

Qt 5.6.0-VS2015 版本安装配置图文教程

作者：Winland0704

〇、前言

因为 Visual Studio 在中国区太盛行了，正好 Qt 5.6.0 版本刚发布，写一篇关于 Qt VS2015 版本的安装配置说明。这里首先介绍一下集成开发环境（Integrated Development Environment, IDE）的概念，一般 IDE 除了直接的代码编辑和项目管理功能，还有非常重要的功能就是编译链接生成目标程序，以及调试目标程序可能的错误。

在 Qt VS 版本开发过程中，可以使用的 IDE 其实有两套：

第一套 IDE，使用 Visual Studio，里面自动集成了编译、链接、调试的功能，通过 Qt Visual Studio Add-in 插件（注意微软耍流氓，VS2015 移除了插件支持功能，需要改用微软 MSDN 网站的 Qt5Package.vsix 扩展包）来集成和管理 Qt 项目。使用 VS 开发的 Qt 项目，如果与 Windows 开发比较紧密，使用了 Qt 库之外的系统类库，那么这些代码的跨平台特性比较差，可能无法在 Unix/Linux 和 Mac OS X 系统编译成功。VS 使用的项目管理文件为 *.sln 和 *.vcproject，这些项目文件只在 Windows 系统的 VS 开发环境可用，不能跨平台。

第二套 IDE，使用 Qt 官方的集成开发环境 QtCreator + 微软的 WinDbg 调试器（内含命令行调试器为 CDB）。Visual Studio 只提供了编译器和链接器独立程序，但 VS 的调试器是它内部集成，无法在外部调用，QtCreator 只能用 WinDbg 包含的命令行调试器 CDB，这就是为什么要用 WinDbg 的原因。QtCreator 一般使用 *.pro 文件管理项目，而且 QtCreator 本身就是跨平台的，这种项目文件通常具有优良的跨平台编译生成特性，所以如果希望开发跨平台的程序，应该使用 QtCreator 和 *.pro 文件管理项目。QtCreator 是 Qt 官方专门定做的集成开发环境，使用 QtCreator 开发 Qt 程序效率更高，信号和槽的关联和自动生成更方便，并且代码补全功能和随时的文档查阅功能也都优于 VS，另外 QtCreator 集成的帮助文档是离线版，不需要联网，而且 Qt 帮助文档内容非常详实，又有很多示例代码，这些优点都是 Visual Studio 无法比拟的。之所以 QtCreator 知名度不高，是因为国内教育环境老掉牙了而已。

本教程会把这两套 IDE 的安装配置都讲一遍，但推荐读者使用 QtCreator 来开发 Qt 程序。

一、工具和开发库下载

笔者使用的系统是 Win7 SP1 32 位系统（系统要打最新的补丁，并且 VS2015 推荐 IE10 及以上浏览器），后面都是 32 位的 Qt 库和调试器。

如果读者是 64 位系统，那么建议选择 64 位的 Qt 库和 WinDbg 调试器。

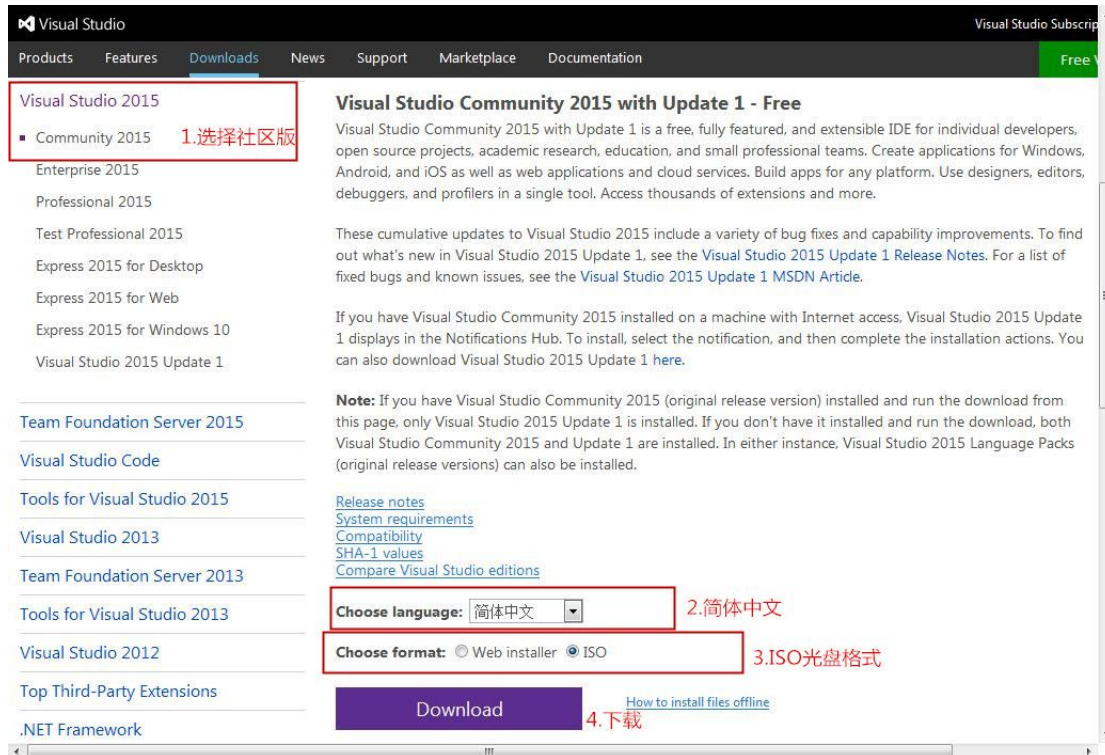
本教程使用到了四个安装程序，罗列如下：

1、Visual Studio Community 2015 with Update 1 免费版

下载地址：

<https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx>

根据下图说明，下载简体中文的 ISO 文件：



得到的文件为 **vs2015.1.com_chs.iso**，大小 5.78GB，要注意磁盘空间够不够。

2、Qt 官方的开发库安装程序

不建议从 www.qt.io 去找安装包，太繁琐，直接用下面的链接就能看到一堆安装包：

http://download.qt.io/official_releases/qt/5.6/5.6.0/

Name	Last modified	Size	Metadata
↑ Parent Directory		-	
submodules/	17-Mar-2016 10:38	-	
single/	15-Mar-2016 14:37	-	
qt-opensource-windows-x86-wiirt-msvc2015-5.6.0.exe	15-Mar-2016 13:55	1.0G	Details
qt-opensource-windows-x86-wiirt-msvc2013-5.6.0.exe	15-Mar-2016 13:54	1.0G	Details
qt-opensource-windows-x86-msvc2015_64-5.6.0.exe	15-Mar-2016 13:53	836M	Details
qt-opensource-windows-x86-msvc2015-5.6.0.exe	15-Mar-2016 13:53	840M	Details
qt-opensource-windows-x86-msvc2013_64-5.6.0.exe	15-Mar-2016 13:52	836M	Details
qt-opensource-windows-x86-msvc2013-5.6.0.exe	15-Mar-2016 13:51	840M	Details
qt-opensource-windows-x86-mingw492-5.6.0.exe	15-Mar-2016 13:51	1.0G	Details
qt-opensource-windows-x86-android-5.6.0.exe	15-Mar-2016 13:50	1.1G	Details
qt-opensource-mac-x64-clang-5.6.0.dmg	15-Mar-2016 13:43	710M	Details
qt-opensource-mac-x64-android-ios-5.6.0.dmg	15-Mar-2016 13:42	2.1G	Details
qt-opensource-mac-x64-android-5.6.0.dmg	15-Mar-2016 13:41	761M	Details
qt-opensource-linux-x64-android-5.6.0.run	15-Mar-2016 13:38	717M	Details
qt-opensource-linux-x64-5.6.0.run	15-Mar-2016 13:38	661M	Details
qt-opensource-...-5.6.0.run	15-Mar-2016 14:02	1.0G	Details

安装包带 wiirt 字样的不要管，对于 32 位系统下载：

qt-opensource-windows-x86-msvc2015-5.6.0.exe

对于 64 位系统下载：

qt-opensource-windows-x86-msvc2015_64-5.6.0.exe

下面示范的是 32 位系统的。

3、Qt 集成到 VS 的插件

Qt 官方原本开发了用于 Visual Studio 集成的插件，即 Qt Visual Studio Add-in，可用于 VS2010 到 VS2013，但是微软耍起流氓来谁都挡不住，VS2015 直接把插件支持的功能移除了，不能再使用插件了。

微软 MSDN 网站提供了一个 Qt5Package 的扩展包，勉强用于支持 Qt5 程序开发，下载地址为：

<https://visualstudiogallery.msdn.microsoft.com/c89ff880-8509-47a4-a262-e4fa07168408>



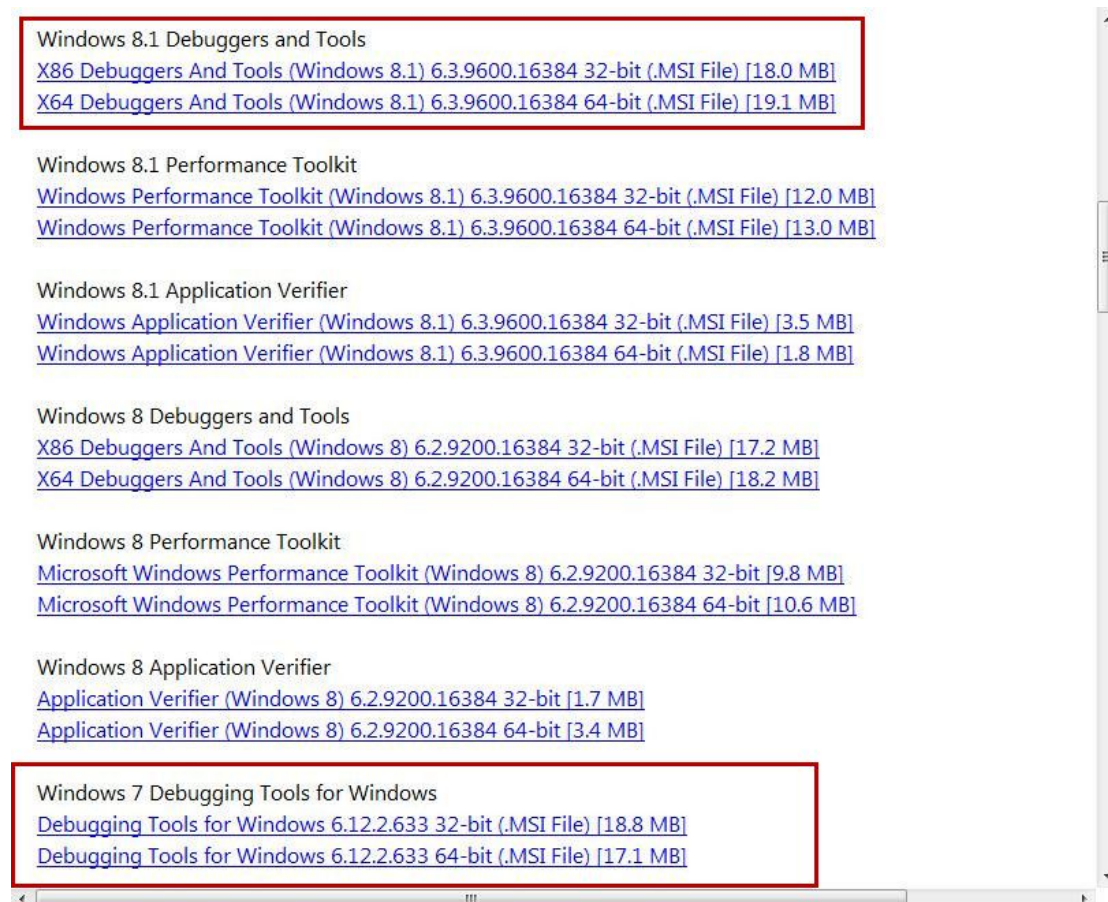
下载后的文件名为 **Qt5Package.vsix** 。

4、WinDbg 调试器

微软官方把 WinDbg 调试器隐含到 Windows SDK 和 Windows Driver Kit 里面了，老外有比较精的把各个操作系统适用的 WinDbg 都抽取出来了，下载页面：

<http://www.codemachine.com/downloads.html>

把这个网页往下拖，找到 Downloads 下面的链接，根据自己的操作系统选择调试器：



笔者下载的是 Win 7 32 位系统的调试器：

[dbg_x86_6.12.2.633.msi](#)

等四个安装包都下载完成，下面开始安装。

二、Visual Studio 2015 安装（必备）

提前说明：

VS2015 安装过程非常漫长，而且 VS2015 默认不安装 VC++ 编译器，需要自定义安装。

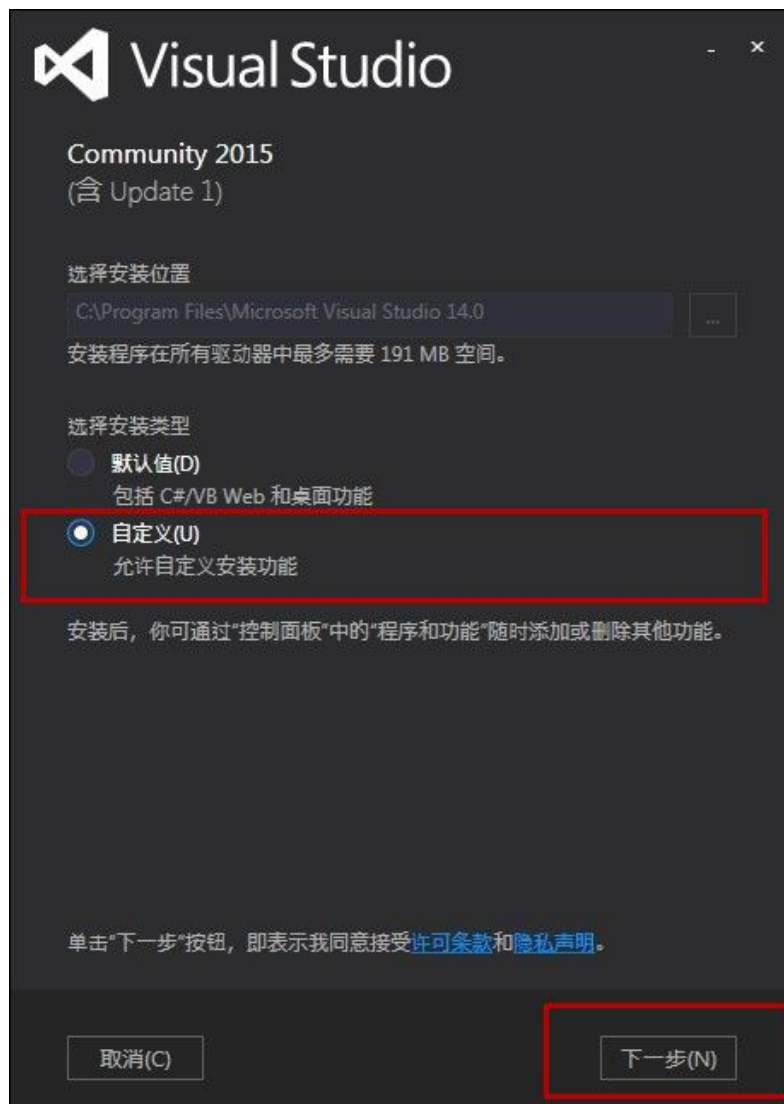
虽然 ISO 文件是离线版，但安装程序还是会从网上下载一些开发组件，比如 Windows 10 SDK 开发包，离线版安装时建议也联好网。

另外安装之前要准备好磁盘空间，最好 25GB 以上空闲空间。

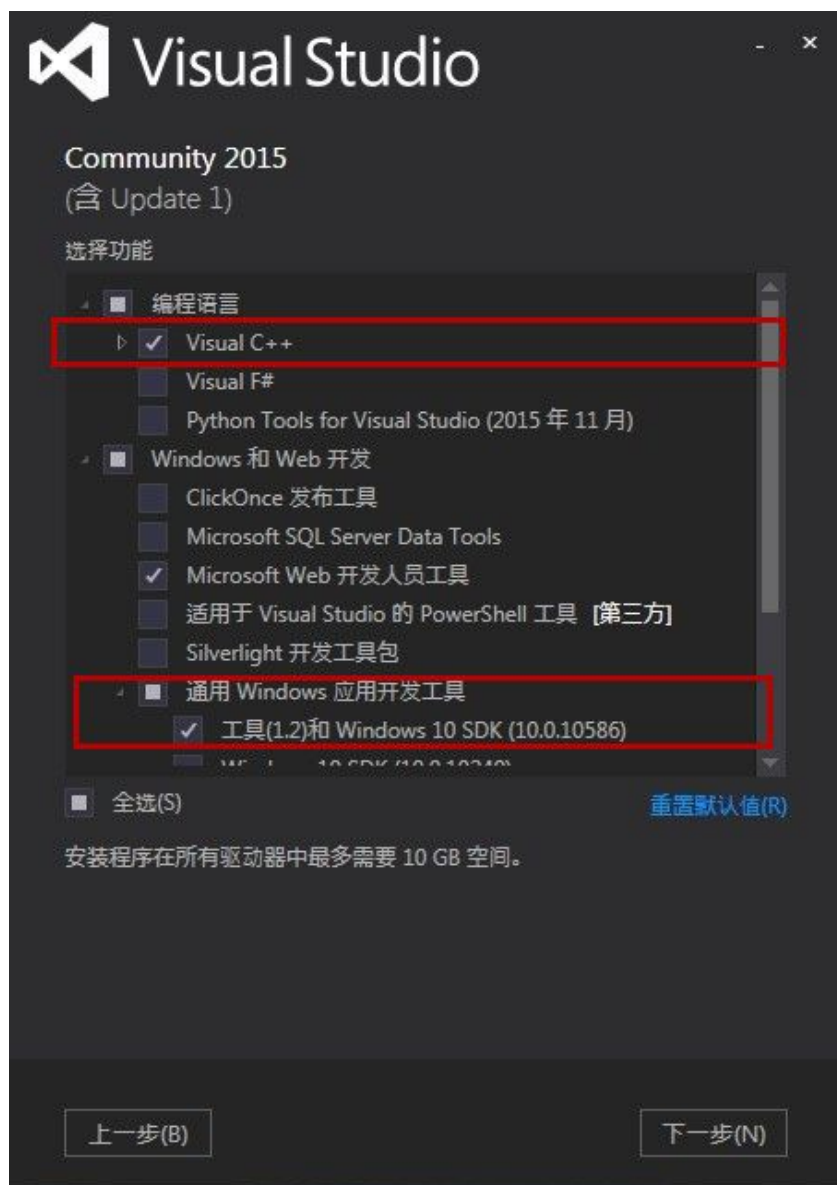
使用虚拟光驱工具（好压就自带虚拟光驱）把 vs2015.1.com_chs.iso 加载到虚拟光驱，然后运行里面的 vs_community.exe：



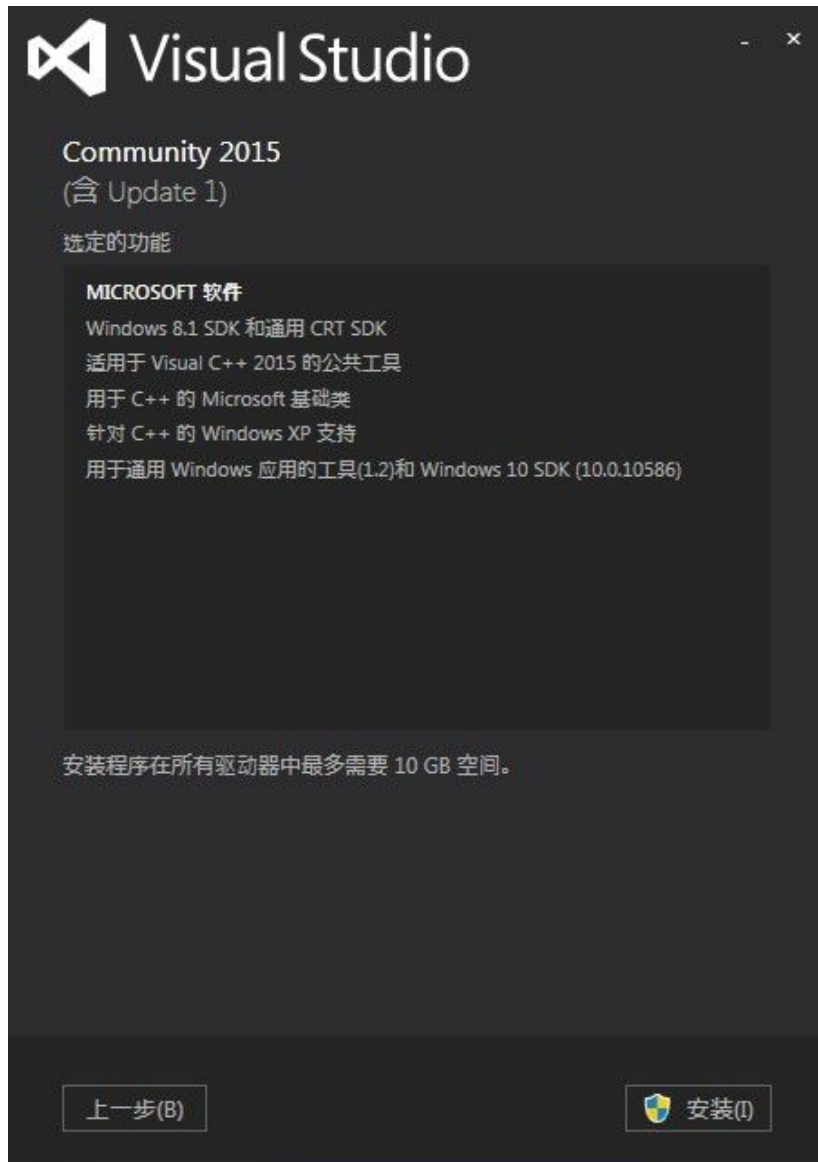
注意不要用默认值安装，VS2015 默认情况下不安装 VC++ 编译器和开发库，一定要选第二个自定义安装：



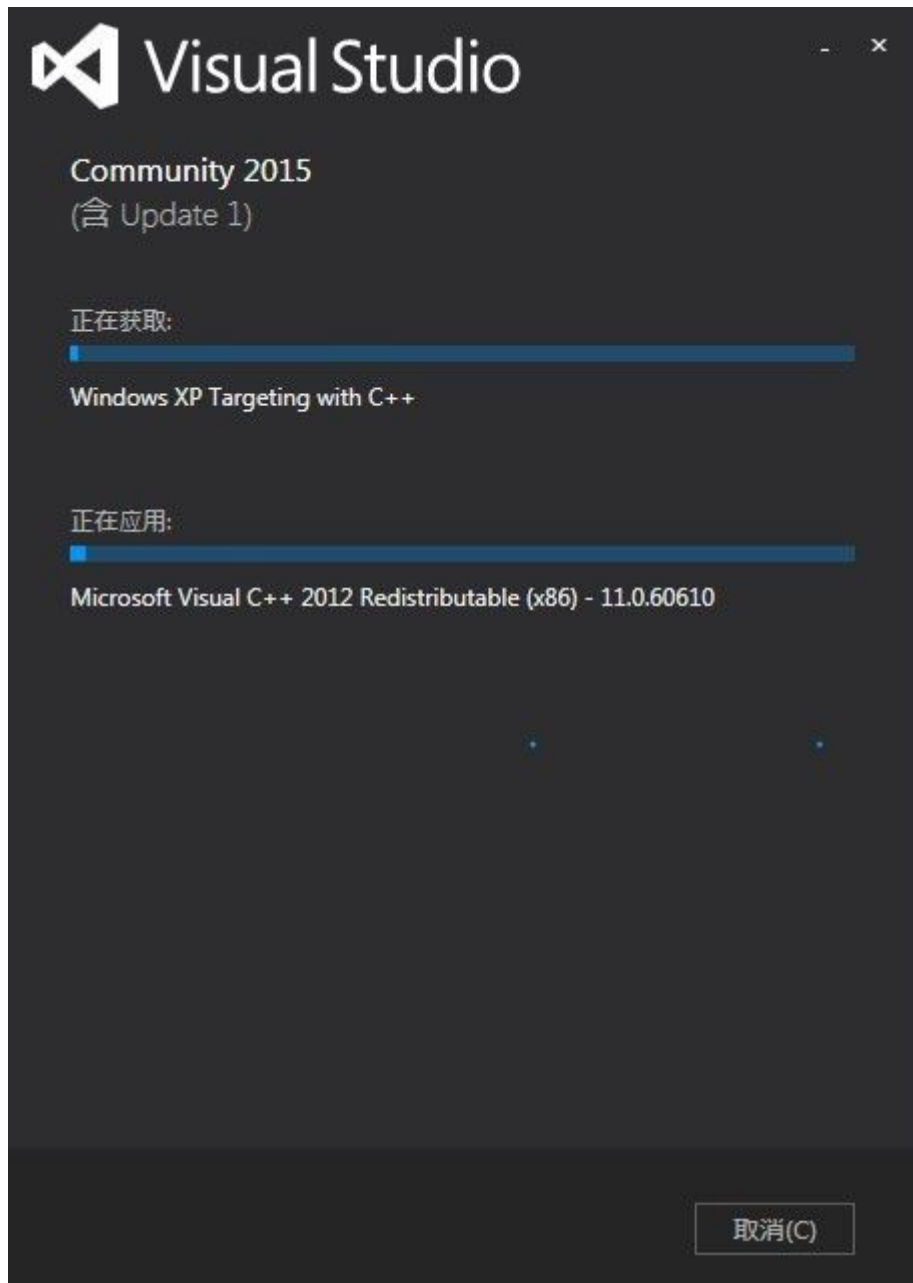
点击下一步，进入下面组件选择界面：



在编程语言里面要选中 **“Visual C++”**，**这个组件是必需的**；
另外建议在下面 Windows 和 Web 开发里面点开 “通用 Windows 应用开发工具”，选择其中一个版本的 Windows SDK （这个 SDK 可以不安装，不影响 Qt 开发）；
其他的组件读者根据自己开发需要来安装（现在 VS 能开发 Android 和 iOS 程序，可以选中 “跨平台移动开发-->Visual C++移动开发” 组件），选好组件之后点击 “下一步”：



确定要安装的组件之后，点击“安装”，



建议睡前挂机安装，时间实在太长，注意保持网络畅通，VS2015 需要联网下载组件。

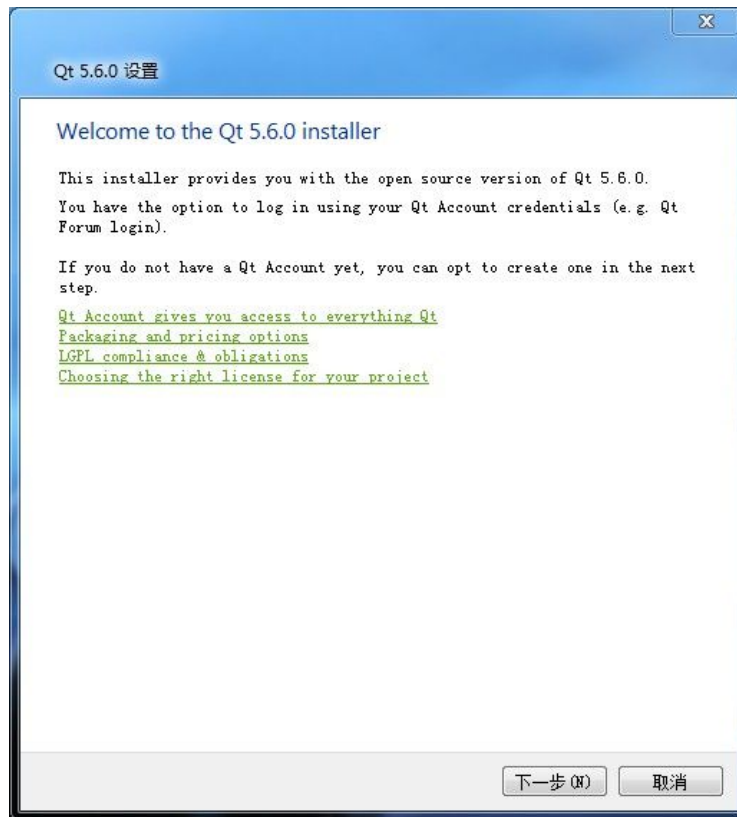
安装 Visual Studio 2015 之后，需要重启系统。

VS2015 首次启动时选择 Visual C++的风格，可以新建一个 MFC 程序，试试 VS2015 是否正常工作。

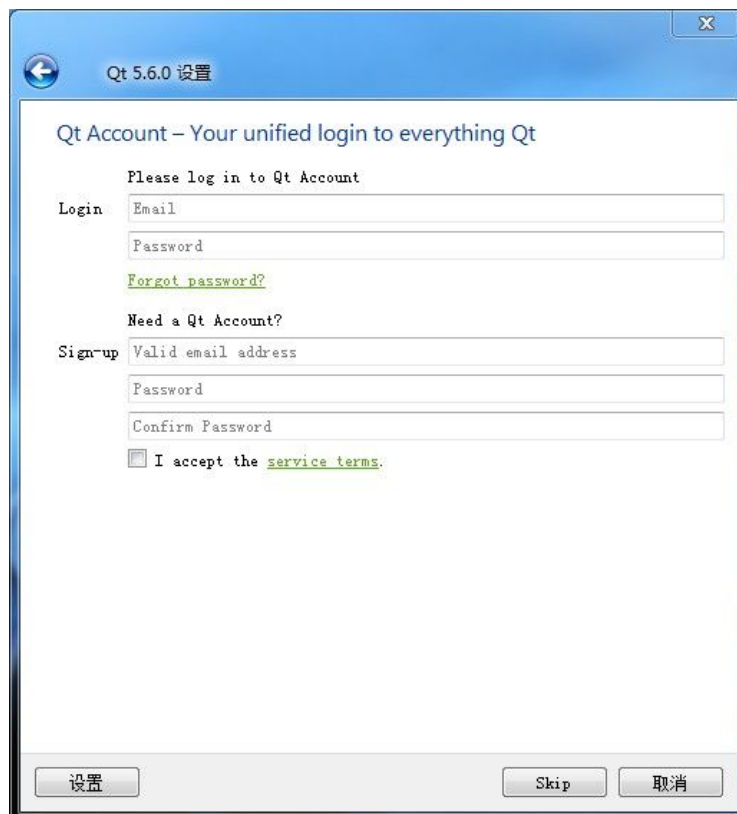
三、Qt 开发库的安装（必备）

注意 Qt 安装的目标磁盘分区要预留足够的剩余空间，如果安装 Qt 开发库和源代码大概需要 5GB，如果只安装开发库大概需要 3GB。

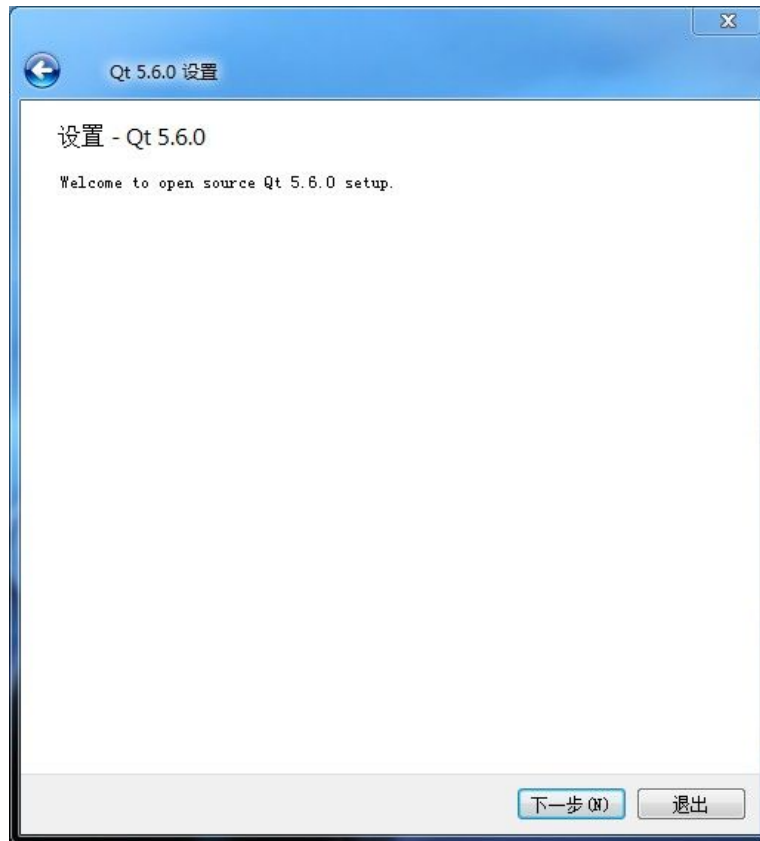
双击运行 qt-opensource-windows-x86-msvc2015-5.6.0.exe 安装程序，进入 Qt 的安装界面：



点击“下一步”，



点击 Skip，跳过账号注册，使用 Qt 开发不需要注册账号。



点击“下一步”，



读者可以自己填写安装路径（**不要用中文路径，必须用全英文的路径**）或者用默认的路径，

注意磁盘上剩余空间需要 5GB 以上。设置好路径后点击“下一步”，



如果以后要学习 Qt 的源代码，点击下方的“全选”按钮，如果不安装源代码，就是用原本“默认”的安装组件，然后点击“下一步”，



在许可协议界面，选择第一个同意许可协议，然后点击“下一步”，



上面界面直接点击“下一步”，



点击“安装”，进入正式的安装过程，



等待安装完成，进入下面界面，



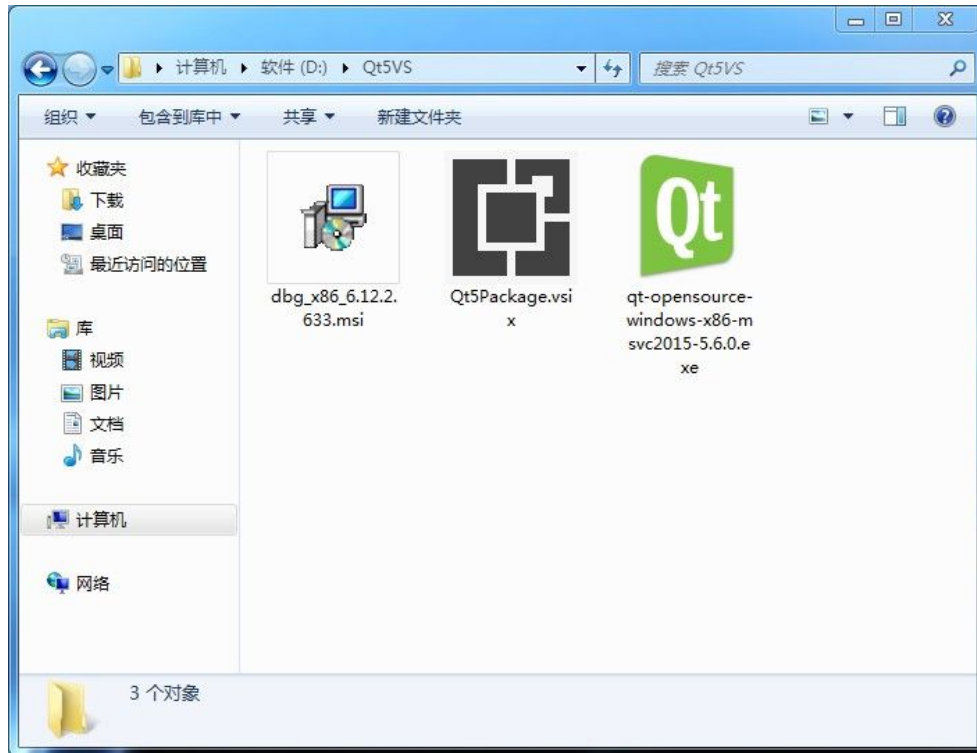
点击“下一步”，



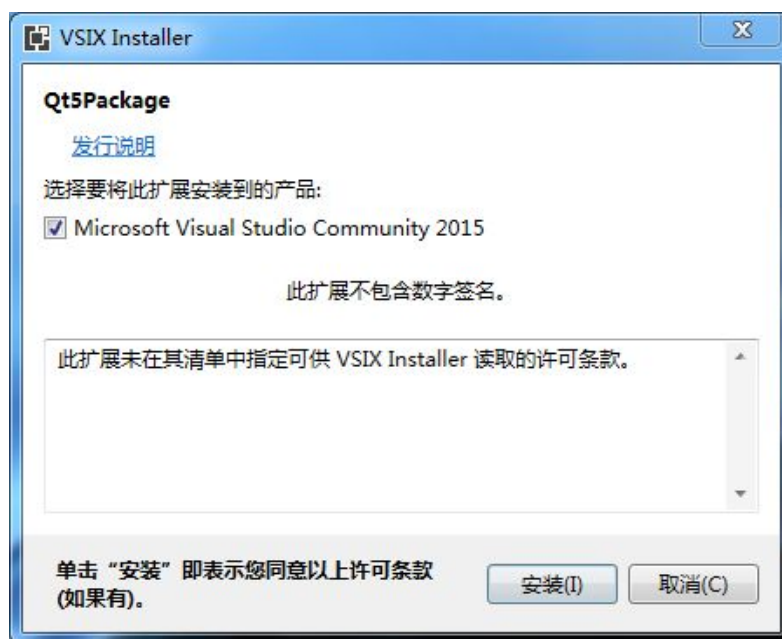
点击“完成”按钮，这样 Qt 开发库的安装就完成了。下面开始配置前文提到的两套 IDE。

四、第一套 IDE 配置，使用 VS 开发 Qt 程序（可选）

如果使用 VS2015 作为默认的 Qt 开发环境，那么需要安装 Qt5Package 扩展包，并进行配置。在 VS2015 安装之后，可以看到之前下载 Qt5Package.vsix 插件有新的图标了，VS2015 已经关联了这类 *.vsix 扩展包：



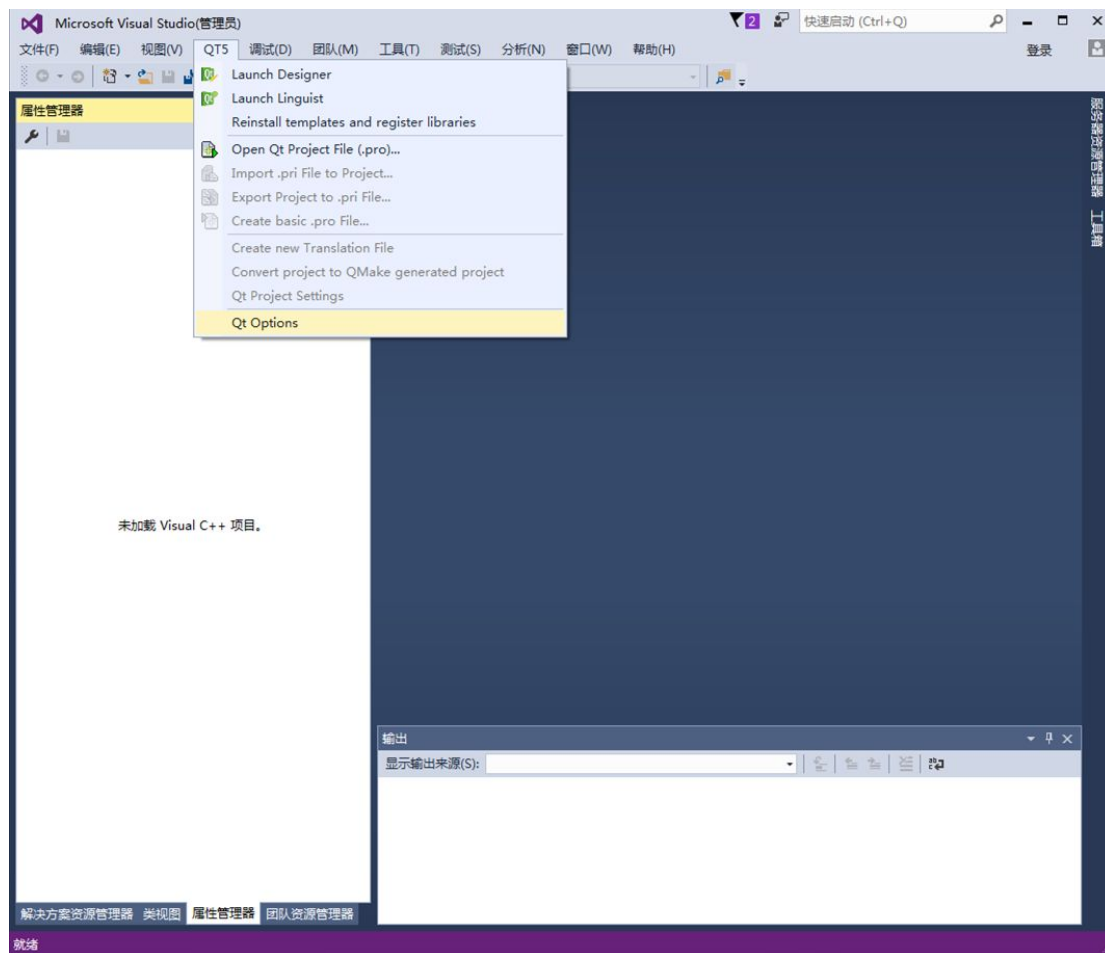
我们双击 `Qt5Package.vsix` 文件，系统会自动调用 VSIX Installer 程序，等待这个 VSIX Installer 程序初始化完成，可以看到：



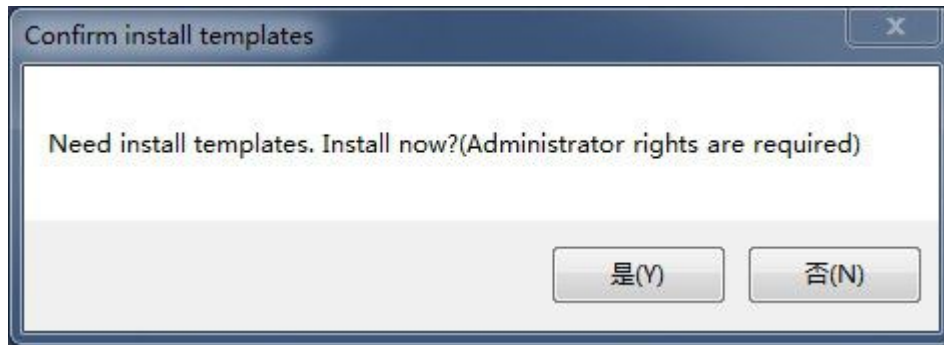
点击“安装”按钮，



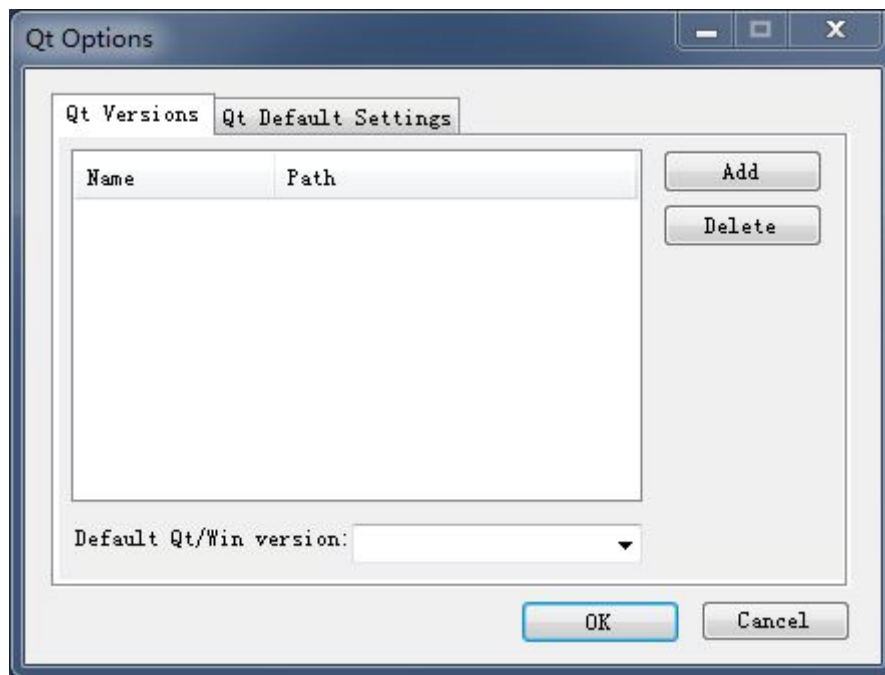
扩展包很快就安装好了，点击“关闭”按钮。
然后我们运行 VS2015，看到新增的 Qt5 菜单，点开该菜单：



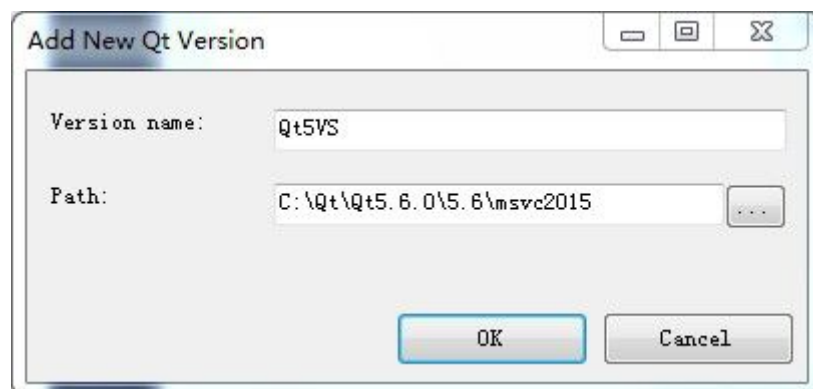
点击“Qt Options”菜单项，首次会自动弹出安装项目模板的对话框（就是上图界面 “Reinstall templates and register libraries” 菜单项的功能）：



点击“是”，现在只是安装了 Qt 在 VS 新建项目时的可选模板，还没配置 Qt 库的路径。之后在 Qt Options 对话框才是配置 Qt 开发库路径的地方：



点击“Add”按钮，看到 Add New Qt Version 对话框：

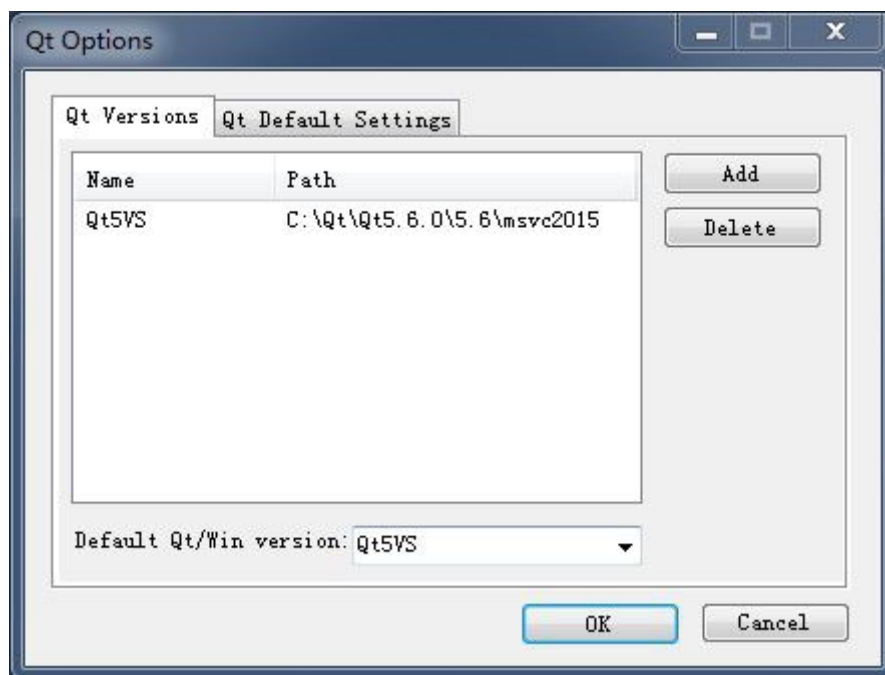


对话框了 Version name 可以按照上面的填写 Qt5VS，下面的 Path 最重要，点击“...”按钮，找到 Qt 安装路径下面 5.6\msvc2015 的文件夹，比如图上的：

C:\Qt\Qt5.6.0\5.6\msvc2015

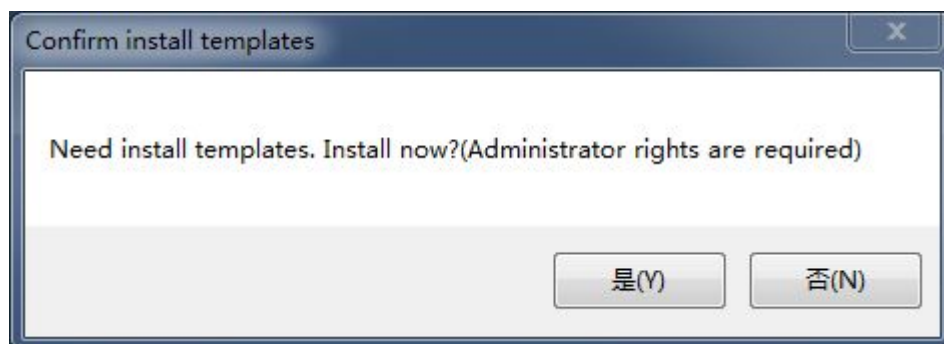
只有路径正确，才能点击 OK 按钮完成添加操作。

点击上面对话框 OK 按钮之后，会回到 Qt Options 对话框：

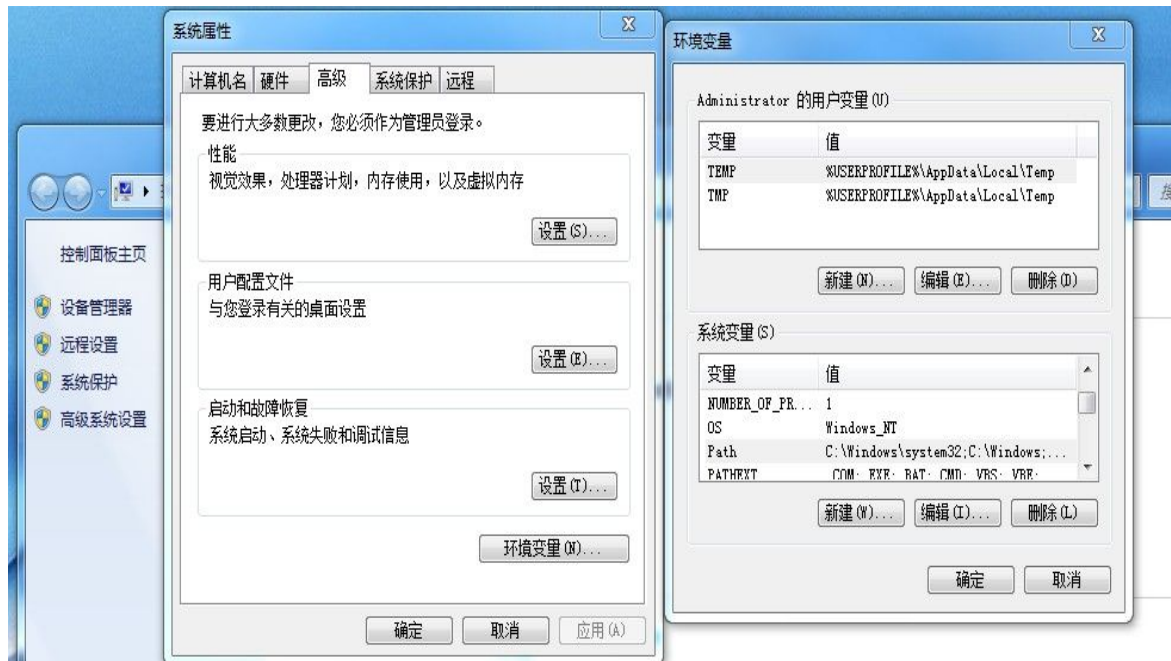


因为只安装了一套 Qt5 32 位的开发库，所以暂时不需要添加其他的，点击 OK 按钮即可。

添加 Qt5 开发库之后，我们再点击 VS2015 菜单 “Qt5” --> “Reinstall templates and register libraries ” 菜单项，将刚才配置的 Qt5 开发库注册给 VS。



然后我们先关闭 VS2015，在桌面右击 “计算机” 图标，打开 “属性”，在系统属性界面点击 “高级系统设置”，



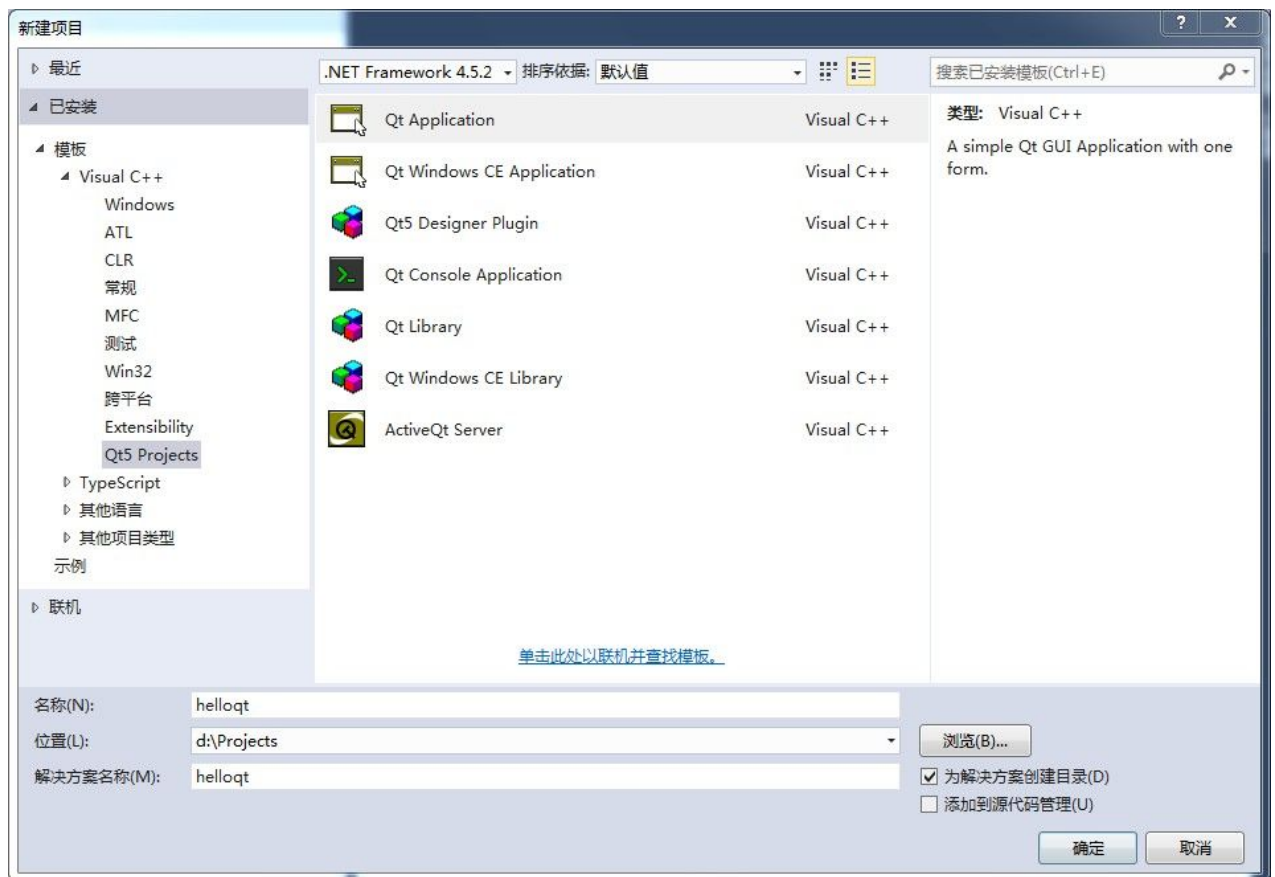
点击“环境变量”按钮，然后编辑系统变量 PATH，
在原有的 PATH 变量后面添加英文分号，并把 Qt 库安装后大量工具和动态链接库所处的 bin
路径添加到环境变量：

C:\Qt\Qt5.6.0\5.6\msvc2015\bin

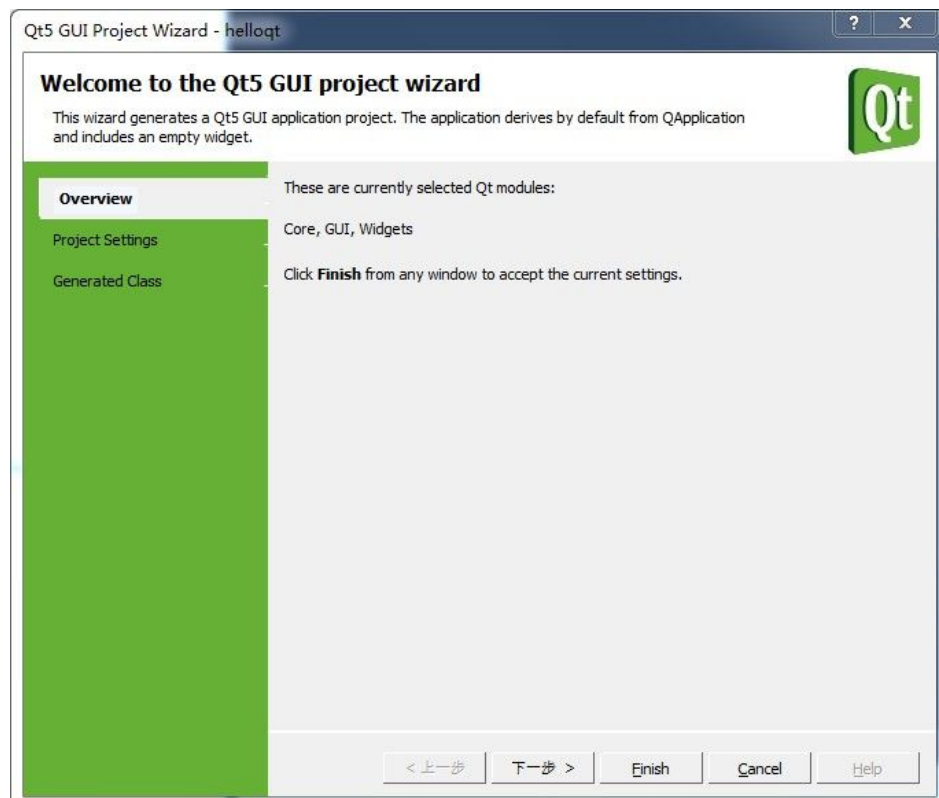


点击确定。

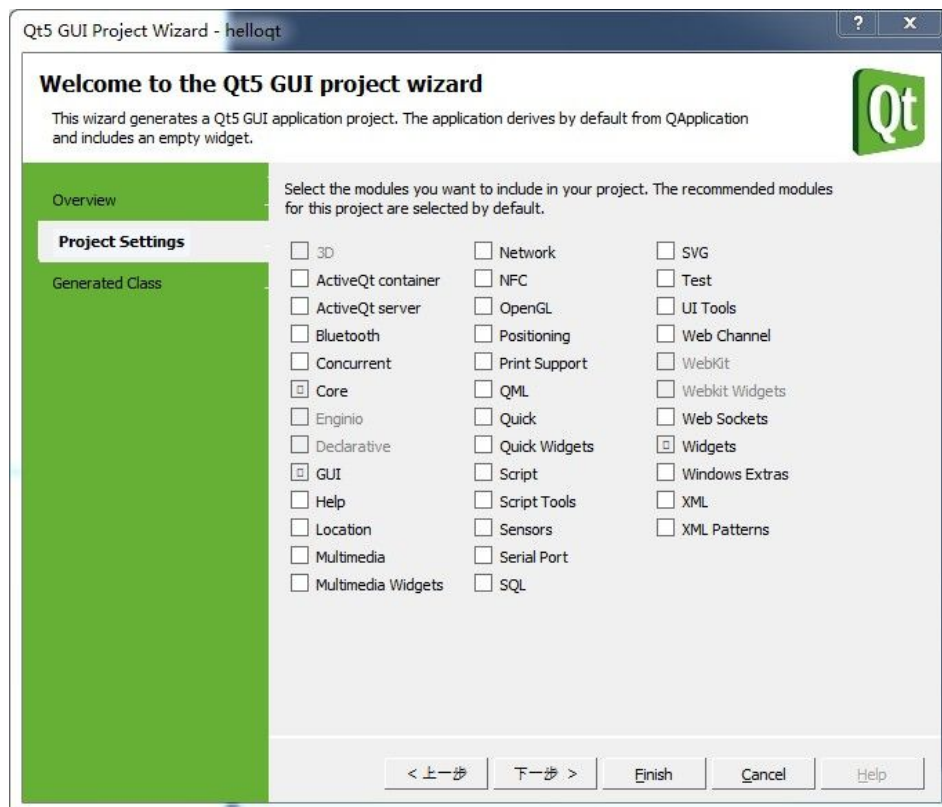
编辑好环境变量之后，重新打开 VS2015，点击菜单“新建”-->“项目”，在新建项目对话框可以看到有 Qt 的项目模板：



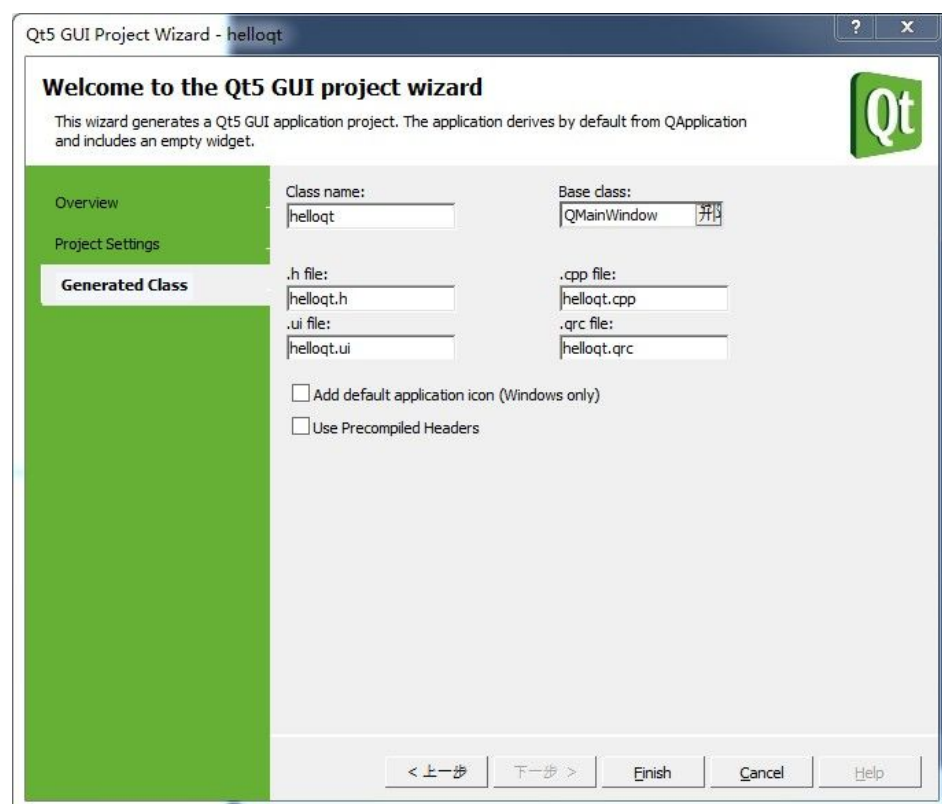
在 Visual C++子树里面有 Qt5 Projects，然后在右边选中 Qt Application，下面名称填写 helloqt
位置填写 d:\Projects ，Qt 项目源码最好全都放在全英文的路径！
解决方案名称也是 helloqt，点击“确定”按钮，进入 Qt 项目向导界面：



点击“下一步”，

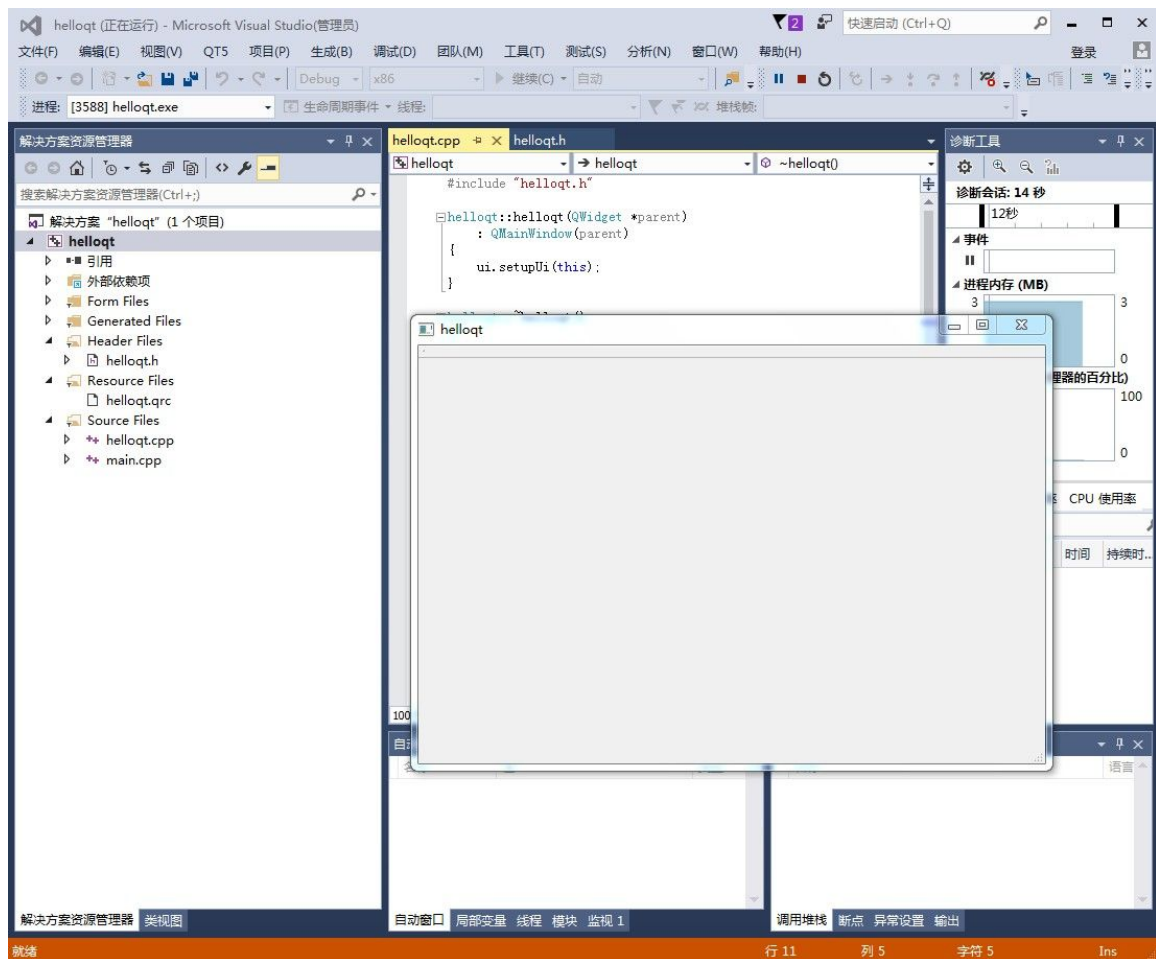


在 Qt 模块选择界面，目前用默认的即可，以后编程会用到更多的模块，点击“下一步”，



这步也用默认的，点击 Finish。

新建好项目之后，点击 VS 工具栏的“本地 Windows 调试器”按钮，就可以生成运行项目了：



目前关于 VS 集成配置的就介绍到这，如果遇到问题可以参考下面两篇文档试试：

<http://my.oschina.net/jthmath/blog/640227>

<http://blog.csdn.net/liuvez123/article/details/50475059>

五、第二套 IDE 配置，使用 QtCreator 开发 Qt 程序（推荐）

下面介绍 QtCreator IDE 的配置工作，VS2015 里面的 VC++编译器、链接器都是独立程序，所以即使不配置 QtCreator，默认也是能自动生成基于 VC++编译器的目标程序，下面内容主要是安装配置 WinDbg 调试器。

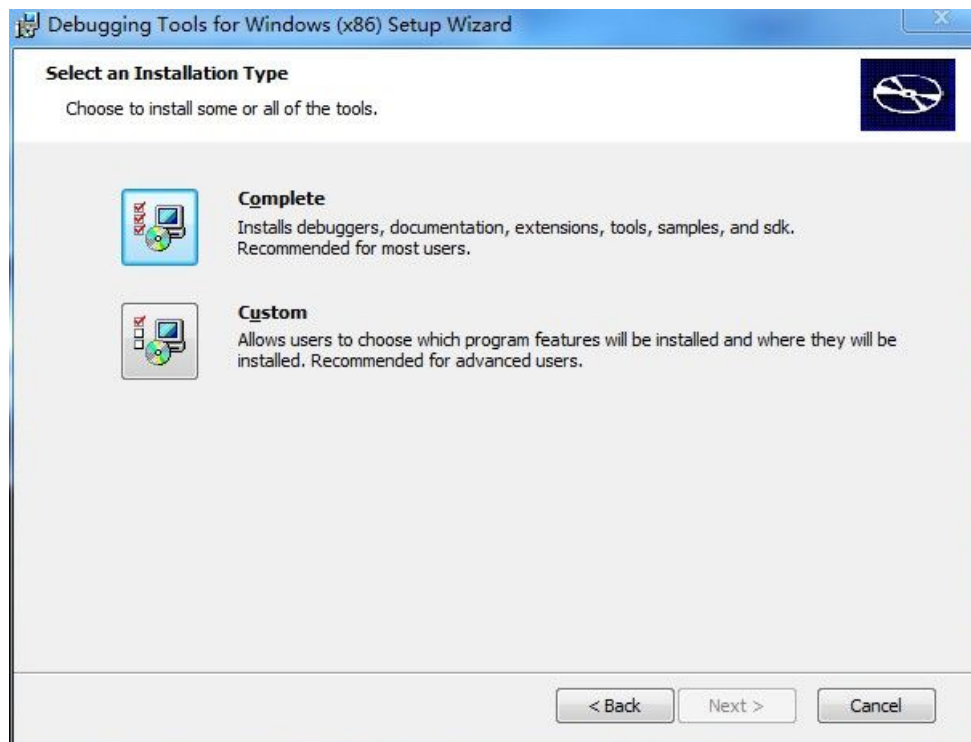
双击运行 dbg_x86_6.12.2.633.msi 文件：



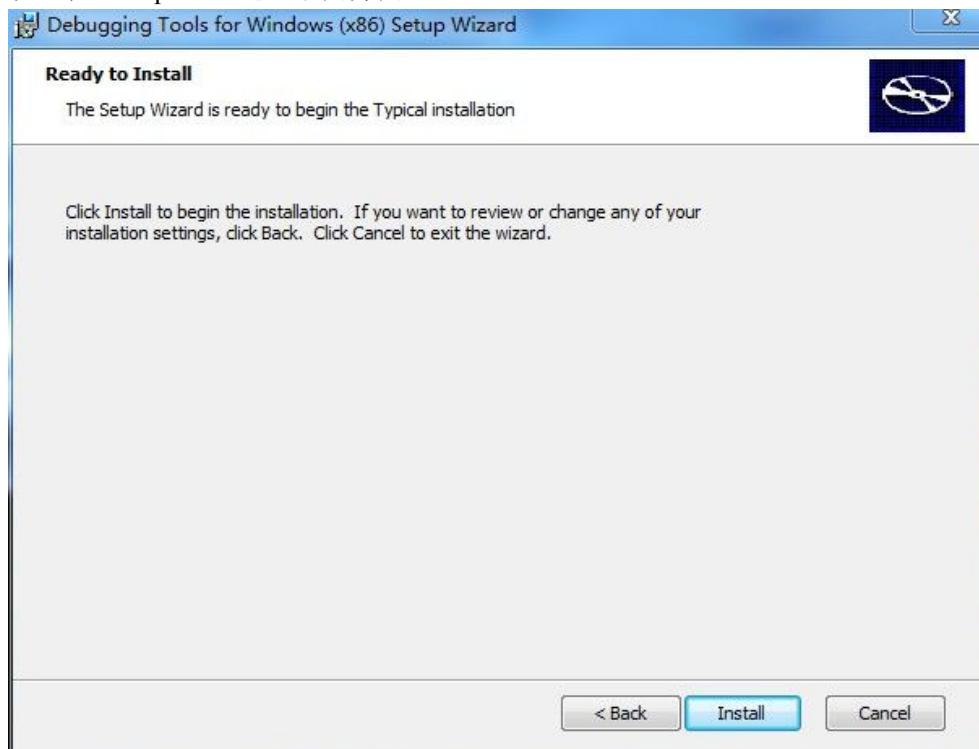
点击 Next,



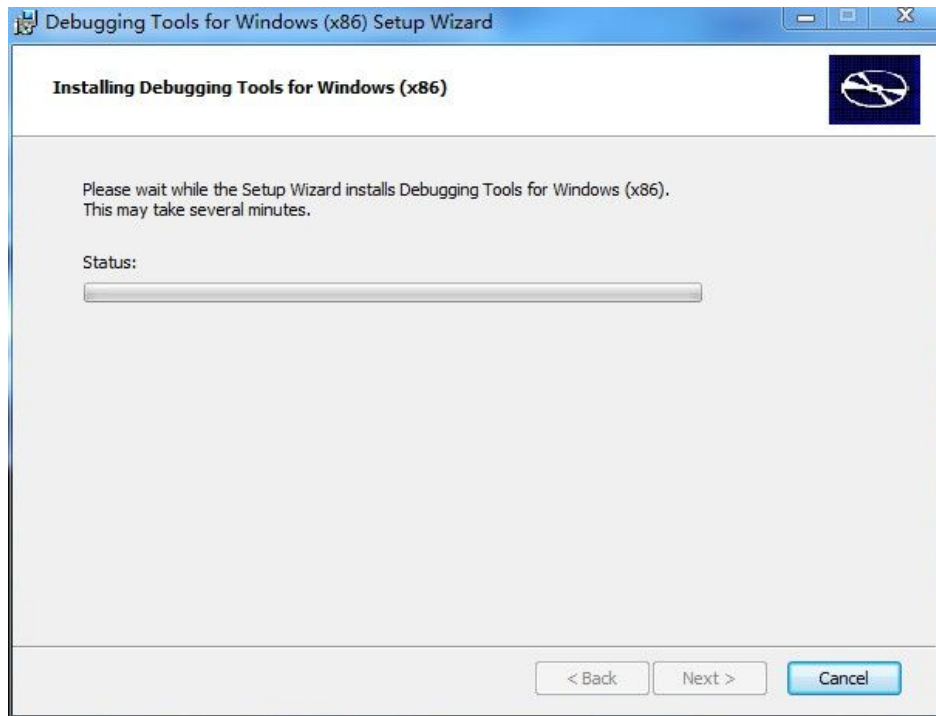
选择 I agree , 然后点击 Next,



点击第一个 Complete，进入下面界面：



点击 Install 按钮，开始安装：



等待安装完成，进入下面界面：

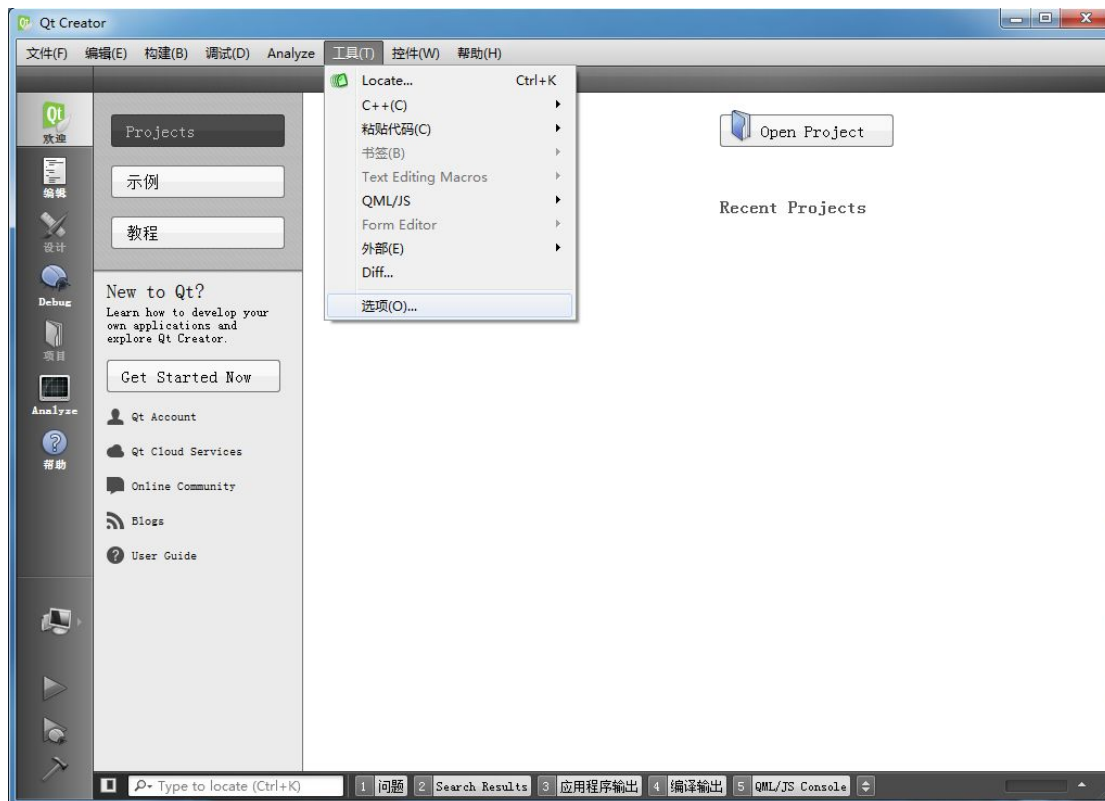


点击 Finish 按钮，完成安装，对于 32 位系统的 WinDbg 调试器，会安装到如下路径：

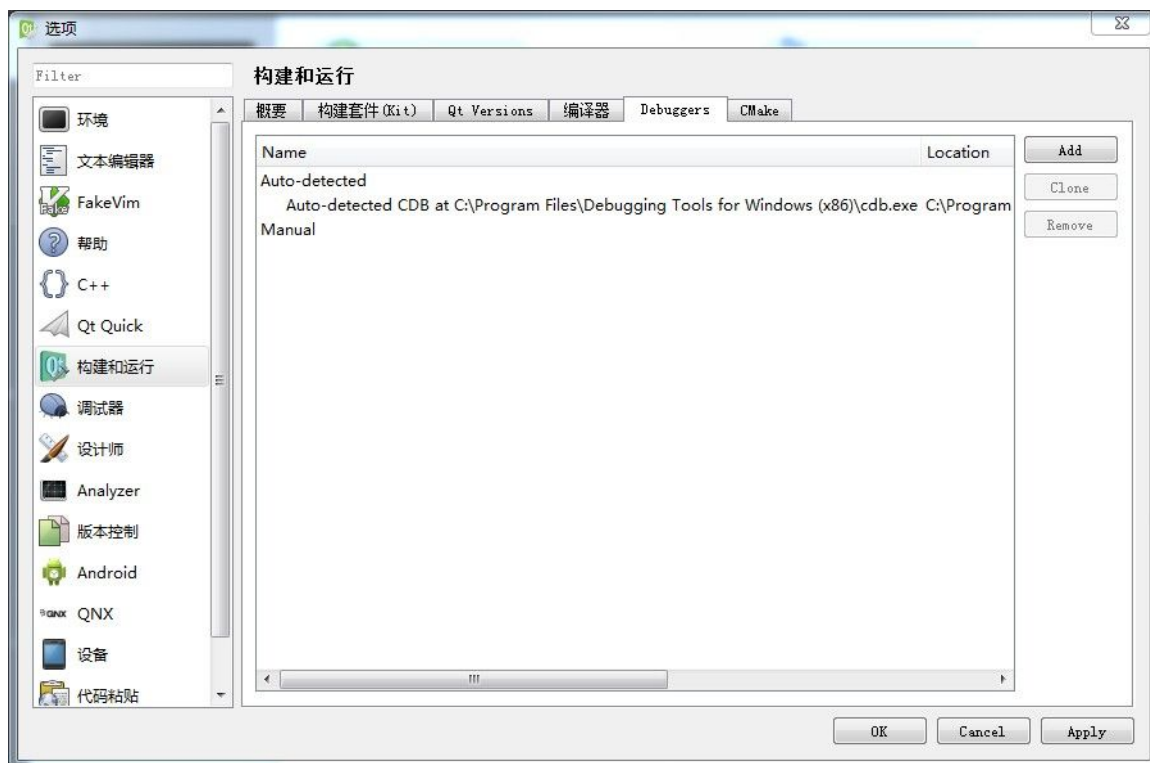
[C:\Program Files\Debugging Tools for Windows \(x86\)](#)

这个文件夹里面既有 windbg.exe 图形化调试器，也有命令行的调试器 cdb.exe，我们等会用到命令行的 cdb.exe 。

下面我们打开集成开发环境 QtCreator（开始菜单-->Qt5.6.0 里面有），

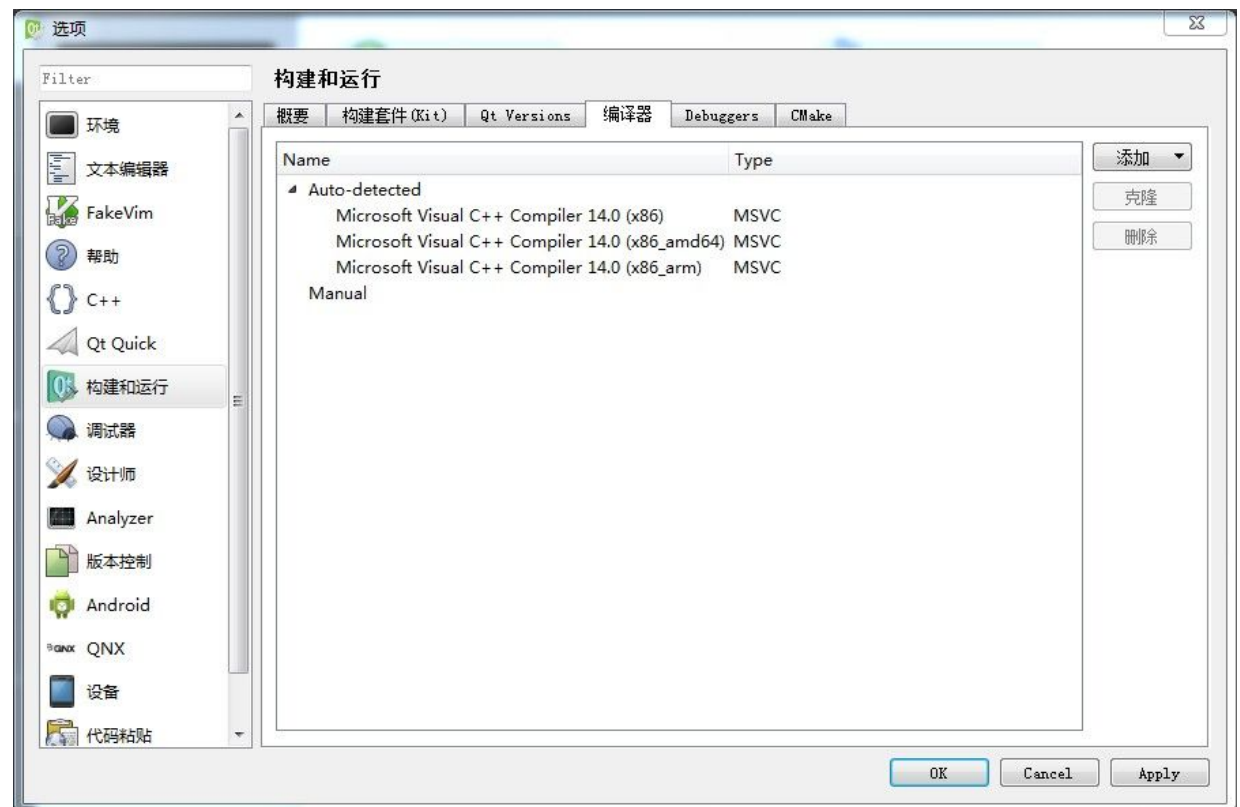


点击上面菜单“工具”-->“选项”，在弹出的选项对话框里，
左边点击“构建和运行”，右边点击“Debuggers”，

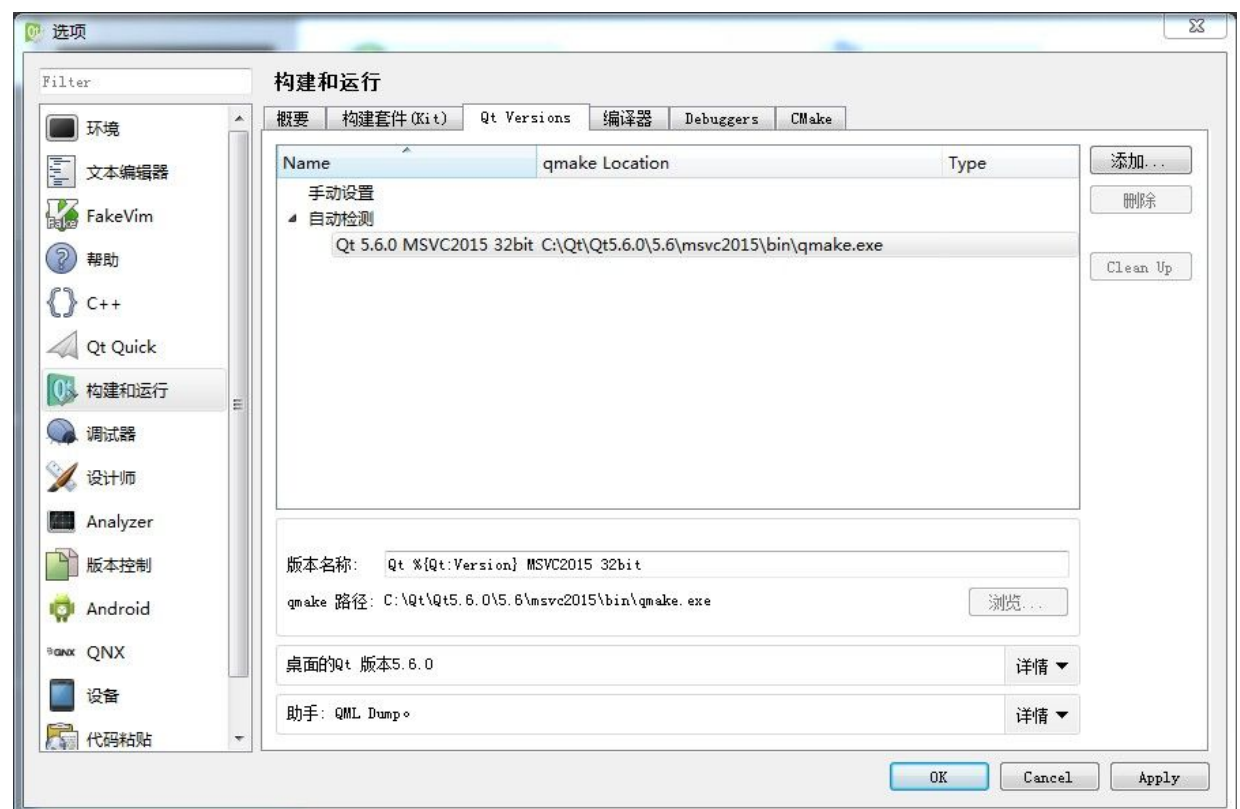


可以看到刚才安装的命令行调试器 cdb.exe 已经自动检测到了，如果没看到“Auto-detected CDB”，那么手动添加一下。

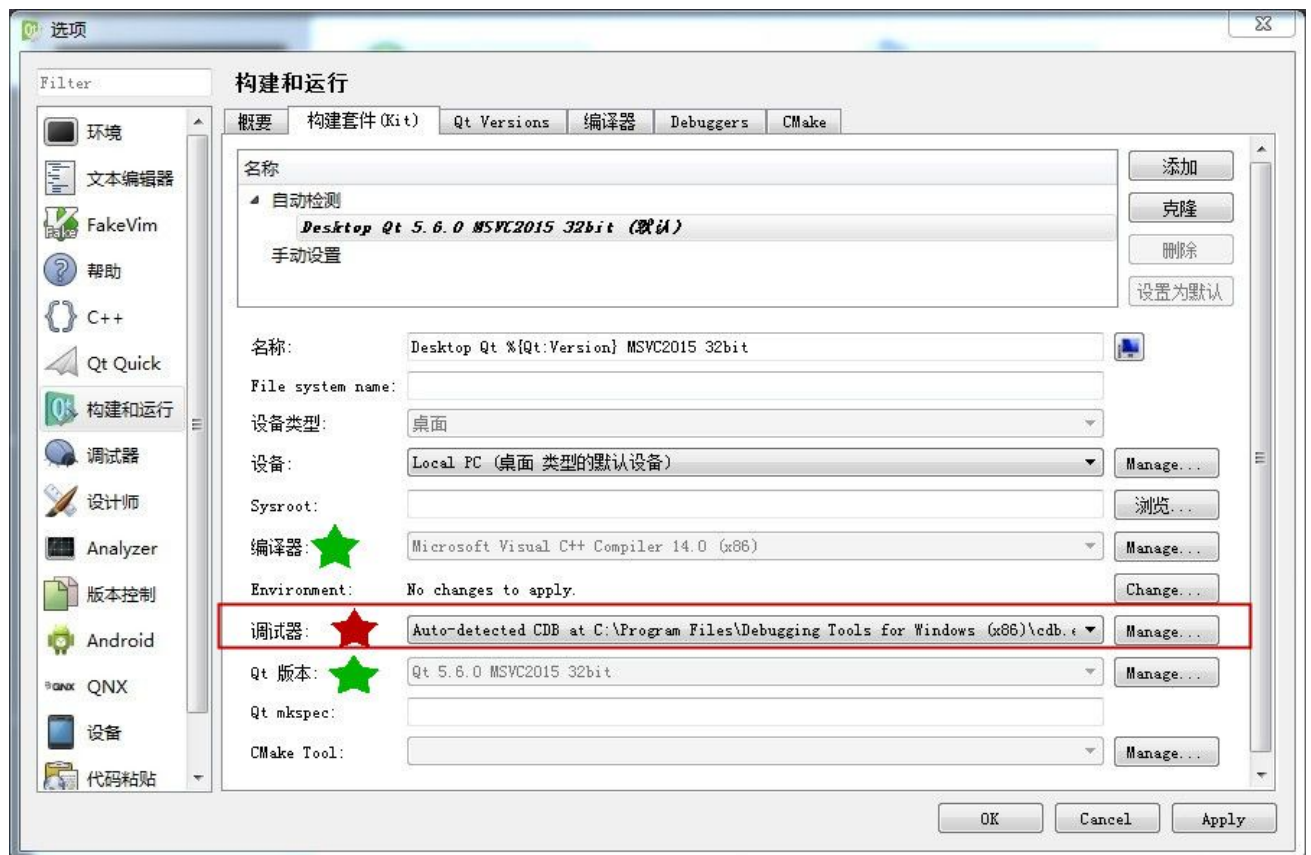
然后我们在上面构建运行配置界面右边，点击“**编译器**”，（链接器不用设置，一般编译器自动调用链接器）



VC++的编译器也是会自动检测到的，有三个，这个也不用自己配置，看看就好了。
如果点击右边顶部的“**Qt Versions**”页面，可以看到自动配置好的 Qt 库路径（不用设置）：



现在调试器、编译器、Qt 开发库都是自动检测好的，剩下一件事需要手动调整：
在上面对话框右边顶部点击 **“构建套件(Kit)”**，
构建套件相当于把指定的编译器（链接器）、调试器、Qt 开发库等全部集合起来，配置成一套 Qt 开发环境：

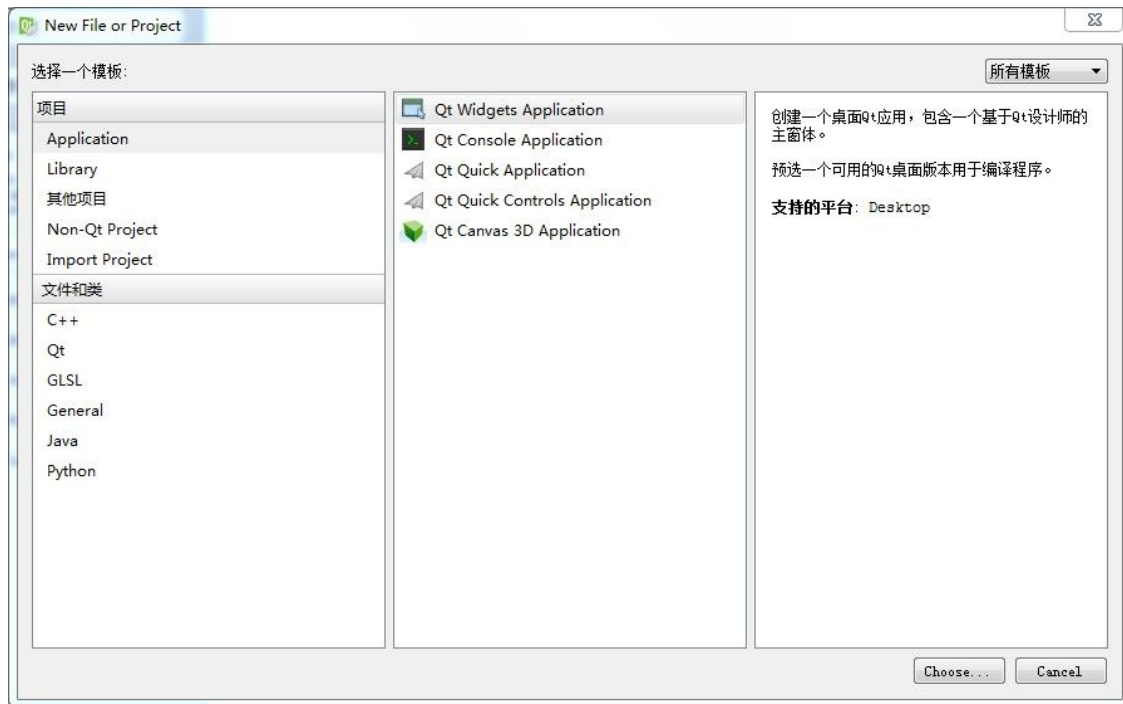


由于调试器是最后安装的，需要在构建套件配置界面选择新安装的调试器，就是刚才看到的 Auto-detected CDB 那个调试器。

然后看到现有的套件 “Desktop Qt 5.6.0 MSVC2015 32bit (默认)” 就处于正常可用的状态，套件名称的左边没有感叹号警告图标了。

选好调试器之后，点击 OK 按钮，配置工作就完成了。

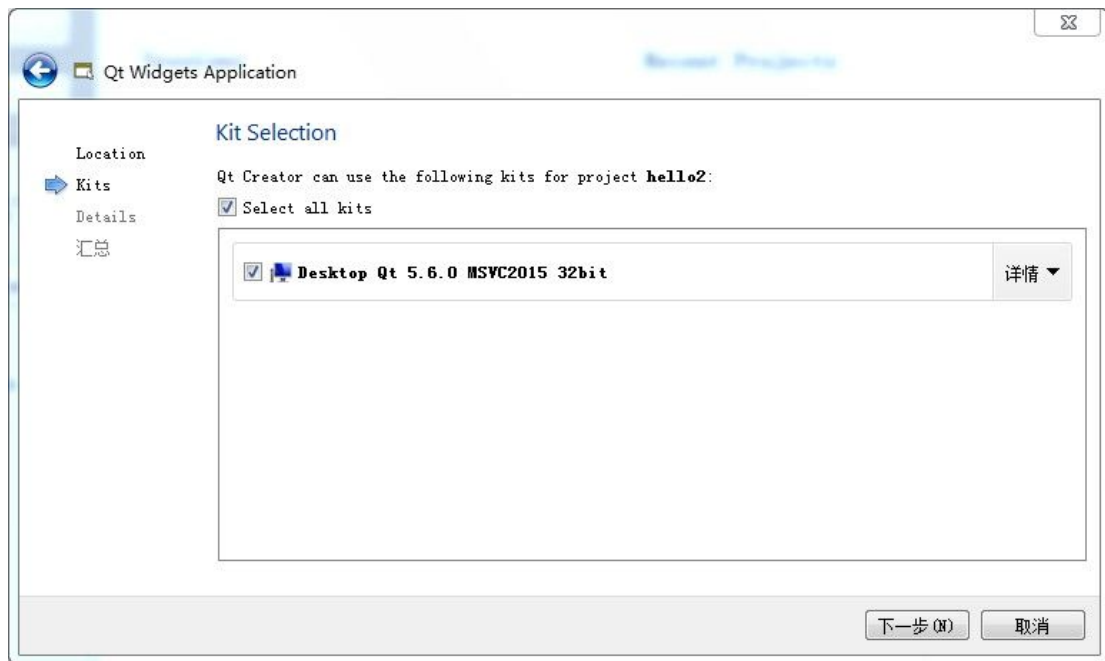
我们点击 QtCreator 菜单 “文件” --> “新建文件或项目”，打开如下对话框：



左边选择默认的 Application，中间选择 Qt Widgets Application，再点击右下角 “Choose...” 按钮，



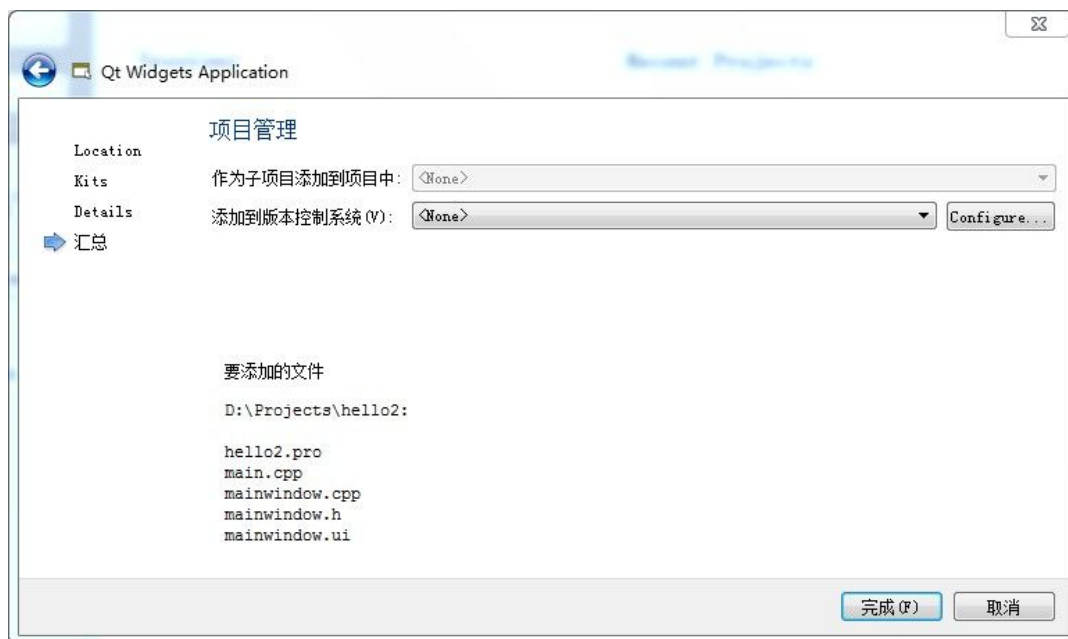
项目名称填 hello2，创建路径填写 D:\Projects ，
注意项目名和路径都必须用英文的，不要用中文名和路径！
点击 “下一步”，



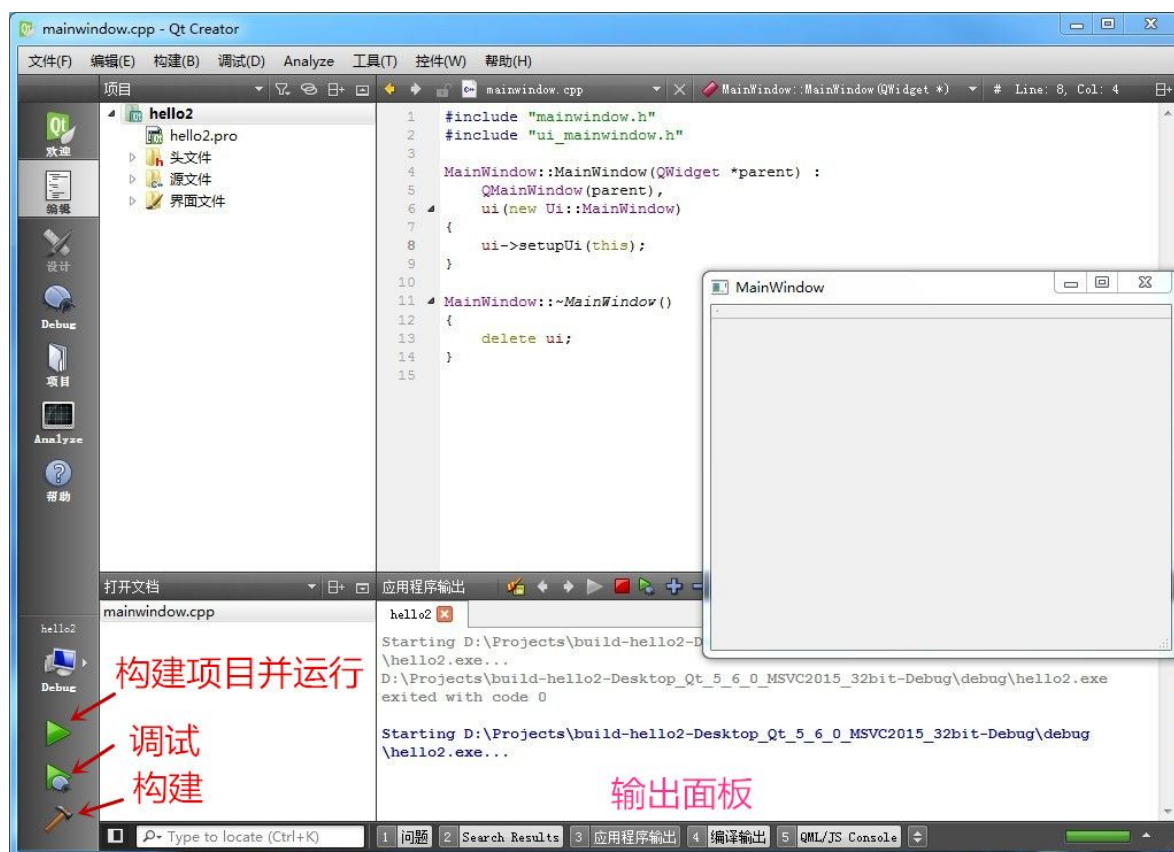
点击“下一步”，



点击下一步，



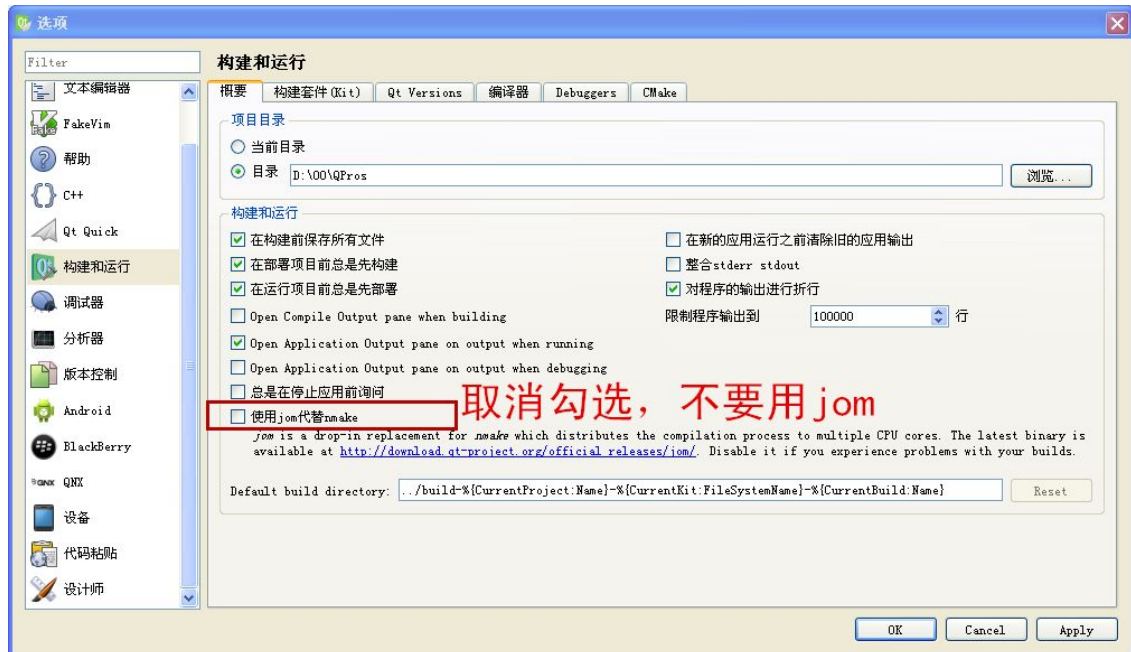
点击“完成”按钮，新建项目后，会进入 QtCreator 代码编辑界面，点击 QtCreator 左下角的绿色三角形按钮，可以生成运行目标程序：



QtCreator 的运行和调试的快捷键是与 Visual Studio 快捷键差不多，F9 打断点，F5 运行调试等待。QtCreator 无论是开发还是生成、调试程序，都比 VS2015 好得多。关于 QtCreator 的使用说明，可以去网上搜集相关书籍，或者看如下网页教程：

<https://lug.ustc.edu.cn/sites/qtguide/>

另外如果遇到操作系统是中文的用户名，导致 QtCreator 编译 Qt 程序失败，那么还是在之前用到的 Qt 选项对话框里面，左边点击“构建和运行”，右边点击“概要”，看到下图界面，取消“使用 jom 代替 nmake”的勾选，然后点击 OK 按钮：



参考资料

1、关于 Qt5Package 扩展包的文章：

<http://my.oschina.net/jthmath/blog/640227>

<http://blog.csdn.net/liuyez123/article/details/50475059>

2、关于 VS2015 安装

[https://msdn.microsoft.com/en-us/library/e2h7fzkw\(v=vs.140\).aspx#bkmk_offline](https://msdn.microsoft.com/en-us/library/e2h7fzkw(v=vs.140).aspx#bkmk_offline)

3、关于 jom 导致无法编译的问题

<http://bbs.csdn.net/topics/390695754>

最后再说一遍，Qt 所有的项目源代码不要放在中文路径！

4、Qt 讨论交流社区

<http://www.qtcn.org>

如果要下载静态版本或其他一堆 Qt 库，可以在社区找到：

<http://www.qtcn.org/bbs/thread-htm-fid-68.html>

5、老外编译的很丰富的各个版本 Qt 库：

<http://www.npcglib.org/~stathis/blog/precompiled-qt4-qt5/>