

# 核心网络

- [全部分类](#)
- [移动开发](#)
- [Web前端](#)
- [架构设计](#)
- [编程语言](#)
- [互联网](#)
- [数据库](#)
- [系统运维](#)
- [云计算](#)
- [研发管理](#)
- [综合](#)

OpenCV - win7+vs2013(2012)+opencv3.0.0 环境配置（以及配置技巧）

## OpenCV - win7+vs2013(2012)+opencv3.0.0 环境配置（以及配置技巧）

分类：[OpenCV](#) | 2015-01-09 17:20 | 34475人阅读

1. opencv 3.0.0 库下载地址，这里的版本是3.0.0，其他的版本配置可能不一样，请大家注意。

<http://jaist.dl.sourceforge.net/project/opencvlibrary/opencv-win/3.0.0-alpha/opencv-3.0.0-alpha.exe>

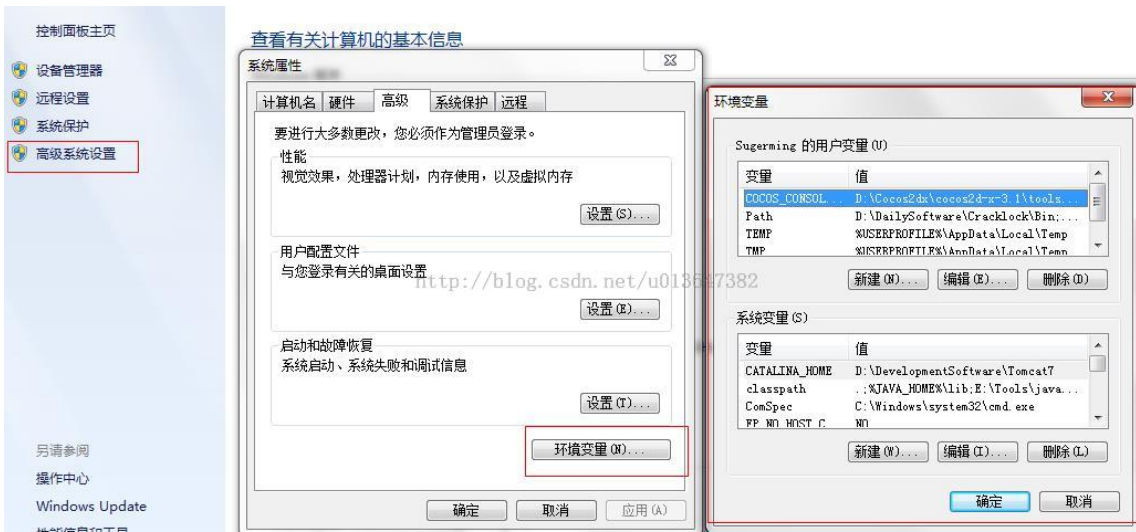
2. 下载完成之后是一个exe文件：



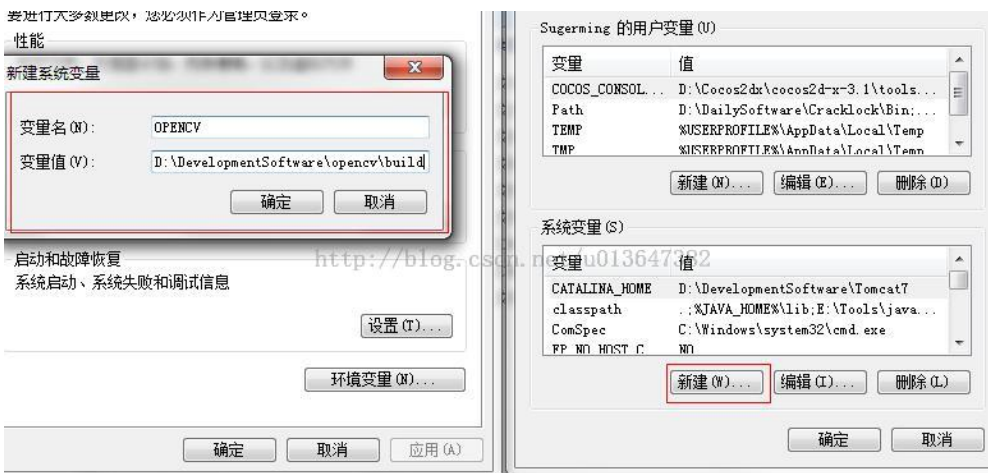
双击解压它，会选择解压目录，我这里选择的是：D:\DevelopmentSoftware，解压之后为：



3. 配置系统环境变量：计算机 > 属性 > 高级系统设置 > 环境变量



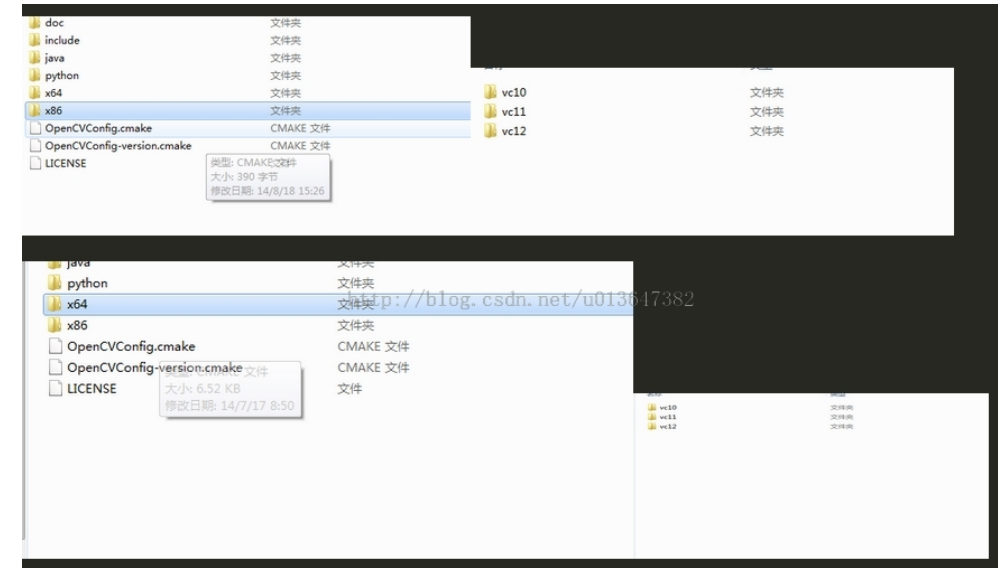
(1) 首先创建opencv的环境变量, 变量名: OPENCV, 变量值: D:\DevelopmentSoftware\opencv\build (你的opencv解压目录的 build目录)



(2) 添加到Path变量。双击 Path, 在变量值末尾添加: %OPENCV%\x86\vc12\bin。(最后没有";", 不要手贱哈)

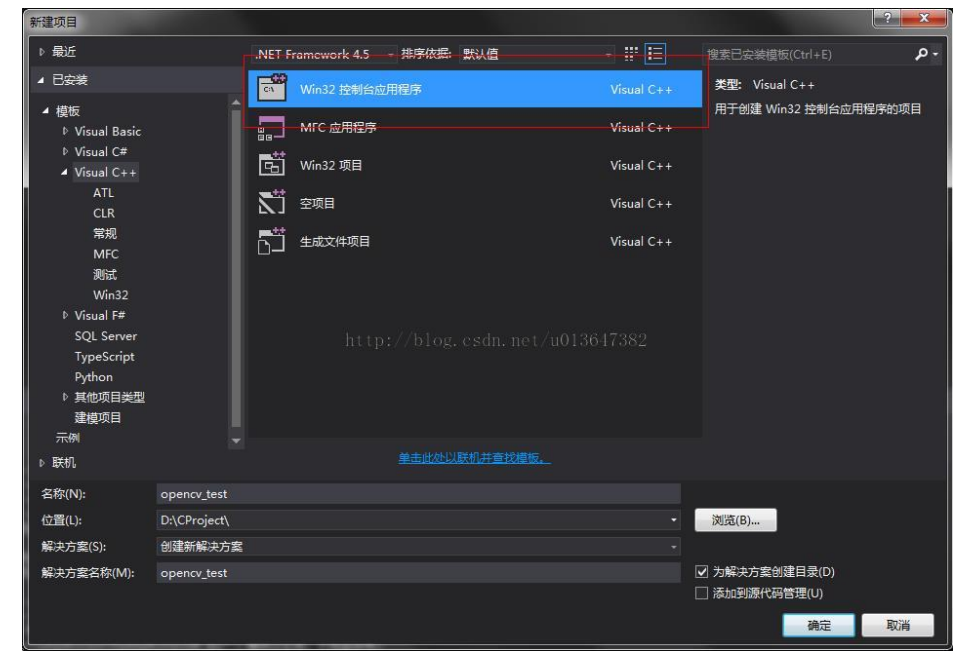


ps: opencv/build目录下有 x64和x86两个目录, 里面都有 vc10, vc11, vc12。这里注意一定要选 x86目录, 因为编译都是32位的。vc12 对应 vs2013, 不是vs2012, 不要被误导了。(vc11->vs2012, vc10->vs2010)

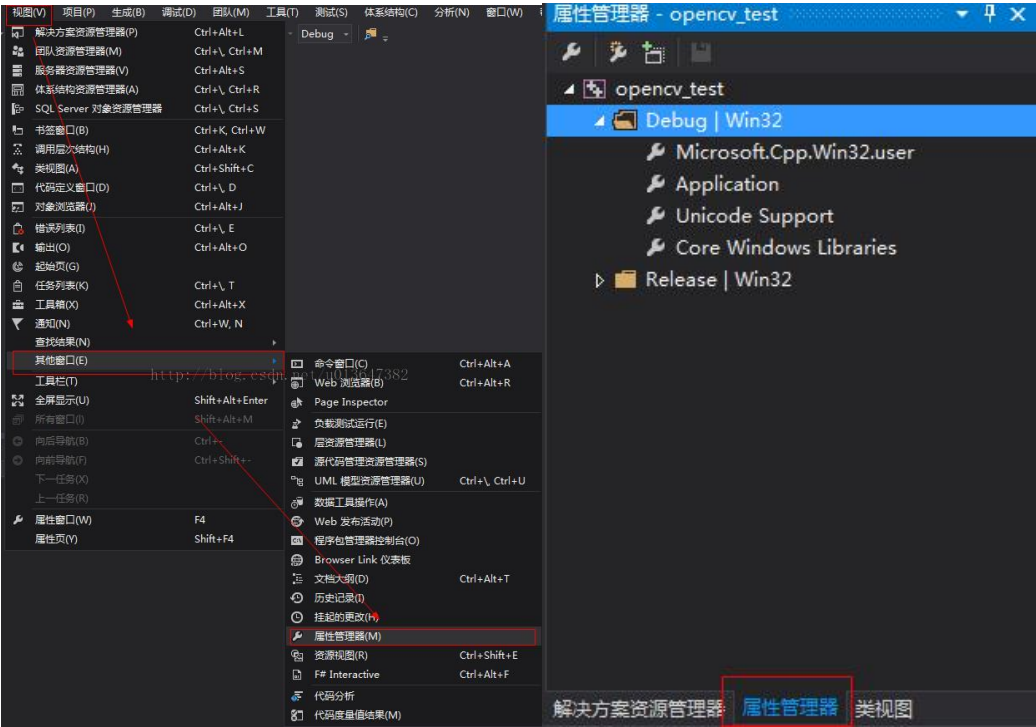


4. 到这里，就可以打开vs2013，开始创建项目了。

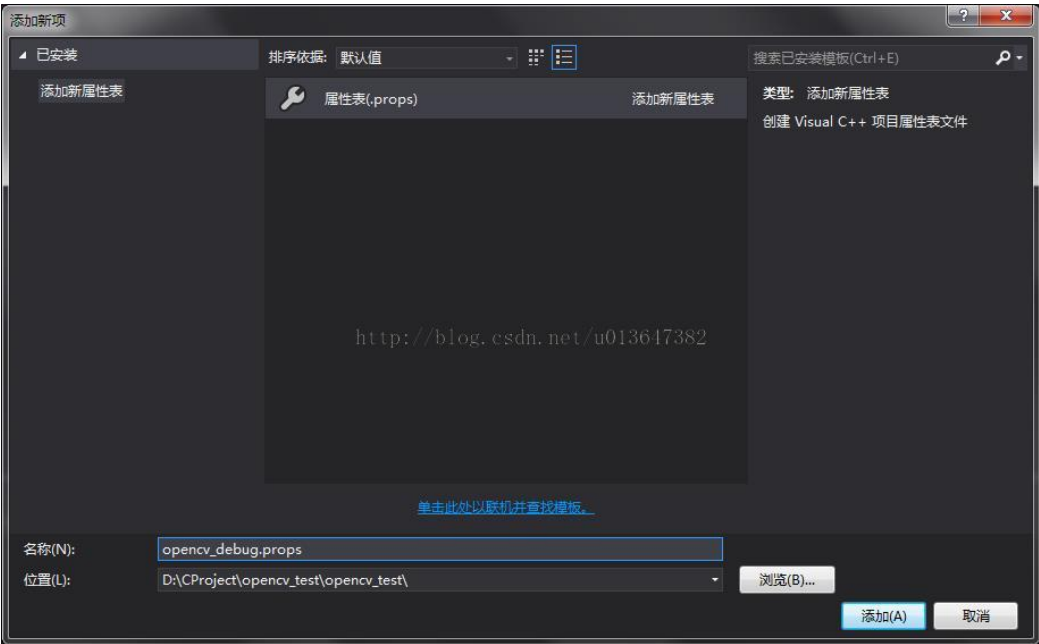
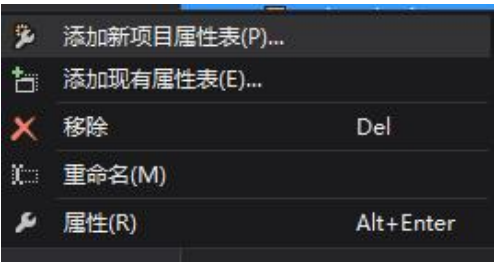
(1) 创建一个win32空项目：

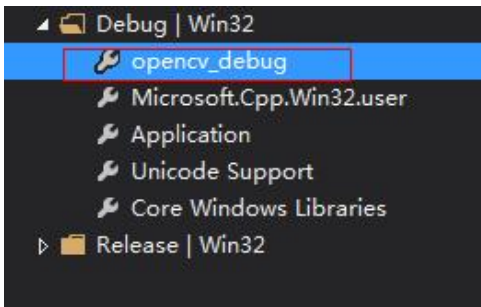


(2) 打开属性管理器：

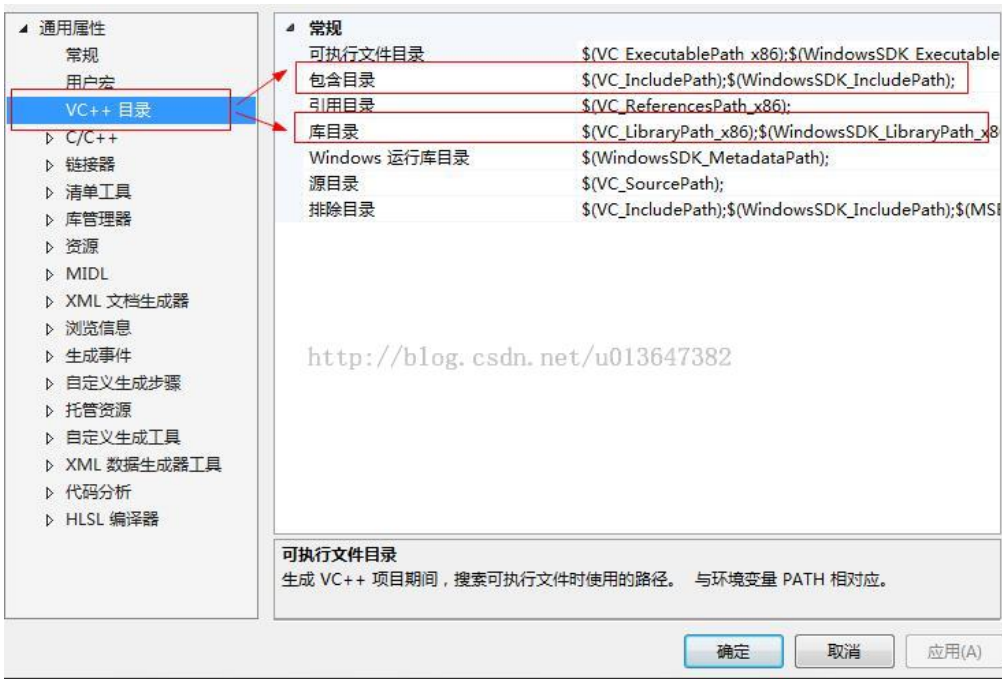


(3) 在debug上右键，添加新项目属性表，创建一个属性表： opencv\_debug.props， 双击它开始编辑：





然后 点击VC++目录，我们需要分别添加包含目录和库目录。

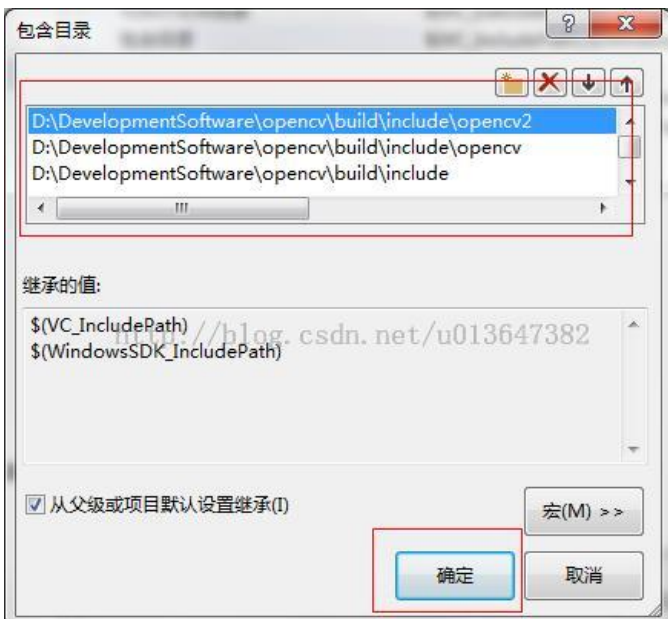


<1> 包含目录 -> 编辑，添加这三个目录到包含目录下：

D:\DevelopmentSoftware\opencv\build\include

D:\DevelopmentSoftware\opencv\build\include\opencv

D:\DevelopmentSoftware\opencv\build\include\opencv2

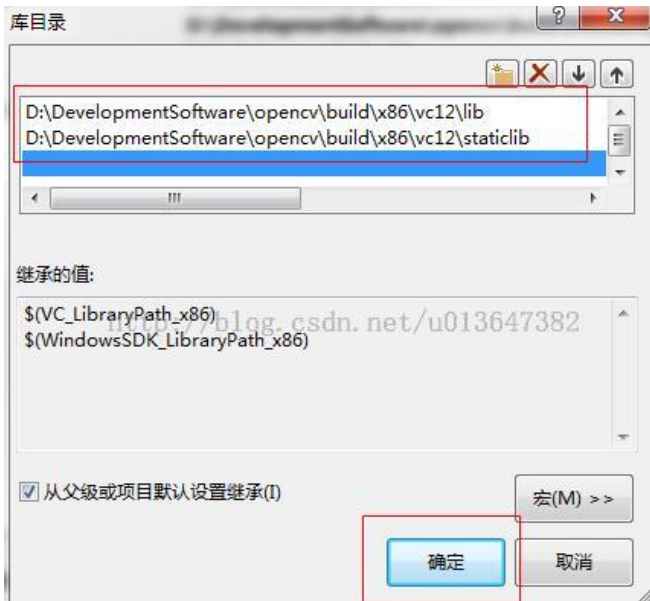


<2> 然后添加库目录：

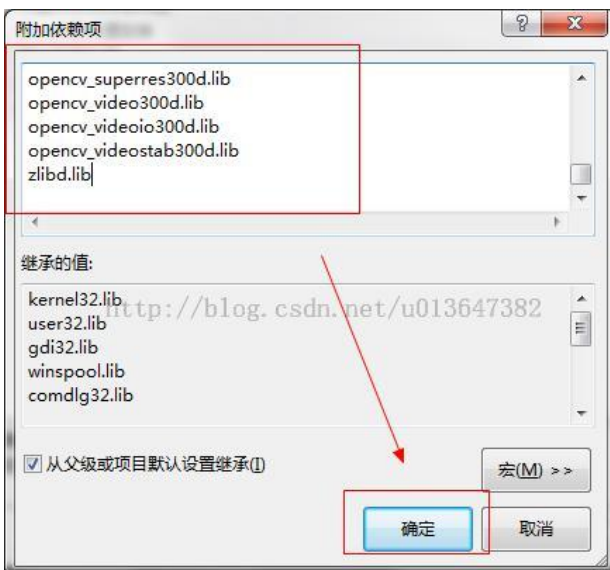
D:\DevelopmentSoftware\opencv\build\x86\vc12\lib



D:\DevelopmentSoftware\opencv\build\x86\vc12\staticlib



<3> 链接器 -> 附加依赖项 -> 编辑



直接拷贝进去:

opencv\_ts300d.lib  
opencv\_world300d.lib  
IlmImfd.lib  
libjasperd.lib  
libjpegd.lib  
libpngd.lib  
libtiffd.lib  
libwebpd.lib

```

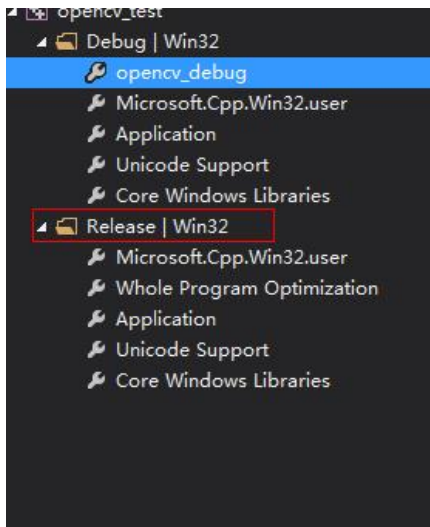
opencv_calib3d300d.lib
opencv_core300d.lib
opencv_features2d300d.lib
opencv_flann300d.lib
opencv_highgui300d.lib
opencv_imgcodecs300d.lib
opencv_imgproc300d.lib
opencv_ml300d.lib
opencv_objdetect300d.lib
opencv_photo300d.lib
opencv_shape300d.lib
opencv_stitching300d.lib
opencv_superres300d.lib
opencv_video300d.lib
opencv_videoio300d.lib
opencv_videostab300d.lib
zlibd.lib

```

这里链接时使用到的库。 这里有两点需要注意：

1. 各个版本的opencv链接的库文件不一样，这里只适用于 opencv3.0.0(alpha)版本。
2. 现在我们添加进的是 debug使用的库， 可以发现每个lib文件文件名后有一个“d”，表示debug。所以下一步我们在添加release版本的时候，添加的链接库和上面的一样，只是后面没有“d”。

(4) 做到这里，debug模式调试就可以使用了，为了使用release，我们需要将 步骤(3)的步骤再做一遍，创建一个opencv\_release.props。只是说我们选择的属性是release。而且刚刚已经说过了，添加链接库的时候添加的是后面没有“d”的。



链接库：

```

opencv_ts300.lib
opencv_world300.lib
IlmImf.lib
ippicvmt.lib
libjasper.lib
libjpeg.lib
libpng.lib
libtiff.lib
libwebp.lib
opencv_calib3d300.lib
opencv_core300.lib
opencv_features2d300.lib
opencv_flann300.lib
opencv_highgui300.lib

```

```

opencv_imgcodecs300.lib
opencv_imgproc300.lib
opencv_ml300.lib
opencv_objdetect300.lib
opencv_photo300.lib
opencv_shape300.lib
opencv_stitching300.lib
opencv_superres300.lib
opencv_video300.lib
opencv_videoio300.lib
opencv_videostab300.lib
zlib.lib

```

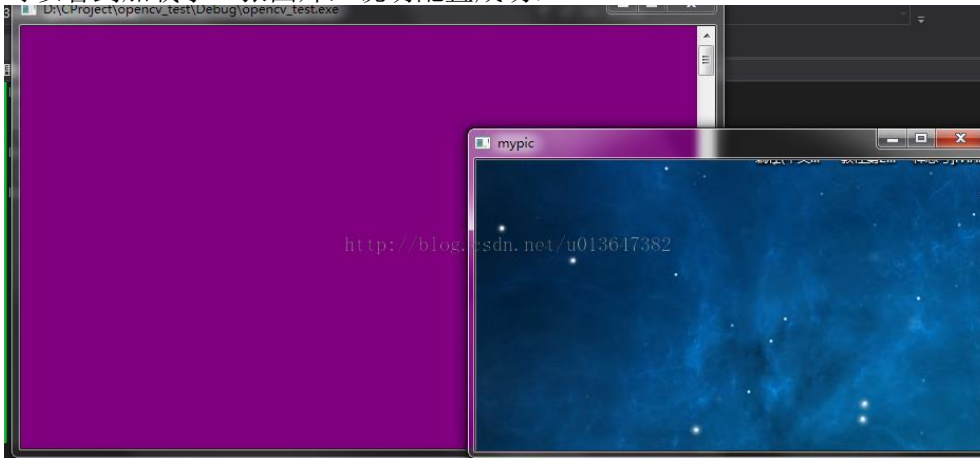
(5) 好了，到这里我们就已经配置完了。可以写个程序来测试一下：

```

view plain copy to clipboard print ?
01. #include <opencv2\opencv.hpp> #include <iostream> #include <string> using namespace cv;using name
    { Mat img = imread("F:\\test.bmp"); if (img.empty()) { cout << "error"; return

```

可以看到加载了一张图片，说明配置成功：



(6) 这里还有一个建议，每次创建项目都要这样配置是不是觉得很麻烦。不要忘了我们刚刚创建的项目配置文件， `opencv_debug.props` 和 `opencv_release.props`，其实这两个文件是可以提取出来。直接到项目文件夹下，搜索配置文件，直接copy出来，下次就可以直接导入。



选择添加现有属性表，选择我们以前创建好的就可以了。

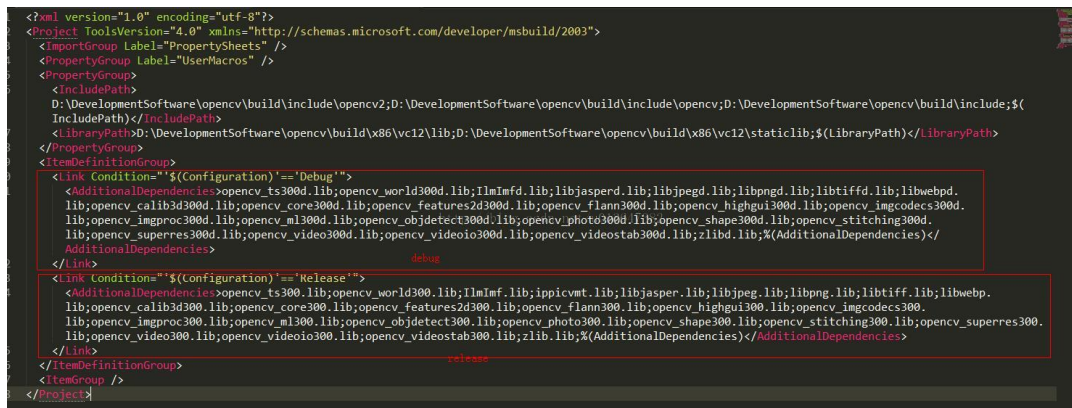




这样配置简单了一点，但是每次需要添加两次，就是debug和release要分别导入。其实我们可以把两个文件写到一个属性表中，这样debug和release都添加这个就可以了。大家自己把配置文件抓出来做拼接，我这里就不带大家做了，很简单，就是在 <Link> 标签中添加了条件判断：

```
<Link Condition="'$(Configuration)'=='Debug'"> .... </Link>
```

```
<Link Condition="'$(Configuration)'=='Release'">.... </Link>
```



```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <ImportGroup Label="PropertySheets" />
  <PropertyGroup Label="UserMacros" />
  <PropertyGroup>
    <IncludePath>
      D:\DevelopmentSoftware\opencv\build\include\opencv2;D:\DevelopmentSoftware\opencv\build\include\opencv;D:\DevelopmentSoftware\opencv\build\include;$(
    </IncludePath>
    <LibraryPath>D:\DevelopmentSoftware\opencv\build\x86\vc12\lib;D:\DevelopmentSoftware\opencv\build\x86\vc12\staticlib;$(LibraryPath)</LibraryPath>
  </PropertyGroup>
  <ItemDefinitionGroup>
    <Link Condition="'$(Configuration)'=='Debug'">
      <AdditionalDependencies>opencv_ts300d.lib;opencv_world300d.lib;Imlmf.lib;libjasper.lib;libjpeg.lib;libpngd.lib;libtiffd.lib;libwebpd.
      lib;opencv_calib3d300d.lib;opencv_core300d.lib;opencv_features2d300d.lib;opencv_flann300d.lib;opencv_highgui300d.lib;opencv_imgcodecs300d.
      lib;opencv_imgproc300d.lib;opencv_ml300d.lib;opencv_objdetect300d.lib;opencv_photo300d.lib;opencv_shape300d.lib;opencv_stitching300d.
      lib;opencv_superres300d.lib;opencv_video300d.lib;opencv_videoio300d.lib;opencv_videostab300d.lib;zlibd.lib;%(AdditionalDependencies)</
      AdditionalDependencies>
    </Link>
    <Link Condition="'$(Configuration)'=='Release'">
      <AdditionalDependencies>opencv_ts300.lib;opencv_world300.lib;Imlmf.lib;ippicvt.lib;libjasper.lib;libjpeg.lib;libpng.lib;libtiff.lib;libwebp.
      lib;opencv_calib3d300.lib;opencv_core300.lib;opencv_features2d300.lib;opencv_flann300.lib;opencv_highgui300.lib;opencv_imgcodecs300.
      lib;opencv_imgproc300.lib;opencv_ml300.lib;opencv_objdetect300.lib;opencv_photo300.lib;opencv_shape300.lib;opencv_stitching300.lib;opencv_superres300.
      lib;opencv_video300.lib;opencv_videoio300.lib;opencv_videostab300.lib;zlib.lib;%(AdditionalDependencies)</AdditionalDependencies>
    </Link>
  </ItemDefinitionGroup>
</Project>
```

好了，终于完结了。真是不容易啊。配置成功了给自己一个赞！

版权声明：本文为博主原创文章，未经博主允许不得转载。

## 文章分类

- [排序算法](#) (8)
- [Android 入门](#) (28)
- [Android 进阶](#) (3)
- [OpenCV](#) (2)
- [趣味题](#) (2)
- [欧拉项目](#) (21)
- [我的工具](#) (4)
- [HDOJ](#) (4)
- [java](#) (3)
- [c#](#) (1)
- [Git](#) (1)
- [Python](#) (1)

## 两天热门文章

- [\[拉开大变革序幕（上）：在浪潮之巅观望Docker\]](#)
- [\[Android Context完全解析，你所不知道的Context的各种细节\]](#)
- [\[【凯子哥带你学Framework】Activity界面显示全解析\]](#)
- [\[用Eclipse进行C++开发时Bianry not found的问题解决\]](#)
- [\[std::deque\]](#)
- [\[Android 一个改善的okHttp封装库\]](#)
- [\[VS2013安装SVN插件\]](#)
- [\[拉开大变革序幕（中）：Docker场景化\]](#)
- [\[【浅墨Unity3D Shader编程】之十 深入理解Unity5中的Standard Shader\(二\)&屏幕油画特效的实现\]](#)
- [\[Chromium硬件加速渲染的OpenGL命令执行过程分析\]](#)
- [\[程序员，这12个问题让经理比你痛苦多了\]](#)
- [\[从头认识 java-7.4 实现多重接口\]](#)
- [\[Volley完全解析\]](#)
- [\[用SBT和Play进行JSON序列化\]](#)

- [\[linux中mmap系统调用原理分析与实现\]](#)
- [\[从头认识java-7.3 接口怎样解耦?\]](#)
- [\[Velt中的编译器配置\]](#)
- [\[HDU 2918 Tobo or not Tobo\]](#)
- [\[C++11新特性之 std::array container\]](#)
- [\[程序员眼中的单词\]](#)

## 最新推荐文章

©2012 [核心网络](#). Powered by [核心网络联系我们](#) | [合作伙伴](#) | [广告服务](#)