# 1 Windows环境搭建

FFMPEG官网：<http://ffmpeg.org/>

## 1.1 下载4.2.1版本源码

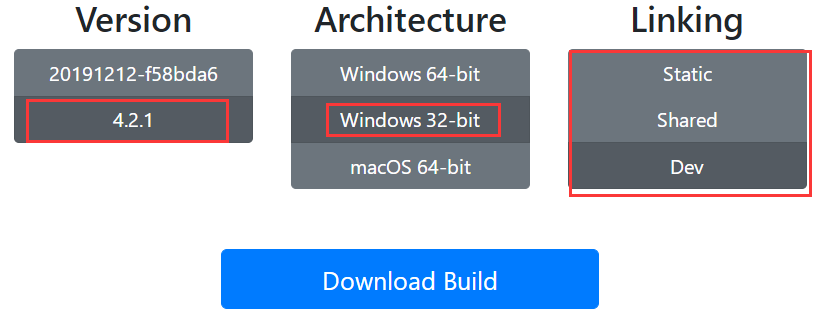
源码：[https://ffmpeg.org/releases/ffmpeg-4.2.1.tar.bz2](https://ffmpeg.org/releases/ffmpeg-4.0.2.tar.bz2)

## 1.2 下载4.2.1编译好的文件

**下载已经编译好的FFMPEG**

网址：<https://ffmpeg.zeranoe.com/builds/>

因为后续项目开发是基于4.2.1的头文件，所以强烈建议大家使用我提供的32位下载地址进行。



因为后续项目开发是基于4.2.1的头文件，所以强烈建议大家使用下述的链接进行下载。

**32位下载地址**：

**Shared**：包含FFMPEG的dll库文件

[https://ffmpeg.zeranoe.com/builds/win32/shared/ffmpeg-4.2.1-win32-shared.zip](https://ffmpeg.zeranoe.com/builds/win32/shared/ffmpeg-4.0.2-win32-shared.zip)

**Static**：包含了FFMPEG的官方文档

[https://ffmpeg.zeranoe.com/builds/win32/static/ffmpeg-4.2.1-win32-static.zip](https://ffmpeg.zeranoe.com/builds/win32/static/ffmpeg-4.0.2-win32-static.zip)

**Dev**：包含FFMPEG的lib文件/头文件，以及example范例。

[https://ffmpeg.zeranoe.com/builds/win32/dev/ffmpeg-4.2.1-win32-dev.zip](https://ffmpeg.zeranoe.com/builds/win32/dev/ffmpeg-4.0.2-win32-dev.zip)

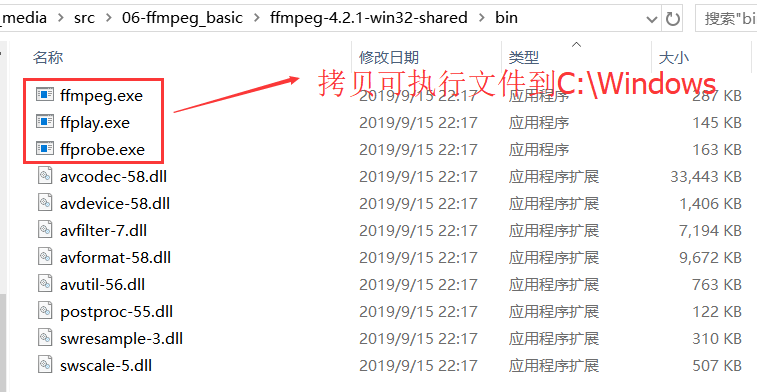
我们目前主要是使用32位的版本。

## 1.3 FFmpeg命令行环境搭建

必须安装步骤进行搭建

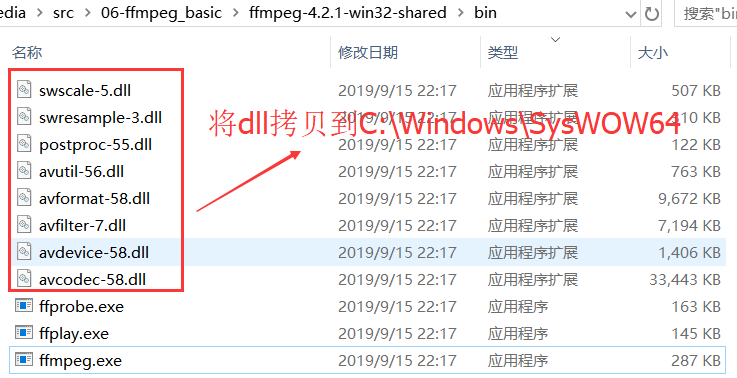
解压ffmpeg-4.2.1-win32-shared.zip

**1. 拷贝可执行文件到C:\Windows**



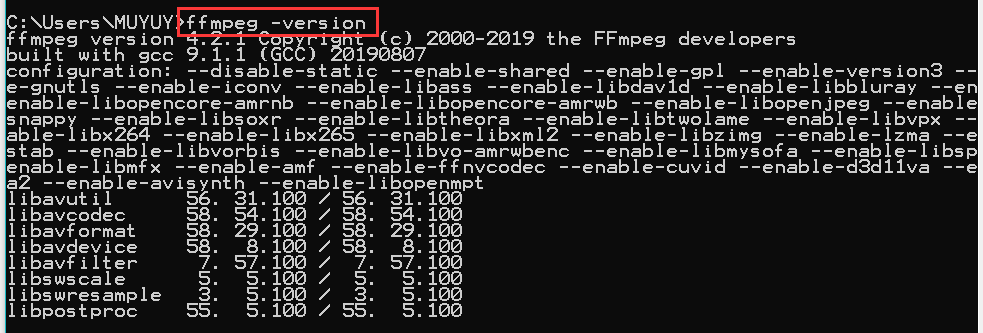
**2. 拷贝动态链接库到C:\Windows\SysWOW64**

**（**WoW64 (Windows On Windows64[1]  )是一个[Windows](https://baike.baidu.com/item/Windows)操作系统的子系统，被设计用来处理许多在32-bit Windows和64-bit Windows之间的不同的问题，使得可以在64-bit Windows中运行32-bit程序。**）**



**3. 打开cmd命令行窗口**

输入ffmpeg -version测试，打印版本号4.2.1即可。



对于命令行实战中遇到FFmpeg版本号不同的时候不用担心，之前录制FFmpeg命令行实战的时候最新的版本是4.1，现在用4.2.1版本测试是没有任何问题。

## 1.4 QT下载和安装

QT官网：<https://www.qt.io/>

### QT下载地址

为了避免QT版本的问题，请大家也尽量使用5.10.1版本。

下载版本：QT版本 5.10.1

下载地址：<http://download.qt.io/official_releases/qt/5.10/5.10.1/>

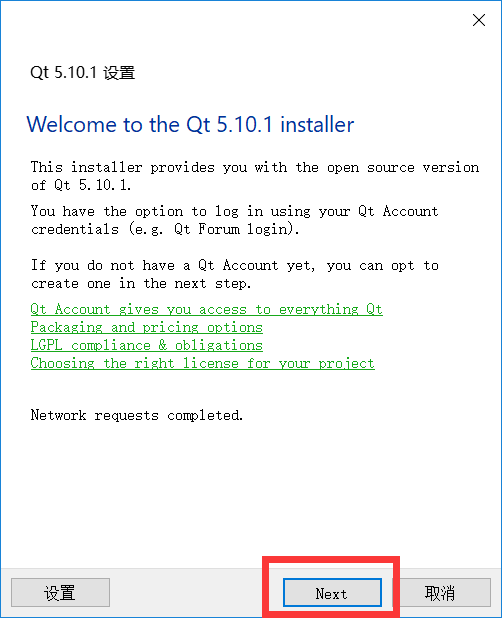
选择该版本



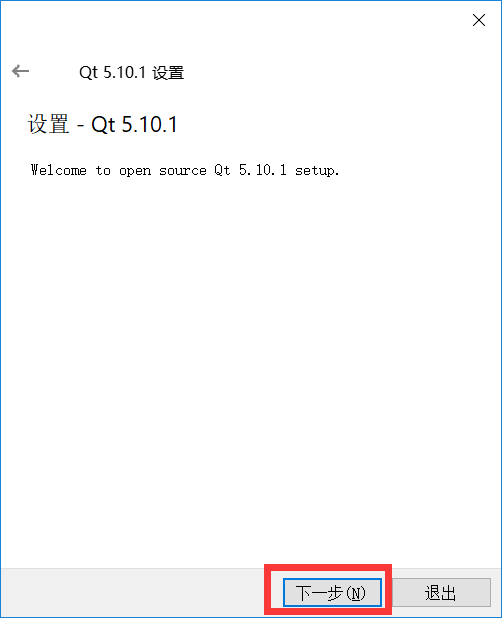
直接下载地址：<http://iso.mirrors.ustc.edu.cn/qtproject/archive/qt/5.10/5.10.1/qt-opensource-windows-x86-5.10.1.exe>

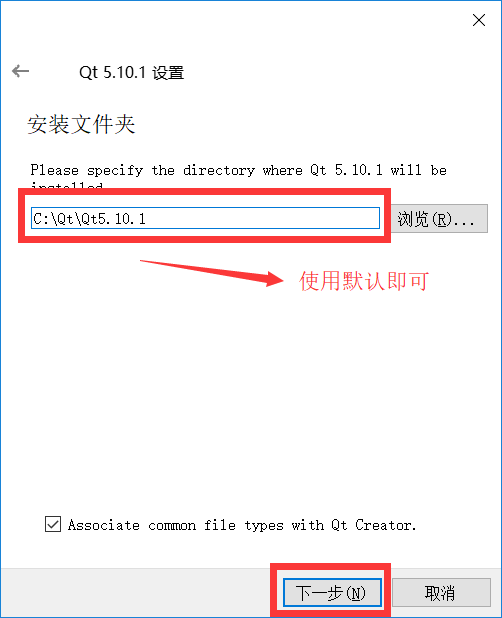
### QT安装

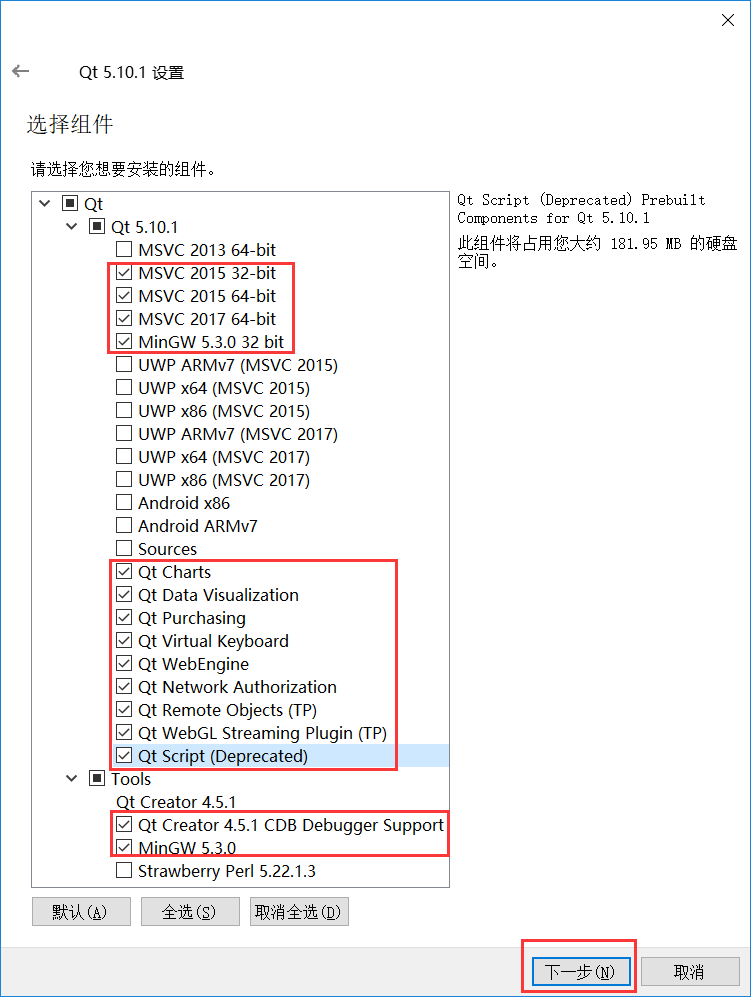
安装安装向导一步步Next（或下一步），



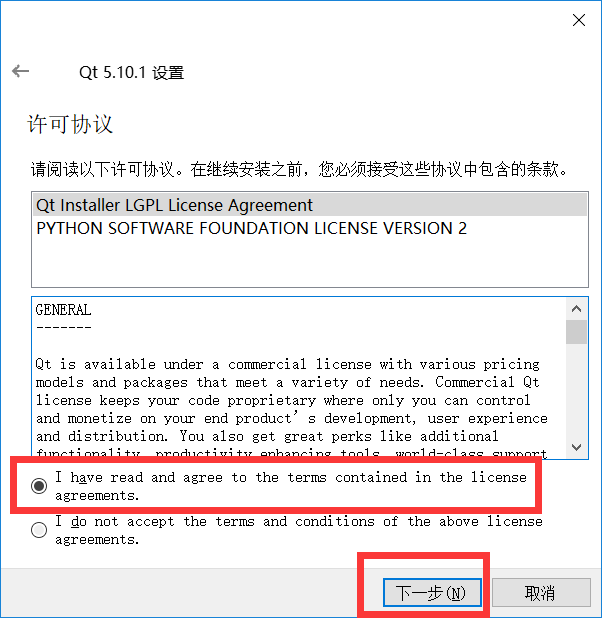


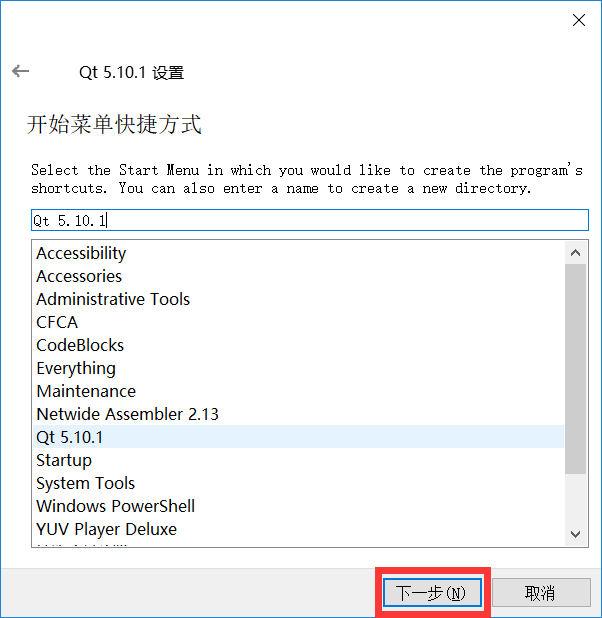


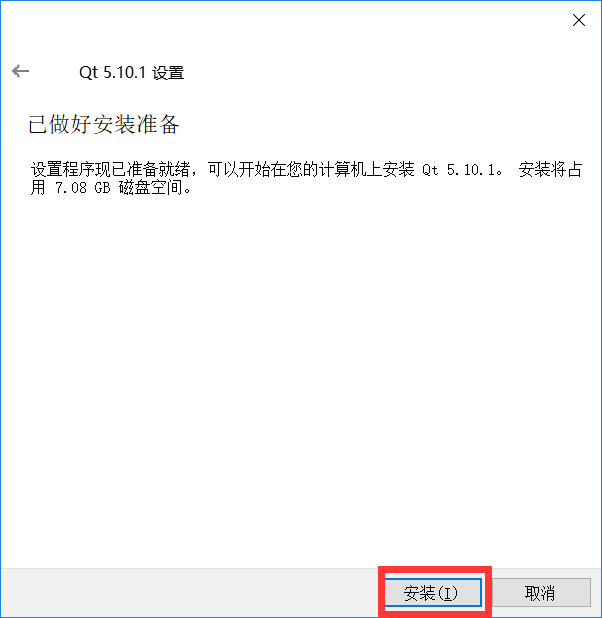




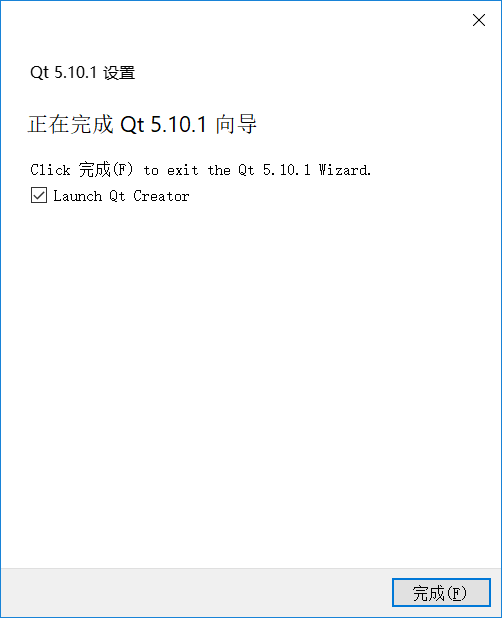
如果你想阅读QT源码，则可以勾上，但比较占用硬盘。







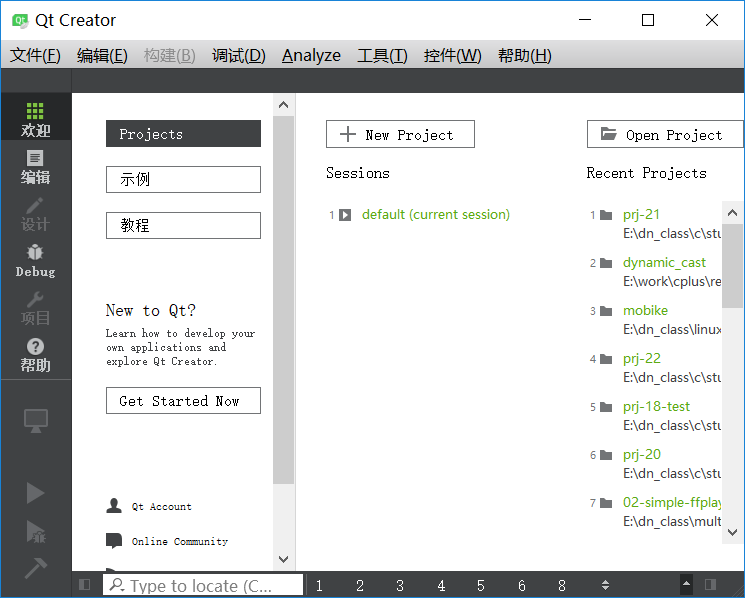
然后等待安装结束。



# 2 测试QT+FFmpeg的使用

## 创建QT工程

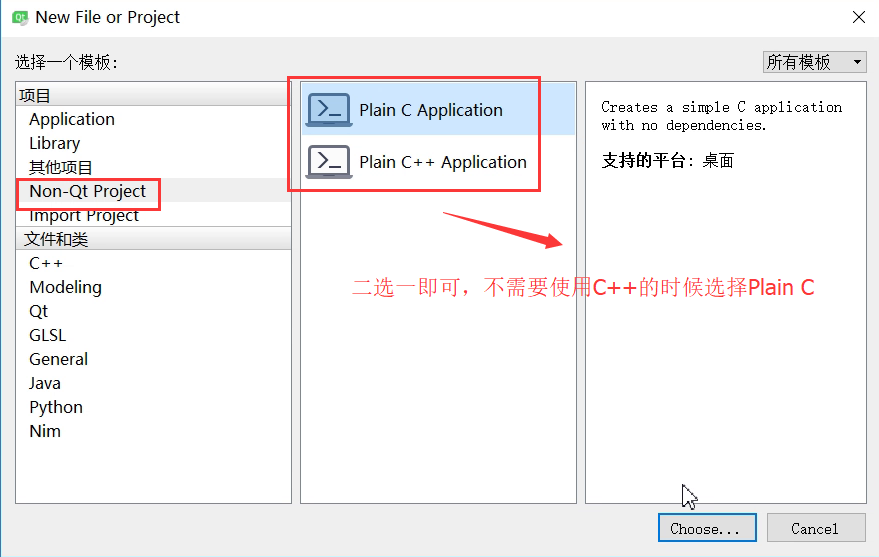
1 刚打开QT Creator的界面



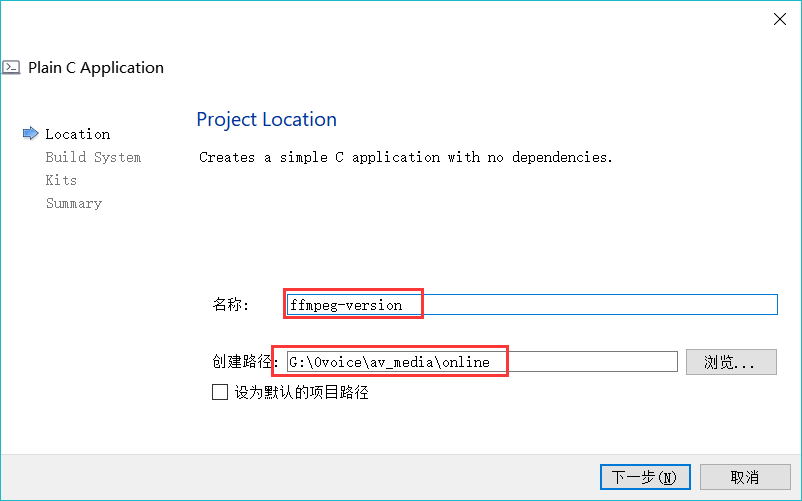
2 新建工程

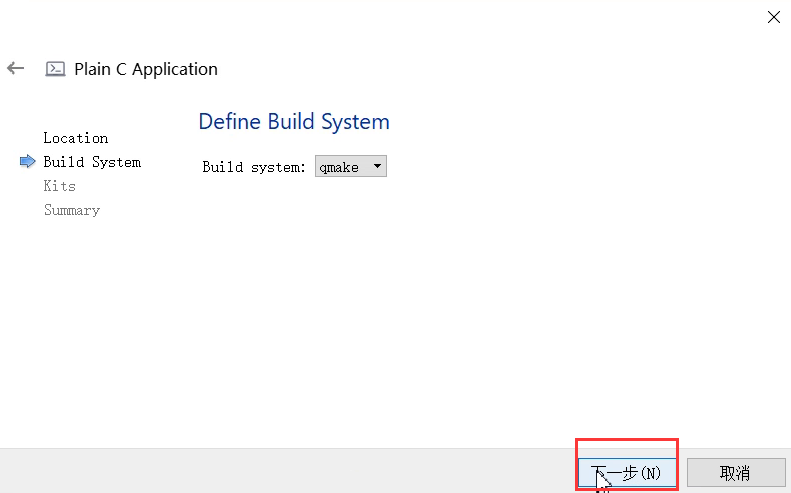


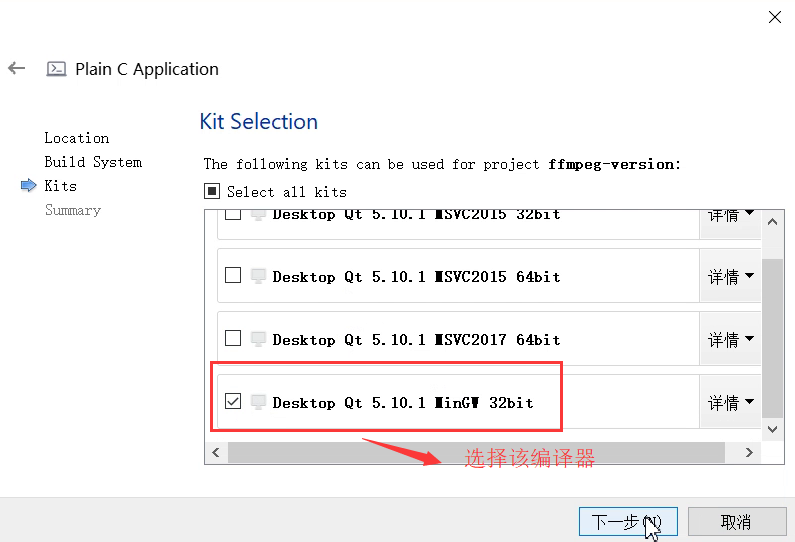
3 选择Non-Qt Project



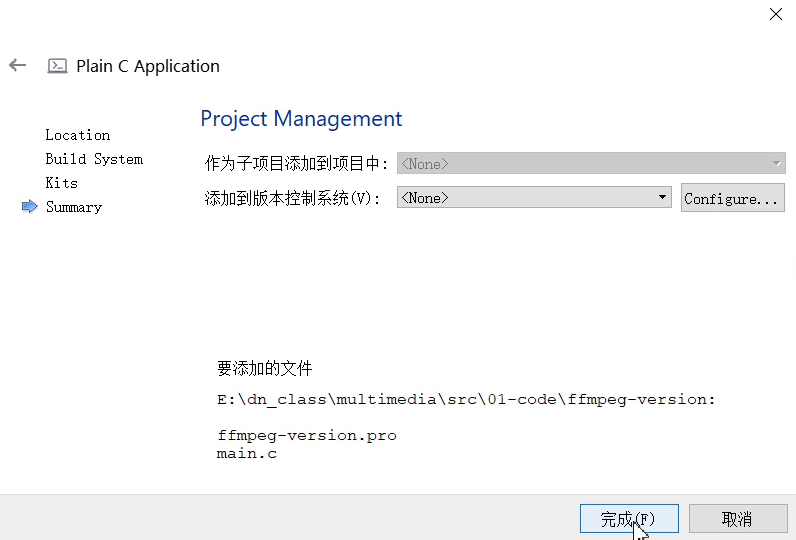
4 填写项目名称以及路径，如下所示就创建了一个叫ffmpeg-version的工程。







根据实际情况进行选择，在rtmp推流项目选择MSVC2015 32bit进行（主要部分lib和dll编译器兼容性的考虑）。

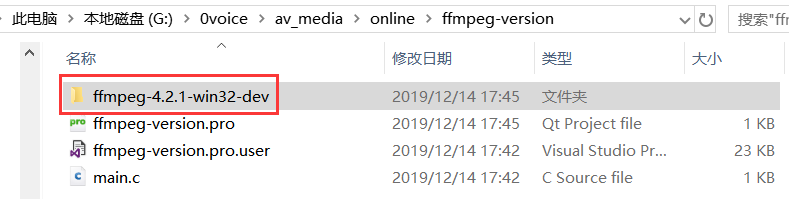


到此创建了一个基本的工程。

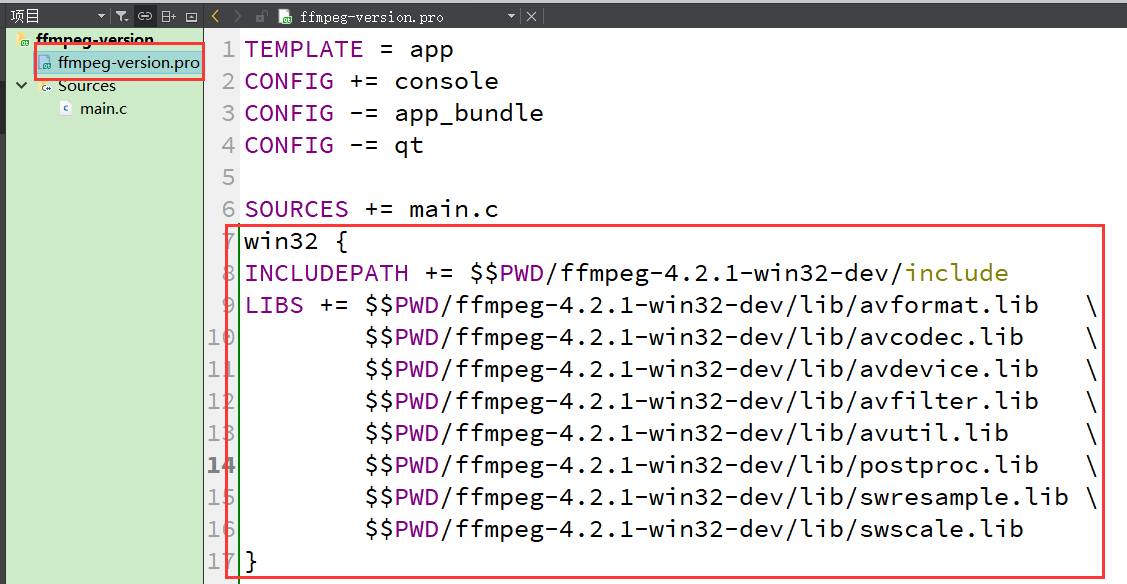
**注意：需要使用C++时则选择。**

## 引用FFMPEG库

将ffmpeg-4.2.1-win32-dev拷贝到ffmpeg-version目录下



在ffmpeg-version.pro里面添加ffmpeg头文件和库文件路径



即是

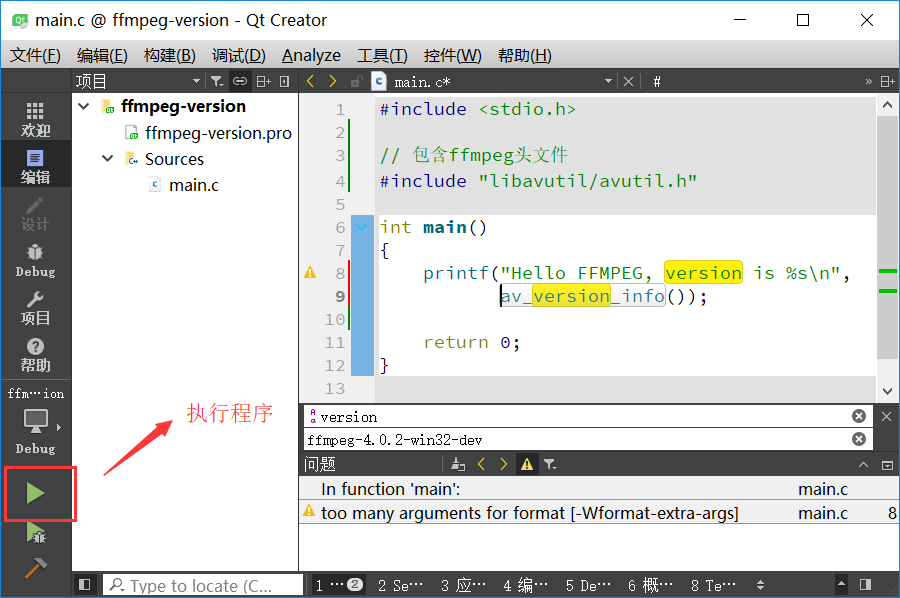
|  |
| --- |
| win32 {  INCLUDEPATH += $$PWD/ffmpeg-4.2.1-win32-dev/include  LIBS += $$PWD/ffmpeg-4.2.1-win32-dev/lib/avformat.lib \  $$PWD/ffmpeg-4.2.1-win32-dev/lib/avcodec.lib \  $$PWD/ffmpeg-4.2.1-win32-dev/lib/avdevice.lib \  $$PWD/ffmpeg-4.2.1-win32-dev/lib/avfilter.lib \  $$PWD/ffmpeg-4.2.1-win32-dev/lib/avutil.lib \  $$PWD/ffmpeg-4.2.1-win32-dev/lib/postproc.lib \  $$PWD/ffmpeg-4.2.1-win32-dev/lib/swresample.lib \  $$PWD/ffmpeg-4.2.1-win32-dev/lib/swscale.lib  } |

**LIBS的多行引用一定要记得带斜杠，否则后续的引用无效。**

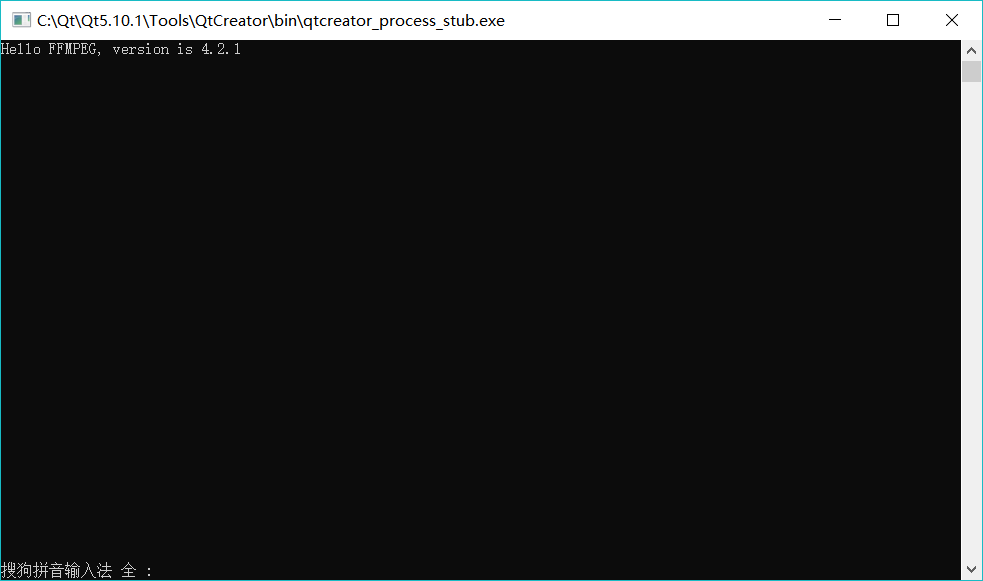
## 修改main.c文件

|  |
| --- |
| #include <stdio.h>  // 包含ffmpeg头文件  #include "libavutil/avutil.h"  int **main**()  {  printf("Hello FFMPEG, version is %s\n", av\_version\_info());  return 0;  } |

## 执行程序



显示 **Hello FFMPEG, version is 4.2.1**



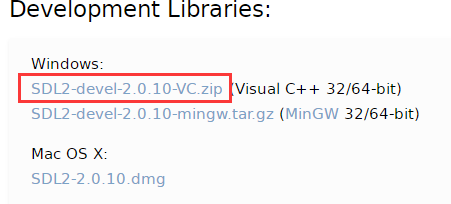
到这里，我们Windows QT+FFMPEG的开发环境就搭建完毕了。

需要注意的是，如果我们没有进行 FFmpeg命令行环境搭建，则这里在执行的时候不会有数据显示，只是黑屏，因为它缺乏对应的dll。

# 3 添加SDL2动态库

下载地址：<https://www.libsdl.org/download-2.0.php>

先直接下载dll和lib使用



将SDL2-devel-2.0.10-VC\SDL2-2.0.10\lib\x86的**SDL2.dll 拷贝到C:\Windows\SysWOW64**

**否则使用ffplay命令时无法显示画面和输出声音。**