georgeca  staff    96B Jan  4 22:11 qdbusviewer.app
-rwxr-xr-x  1 georgeca  staff    78K Nov  6 07:19 qdbusxml2cpp
-rwxr-xr-x  1 georgeca  staff   137K Nov  6 10:49 qdistancefieldgenerator
-rwxr-xr-x  1 georgeca  staff    74M Nov  6 10:52 qdoc
-rwxr-xr-x  1 georgeca  staff   8.1M Nov  6 11:17 qgltf
-rwxr-xr-x  1 georgeca  staff   167K Nov  6 10:51 qhelpgenerator
-rwxr-xr-x  1 georgeca  staff   125K Nov  6 07:18 qlalr
-rwxr-xr-x  1 georgeca  staff   6.2M Nov  6 07:16 qmake
drwxr-xr-x  3 georgeca  staff    96B Jan  4 22:11 qml.app
-rwxr-xr-x  1 georgeca  staff   908K Nov  6 10:36 qmlcachegen
-rwxr-xr-x  1 georgeca  staff   109K Nov  6 10:36 qmleasing
-rwxr-xr-x  1 georgeca  staff   476K Nov  6 10:36 qmlformat
-rwxr-xr-x  1 georgeca  staff   405K Nov  6 10:36 qmlimportscanner
-rwxr-xr-x  1 georgeca  staff   1.0M Nov  6 10:36 qmllint
-rwxr-xr-x  1 georgeca  staff   134K Nov  6 10:36 qmlmin
-rwxr-xr-x  1 georgeca  staff   171K Nov  6 10:36 qmlplugindump
-rwxr-xr-x  1 georgeca  staff   112K Nov  6 10:36 qmlpreview
-rwxr-xr-x  1 georgeca  staff   212K Nov  6 10:36 qmlprofiler
-rwxr-xr-x  1 georgeca  staff    71K Nov  6 10:36 qmlscene
-rwxr-xr-x  1 georgeca  staff    28K Nov  6 10:36 qmltestrunner
-rwxr-xr-x  1 georgeca  staff   109K Nov  6 10:24 qmltyperegistrar
-rwxr-xr-x  1 georgeca  staff   414K Nov  6 10:51 qscxmlc
-rw-rw-rw-  1 georgeca  staff    87B Jan  4 22:12 qt.conf
-rwxr-xr-x  1 georgeca  staff    78K Nov  6 10:49 qtattributionsscanner
-rwxr-xr-x  1 georgeca  staff    82K Nov  6 10:51 qtdiag
-rwxr-xr-x  1 georgeca  staff    40K Nov  6 10:50 qtpaths
-rwxr-xr-x  1 georgeca  staff    33K Nov  6 10:49 qtplugininfo
-rwxr-xr-x  1 georgeca  staff    41K Nov  6 07:18 qvkgen
-rwxr-xr-x  1 georgeca  staff   860K Nov  6 07:17 rcc
-rwxr-xr-x  1 georgeca  staff   407K Nov  6 10:52 repc
-rwxr-xr-x  1 georgeca  staff    47K Oct 27 10:02 syncqt.pl
-rwxr-xr-x  1 georgeca  staff   668K Nov  6 07:17 tracegen
-rwxr-xr-x  1 georgeca  staff   524K Nov  6 07:19 uic
-rwxr-xr-x  1 georgeca  staff    79K Nov  6 10:58 xmlpatterns
-rwxr-xr-x  1 georgeca  staff    19K Nov  6 10:58 xmlpatternsvalidator
```

4.2. Note the path to the ‘macdeployqt’ :

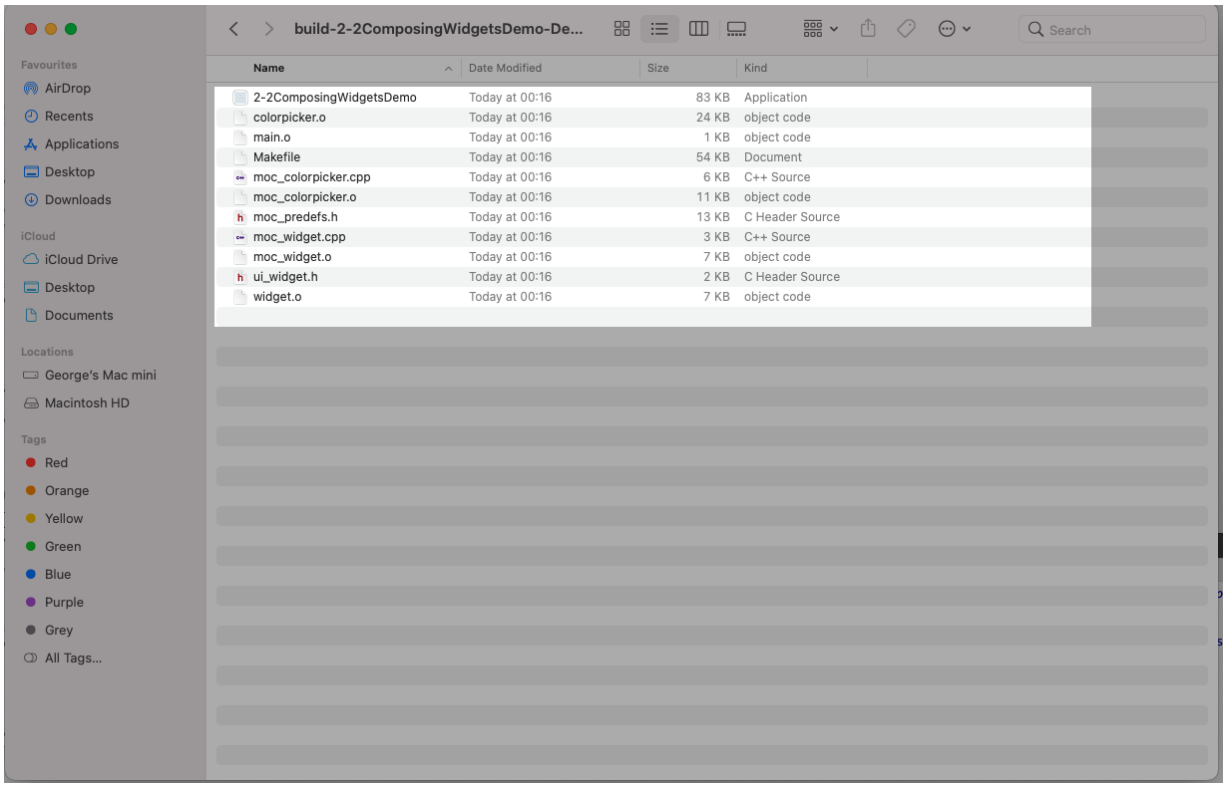
```
georgeca@Georges-Mini bin % pwd
/Users/georgeca/Qt/5.15.2/clang_64/bin
```

4.3. Go to the build directory

This directory is generated after building the Qt code (you did this at Step 3)



using the GUI



Same thing, but using the command line:

```
georgeca@Georges-Mini % ls -lh
total 0
drwxr-xr-x  11 georgeca  staff   352B Jan  4 23:59 2-2ComposingWidgetsDemo
drwxr-xr-x  14 georgeca  staff   448B Jan  5 00:16 build-2-2ComposingWidgetsDemo-Desktop_Qt_5_15_2_clang_64bit-Release
georgeca@Georges-Mini % cd build-2-2ComposingWidgetsDemo-Desktop_Qt_5_15_2_clang_64bit-Release
georgeca@Georges-Mini build-2-2ComposingWidgetsDemo-Desktop_Qt_5_15_2_clang_64bit-Release % ls -lh
total 288
drwxr-xr-x  3 georgeca  staff    96B Jan  5 00:16 2-2ComposingWidgetsDemo.app
-rw-r--r--  1 georgeca  staff    53K Jan  5 00:16 Makefile
-rw-r--r--  1 georgeca  staff    23K Jan  5 00:16 colorpicker.o
-rw-r--r--  1 georgeca  staff   1.2K Jan  5 00:16 main.o
-rw-r--r--  1 georgeca  staff   5.7K Jan  5 00:16 moc_colorpicker.cpp
-rw-r--r--  1 georgeca  staff    10K Jan  5 00:16 moc_colorpicker.o
-rw-r--r--  1 georgeca  staff    13K Jan  5 00:16 moc_predefs.h
-rw-r--r--  1 georgeca  staff   3.4K Jan  5 00:16 moc_widget.cpp
-rw-r--r--  1 georgeca  staff   6.6K Jan  5 00:16 moc_widget.o
-rw-r--r--  1 georgeca  staff   1.5K Jan  5 00:16 ui_widget.h
-rw-r--r--  1 georgeca  staff   6.7K Jan  5 00:16 widget.o
```

4.4. Execute ‘macdeployqt’ against the application in the build directory

```
georgeca@Georges-Mini build-2-2ComposingWidgetsDemo-Desktop_Qt_5_15_2_clang_64bit-Release % /Users/georgeca/Qt/5.15.2/clang_64/bin/macdeployqt 2-2ComposingWidgetsDemo.app
```

If you wish to include a custom library in the application bundle, then copy the library into the bundle manually, after the bundle is created.

In order to place the bundle into a disk image for easy distribution, just add the **‘-dmg’ parameter** to the command.

```
georgeca@Georges-Mini build-2-2ComposingWidgetsDemo-Desktop_Qt_5_15_2_clang_64bit-Release % /Users/georgeca/Qt/5.15.2/clang_64/bin/macdeployqt 2-2ComposingWidgetsDemo.app -dmg
```

`macdeployqt` also supports these options:

- -verbose=<0–3>

> *0 = no output, 1 = error/warning (default), 2 = normal, 3 = debug*

- -no-plugins

> *Skip plugin deployment*

- -dmg

> *Create a .dmg disk image*

- -no-strip

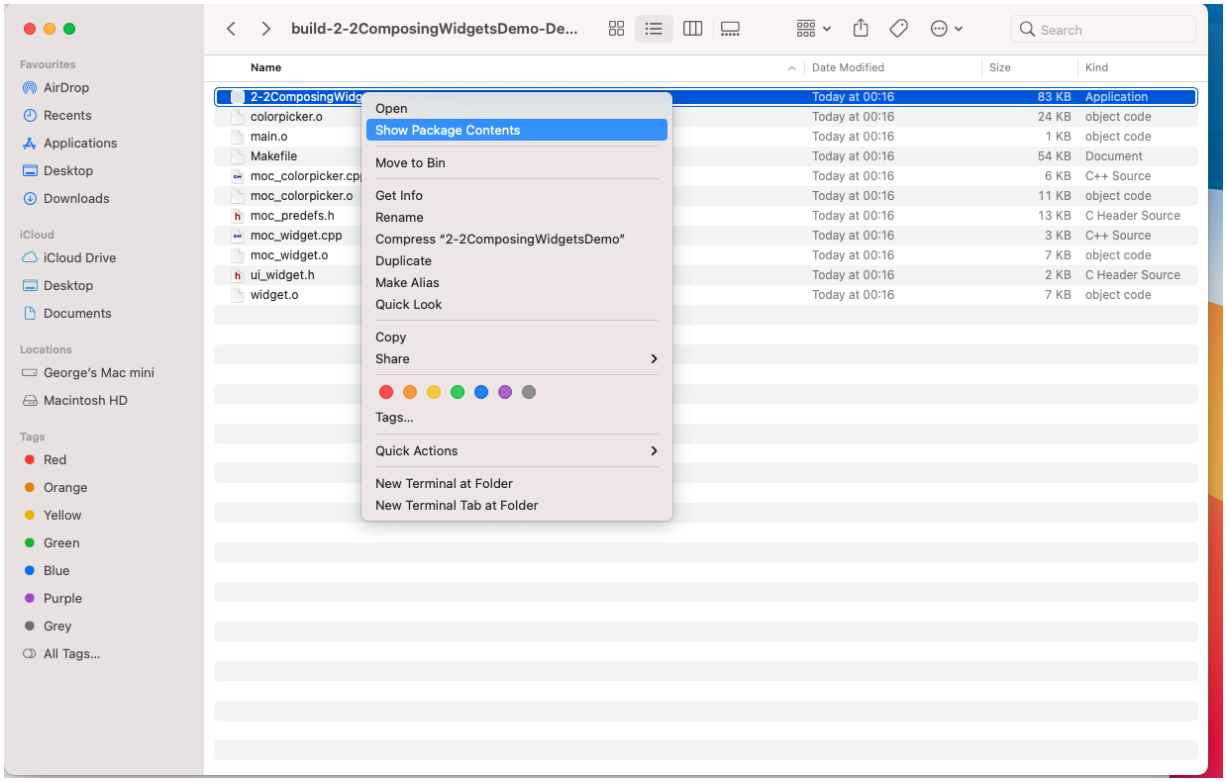
> *Don’t run ‘strip’ on the binaries*

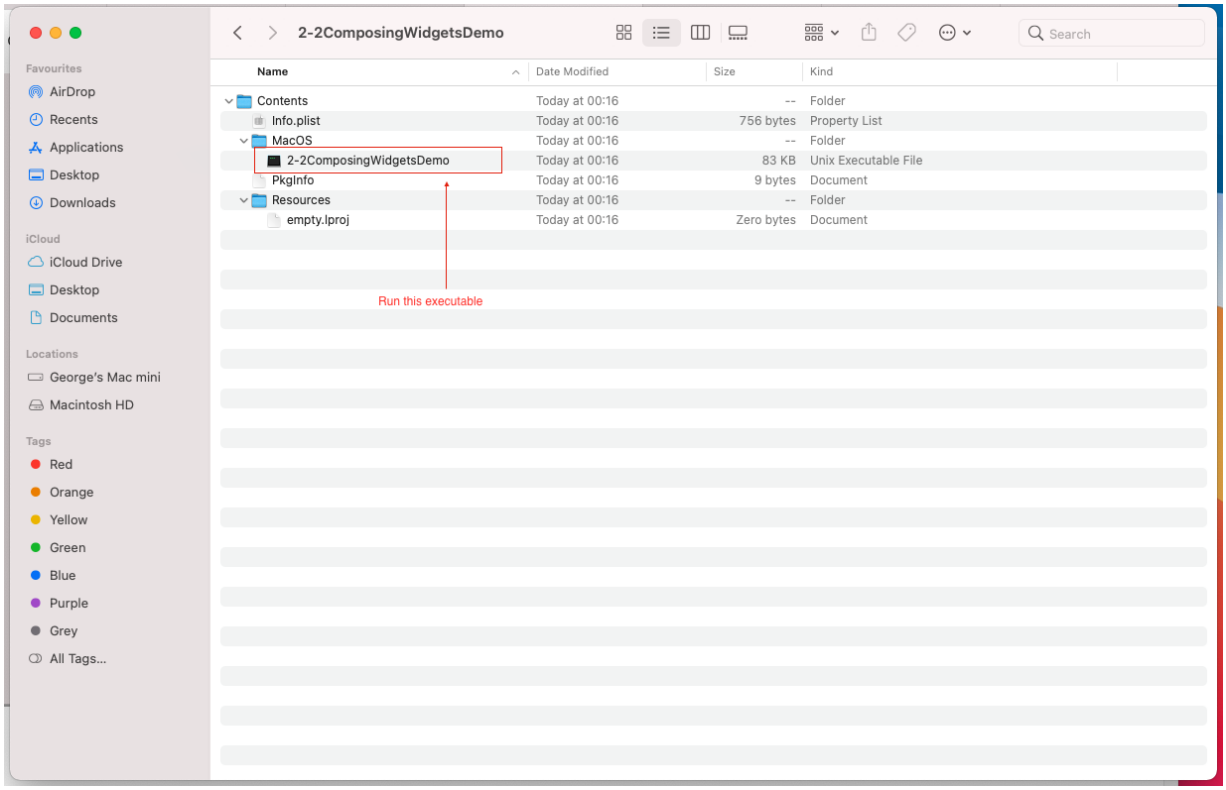
- -qmldir=

> *Deploy imports used by .qml files in the given path*

Test and run the application

Verify the contents of the bundle .app

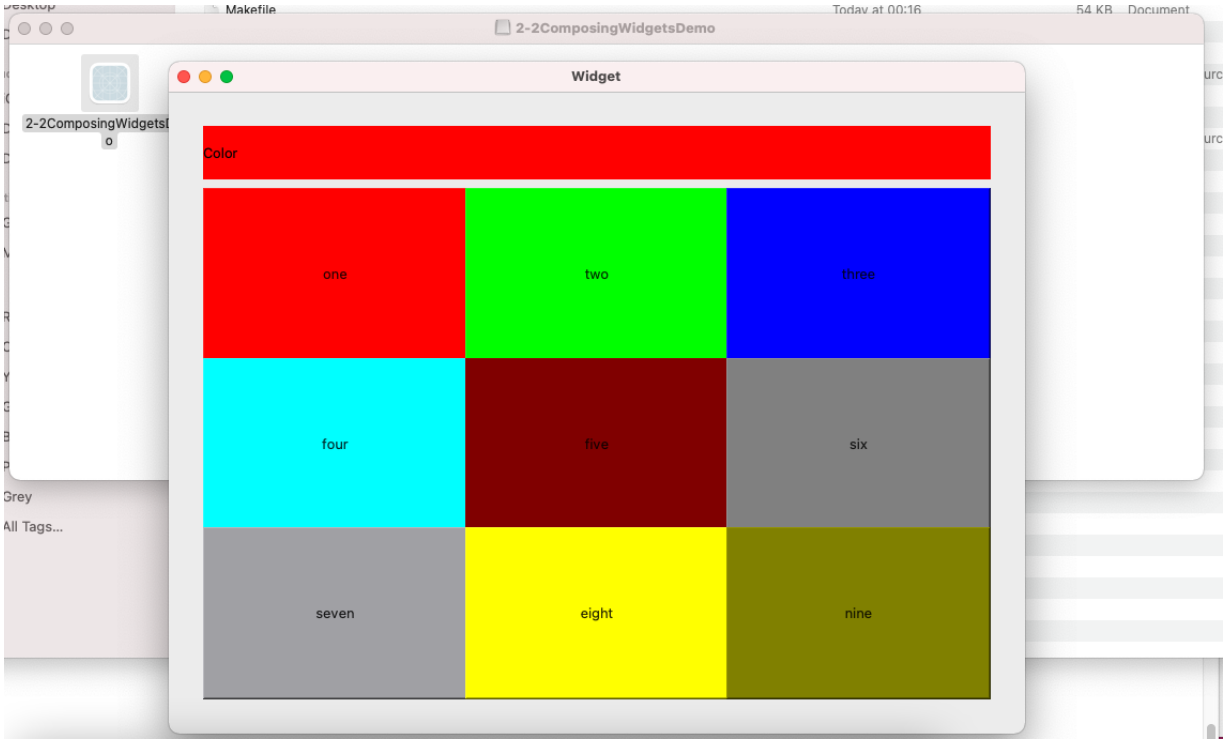




Test run the disk image .dmg

```
georgeca@Georges-Mini build-2-2ComposingWidgetsDemo-Desktop_Qt_5_15_2_clang_64bit-Release % /Users/georgeca/Qt/5.15.2/clang_64/bin/macdeployqt 2-2ComposingWidgetsDemo.app -dmg
georgeca@Georges-Mini build-2-2ComposingWidgetsDemo-Desktop_Qt_5_15_2_clang_64bit-Release % ls -lh
total 23648
drwxr-xr-x  3 georgeca  staff    96B Jan  5 00:16 2-2ComposingWidgetsDemo.app
-rw-r--r--@ 1 georgeca  staff    11M Jan  5 01:07 2-2ComposingWidgetsDemo.dmg
-rw-r--r--  1 georgeca  staff    53K Jan  5 00:16 Makefile
-rw-r--r--  1 georgeca  staff    23K Jan  5 00:16 colorpicker.o
-rw-r--r--  1 georgeca  staff   1.2K Jan  5 00:16 main.o
-rw-r--r--  1 georgeca  staff   5.7K Jan  5 00:16 moc_colorpicker.cpp
-rw-r--r--  1 georgeca  staff    10K Jan  5 00:16 moc_colorpicker.o
-rw-r--r--  1 georgeca  staff    13K Jan  5 00:16 moc_predefs.h
-rw-r--r--  1 georgeca  staff   3.4K Jan  5 00:16 moc_widget.cpp
-rw-r--r--  1 georgeca  staff   6.6K Jan  5 00:16 moc_widget.o
-rw-r--r--  1 georgeca  staff   1.5K Jan  5 00:16 ui_widget.h
-rw-r--r--  1 georgeca  staff   6.7K Jan  5 00:16 widget.o
```

mount the image to access the inside application



run the application in the .dmg

Alternative solutions

You can also deploy Qt cross-platform applications to macOS using the **Qt Installer Framework.**

> *The **steps** are similar to the ones I already described in the old articles concerning the deployment on **Windows** or **Linux Operating Systems** using the Qt Installer Framework. The sole difference is that, in this case of deploying apps for the macOS,*

> *you will be using* ***macdeployqt*** *instead of* ***windeployqt*** *or* ***linuxdeployqt****.*

How to Deploy Your Qt Cross-Platform Applications to Windows Operating…

This tutorial explains how to deploy cross-platform projects made with Qt on Windows using the Qt…

george-calin.medium.com

How to Deploy Your Qt Cross-Platform Applications to Linux Operating System…

This tutorial explains how we can deploy the projects you worked on with the Qt framework (Qt…

medium.com

Read more …

Visit the Qt documentation for the *complete reference about the deployment* of Qt cross-platform applications to *macOS*.

Qt for macOS — Deployment

This document describes how to create a macOS bundle and make sure that the application finds th…

doc.qt.io

Qt Cross Platform Mac Development Development Mac

Learn more.

Medium is an open platform where 170 million readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. Learn more

Make Medium yours.

Follow the writers, publications, and topics that matter to you, and you'll see them on your homepage and in your inbox. Explore

Write a story on Medium.

If you have a story to tell, knowledge to share, or a perspective to offer — welcome home. It's easy and free to post your thinking on any topic. Start a blog

About Write Help Legal