



ap04– QT Quick Start

Version 1.1 (2023.01.01)

目 录

1 Overview	4
1.1 Qt brief introduction	4
1.2 Real time location system function.....	4
1.3 Qt version.....	4
2 Qt installation and deployment.....	5
2.1 QT install online	5
2.2 QT install offline.....	8
2.3 Open the project	8
2.4 Path configuration	9
3 Publish executable file	10
3.1 DLL file missing issue.....	10
3.2 Solutions.....	10
3.3 Publish executable file by Windeployqt tools.....	10
4 Qt Installer Framework	15
4.1 Qt version.....	15
4.2 Overview.....	15
4.3 Configuration.....	15
5 Document Management Information Sheet	22

DISCLAIMER

YCHIOT has the right to update the product description without informing the customer. Changes in functions and specifications will be published in product errata or new versions of documents as much as possible. It is recommended that customers login to YCHIOT official website www.ychiot.com to download the latest product description documents.

LIFE SUPPORT POLICY

YCHIOT products are not authorized to be used in high safety areas (such as places where there is danger to life), because serious personal injury or death may be caused if the products are operated incorrectly. If a customer uses or sells YCHIOT products to high security areas, the customer needs to bear all the responsibilities; If this product is used in the field of high security, the customer needs to agree that YCHIOT and its agents are completely irresponsible.



Note! Electrostatic sensitive equipment. When using the product, take precautions to prevent permanent damage.

REGULATORY CERTIFICATION

All users who use this module for product development must obtain the approval of the local radio supervision and management department before marketing or selling the product, and the customer must assume all responsibilities for obtaining the approval from the relevant authorities.

1 Overview

1.1 Qt brief introduction

This section describes the use of PC PCs. This host computer software uses QT 5. 11. 3 developed by MinGM, written in C++. Qt (pronounced "cute") is cross-platform software for creating graphical user interfaces as well as cross-platform applications that run on various software and hardware platforms such as Linux, Windows, macOS, Android or embedded systems with little or no change in the underlying codebase while still being a native application with native capabilities and speed.

1.2 Real time location system function

The main functions realized by this upper computer are:

1. Establish a connection with the virtual serial port of the UWB module, [Virtual COM Port / Ethernet](#).
2. Read the TOF / Toda report message from the UWB module.
3. A list of base stations, in which you can set the actual placement of the base station.
4. a list of labels, which can show the distance of the tag from the base station, as well as the location of the tag (XYZ coordinates).
5. Map display, support custom import of a PNG format map, can achieve zoom and coordinate fine-tuning.
6. Other parameter settings.

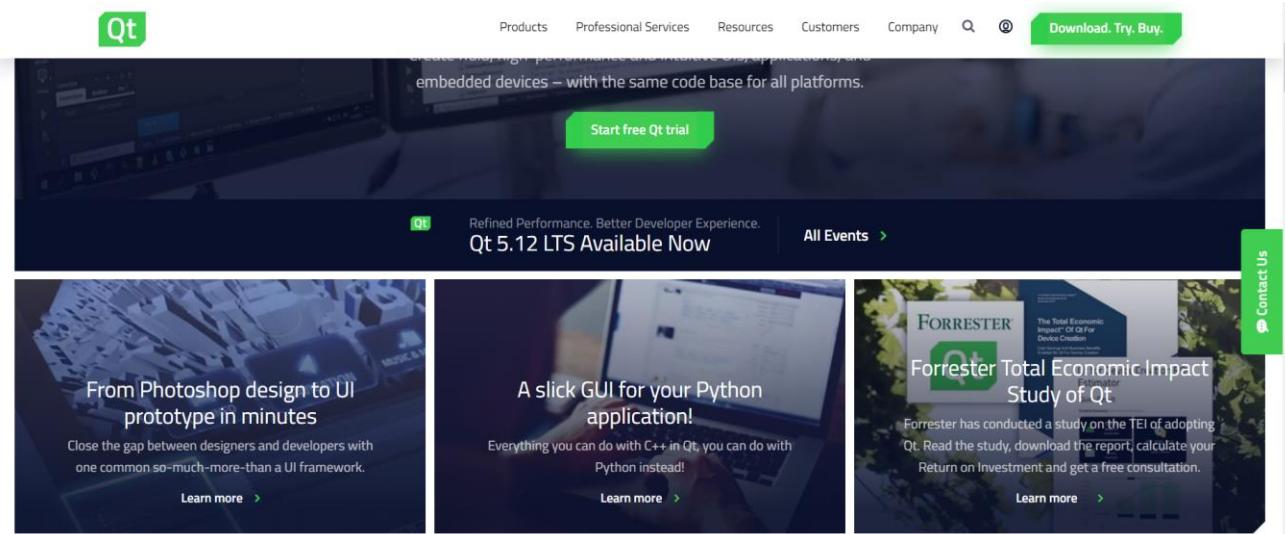
1.3 Qt version

- Qt Creator 4.8.0
- Based on Qt 5.11.3 (MinGW 5.3.0 32bit)

2 Qt installation and deployment

2.1 QT install online

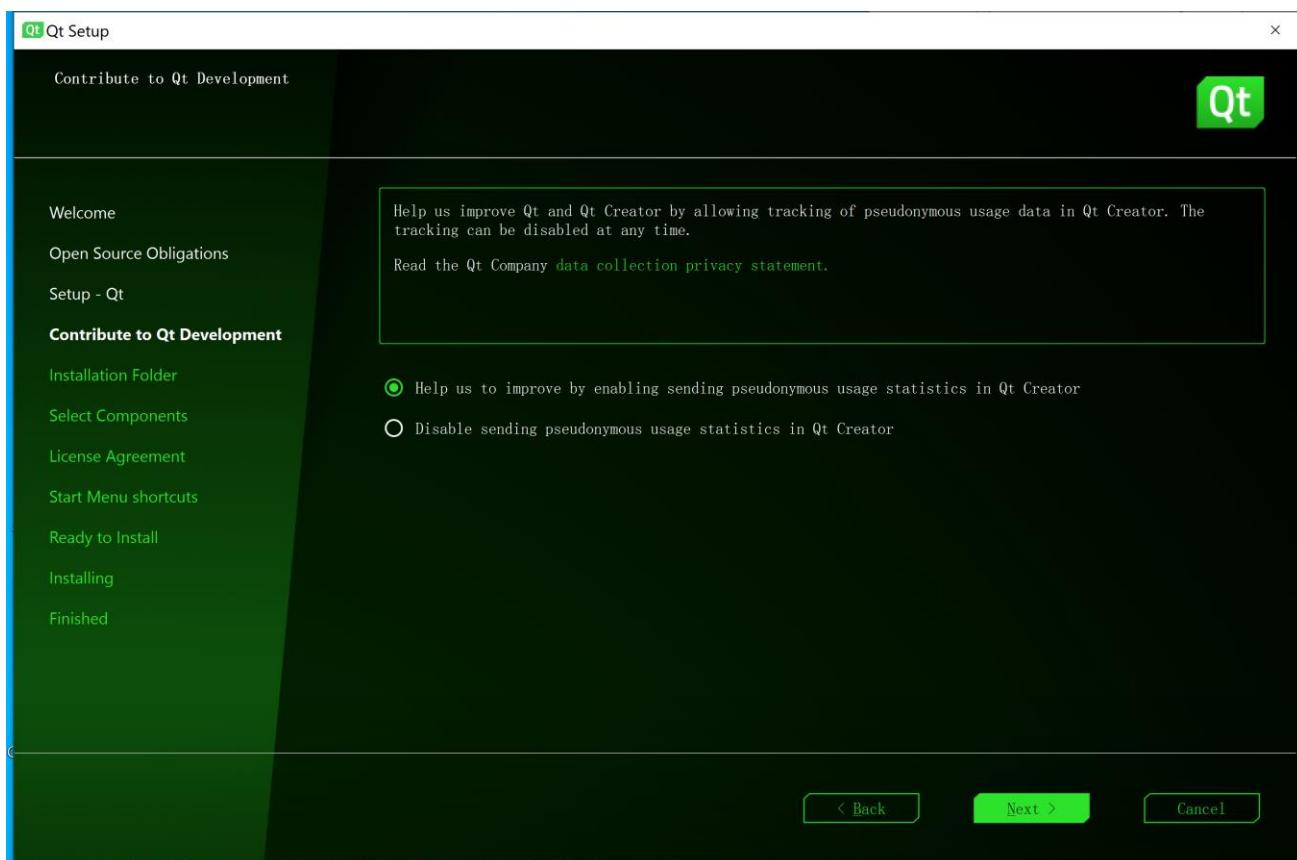
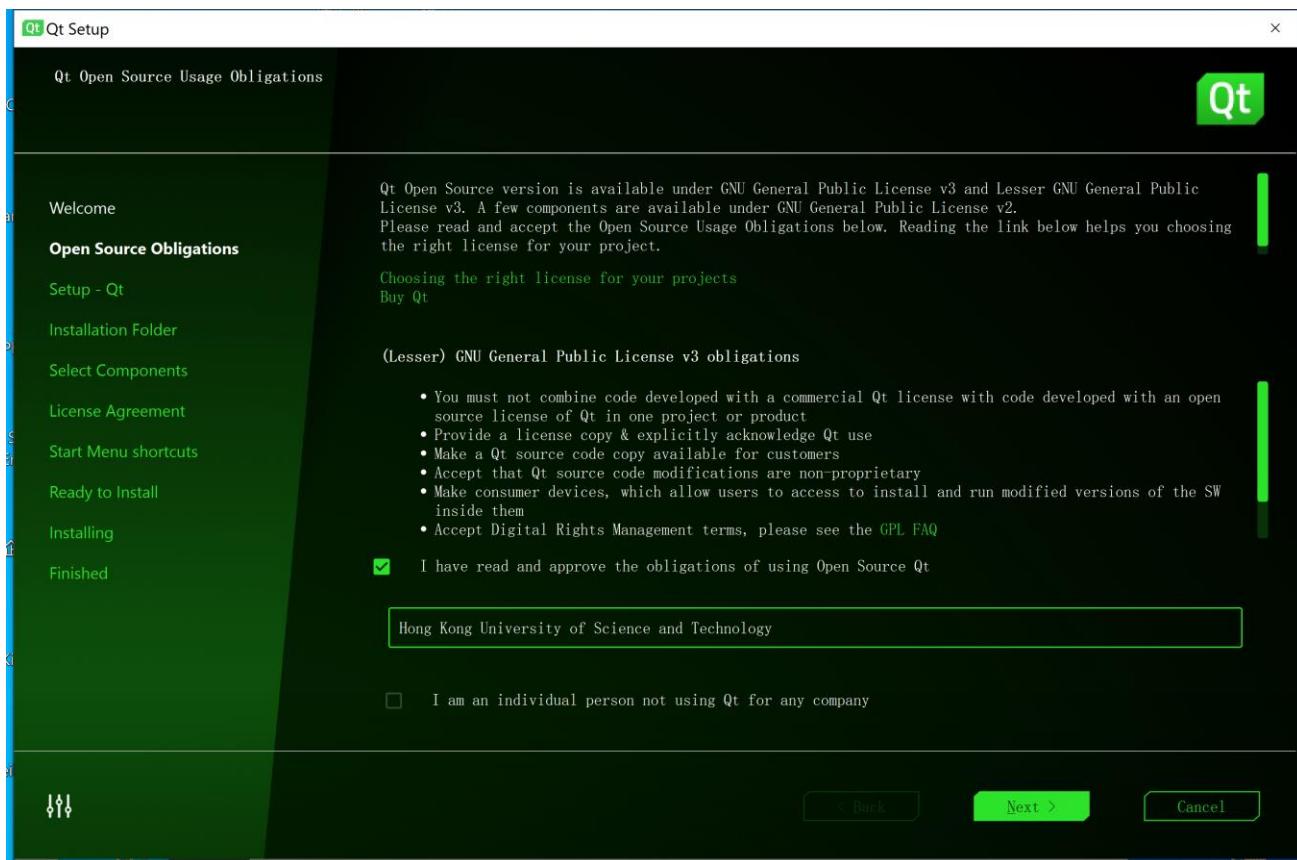
Visit the <https://www.qt.io/> to download the online installation package, this section uses qt-unified-windows-x86-3.0.6-online as an example.

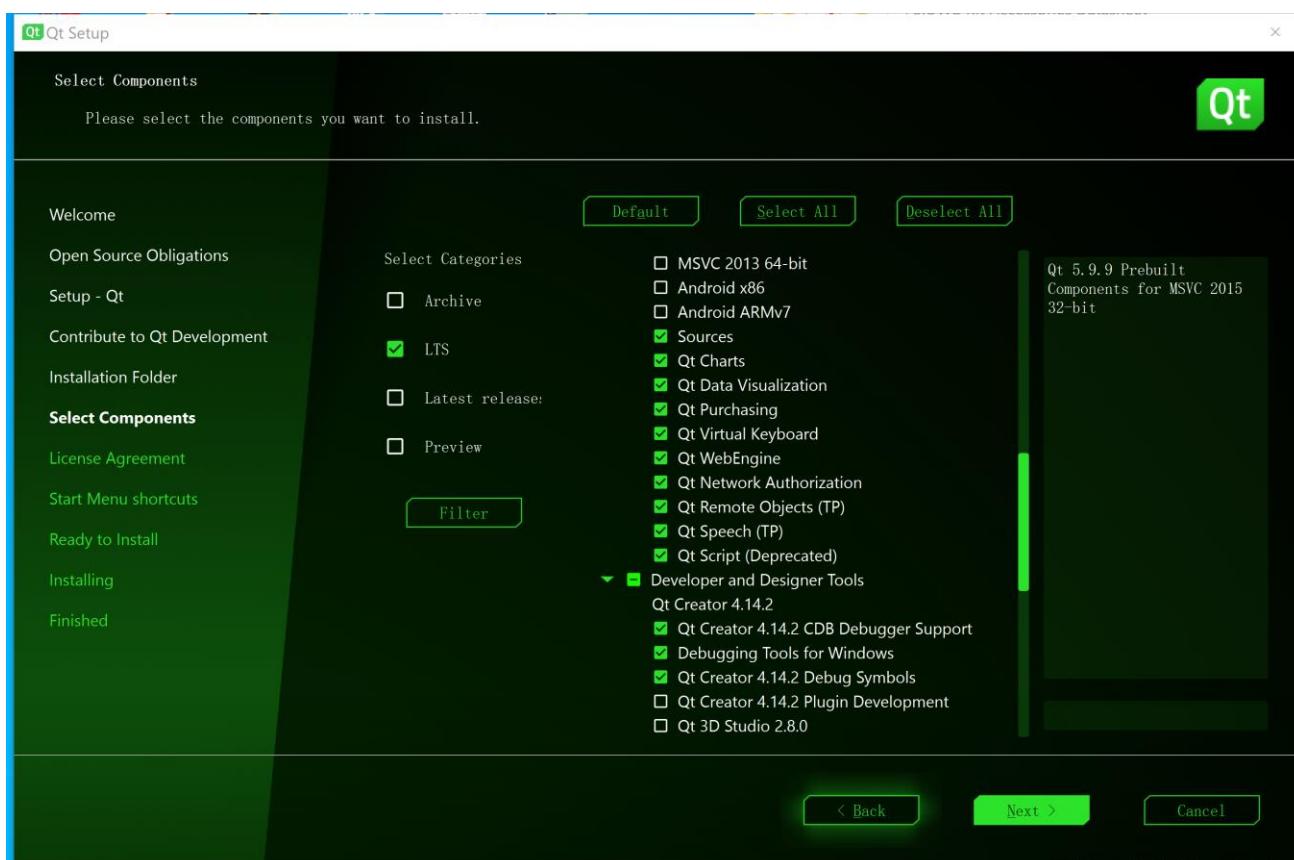
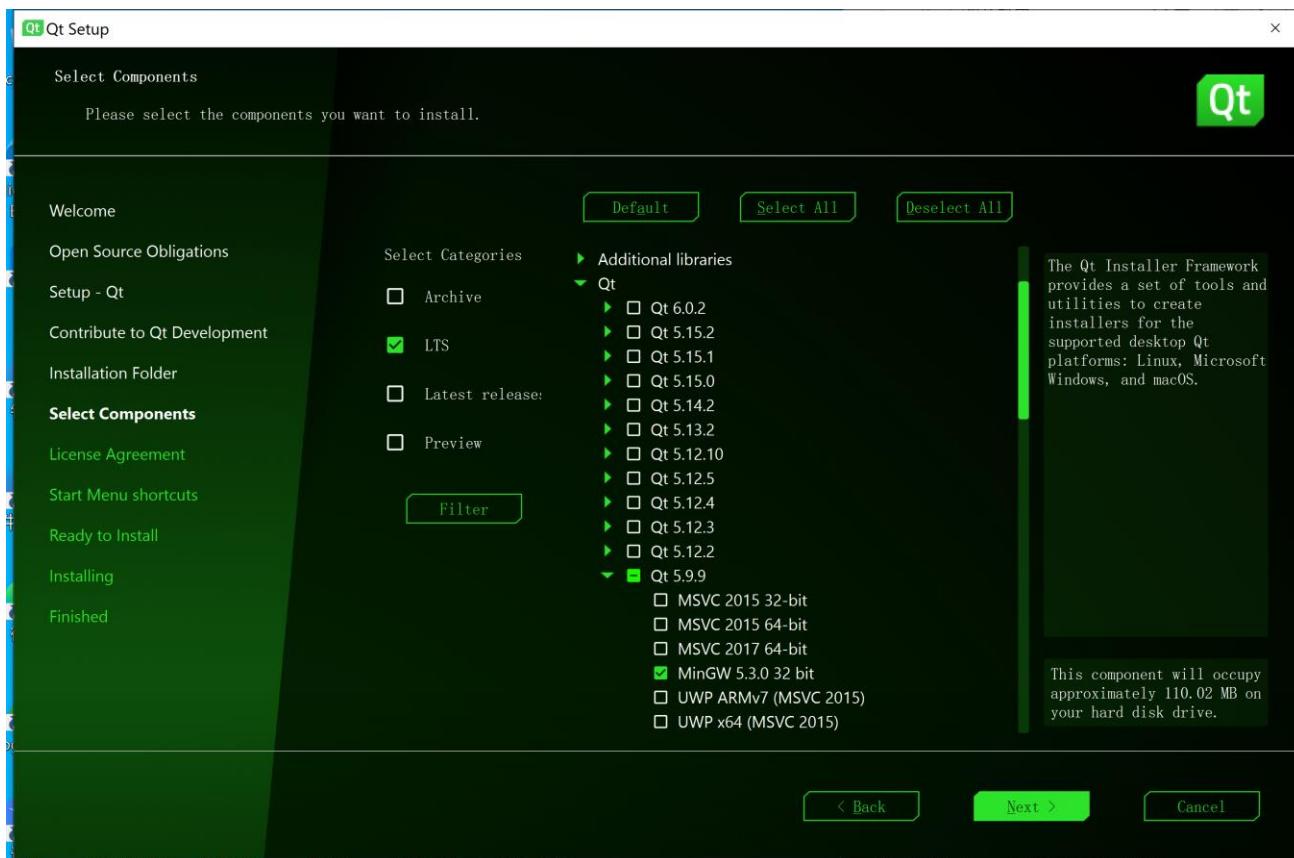


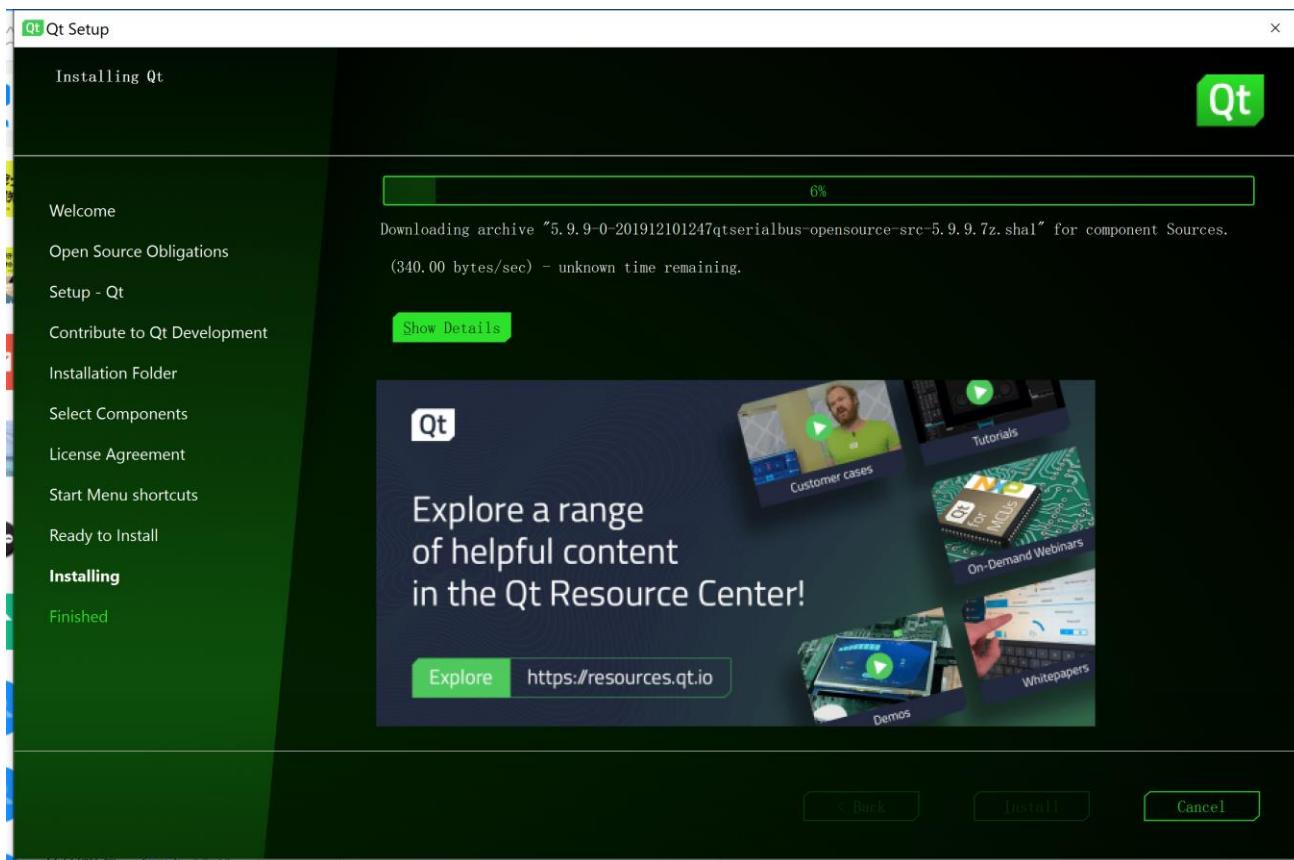
2.1.1 steps

Open the file, qt-unified-windows-x86-3.0.6-online, for online installation. Take 5 Version 11.3 is an example. The components that need to be selected are:

- Qt Creator 4.8.0-rc1 (optional)
- Qt 5.11.3 (**REQUIRED**)
- Qt Creator 4.8.0 CDB Debugger Support (optional)
- MinGW 5.3.0 32bit (**REQUIRED**)
- Qt Installer Framework 3.0 (optional)







2.2 QT install offline

Taking the Qt 5.9.9 version as an example, we open the link:

<https://download.qt.io/archive/qt/>

Name	Last modified	Size	Metadata
Parent Directory		-	
5.9.9/	16-Dec-2019 15:07	-	

Select [qt-opensource-windows-x86-5.9.9.exe](#) to download and install, the specific installation steps are the same as section 2.1.

2.3 Open the project

Unzip the package and open the TREKdisplay (Qt Project file).

myhelper	2018-08-07 15:04	C++ Header file	3 KB
rc	2014-08-15 15:14	QRC 文件	1 KB
RTLSDisplayApplication	2018-08-07 15:50	C++ Source file	5 KB
RTLSDisplayApplication	2016-01-21 21:46	C++ Header file	3 KB
TREKanc config	2015-02-04 20:21	XML 文档	1 KB
TREKdisplay	2018-08-11 12:07	Qt Project file	2 KB
TREKdisplay.pro.user	2019-01-06 11:48	每用户项目选项文...	24 KB
TREKdisplay.pro.user.4.8-pre1	2019-01-05 10:03	8-PRE1 文件	24 KB
TREKdisplay.pro.user.38a701d	2017-10-29 21:01	38A701D 文件	24 KB

Figure 2.3 Opening the project

2.4 Path configuration

Select a directory as the folder for the compilation output, note that there are no characters other than English in the path.

**Figure 2.4.1 Path configuration**

Click Run and wait a while for the compilation to complete. After that, the project code is complied and running.

**Figure 2.4.2 Running the project**

3 Publish executable file

The dynamic link library method used by the Qt official development environment requires copying many DLL files when publishing the generated exe program, and if the dll file is missing, the exe will not run normally on other computers. Therefore, the official development environment of Qt comes with a release tool: `windeployqt.exe`.

3.1 DLL file missing issue

The QT source code is run under Release or Debug, and the generated .exe file cannot be run, and multiple error dialog boxes pop up, as shown in the following figure.

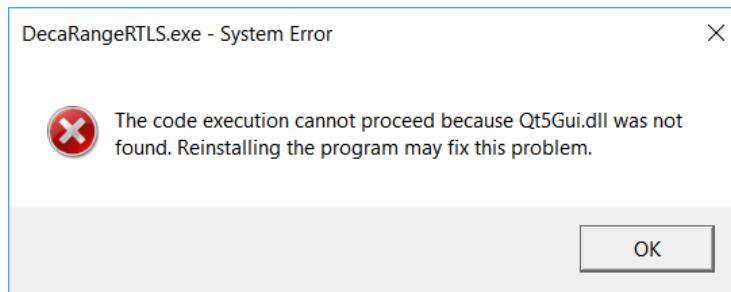


Figure 3.1 .dll file missing issue

3.2 Solutions

Under the corresponding folder, you need to add the corresponding .dll file.

3.3 Publish executable file by Windeployqt tools

1. Release compilation. Under Projects -> Build Settings, select Release (if you think this is the last version, otherwise choose Debug). Then click Run in the lower left corner.

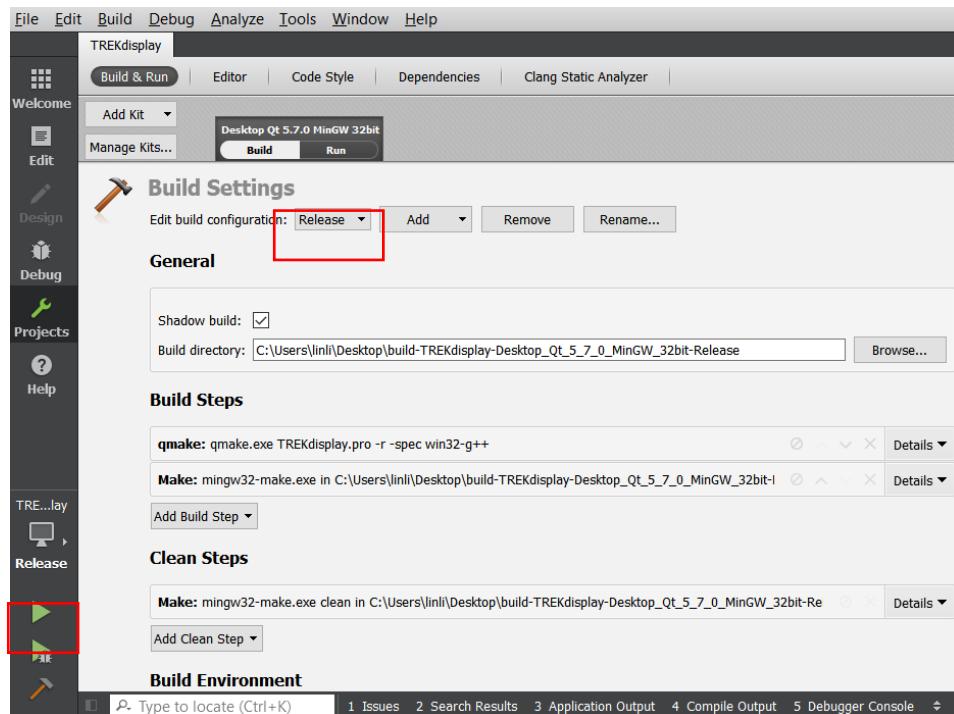


Figure 3.1 Compile the project

2. Go to the compiled Release folder and find the DecaRangeRTLS .exe

Name	Date modified	Type	Size
debug	4/30/2017 8:07 PM	File folder	
release	4/30/2017 9:23 PM	File folder	
.qmake.cache	4/30/2017 8:06 PM	CACHE File	0 KB
.qmake.stash	4/30/2017 8:06 PM	STASH File	1 KB
Makefile	4/30/2017 8:07 PM	File	24 KB
Makefile.Debug	4/30/2017 8:07 PM	DEBUG File	562 KB
Makefile.Release	4/30/2017 8:07 PM	RELEASE File	563 KB
object_script.DecaRangeRTLS.Debug	4/30/2017 8:07 PM	DEBUG File	1 KB
object_script.DecaRangeRTLS.Release	4/30/2017 8:07 PM	RELEASE File	1 KB
TREKview_config	4/30/2017 8:08 PM	XML Document	0 KB
ui_connectionwidget	4/30/2017 8:07 PM	C++ Header file	3 KB
ui_GraphicsWidget	4/30/2017 8:07 PM	C++ Header file	5 KB
ui_mainwindow	4/30/2017 8:07 PM	C++ Header file	5 KB
ui_ViewSettingsWidget	4/30/2017 8:07 PM	C++ Header file	20 KB

Figure 3.2 The release folder

3. Copy this .exe to a new separate folder for publishing, such as the :D:\UWB folder.

名称	修改日期
connectionwidget.o	2019-01-06 11:47
DecaRangeRTLS	2019-01-06 11:48
GraphicsView.o	2019-01-06 11:47
GraphicsWidget.o	2019-01-06 11:47
iconhelper.o	2019-01-06 11:47
main.o	2019-01-06 11:47
mainwindow.o	2019-01-06 11:47
MinimapView.o	2019-01-06 11:47
moc_AbstractTool	2019-01-06 11:48
moc_AbstractTool.o	2019-01-06 11:48
moc_connectionwidget	2019-01-06 11:48
moc_connectionwidget.o	2019-01-06 11:48
moc_GraphicsView	2019-01-06 11:48
moc_GraphicsView.o	2019-01-06 11:48
moc_GraphicsWidget	2019-01-06 11:48
moc_GraphicsWidget.o	2019-01-06 11:48

Figure 3.3.3 DecaRangeRTLS .exe awaiting release

4. Take the official Qt 5.11.3+MinGW development environment as an example to find Qt 5.11.3 for Desktop (MinGW 5.3.0 32 bit).

**图 3.3.4 Qt 5.11.3 for Desktop**

5. You can open the Qt command line, from where you can execute the `windeployqt` tool.
 6. Open the Qt command line from the Start menu and enter the command: `cd /d D:\UWB`

Setting up environment for Qt usage...

```
D:\Qt\5.11.3\mingw53_32>cd /d D:\UWB
```

Figure 3.3.5 Qt command

7. Use the `windeployqt` tool command: [windeployqt DecaRangeRTLS.exe](#)

Setting up environment for Qt usage...

```
D:\Qt\5.11.3\mingw53_32>cd /d D:\UWB  
  
D:\UWB>windeployqt DecaRangeRTLS.exe  
D:\UWB\DecaRangeRTLS.exe 32 bit, release executable  
Adding Qt5Svg for qsvgicon.dll  
Direct dependencies: Qt5Core Qt5Gui Qt5SerialPort Qt5Widgets Qt5Xml  
All dependencies : Qt5Core Qt5Gui Qt5SerialPort Qt5Widgets Qt5Xml  
To be deployed : Qt5Core Qt5Gui Qt5SerialPort Qt5Svg Qt5Widgets Qt5Xml  
Qt5Core.dll is up to date.  
Qt5Gui.dll is up to date.  
Qt5SerialPort.dll is up to date.  
Qt5Svg.dll is up to date.  
Qt5Widgets.dll is up to date.  
Qt5Xml.dll is up to date.  
libGLESV2.dll is up to date.  
libEGL.dll is up to date.  
D3Dcompiler_47.dll is up to date.  
opengl32sw.dll is up to date.  
libgcc_s_dw2-1.dll is up to date.  
libstdc++-6.dll is up to date.  
libwinpthread-1.dll is up to date.  
Patching Qt5Core.dll...  
qsvgicon.dll is up to date.  
qgif.dll is up to date.  
qicns.dll is up to date.  
qico.dll is up to date.  
qjpeg.dll is up to date.  
qsvg.dll is up to date.  
qtga.dll is up to date.  
qtiff.dll is up to date.  
qwbmp.dll is up to date.  
qwebp.dll is up to date.  
qwindows.dll is up to date.  
qwindowsvistastyle.dll is up to date.  
Creating qt_arqm...  
Creating qt_bgqm...  
Creating qt_caqm...  
Creating qt_csqm...  
Creating qt_daqm...  
Creating qt_deqm...  
Creating qt_enqm...  
Creating qt_esqm...  
Creating qt_fiqm...
```

```
Creating qt_fr.qm...
Creating qt_gd.qm...
Creating qt_he.qm...
Creating qt_hu.qm...
Creating qt_it.qm...
Creating qt_ja.qm...
Creating qt_ko.qm...
Creating qt_lv.qm...
Creating qt_pl.qm...
Creating qt_ru.qm...
Creating qt_sk.qm...
Creating qt_uk.qm...
```

D:\UWB>

8. Copy the lapack_win32_MT.dll and blas_win32_MT to get the complete .dll project package

iconengines	2019-01-06 13:58	文件夹
imageformats	2019-01-06 13:58	文件夹
platforms	2019-01-06 13:58	文件夹
styles	2019-01-06 13:58	文件夹
translations	2019-01-06 13:58	文件夹
blas_win32_MT.dll	2015-07-09 15:33	应用程序扩展
D3Dcompiler_47.dll	2014-03-11 18:54	应用程序扩展
DecaRangeRTLS	2019-01-06 11:48	应用程序
lapack_win32_MT.dll	2015-07-09 15:33	应用程序扩展
libEGL.dll	2018-11-26 18:04	应用程序扩展
libgcc_s_dw2-1.dll	2015-12-29 6:25	应用程序扩展
libGLESV2.dll	2018-11-26 18:04	应用程序扩展
libstdc++-6.dll	2015-12-29 6:25	应用程序扩展
libwinpthread-1.dll	2015-12-29 6:25	应用程序扩展
opengl32sw.dll	2016-06-14 21:08	应用程序扩展
Qt5Core.dll	2019-01-06 13:58	应用程序扩展
Qt5Gui.dll	2018-11-26 18:08	应用程序扩展
Qt5SerialPort.dll	2018-11-26 18:25	应用程序扩展
Qt5Svg.dll	2018-11-26 19:03	应用程序扩展
Qt5Widgets.dll	2018-11-26 18:10	应用程序扩展
Qt5Xml.dll	2018-11-26 18:05	应用程序扩展
TREKanc_config	2018-12-27 20:17	XML 文档
TREKtag_config	2018-12-27 20:17	XML 文档
TREKview_config	2019-01-06 13:59	XML 文档

4 Qt Installer Framework

4.1 Qt version

- Qt Creator 4.8.0
- Based on Qt 5.11.3
- Qt Installer Framework 3.0

4.2 Overview

Applications developed using the Qt library generally have two ways to publish:

1. Statically compiled release. This way allows the program to compile all the Qt core libraries into an executable when compiling. Its advantage is that it is simple and single, all the dependent libraries are concentrated together, and its disadvantages are also obvious, the executable program volume is large, and the Qt core library alone adds up to more than ten megabytes.
2. Make an installation package release. The principle of this method is also simple, that is, the executable program and its dependent library files are packaged and compressed together to make an installation package for distribution.

There are many tools for making installation packages, and the Qt installer framework can be completed into an installation package using Qt's official installation package production framework.

4.3 Configuration

4.3.1 Install QtInstallerFramework-win-x86

Download and follow the step-by-step process to complete the installation.

4.3.2 Generate an executable file

Refer to Section 3.3.1 to generate the correct executable .exe file.

4.3.3 Create a directory

1. Under C:\Qt\QtIFW2.0.3\bin, create a sc folder, and create two more folders under the sc folder, config and packages;

2. The config xml file in the config folder is copied from C:\Qt\QtIFW2.0.3\examples\tutorial\config;
3. Copy all files in the folder of C:\Qt\QtIFW2.0.3\examples\tutorial\packages to C:\Qt\QtIFW2.0.3\bin\sc\packages;
4. The result is shown in the following figure:

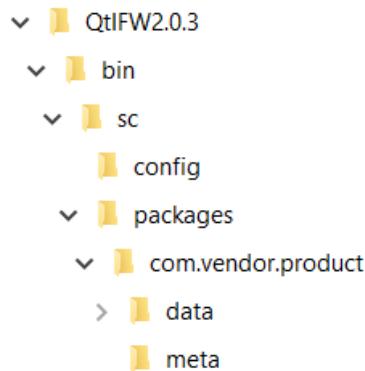


Figure 4.3.3 Directory structure

4.3.4 config.xml

Create a config .xml file under the config folder, which is used to customize the UI and behavior of the installation package, and its basic format is as follows:

```

<?xml version="1.0"?>
<Installer>
  <Name>Some Application</Name>
  <Version>1.0.0</Version>
  <Title>Some Application Setup</Title>
  <Publisher>Your Company</Publisher>
  <ProductUrl>http://www.your-fantastic-company.com</ProductUrl>
  <InstallerWindowIcon>installericon</InstallerWindowIcon>
  <InstallerApplicationIcon>installericon</InstallerApplicationIcon>
  <Logo>logo.png</Logo>
  <Watermark>watermark.png</Watermark>
  <RunProgram></RunProgram>
  <RunProgramArguments></RunProgramArguments>
  <RunProgramDescription></RunProgramDescription>
  <StartMenuDir>Some Application Entry Dir</StartMenuDir>
  <UninstallerName>SDKMaintenanceTool</UninstallerName>
  <AllowNonAsciiCharacters>true</AllowNonAsciiCharacters>
  <Background>background.png</Background>

  <TargetDir>@homeDir@/testinstall</TargetDir>
  <AdminTargetDir>@rootDir@/testinstall</AdminTargetDir>
  <RemoteRepositories>
  
```

```

<Repository>
<Url>http://www.your-repo-location/packages/</Url>
</Repository>
</RemoteRepositories>
</Installer>

```

Among them, the name and version child elements are required, the other child elements are optional, and the order can be arbitrary. The meaning of these configuration items can be found on the link on the official website.

<http://doc.qt.io/qtinstallerframework/ifw-globalconfig.html>

4.3.5 packages folder

Notice that under the packages folder, each component needs to create two subfolders, meta and data. The meta directory is used to store some configuration files to specify the installation and deployment process. In the meta folder, you need at least one package .xml and all resource files referenced in the file, such as script files, interface resources, translation files, etc. The package .xml file describes the basic information of a component in the following format:

```

<?xml version="1.0"?>
<Package>
    <DisplayName>QtGui</DisplayName>
    <Description>Qt gui libraries</Description>
    <Description xml:Lang="de_de">Qt GUI Bibliotheken</Description>
    <Version>1.2.3</Version>
    <ReleaseDate>2009-04-23</ReleaseDate>
    <Name>com.vendor.root.component2</Name>
    <Dependencies>com.vendor.root.component1</Dependencies>
    <Virtual>false</Virtual>
    <Licenses>
        <License name="License Agreement" file="license.txt" />
    </Licenses>
    <Script>installscript.qs</Script>
    <UserInterfaces>
        <UserInterface>specialpage.ui</UserInterface>
        <UserInterface>errorpage.ui</UserInterface>
    </UserInterfaces>
    <Translations>
        <Translation>sv_se.qm</Translation>
        <Translation>de_de.qm</Translation>
    </Translations>
    <DownloadableArchives>component2.7z, component2a.7z</DownloadableArchives>
    <AutoDependOn>com.vendor.root.component3</AutoDependOn>

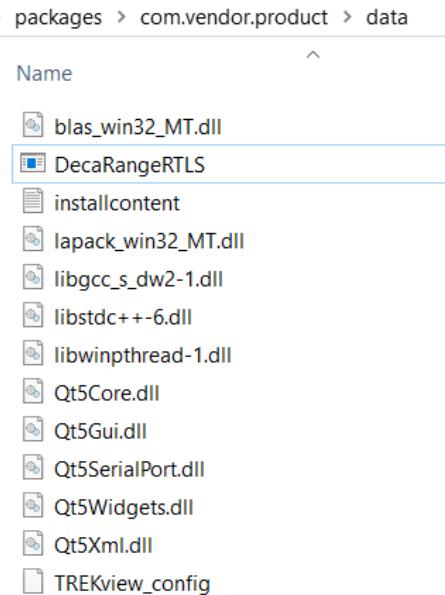
```

```
<SortingPriority>123</SortingPriority>
<UpdateText>This changed compared to the last release</UpdateText>
<Default>false</Default>
<ForcedInstallation>false</ForcedInstallation>
<Essential>false</Essential>
<Replaces>com.vendor.root.component2old</Replaces>
</Package>
```

The license section specifies the terms of the agreement, and the file attribute specifies the file in which the terms of the agreement are located. The UserInterfaces and Translations sections specify interface resource files and translation files, respectively. The Script section specifies the script file, where the installation process can be customized in more detail. The names and values of other elements in the sample are relatively easy to understand, and it is not difficult to modify them. All options can be viewed on the official website.

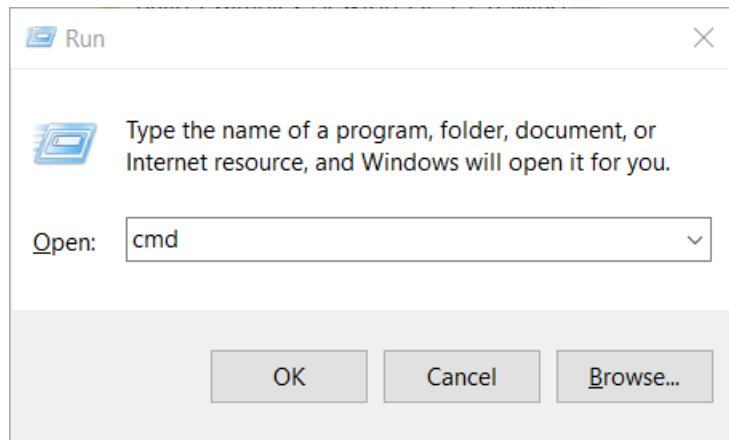
<http://doc.qt.io/qtinstallerframework/ifw-component-description.html#package-information-file-syntax>

Copy all files under the data subfolder, as shown in the following figure.



4.3.6 run the command

Run it with Win+R and enter "cmd" as shown in the image below

**Figure 4.3.6 run the cmd**

4.3.7 Go to the specified folder

Enter the command, cd C:\Qt\QtIFW2.0.3\bin,

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\linli>cd C:\Qt\QtIFW2.0.3\bin

C:\Qt\QtIFW2.0.3\bin>
```

Figure 4.3.7 Go to the bin folder

4.3.8 binarycreator.exe

Enter the command:

```
binarycreator.exe -c sc\config\config.xml -p sc\packages my_installer.exe -v
```

The result is as follows:

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Qt\QtIFW2.0.3\bin>binarycreator.exe -c sc\config\config.xml -p sc\packages
my_installer.exe -v
[0] Parsed arguments, ok.
[3]
Collecting information about available packages...
[5] found subdirectory 'com.vendor.product'
[6] - it provides the package com.vendor.product - 0.1.0-1
[8] Copying component data for "com.vendor.product"
[11] Compressing data directory "build-TREKdisplay-Desktop_Qt_5_7_0_MinGW_32bit-
Release"
```

```
[1050] Compressing files found in data directory:  
("C:/Qt/QtIFW2.0.3/bin/sc/packages/com.vendor.product/data/installcontent.txt")  
[1063] Hash is stored in "C:/Users/linli/AppData/Local/Temp/binarycreator-  
19Mfi9/com.vendor.product/0.1.0-1build-TREKdisplay-Desktop_Qt_5_7_0_MinGW_32bit-  
Release.7z.sha1"  
[1068] Creating hash of archive "C:/Users/linli/AppData/Local/Temp/binarycreator-  
19Mfi9/com.vendor.product/0.1.0-1build-TREKdisplay-Desktop_Qt_5_7_0_MinGW_32bit-  
Release.7z"  
[1077] Generated sha1 hash: "d63a8b0b8cde3134bc9eacbfd93d563980e207a4"  
[1084] Hash is stored in "C:/Users/linli/AppData/Local/Temp/binarycreator-  
19Mfi9/com.vendor.product/0.1.0-1content.7z.sha1"  
[1087] Creating hash of archive "C:/Users/linli/AppData/Local/Temp/binarycreator-  
19Mfi9/com.vendor.product/0.1.0-1content.7z"  
[1092] Generated sha1 hash: "c8d32a476764d3d60e341d04e3e809dc6b6d9525"  
[1100] Copy meta data for package 'com.vendor.product' using  
'sc/packages/com.vendor.product/meta/package.xml'.  
[1105] calculate size of directory: C:/Users/linli/AppData/Local/Temp/binarycreator-  
19Mfi9/com.vendor.product/data  
[1145] Copying associated script file  
'sc/packages/com.vendor.product/meta/installscript.qs'  
[1150] done.  
  
[1152] Copying associated user interface file  
'sc/packages/com.vendor.product/meta/page.ui'  
[1159] done.  
  
[1161] Copying associated license file  
'sc/packages/com.vendor.product/meta/license.txt'  
[1168] done.  
  
[1174] Begin to copy configuration file and data.  
[1176] Copying associated configuration file  
'C:/Qt/QtIFW2.0.3/bin/sc/config/config.xml'  
[1180] done.  
  
[1185] Read dom element: <Name>Your application</Name>.  
[1186] Read dom element: <Version>1.0.0</Version>.  
[1188] Read dom element: <Title>Your application Installer</Title>.  
[1190] Read dom element: <Publisher>Your vendor</Publisher>.  
[1192] Read dom element: <StartMenuDir>Super App</StartMenuDir>.  
[1194] Read dom element: <TargetDir>@HomeDir@/InstallationDirectory</TargetDir>.  
[1197] done.  
  
[1246] Creating the binary  
[1414] Creating resource archive for "com.vendor.product"  
[1416] Appending C:/Users/linli/AppData/Local/Temp/binarycreator-  
19Mfi9/com.vendor.product/0.1.0-1build-TREKdisplay-Desktop_Qt_5_7_0_MinGW_32bit-  
Release.7z (661.19 KiB)
```

```
[1421] Appending C:/Users/linli/AppData/Local/Temp/binarycreator-19Mfi9/com.vendor.product/0.1.0-1build-TREKdisplay-Desktop_Qt_5_7_0_MinGW_32bit-Release.7z.sha1 (40.00 bytes)
[1426] Appending C:/Users/linli/AppData/Local/Temp/binarycreator-19Mfi9/com.vendor.product/0.1.0-1content.7z (217.00 bytes)
[1432] Appending C:/Users/linli/AppData/Local/Temp/binarycreator-19Mfi9/com.vendor.product/0.1.0-1content.7z.sha1 (40.00 bytes)
[1475] Cleaning up...
```

C:\Qt\QtFW2.0.3\bin>

At this point, the installation package file has been generated.

4.3.9 Test

Under the C:\Qt\QtFW2.0.3\bin folder, a [my_installer.exe](#) file is generated, this is the installation package.

Name	Date modified	Type
sc	4/30/2017 8:23 PM	File folder
archivegen	5/2/2016 9:16 PM	Application
binarycreator	5/1/2017 10:36 AM	Application
cd	5/1/2017 10:36 AM	File
devtool	5/2/2016 9:17 PM	Application
installerbase	5/2/2016 9:16 PM	Application
Microsoft	5/1/2017 10:36 AM	File
my_installer	5/1/2017 10:31 AM	Application
repogen	5/2/2016 9:17 PM	Application

Figure 4.3.9 installation package

5 Document Management Information Sheet

Subject		YCHIOT QT quick start
Version	V1.1	
Reference documents	https://blog.csdn.net/sinat_36264666/article/details/73305712 https://www.qt.io/	
Creation time	2018/06/01	
Founder	Lynn	
Latest release date	2023/01/01	

Modifier	Date	Document change history
Lynn	2019-01-06	<u>V1.0</u> [Cancelled] aps001-QT generates exe file library missing issue [Cancelled] aps002-QT installation package creation framework
Lynn	2023-01-01	<u>V1.1</u> Added Section 2.2 Offline Installation Method Modified format to YCHIOT 2023 document vision