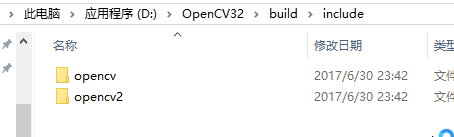
1. 安装vs 2015 社区版

安装使用自定义安装，选上所有visual C++ 语言

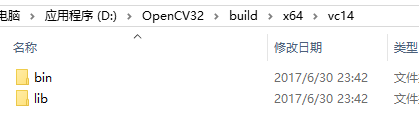
2.安装opencv32(exe 文件 直接执行/解压到自定义目录 e.g D:\OpenCV32\)

主要目录:

D:\OpenCV32\build\include (下面有opencv opencv2两个子目录)



D:\OpenCV32\build\x64\vc14 (下面有 bin和 lib目录)

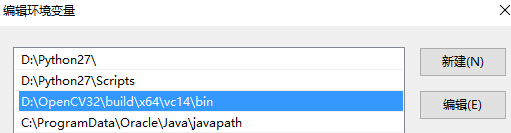


OpenCV32\build\x64\vc14\bin 目录下有：

opencv\_world320.dll

opencv\_world320d.dll

所有要把目录OpenCV32\build\x64\vc14\bin 加入系统变量PATH



OpenCV32\build\x64\vc14\lib 目录下有：

opencv\_world320.lib

opencv\_world320d.lib

链接器需要！

1. 新建工程 选择 win32控制台程序

测试程序：

#include "stdafx.h"

#include <iostream>

#include <opencv2/core/core.hpp>

#include <opencv2/highgui/highgui.hpp>

using namespace cv;

using namespace std;

int main()

{

Mat img = imread("D:\\项目\\opencvTest\\opencvTest\\lena.jpg");

if (img.empty())

{

cout << "error";

return -1;

}

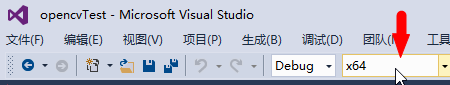
imshow("Lena", img);

waitKey();

return 0;

}

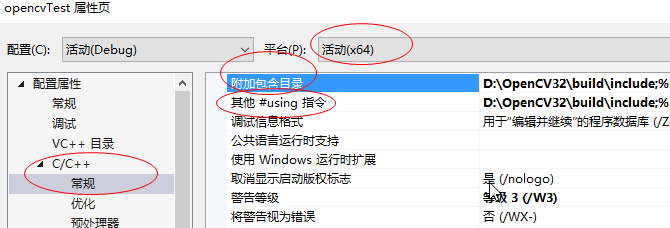
1. 选择x64 (因为opencv3.2只有x64)



1. 配置工程 添加OpenCV头文件和库路径

右键项目--属性

C/C++ 标签 附加包含目录 和 #using指令 选择openCV对应头文件目录 <OpenCV>/build/include

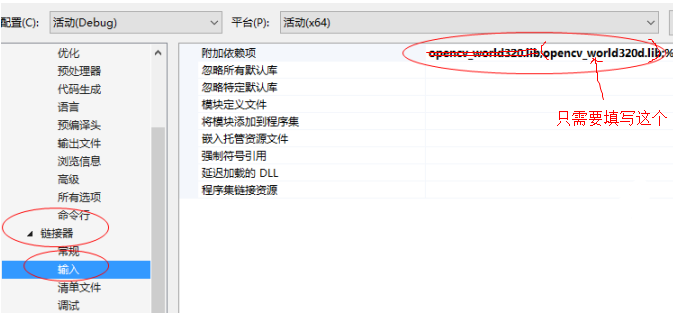


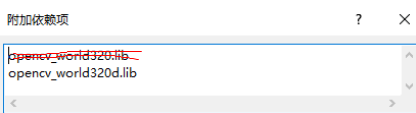
链接器--常规 附加库目录 选择 <OpenCV>/build/x64/vc14/lib 目录



链接器--输入 附加依赖项 输入 ~~opencv\_world320.lib~~ opencv\_world320d.lib (每行写一个，换行)

(opencv\_world320.lib opencv\_world320d.lib 位于<OpenCV>/build/x64/vc14/lib下 注意！320d是debug版本 如果当前设置是debug模式，那么应该“只”选opencv\_world320d.lib )



 (注意 如果是)

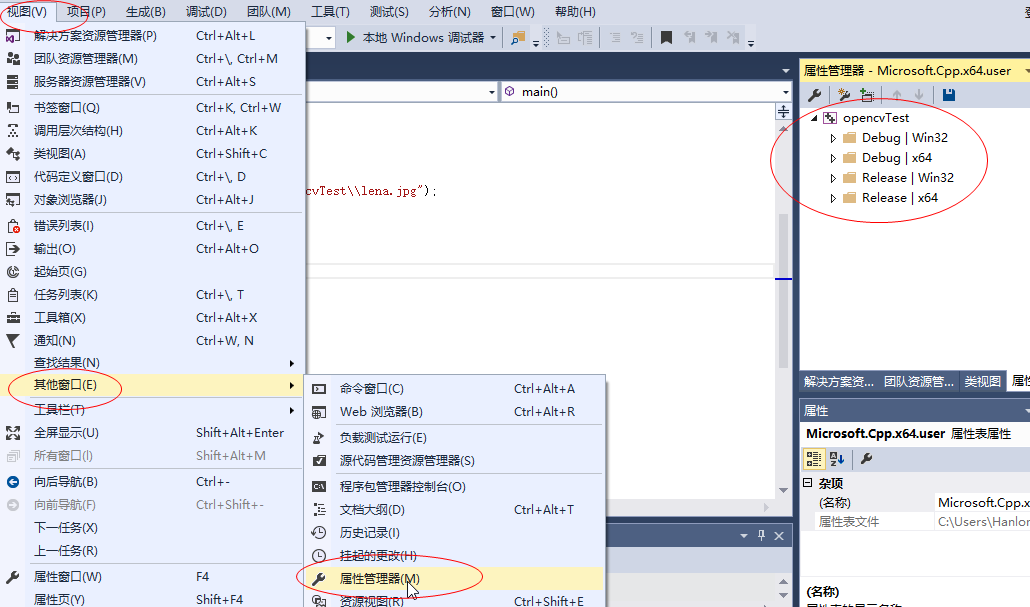
6. 编译执行

7. 全局！应用到其他工程

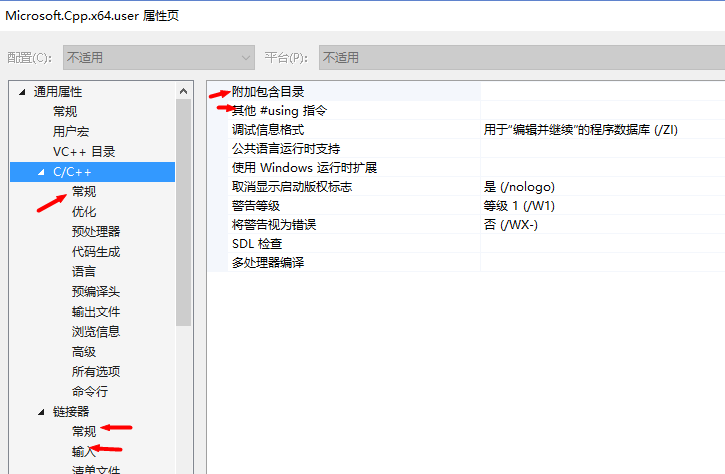
<http://blog.chinaunix.net/uid-21375345-id-2953297.html>

这里的设置是针对全局的！ 注意分为win32 x64 而且再分为Debug Release版本

如果原来项目只配置了x64 debug版本，执行下面步骤后，切换到release版本时，这些配置也会应用过去(否则切换到release版本又要重新设置上面的)



这个就是用户属性界面！(全局设置)



8.其他问题：

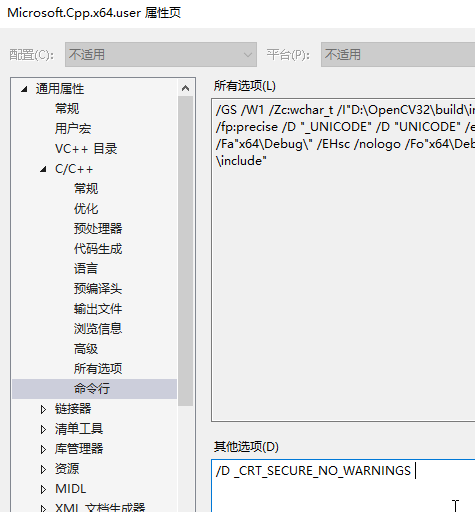
A. [解决\_CRT\_SECURE\_NO\_WARNINGS 警告](http://blog.csdn.net/iesneaker/article/details/6328278)

我们在程序中使用fopen等CRT函数，就会出现一些警告信息

解决方法：  
 右击工程 - 属性 - 配置属性 - C/C++  - 命令行

命令行增加 /D \_CRT\_SECURE\_NO\_WARNINGS

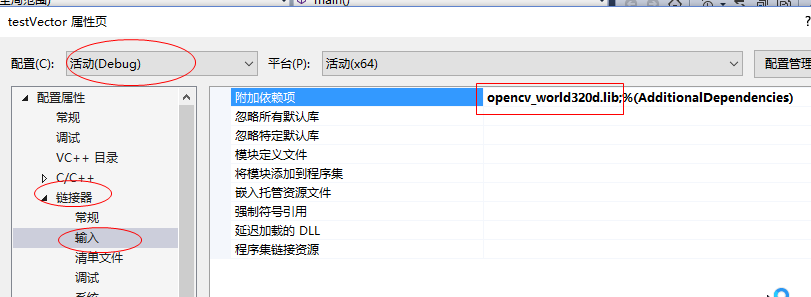
添加到全局(在user属性页面)



B. CascadeClassifier::detectMultiScale 或者 cv::findNonZero(m, x);

遇到 \*\_Pnext != 0; \*\_Pnext = (\*\_Pnext)->\_Mynextiter) 指针异常 程序崩溃

在 链接器--输入 附加依赖项 输入 对应release模式和debug模式 只能分别选择 opencv\_world320.lib 或者 opencv\_world320d.lib



测试程序：

#include "stdafx.h"

#include <opencv2/opencv.hpp>

using namespace cv;

using namespace std;

int main()

{

//cv::namedWindow("hello");

int size = 12;

std::vector<cv::Point> x;

cv::Mat m(1, size, CV\_8U, cv::Scalar(128));

cv::findNonZero(m, x); // 在矩阵中找到非零的元素 返回坐标点(Point)数组(Vector)

printf("findNonZero x %zd " , x.size() ); // 程序异常时这里打印负数

//cv::waitKey(); // 如果没有创建窗口（namedWindow imshow等） 这句话不会等待

//system("pause");

return 0;

}

C. 编译出错！ 预编译头文件来自编译器的早期版本，或者预编译头为 C++ 而在 C 中使用它(或相反)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 严重性 | 代码 | 说明 | 项目 | 文件 | 行 |
| 错误 | C1853 | “x64\Debug\face\_detect.pch”预编译头文件来自编译器的早期版本，或者预编译头为 C++ 而在 C 中使用它(或相反) | face\_detect | d:\项目\opencvtest\face\_detect\face\_detect\hog.c | 1 |

项目中混合使用C和C++文件导致

即使使用了extern “C”{} 也会导致这个

选取C文件，多个C文件用Ctrl同时选上 右键-属性

