## 另辟蹊径-利用MSYS2安装MinGW+Qt开发环境

（含32位-64位-动态库-静态库-qwt-opencv等等）

By Winland, 2015.04.04

本文介绍在Windows系统里 MSYS2 环境和 MinGW-w64 编译套件安装、Qt开发环境安装。

一、MSYS2及其安装使用

1、MSYS2介绍

这得从 MinGW 和 MSYS说起，原本 GNU 工具只在 Linux/Unix 系统里才有，随着 Windows 系统的广泛使用， 为了在 Windows 系统里可以使用 GNU 工具，诞生了 MinGW（Minimalist GNU for Windows） 项目，利用 MinGW 就可以生成 Windows 里面的 exe 程序和 dll 链接库。MinGW只是编译工具集没有类似Unix命令行环境，所以为了进一步在Windows系统上使用类Unix环境，MinGW项目组衍生了MSYS子项目，该项目从Cygwin派生，构造了一个类Unix命令行环境，并提供POSIX功能支持。

MinGW本身仅支持生成32位程序，而MinGW-w64项目相当于是MinGW的升级版，同时支持生成32位和64位程序。MSYS2也是MSYS的衍生版，使用MinGW-w64编译工具集。另外MSYS2还有一个非常强大的功能，它从Arch Linux引入了软件包管理器pacman工具，使得MSYS2功能扩展非常方便，并且可以通过pacman来解决各个软件和库之间的依赖关系，自动安装需要的依赖软件包。

MSYS2项目还打包了非常多的开发库和开发工具，能移植到Windows上的估计都打包了，而且还可以进行扩展。因此通过MSYS2的pacman工具安装诸如MinGW-w64编译工具集、Qt开发库和QtCreator、qwt绘图插件、opencv开发库、ffmpeg、gstreamer、openssl、sqlite、postgresql、gtk、crypto++、mesa、openal、tcl/tk、vlc、zlib、SDL等，都是易如反掌。对于python、perl、ruby、lua脚本环境， git、mercurial、cvs等版本控制软件，cmake、clang、llvm等编译生成工具，也可以通过pacman来安装。

MSYS2 主页

<http://msys2.github.io/>

<http://sourceforge.net/projects/msys2/>

MSYS2详细安装指南

<http://sourceforge.net/p/msys2/wiki/MSYS2%20installation/>

2、安装MSYS2

首先从MSYS2项目主页或SourceForge下载MSYS2的基本安装包：

32位系统下载 msys2-i686-20150202.exe，

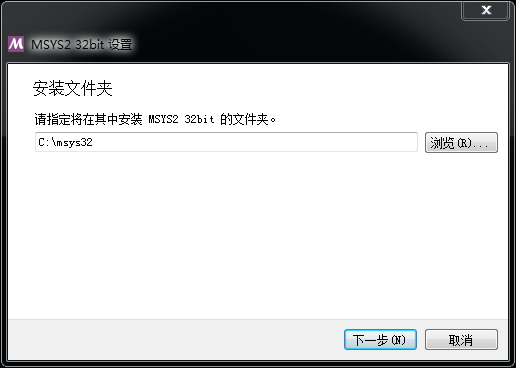
64位系统下载 msys2-x86\_64-20150202.exe。

请记住 i686 对应32位系统，x86\_64对应64位系统，以后装软件都是这种对应关系。

下面以32位的msys2-i686-20150202.exe示范安装过程：

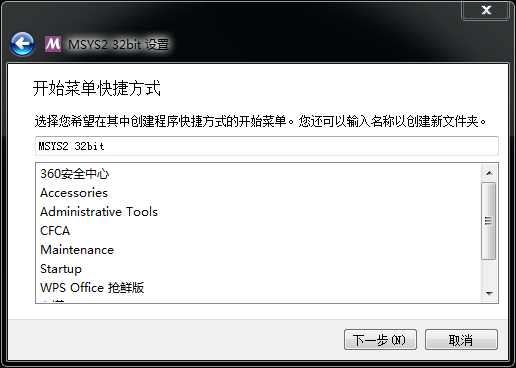


点击“下一步”，



注意上面选择安装位置，路径里不要有任何中文、特殊字符、空格等等，一般装在磁盘根的msys32或msys64目录。**注意磁盘剩余空间要有10GB以上剩余空间（硬盘空间不足的不要模仿）**，那个分区剩余空间比较大就装哪。MSYS2本身不大，但是后续如果装Qt库、opencv之类的，会越装越多，所以要留足空间。Qt动态库安装后是2.7GB，Qt静态库也是差不多2.7GB，非常占空间。

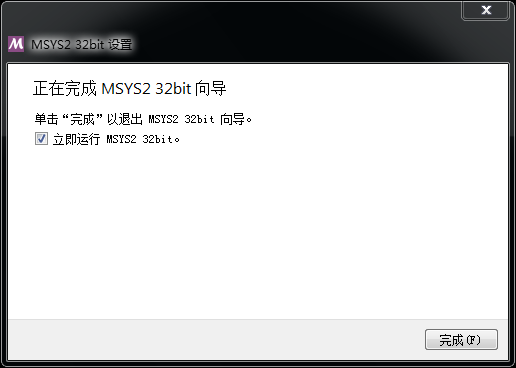
设置好安装路径，点击“下一步”，



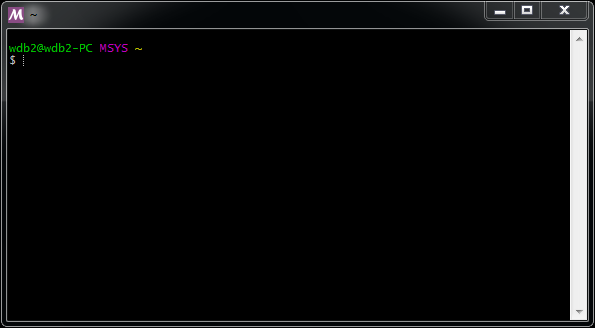
点击下一步，



等待安装结束，

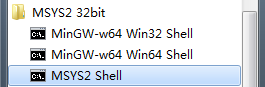


点击完成，看到MSYS2命令行：



先关闭该命令行，现在还干不了什么。

安装后，在开始菜单-->MSYS2 32bit-->



里面有三个命令行工具：

第一个 **MinGW-w64 Win32 Shell**是指32位程序开发环境，在32位和64位Windows里都能用。

第二个 **MinGW-w64 Win64 Shell** 是指64位程序开发环境，仅在64位Windows里使用。

第三个 **MSYS2 Shell**是MSYS2环境管理命令行，可以安装卸载MSYS2环境里软件、管理文件系统、执行脚本等等。

只有在头两个 MinGW-w64的命令行里才能使用gcc、g++等编译工具。**MSYS2 Shell** 一般仅用于软件包安装管理和更新，另外两个Shell才是开发环境，不要弄错了。

3、更新MSYS2系统环境

MSYS2环境的软件和升级都是在线方式的，需要联网下载。所以需要一个可靠的软件源，软件源配置文件在 msys32\etc\pacman.d\ 文件夹里三个：

mirrorlist.mingw32、mirrorlist.mingw64、mirrorlist.msys

更换软件源的话只需要更改三个文件里的**Server =** 右边的服务器地址，用写字板或其他编辑器打开编辑就行了，里面都是文本（注意备份旧的文件）。

软件源可以使用默认的SourceForge官方地址，也可以自定义，比如使用爱尔兰的源

①mirrorlist.mingw32中使用

ftp://ftp.heanet.ie/mirrors/download.sourceforge.net/pub/sourceforge/m/ms/msys2/REPOS/MINGW/i686

②mirrorlist.mingw64中使用

ftp://ftp.heanet.ie/mirrors/download.sourceforge.net/pub/sourceforge/m/ms/msys2/REPOS/MINGW/x86\_64

③mirrorlist.msys中使用

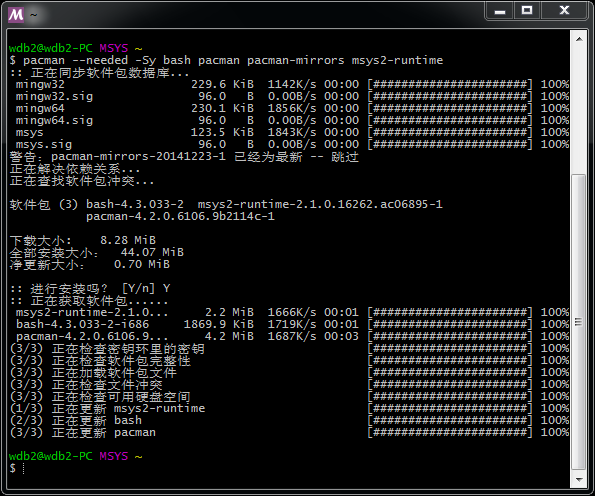
ftp://ftp.heanet.ie/mirrors/download.sourceforge.net/pub/sourceforge/m/ms/msys2/REPOS/MSYS2/$arch

接下来打开开始菜单里的**MSYS2 Shell**进行MSYS2系统更新：

（1）更新软件数据库

**pacman --needed -Sy bash pacman pacman-mirrors msys2-runtime**

有询问就输入 Y，继续执行。

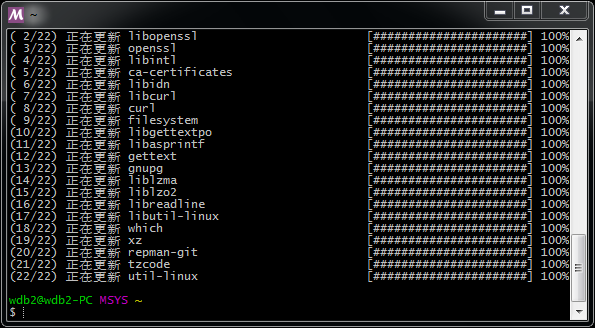


**如果出现下载软件包错误，那就重复执行刚才的命令，直到成功为止。**

（2）关闭旧的MSYS2命令行，重新打开**MSYS2 Shell，**更新软件

**pacman -Su**

有询问就输入 Y，继续安装



**如果出现下载软件包错误，那就重复执行刚才的命令，直到成功为止。**

更新成功之后，关闭MSYS2命令行，完成MSYS2系统更新。

二、安装MinGW-w64编译套件

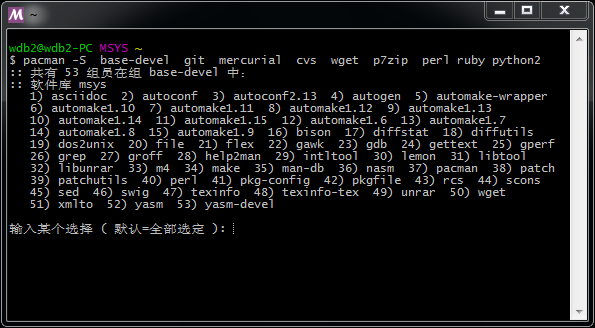
默认情况下MSYS2只安装了基本环境，开发工具集没装，需要手动来安装，下面来安装。

打开开始菜单的 **MSYS2 Shell**，执行下面命令：

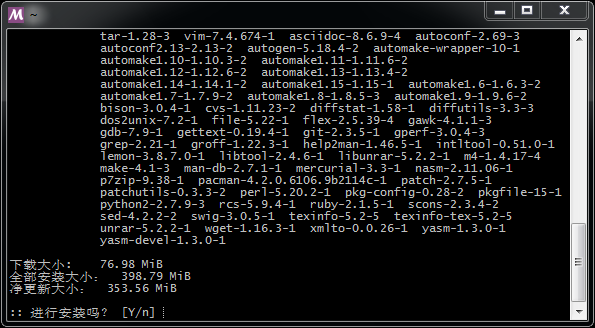
1、安装基本开发环境和工具

**pacman -S base-devel git mercurial cvs wget p7zip perl ruby python2**

说明pacman -S 是安装软件包的意思，base-devel 是基本开发工具和库，git mercurial cvs 三个是版本控制软件，可用于下载和发布项目源代码， wget 是下载工具，p7zip是解压缩工具，perl ruby python2三个是脚本编程环境。



遇到“输入某个选择”，不打字，直接按Enter键，全部安装。



输入 Y，进行安装。

如果出现下载软件错误，就重复执行刚才的命令，直到成功为止。

安装成功后，继续下面小节的安装。

2、安装MinGW-w64

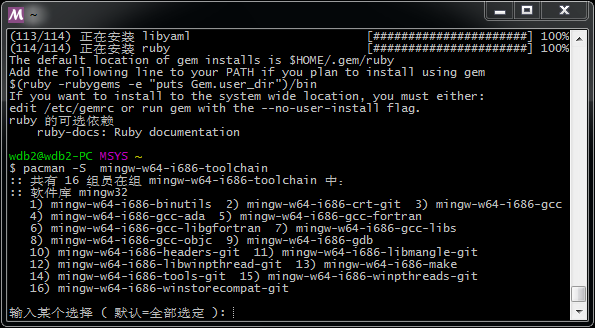
对于32位系统，执行：

**pacman -S mingw-w64-i686-toolchain**

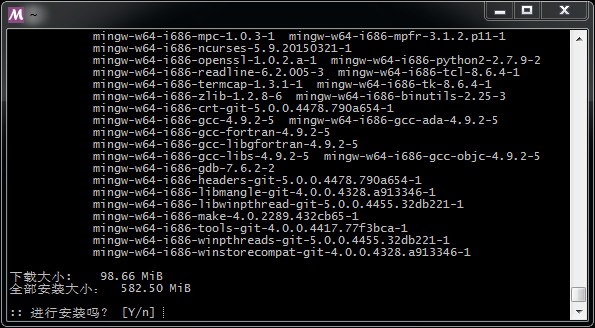
对于64位系统，执行：

**pacman -S mingw-w64-x86\_64-toolchain**

注：32位系统只能编译运行32位程序，64位系统既可以编译运行64位程序，也可以编译运行32位程序。如果希望在64位系统同时编译32位和64位系统，可以把上面两个工具链都安装。一般情况下只需要一个。



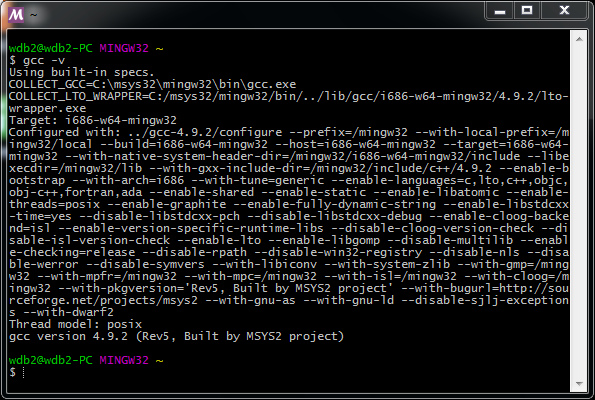
遇到“输入某个选择”，直接按Enter键，全部安装：



输入 Y，进行安装。

如果出现下载错误，就重复执行刚才的命令，直到成功为止。

安装成功后关闭旧的命令行。在开始菜单里打开 **MinGW-w64 Win32 Shell** （64位系统是MinGW-w64 Win64 Shell），输入 gcc -v 就可以查看是否安装好了MinGW-w64编译工具集：



上面安装了MinGW-w64工具集和基本的开发库，还有perl、python、ruby等脚本编程环境。接下来我们来安装Qt开发环境。

三、安装Qt开发环境

Qt开发环境默认情况下都是使用动态链接库的，我们先安装动态库版本Qt和集成开发环境QtCreator，然后安装静态Qt库。

**注意：动态Qt库可以用于遵循LGPL的商业闭源软件，也可以用于GPL开源软件，而静态Qt库只能用于GPL开源软件，这个规则要记牢！**

1、安装动态Qt库和QtCreator

打开开始菜单里的**MSYS2 Shell，**

32位系统执行：

**pacman -S mingw-w64-i686-qt5 mingw-w64-i686-qt-creator**

64位系统执行：

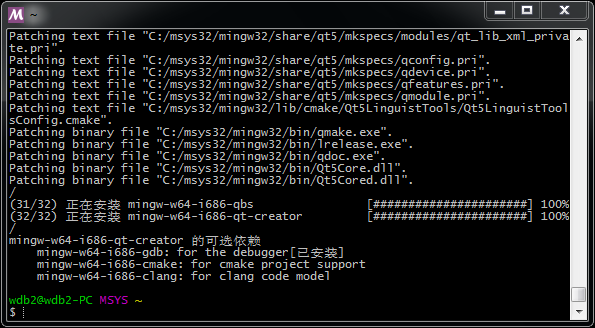
**pacman -S mingw-w64-x86\_64-qt5 mingw-w64-x86\_64-qt-creator**

遇到询问就输入 Y 进行安装。

Qt开发库下载时是500MB左右，比较费时间，需要等待，如果下载过程中出现错误之类的，重复执行刚才的命令，直到成功安装为止。

软件管理工具pacman自动使用断点续传，不会重复下载，所以可放心地重复执行命令直到成功安装。

安装Qt时，有一个软件包 mingw-w64-i686-fontconfig ，安装配置得非常慢，要耐心等它安装好，不要以为pacman坏了或什么的，这个包配置就是慢；还有就是 mingw-w64-i686-qt5 这个包本身非常大，安装起来也比较慢，需要耐心等待。



安装好了之后，需要在 **MinGW-w64 Win32 Shell** （64位系统是MinGW-w64 Win64 Shell）启动Qt开发工具，比如：

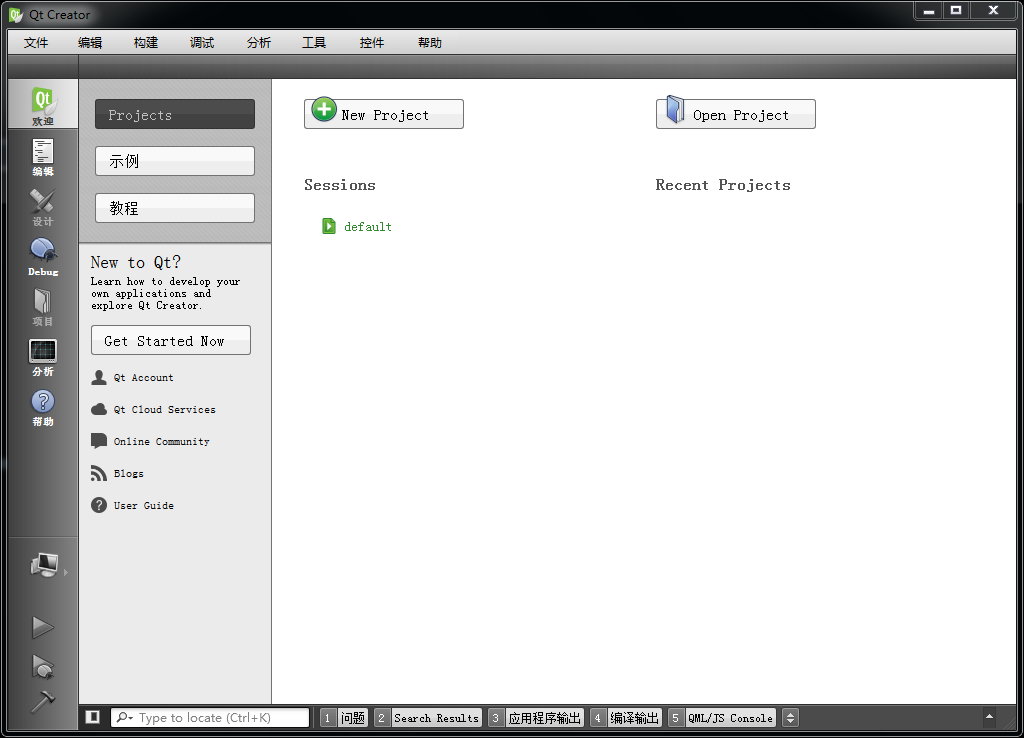
qtcreator &

assistant &

designer &

linguist &

说明：qtcreator是集成开发环境的命令名称，后面接 空格 和 & 是指后台启动新进程，而不需要占用 Shell命令前台。QtCreator 启用时需要加载插件需要些时间，启动之后如下图所示：



这里就不示范动态库的HelloWorld编译生成了。

就算是最简单的Qt版Hellworld，它在Windows系统里依赖的动态链接库也是很多的，可能有十几个，所以用于发布程序很费劲。下面介绍Qt静态库安装和示范。

2、Qt静态库安装

打开开始菜单里的**MSYS2 Shell，**

32位系统执行：

**pacman -S mingw-w64-i686-qt5-static**

64位系统执行：

**pacman -S mingw-w64-x86\_64-qt5-static**

遇到询问就输入 Y 进行安装。

静态Qt库下载时大小也是500MB左右，安装后也是2.7GB左右，下载和安装都很费时间。如果下载过程中出错，就重复执行上面命令直到成功为止。

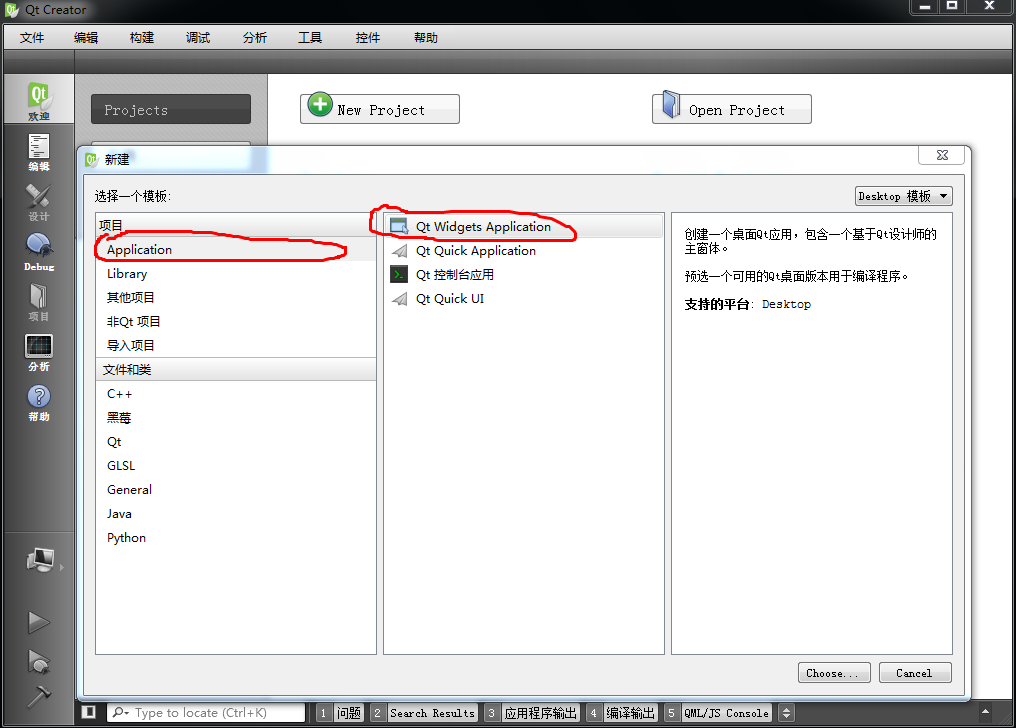
上面安装的是完整的静态Qt库，包括Debug调试版和Release发行版，不建议使用静态库生成调试版程序，因为一个Helolworld都有280MB，静态库生成的Release版Helloworld只有11MB，并且不依赖额外的dll，一个exe走遍天下，所以适合发布程序。但是对于静态库，必须用GPL开源许可证发布程序。LGPL仅适用于动态链接库。

3、QtCreator生成静态链接Qt程序示范

打开**MinGW-w64 Win32 Shell** （64位系统是MinGW-w64 Win64 Shell），执行命令：

**qtcreator &**

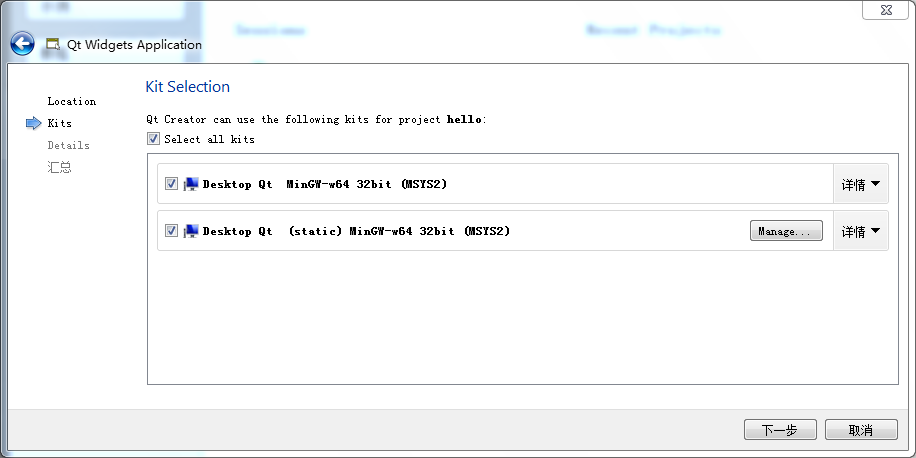
在QtCreator点击菜单“文件”-->“新建文件或项目”，



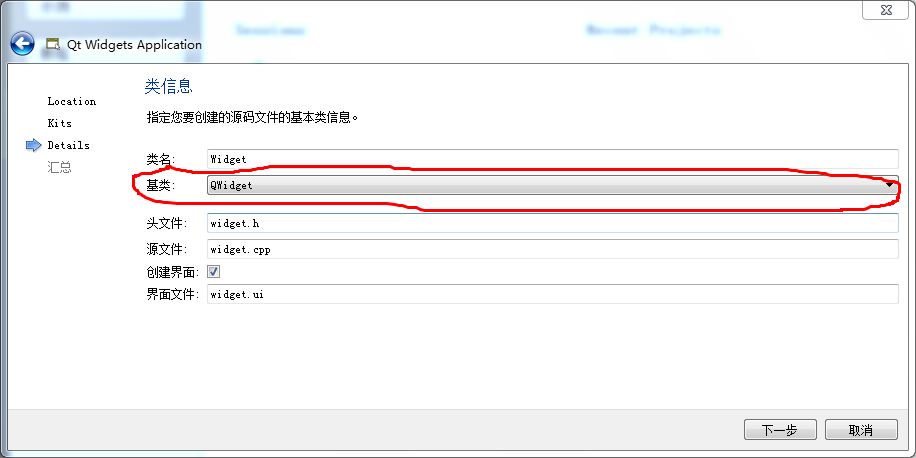
新建 Qt Widgets Application，



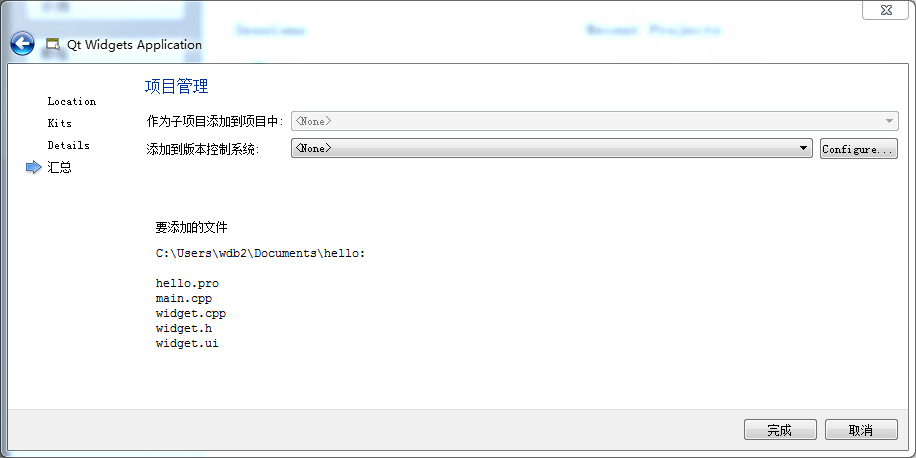
项目名称设为hello，其他的不用变，点“下一步”：



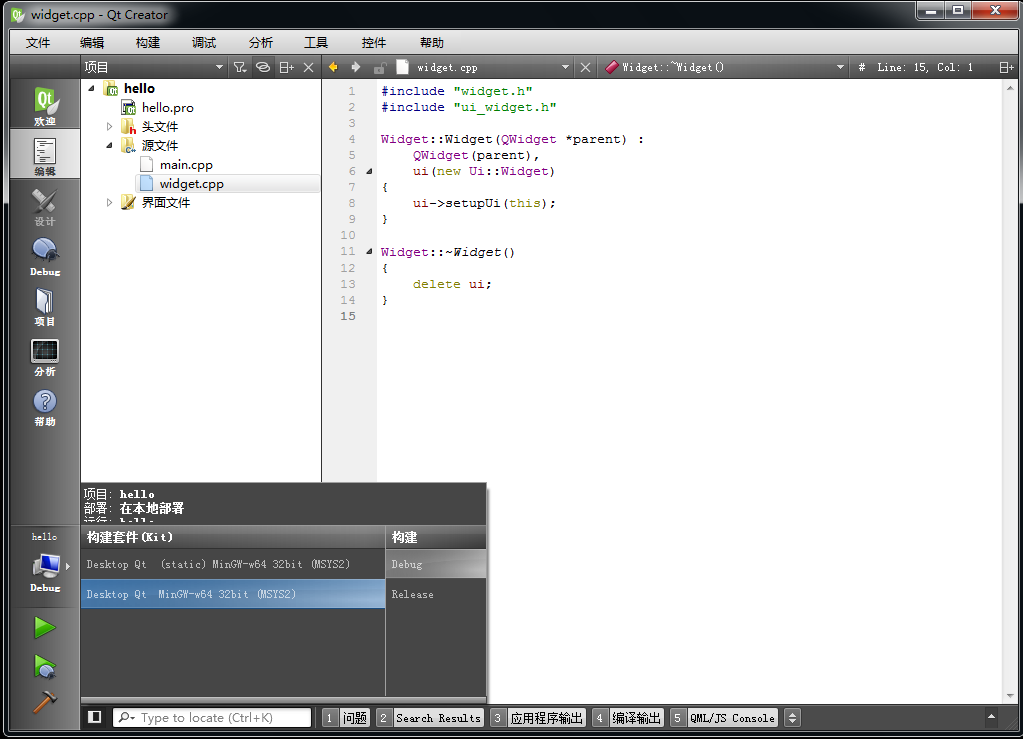
Kit Selection是选择Qt套件的意思，前面装了静态和动态版本，所以有两个，都选上，进入下一步：



基类选择QWidget，进入下一步：



点击完成，进入项目代码编辑界面：



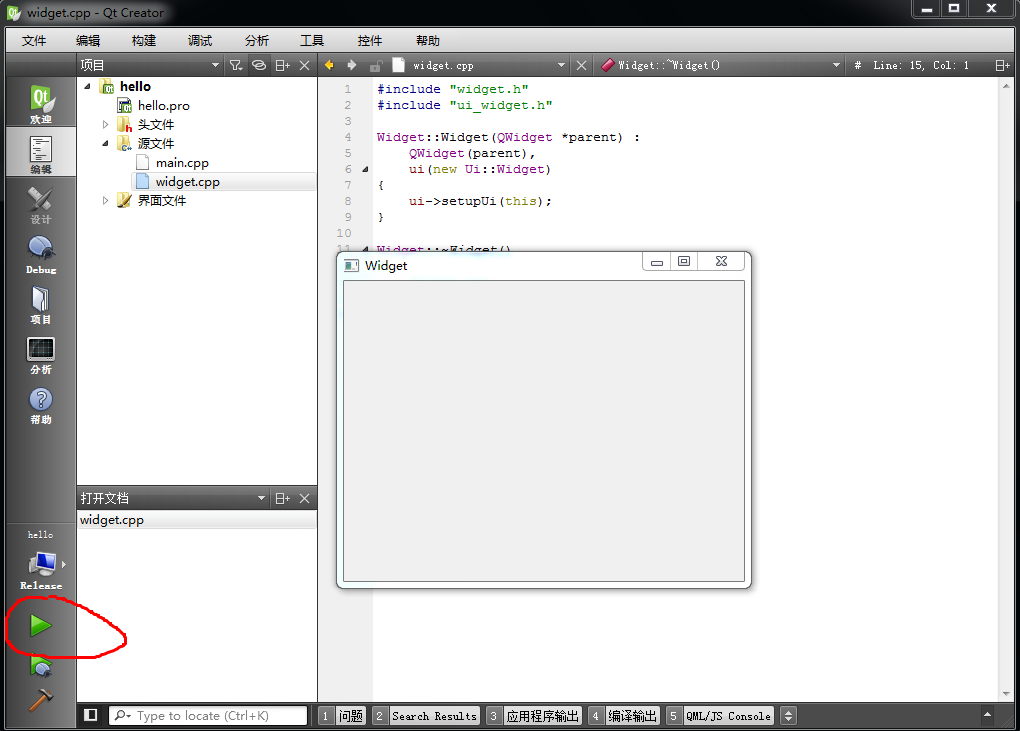
将左下角Debug字样的按钮点开，可以看到两个Qt套件，带有 (static) 的是静态库，没有static字样的是动态库。

一般动态库配合Debug和Release构建模式使用；

静态库配合Release构建模式使用，尽量不要使用静态库生成Debug版程序，因为太大了。

我们选择静态库套件的Release模式构建，然后点击左下角“Debug”字样下面的三角图标按钮，

构建并运行成功程序。



Win7系统默认在

C:\Users\用户名\Documents\build-hello-Desktop\_Qt\_static\_MinGW\_w64\_32bit\_MSYS2-Release\release

文件夹里可以找到 hello.exe ，11.5MB大小，这个可比动态库编译的发布省事多了。

（尽量不要用静态库生成Debug版程序，不仅生成慢，而且巨无霸，我试了一下是270MB。）

注意：对于Qt静态库生成的QtQuick应用程序，运行时有bug（动态链接库没这个问题）：

QQmlApplicationEngine failed to load component

<qrc:/main.qml:2> module "QtQuick.Controls" is not installed

<qrc:/main.qml:1> module "QtQuick" is not installed

<qrc:/main.qml:2> module "QtQuick.Controls" is not installed

<qrc:/main.qml:1> module "QtQuick" is not installed

这是Qt库的bug，不是MSYS2项目能改变的。有临时的解决方法，参考：

<http://tieba.baidu.com/p/3537364638>

详情请见帖子的从16楼到21楼。

四、其他扩展开发库安装

MSYS2项目里的软件和开发库远不止上面那些，之前提到了一大堆，下面首先介绍pacman通用的搜索安装方法，示范一下 qwt 绘图插件和opencv的安装。

1、通用的软件搜索安装方法（以qwt为例）

打开开始菜单里的**MSYS2 Shell，**

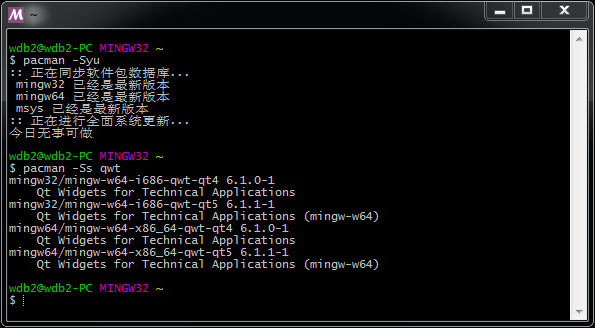
（1）一般首先更新软件仓库和升级系统，这样保证能获知新发布的软件包名称

**pacman -Syu**

（2）检索想要的软件包名称

**pacman -Ss qwt**

说明pacman 的-Ss选项就是查询远程软件仓库里的软件包名，qwt就是我们想装的软件名称。



可以看到有四个顶头没缩进的行，mingw32是软件类别，代表32位软件，里面有 qt4的和qt5的两个版本qwt，最末尾的数字 6.1.0-1 是qwt软件版本。

有4个字符缩进的行是上一行软件包的描述。

Mingw64也是软件类别，代表64位软件，里面也有qt4和qt5两个版本的qwt，末尾的数字 6.1.0-1 也是qwt的版本。

（3）安装所需的软件包

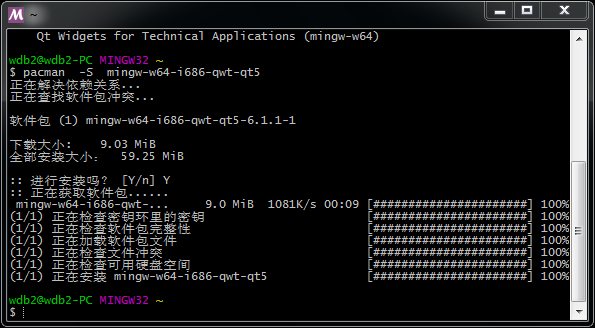
安装时只是需要软件包名，不需要输入软件类别和最末尾的数字版本号，安装32位qt5的qwt命令就是：

**pacman -S mingw-w64-i686-qwt-qt5**

如果安装64位qt5的qwt命令就是：

**pacman -S mingw-w64-x86\_64-qwt-qt5**

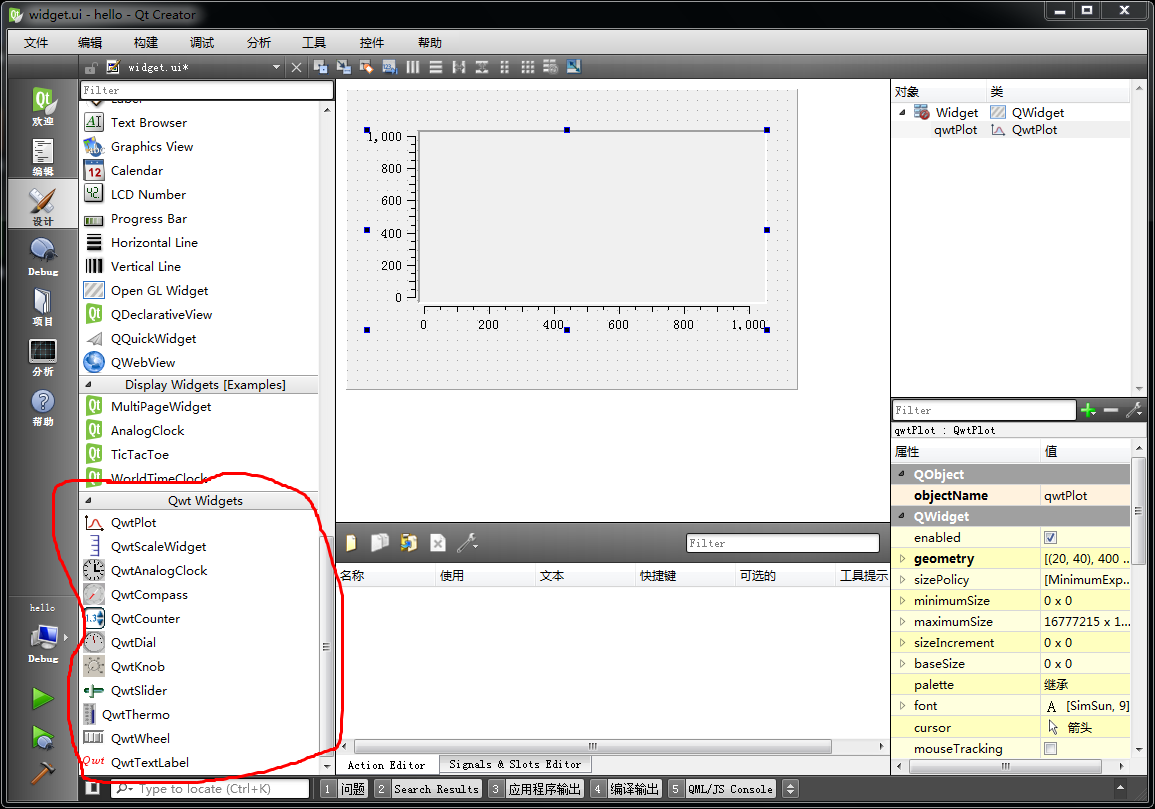
安装时遇到提示是否安装就输入 Y 执行安装。



然后我们查看 qwt 安装后的效果，需要打开开始菜单里的开发命令行**MinGW-w64 Win32 Shell** （64位系统是MinGW-w64 Win64 Shell），执行命令：

qtcreator &

对于QtCreator，打开一个普通窗体项目，打开界面文件 \*.ui，进入设计界面，把左边的控件（或叫部件）列表拖到最底下：



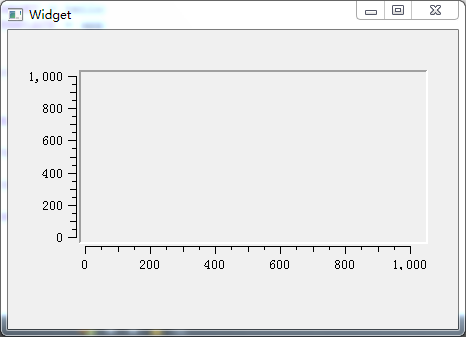
看到可用的QWT控件，拖到界面窗体里面用，并且要在项目文件 .pro 里面加两行：

**CONFIG+=qwt**

**INCLUDEPATH+=/msys32/mingw32/include/qwt**

64位系统把32改成64就行了。

**另外注意 QWT 只有动态库版本，得按照动态库方式发布。**带有QWT控件的程序运行后类似下面这个：

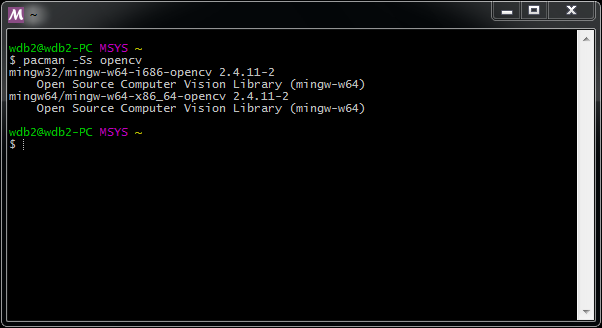


2、安装opencv

与上面过程类似，因为刚升级过系统就不需要执行 pacman -Syu ，直接两步就行了，打开开始菜单 **MSYS2 Shell，**

（1）检索 opencv

**pacman -Ss opencv**



（2）安装

32位系统安装i686的软件包：

**pacman -S mingw-w64-i686-opencv**

64位系统安装 x86\_64 的软件包：

**pacman -S mingw-w64-x86\_64-opencv**

装好就可以使用了。

如果要在Qt程序里使用，那么需要在 .pro 文件里写上

LIBS+=-lopencv\_calib3d\

-lopencv\_contrib\

-lopencv\_core\

-lopencv\_features2d\

-lopencv\_flann\

-lopencv\_gpu\

-lopencv\_highgui\

-lopencv\_imgproc\

-lopencv\_legacy\

-lopencv\_ml\

-lopencv\_nonfree\

-lopencv\_objdetect\

-lopencv\_ocl\

-lopencv\_photo\

-lopencv\_stitching\

-lopencv\_superres\

-lopencv\_ts\

-lopencv\_video\

-lopencv\_videostab\

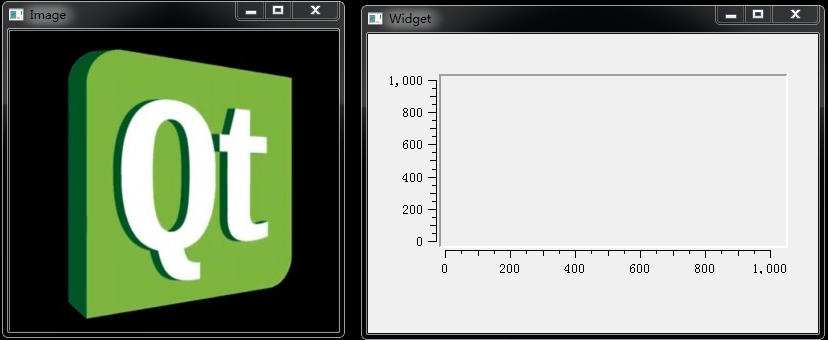
-lopencv\_viz

上面链接了opencv的20个模块，一般情况下项目代码里用了几个链接几个，不需要像上面全部链接。上面就是穷举了opencv的所有模块，并且都链接了。

源代码里引用opencv的话，可以如下所示

|  |
| --- |
| #include"widget.h"  #include<QApplication>  **#include<opencv/cv.h>**  **#include<opencv2/opencv.hpp>**  **#include<opencv/cxcore.h>**  **#include<opencv/highgui.h>**  intmain(intargc,char\*argv[])  {  QApplicationa(argc,argv);  Widgetw;  w.show();  //  //声明IplImage指针  IplImage\*pImg;  //载入图片  pImg=cvLoadImage("/1.jpg",1);  //创建窗口  cvNamedWindow("Image",1);  //显示图像  cvShowImage("Image",pImg);  //等待按键  cvWaitKey(0);  //销毁窗口  cvDestroyWindow("Image");  //释放图像  cvReleaseImage(&pImg);  //  returna.exec();  } |

包含头文件的opencv文件夹和opencv2文件夹在MSYS2系统包含路径里（32位系统是 /msys32/mingw32/include/），所以可以直接用尖括号 <> 引用，QtCreator会自动补全头文件路径和名称的。上面会弹出两个窗口，一个是Qt的主窗口，一个是opencv显示图像的窗口，两个都正常工作：



左边是opencv显示图片的窗口（图片文件1.jpg在磁盘根目录），右边是上面带有qwt控件的Qt窗口。

**注意opencv也是只有动态库版本的，所以opencv+Qt的程序必须按照动态库来发布。**

示范就到这里，其他的如ffmpeg、openssl之类的也可以类似安装。

对于动态库发布Qt程序，可以借助Dependency Walker工具打开 exe 文件，查看依赖哪些dll，然后从 msys32\mingw32\bin 文件夹里找对应的dll复制到 exe一块发布。或者直接运行exe，看它提示要哪些dll，一个个复制。

<http://www.dependencywalker.com/>

不过动态库发布方式有点坑爹，上面带有opencv和qwt的发行版hello.exe ，需要的dll数目是34个，共计58MB：

|  |
| --- |
| \build-hello-Desktop\_Qt\_MinGW\_w64\_32bit\_MSYS2-Release\release 的目录  2015/04/03 21:32 33,280 hello.exe  2015/02/14 05:56 76,571 libbz2-1.dll  2015/02/15 06:18 664,653 libfreetype-6.dll  2015/03/22 16:34 119,476 libgcc\_s\_dw2-1.dll  2015/03/24 20:56 1,372,607 libglib-2.0-0.dll  2014/08/31 22:23 287,521 libHalf-2\_2.dll  2015/03/22 19:14 358,127 libharfbuzz-0.dll  2015/02/14 06:37 1,067,193 libiconv-2.dll  2014/11/11 00:32 25,352,575 libicudt54.dll  2014/11/11 00:32 2,304,620 libicuin54.dll  2014/11/11 00:32 1,487,727 libicuuc54.dll  2014/08/31 22:23 48,527 libIex-2\_2.dll  2014/09/01 20:32 2,791,882 libIlmImf-2\_2.dll  2014/08/31 22:23 32,660 libIlmThread-2\_2.dll  2014/08/31 22:23 70,787 libImath-2\_2.dll  2015/02/15 18:33 139,177 libintl-8.dll  2015/01/30 05:40 362,433 libjasper-1.dll  2014/08/25 16:58 66,202 libjbig-0.dll  2015/01/18 21:45 436,206 libjpeg-8.dll  2015/02/28 06:34 173,896 liblzma-5.dll  2015/03/06 16:30 2,038,307 libopencv\_core2411.dll  2015/03/06 16:30 397,639 libopencv\_highgui2411.dll  2014/09/27 16:06 254,165 libpcre16-0.dll  2015/03/29 18:57 249,293 libpng16-16.dll  2015/03/22 16:34 1,031,444 libstdc++-6.dll  2014/12/06 17:30 495,609 libtiff-5.dll  2015/02/28 02:47 65,949 libwinpthread-1.dll  2015/04/03 18:44 4,358,144 Qt5Core.dll  2015/02/26 13:56 4,780,544 Qt5Gui.dll  2015/02/26 13:57 351,232 Qt5PrintSupport.dll  2015/02/26 13:57 328,192 Qt5Svg.dll  2015/02/26 13:56 6,470,656 Qt5Widgets.dll  2015/03/29 18:18 1,116,672 qwt.dll  2015/03/05 15:03 2,423,207 tbb.dll  2015/02/14 06:44 94,500 zlib1.dll  35 个文件 61,701,673 字节 |

如果发布到其他计算机里使用，另外还需要 plugins/platforms 里面的qwindows.dll和qminimal.dll，如果用了图片或数据库之类的Qt插件模块，还得复制这些插件，算起来至少是36个dll了。

动态库也就只能这样了。纯Qt5的程序比较好办，静态库链接，一个exe大概十几兆就够了。

附录一：MSYS2 软件源链接

MSYS2 主页  
<http://msys2.github.io/>  
<http://sourceforge.net/projects/msys2/>  
  
MSYS2详细安装指南  
<http://sourceforge.net/p/msys2/wiki/MSYS2%20installation/>   
安装Qt

<http://qt-project.org/wiki/MinGW-64-bit>   
  
国际镜像  
大爱尔兰帝国有几个速度不错，东西也全：  
<ftp://ftp.heanet.ie/mirrors/download.sourceforge.net/pub/sourceforge/m/ms/msys2/>  
<http://ftp3.ie.freebsd.org/pub/download.sourceforge.net/pub/sourceforge/m/ms/msys2/>  
  
英国肯特大学，速度很快，MSYS2的东西可能不全  
<http://www.mirrorservice.org/sites/download.sourceforge.net/pub/sourceforge/m/ms/msys2/>  
  
日本北陸先端科学技術大学院大学  
<ftp://ftp.jaist.ac.jp/pub/sourceforge/m/ms/msys2/>  
  
国内镜像  
北京理工大学  
<http://mirror.bit.edu.cn/msys2/>  
  
中国科学技术大学  
<http://mirrors.ustc.edu.cn/msys2/>

附录二：Pacman使用教程链接

简易教程

<http://wiki.51osos.com/wiki/Pacman>

较完备的教程

<https://wiki.archlinux.org/index.php/Pacman_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)>

目前MSYS中的 QtCreator版本：  
qt-creator 4.1.0 based on Qt 5.6.1  
在Win7的32位系统中运行失败，出现 Segmentation fault 错误，  
而其他Qt工具如 assistant、designer等工具都没有问题。  
mateczek找到了问题所在：  
[https://forum.qt.io/topic/71150/qt-creator-msys2-segmentaion-fault](http://jump.bdimg.com/safecheck/index?url=rN3wPs8te/rA4bR1s4XyYNElSb1leOYYqzSWM2tbNXR7YRzmLt7jeiM94UN+TOrmBOQM3ag8bfPEzqJnJN/ezrEDTJ73h+JoYT6xDfpVPNq/HFZDEuGFZxwkX11w4M7n4r+WxMrgWfud3lT5q2A3cMkB/x3JZmqPJjOeTV6QVKNdWmYshQ/elUvG76QD8+mrVmYQFUeMaCgwPGbuJnYGNA==)  
就是QtCreator的插件 QmlProfiler 出错。  
  
解决方法：  
第一步，在 mingw32 命令行执行命令：  
**qtcreator -noload all -load Core**  
这样QtCreator不加载任何插件，可以正常启动。  
第二步，打开的QtCreator里面菜单 “帮助”--》“关于插件”，  
**然后在插件选中对话框里，把 QmlProfiler 一行的勾去掉，不选中：**  
  
然后关闭qtcreator。  
以后正常启动 qtcreator 就行了，也不要再用什么QmlProfiler ，这是坑爹插件。

