

OpenCV Installation in Microsoft Visual Studio 2005

(rewritten by : Sunu Wibirama)

I. Installation

OpenCV is an open source library for [computer vision](#) development. There are some very interesting sample applications included with this library. Before we can start we need to be able to build the library.

Requirements

- [OpenCV Library](#)
- [Visual Studio 2005 Express C++](#)
- [Windows Platform SDK](#)

Optional

- [Visual Studio 2005 Express C++ SP1](#)

We are going to be using [Microsoft Visual Studio](#) for this install. The reasons for this include Support from OpenCV, Lots of Internet Articles for creating software, MSDN, Official Software and its FREE.

Step 1 - Installing Visual C++ and Platform SDK

Visit <http://msdn.microsoft.com/vstudio/express/visualc/usingpsdk/> and follow Microsofts Guide to setup Visual C++ and the Platform SDK.

When you get to Step 3, add this line to "Include Files" (assume default install location)
C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include\mf

This is because OpenCV requires some files that are inside that folder.

Step 2 - Update Visual C++ with Service Pack 1 (Optional)

Service Packs for the each of the Visual Studio products are available on one page from Microsoft.

[Visual Studio 2005 Express SP1](#)

The specific update that you will require is called "C++ 2005 [Express Edition](#) SP1 - VS80sp1-KB926748-X86-INTL.exe"

Step 3 - Install OpenCV

Go get [OpenCV Library](#) and download [OpenCV 1.0 for Windows](#) (I use OpenCV beta 5.0a)

Installation is straight forward.

Step 4 - Customising Visual C++ for use with OpenCV

We need to setup the directories for Open Visual C++

- Open Visual C++
- Choose menu "Tools" and select "Options"
- In "Projects and Solutions" and go to "VC++ Directories"
- Show directories for "Library Files"
- Add "C:\Program Files\OpenCV\lib" to the list of directories

Step 5 - Open "OpenCV Workspace .NET 2005"

In the start menu a folder called "Open CV" was created when you installed OpenCV.

Inside the folder "OpenCV" there is a shortcut called "OpenCV Workspace .NET 2005"

Open "OpenCV Workspace .NET 2005"

This will load OpenCV solution within Visual C++

There is one last thing you need to do, to be able to compile OpenCV.

- Choose menu "Build" and Select "Configuration Manager"
- Change "Active solution configuration" to either "Debug" or "Release"

Step 6 - Build

Now that everything is installed and setup; building OpenCV should be a breeze.

- Choose menu "Build" and Select "Build Solution"

Building OpenCV can take anywhere from a few minutes to an hour.

When I built OpenCV it took about 10 minutes.

II. Creating Application at A Glance

Now, we will try to create application based on OpenCV at a glance. First, we will try to create console application (without Graphical User Interface / GUI)

1. Create new console project by following the picture below

Creating the Project
A project is initially created by selecting:
File -> New -> Project

Create a "Win32 Console Application"

Make it an "Empty Project" by selecting the box under "Application Settings"

The image shows two screenshots from Visual Studio. The top screenshot is the 'New Project' dialog box. In the 'Project Types' list, 'Win32 Console Application' is selected. The 'Name' field is 'test1' and 'Solution Name' is 'test1'. The bottom screenshot is the 'Win32 Application Wizard - test1' dialog box. Under 'Application type', 'Console application' is selected. Under 'Additional options', the 'Empty project' checkbox is checked. The 'Finish' button is highlighted.

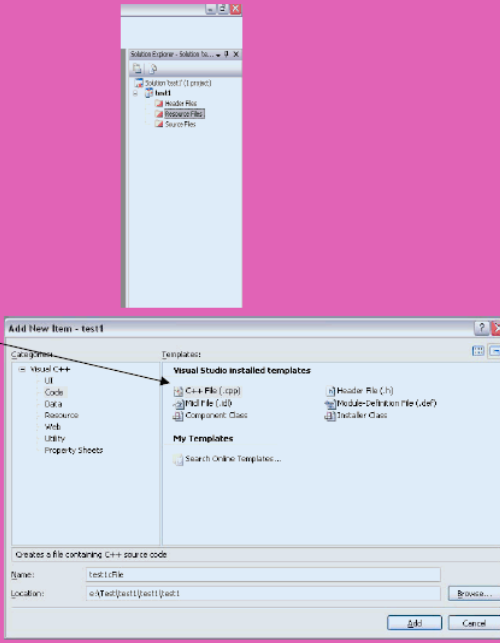
2. Right click and add new file

Create the First File

Right Click the **“Source Files”** Folder under the project name (“Test1” in this case)
Add -> Add new Item

Select **“C++ file(.cpp)”** and give it a name

Creating a file makes it possible to set **“Additional Include Directives”** in the C/C++ pane under the project properties.

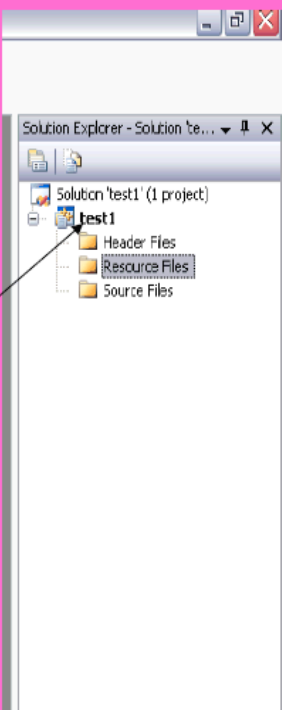


3. Set additional properties to include OpenCV directory and library, follow these steps :

In order to build projects using OpenCV the required libraries and directives must be included in the project's properties

Open the Properties Pane

Right Click the name of the project and select **“Properties”**
 (“Test1” in this case)



Set Additional Include Directives

Under the C/C++ tab select **“General”**

Select the **“Additional Include Directories”**

Add the full path to each of the folders which contain **“.h”** files required to use OpenCV

Be sure to include trailing **“\”**

Utilized Directives

```
D:\OpenCV\cvaux\include\
D:\OpenCV\cxcore\include\
D:\OpenCV\cv\include\
D:\OpenCV\otherlibs\highgui\
D:\OpenCV\otherlibs\cvcam\include\
```

```
..\..\cvaux\include\
..\..\cxcore\include\
..\..\cv\include\
..\..\otherlibs\highgui\
..\..\otherlibs\cvcam\include\
```

Set Additional Dependencies

Under the Linker tab select **“Input”**

Select the **“Additional Dependencies”**

Add the full path to each of the **“.lib”** files required to use OpenCV

Be sure to keep the paths in quotes

Utilized Dependencies

```
"D:\OpenCV\lib\cv.lib"
"D:\OpenCV\lib\cvaux.lib"
"D:\OpenCV\lib\cxcore.lib"
"D:\OpenCV\lib\cvcam.lib"
"D:\OpenCV\lib\highgui.lib"
```

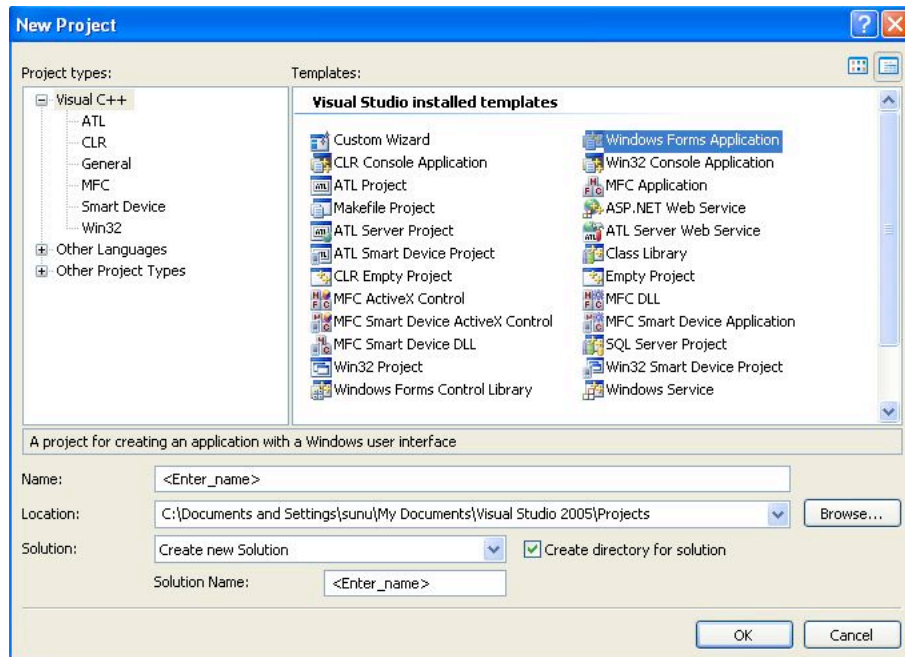
```
“..\..\lib\cv.lib“
“..\..\lib\cvaux.lib“
“..\..\lib\cxcore.lib“
“..\..\lib\cvcam.lib“
“..\..\lib\highgui.lib“
```

Try to write some simple syntax and compile it (check whether it works or not) :
`cvNamedWindow(“test”, 1);`

How about application with GUI ? We will show you in the next section

III. OpenCV Application with Graphical User Interface

1. Click File >> New >> Project >> Visual C++ >> Windows Forms Application (don't forget to enter your application's name)



2. Set additional directories and libraries, as you do in creating console application (step 3 previous section).
3. The main important thing, set your general properties of Common Language Runtime Support. To do this, select your project at the left windows, right click >> Properties >> Configuration Properties >> General >> Common Language Runtime Support
Change from "Pure MSIL Common Language Runtime Support" to "Common Language Runtime Support"and happy coding !

