

HOME

ABOUT

BOOKS

CONTACT

DONATE

DOWNLOADS

technical-recipes.com

3RD PARTY TOOLS

ALGORITHMS

BOOST

C# / .NET / WPF

C++ / MFC / STL

DATABASES

DESIGN PATTERNS

DEVELOPMENT ENVIRONMENTS

ENCRYPTION AND SECURITY

GENETIC ALGORITHMS

GRAPH ALGORITHMS

IMAGE DETECTION

INSTALLERS

JAVA

LINUX

OPENCV

OPTIMIZATION

PYTHON

SERIALIZATION

SMART POINTERS

VERSION CONTROL

WEB DEVELOPMENT

WINDOWS

[Home](#) » [3rd Party Tools](#) » Configuring Code::Blocks to use OpenCV in Linux Environments

Configuring Code::Blocks to use OpenCV in Linux Environments

👤 Andy 📅 25 December 2014 💎 3rd Party Tools, C++ / MFC / STL, Development Environments, Image Detection, OpenCV 💬 10 Comments

Tweet

G+

Like 0

Pinterest Save

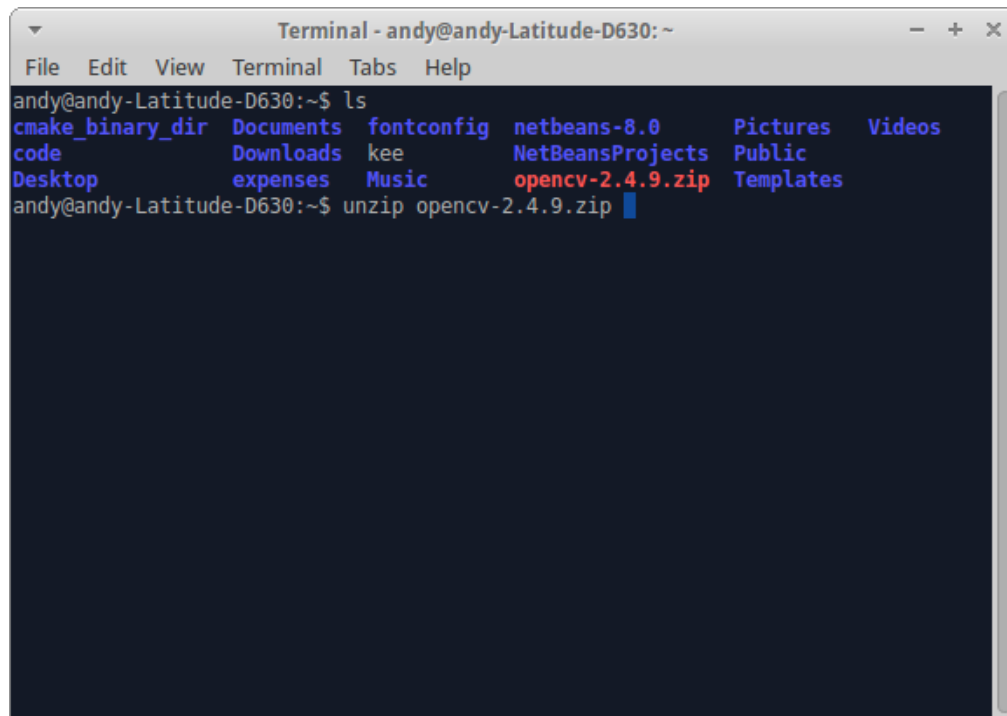
A quick guide to setting up and installing OpenCV for using in the Code::Blocks integrated development environment in Linux. The version of Linux I am currently using is Ubuntu 14.04. At the time of writing the version of OpenCV for Linux used is 2.4.9. (I had originally tried version 2.4.10 but had problems compiling it with the version of gcc I had (4.8.2), so I reverted to 2.4.9 instead.)

Step 1: Download and extract OpenCV for Linux

Versions of OpenCV can be downloaded from here:

<http://opencv.org/downloads.html>

Save it to the location of your choice. Open a command prompt, navigate to the download location and unzip:

A terminal window titled "Terminal - andy@andy-Latitude-D630: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the output of the 'ls' command, listing various directories and files including 'cmake_binary_dir', 'Documents', 'fontconfig', 'netbeans-8.0', 'Pictures', 'Videos', 'code', 'Downloads', 'kee', 'NetBeansProjects', 'Public', 'Desktop', 'expenses', 'Music', 'opencv-2.4.9.zip', and 'Templates'. The 'opencv-2.4.9.zip' file is highlighted in red. The next command entered is 'unzip opencv-2.4.9.zip' with a blue cursor at the end.

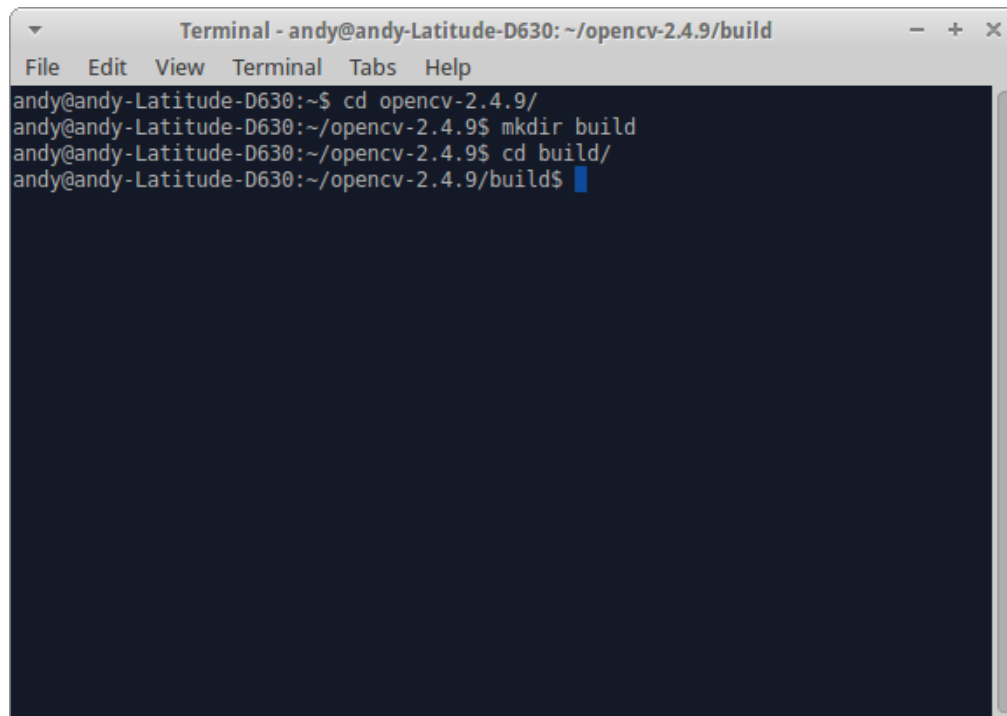
```
andy@andy-Latitude-D630:~$ ls
cmake_binary_dir  Documents  fontconfig  netbeans-8.0  Pictures  Videos
code              Downloads  kee         NetBeansProjects  Public
Desktop          expenses  Music      opencv-2.4.9.zip  Templates
andy@andy-Latitude-D630:~$ unzip opencv-2.4.9.zip
```

Step 2: Create an OpenCV build directory

Navigate to the OpenCV directory, and use

```
1 | mkdir
```

to create the build directory:

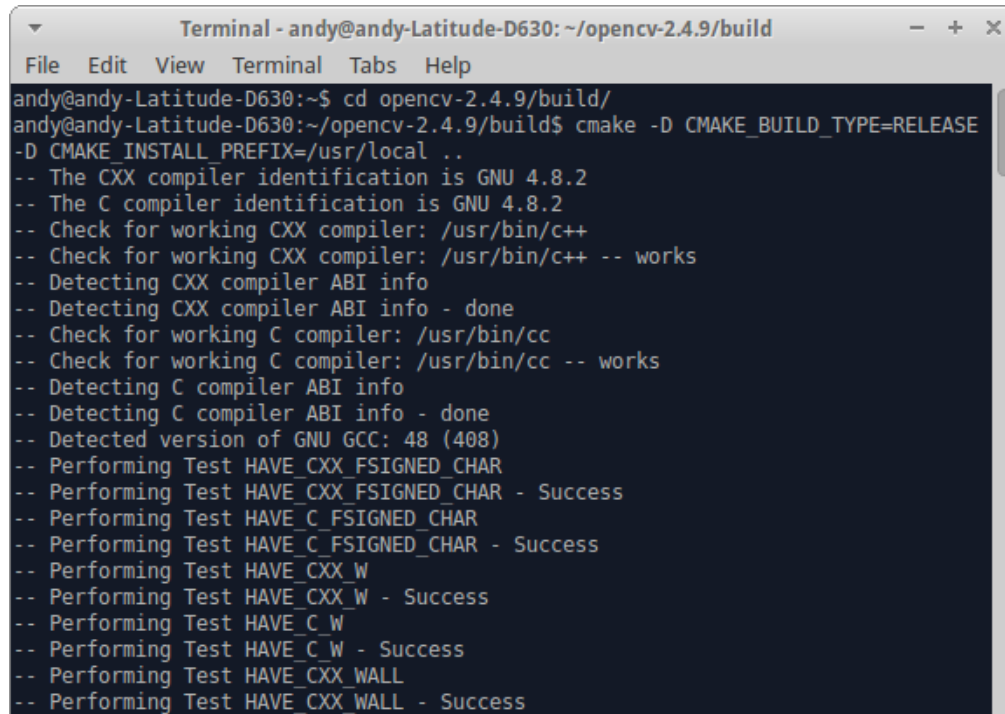
A terminal window titled "Terminal - andy@andy-Latitude-D630: ~/opencv-2.4.9/build" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the following commands and output:

```
andy@andy-Latitude-D630:~$ cd opencv-2.4.9/  
andy@andy-Latitude-D630:~/opencv-2.4.9$ mkdir build  
andy@andy-Latitude-D630:~/opencv-2.4.9$ cd build/  
andy@andy-Latitude-D630:~/opencv-2.4.9/build$
```

Step 3: Use CMake to build OpenCV

Navigate to the 'build' directory created and run the following CMake command:

```
1 | cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..
```

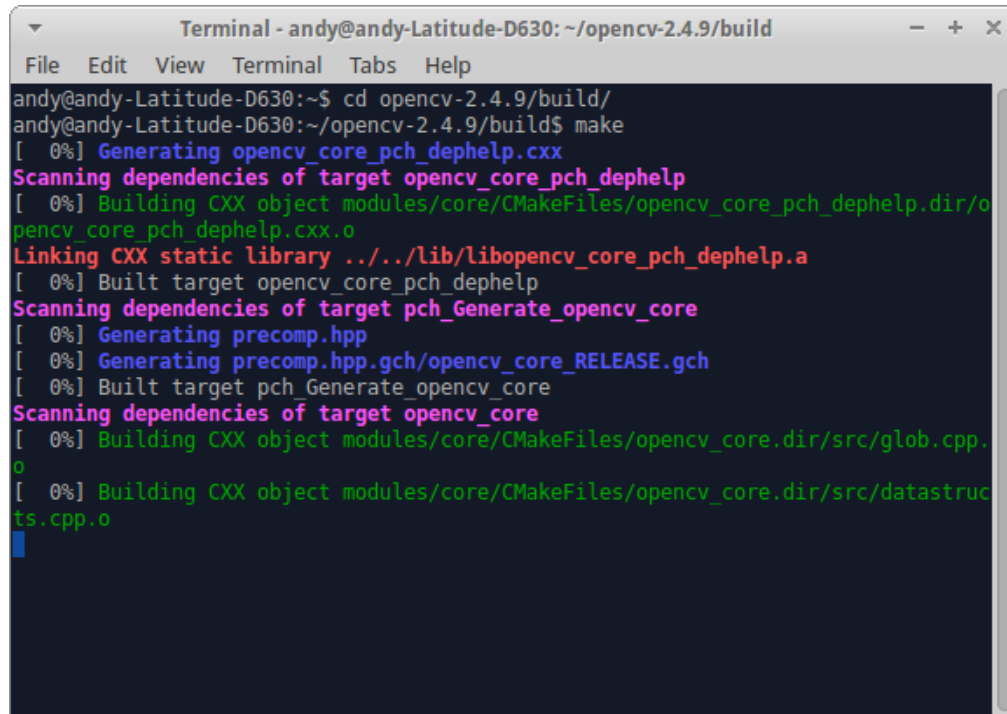


```
Terminal - andy@andy-Latitude-D630: ~/opencv-2.4.9/build
File Edit View Terminal Tabs Help
andy@andy-Latitude-D630:~$ cd opencv-2.4.9/build/
andy@andy-Latitude-D630:~/opencv-2.4.9/build$ cmake -D CMAKE_BUILD_TYPE=RELEASE
-D CMAKE_INSTALL_PREFIX=/usr/local ..
-- The CXX compiler identification is GNU 4.8.2
-- The C compiler identification is GNU 4.8.2
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detected version of GNU GCC: 48 (408)
-- Performing Test HAVE_CXX_Fsigned_char
-- Performing Test HAVE_CXX_Fsigned_char - Success
-- Performing Test HAVE_C_Fsigned_char
-- Performing Test HAVE_C_Fsigned_char - Success
-- Performing Test HAVE_CXX_W
-- Performing Test HAVE_CXX_W - Success
-- Performing Test HAVE_C_W
-- Performing Test HAVE_C_W - Success
-- Performing Test HAVE_CXX_WALL
-- Performing Test HAVE_CXX_WALL - Success
```

And then run

```
1 | make
```

command, which can take a while to complete:

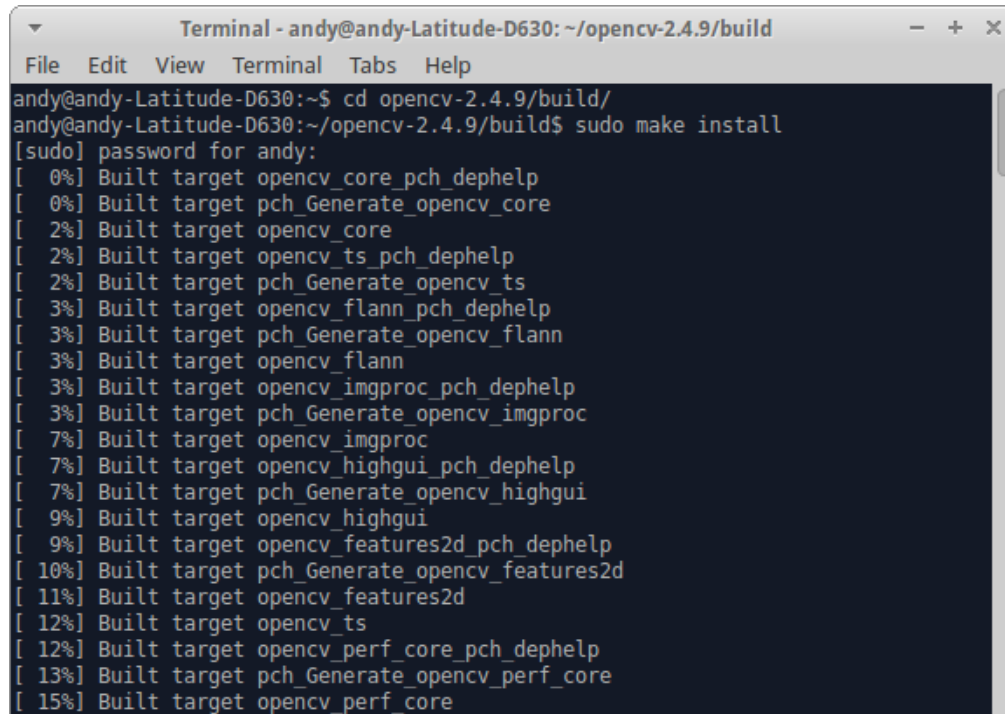


```
Terminal - andy@andy-Latitude-D630: ~/opencv-2.4.9/build
File Edit View Terminal Tabs Help
andy@andy-Latitude-D630:~$ cd opencv-2.4.9/build/
andy@andy-Latitude-D630:~/opencv-2.4.9/build$ make
[ 0%] Generating opencv_core_pch_dephelp.cxx
Scanning dependencies of target opencv_core_pch_dephelp
[ 0%] Building CXX object modules/core/CMakeFiles/opencv_core_pch_dephelp.dir/opencv_core_pch_dephelp.cxx.o
Linking CXX static library ../../lib/libopencv_core_pch_dephelp.a
[ 0%] Built target opencv_core_pch_dephelp
Scanning dependencies of target pch_Generate_opencv_core
[ 0%] Generating precomp.hpp
[ 0%] Generating precomp.hpp.gch/opencv_core_RELEASE.gch
[ 0%] Built target pch_Generate_opencv_core
Scanning dependencies of target opencv_core
[ 0%] Building CXX object modules/core/CMakeFiles/opencv_core.dir/src/glob.cpp.o
[ 0%] Building CXX object modules/core/CMakeFiles/opencv_core.dir/src/datastructures.cpp.o
```

Followed by the

```
1 | sudo make install
```

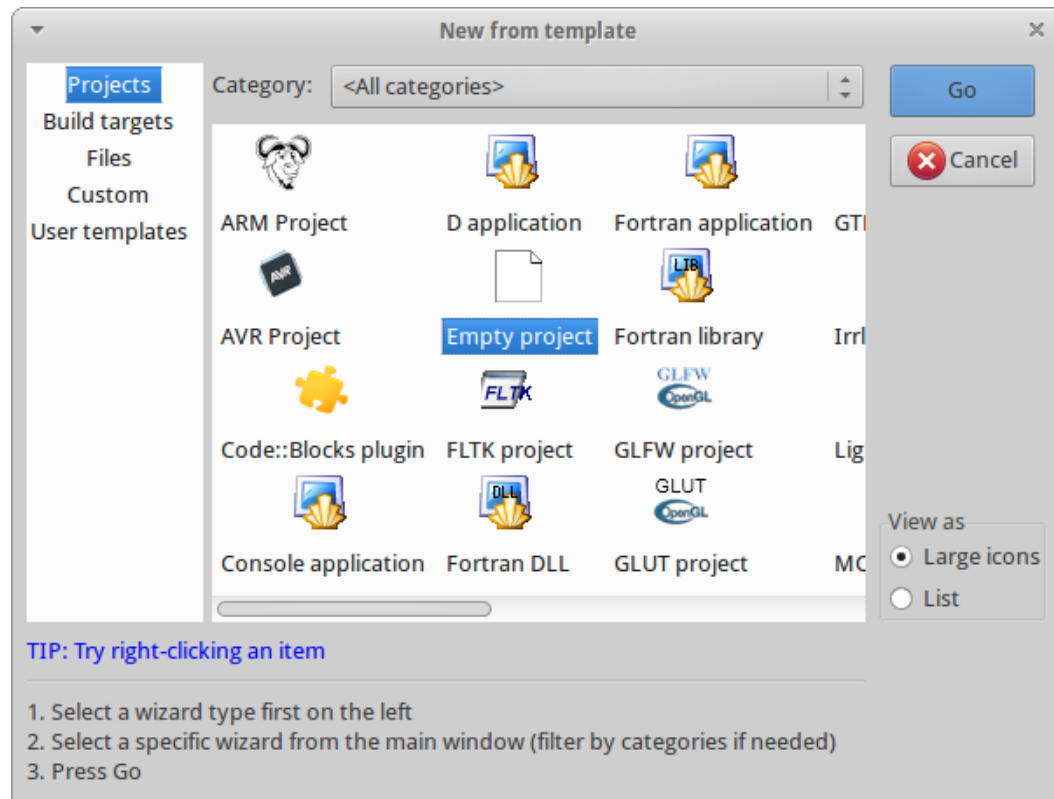
command:



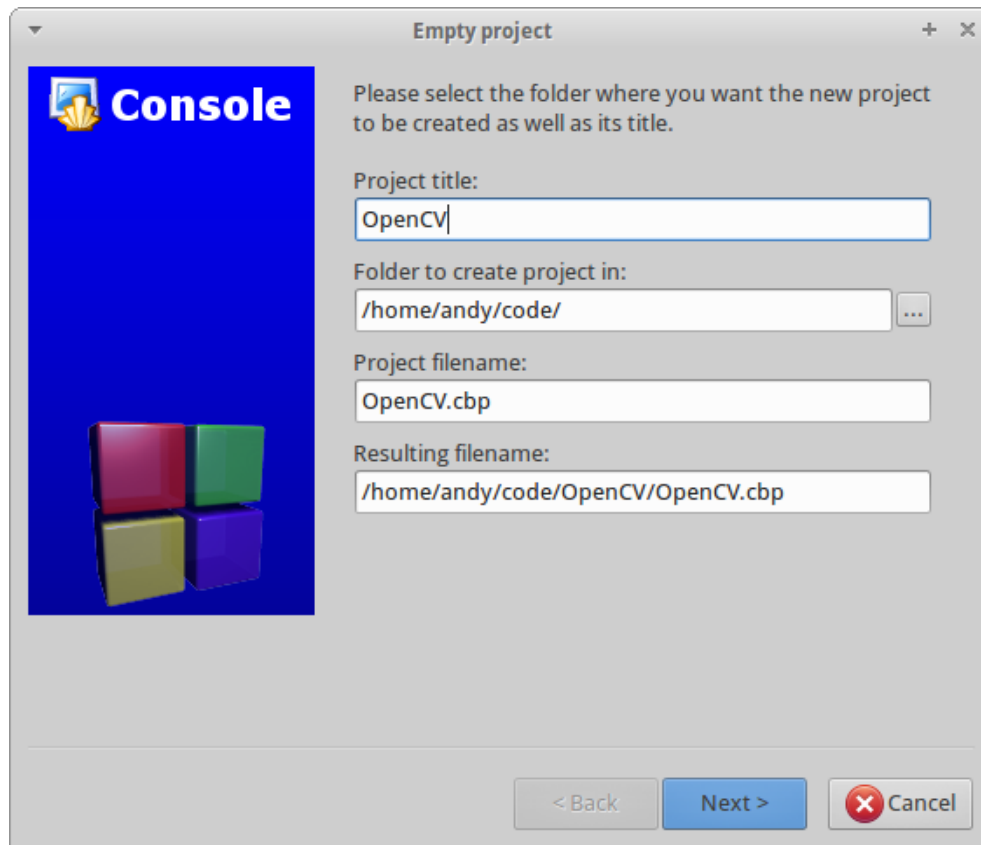
```
Terminal - andy@andy-Latitude-D630: ~/opencv-2.4.9/build
File Edit View Terminal Tabs Help
andy@andy-Latitude-D630:~$ cd opencv-2.4.9/build/
andy@andy-Latitude-D630:~/opencv-2.4.9/build$ sudo make install
[sudo] password for andy:
[ 0%] Built target opencv_core_pch_dephelp
[ 0%] Built target pch_Generate_opencv_core
[ 2%] Built target opencv_core
[ 2%] Built target opencv_ts_pch_dephelp
[ 2%] Built target pch_Generate_opencv_ts
[ 3%] Built target opencv_flann_pch_dephelp
[ 3%] Built target pch_Generate_opencv_flann
[ 3%] Built target opencv_flann
[ 3%] Built target opencv_imgproc_pch_dephelp
[ 3%] Built target pch_Generate_opencv_imgproc
[ 7%] Built target opencv_imgproc
[ 7%] Built target opencv_highgui_pch_dephelp
[ 7%] Built target pch_Generate_opencv_highgui
[ 9%] Built target opencv_highgui
[ 9%] Built target opencv_features2d_pch_dephelp
[10%] Built target pch_Generate_opencv_features2d
[11%] Built target opencv_features2d
[12%] Built target opencv_ts
[12%] Built target opencv_perf_core_pch_dephelp
[13%] Built target pch_Generate_opencv_perf_core
[15%] Built target opencv_perf_core
```

Step 4: Create a Code::Blocks Project

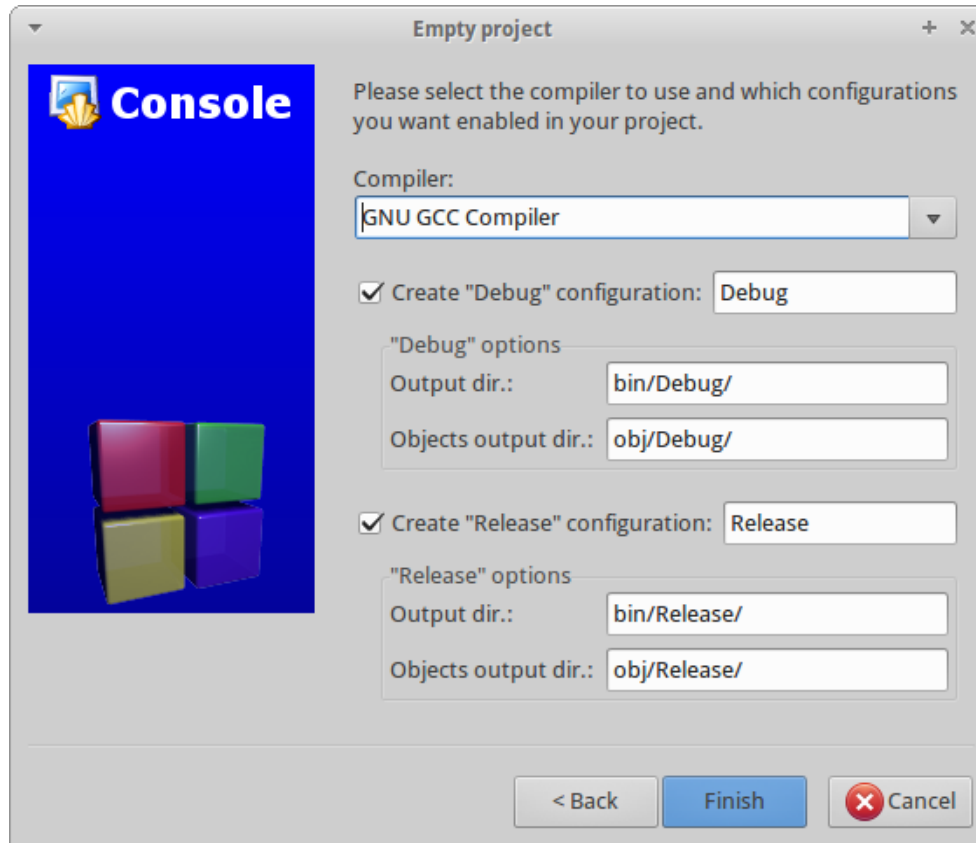
Open Code::Blocks and select File > New > Project. Select Empty Project:



Give the project a name:



And finish with the default settings:



Once the project is created, select File > New > Empty File, to add the main.cpp file to the project. Copy and paste the following code example to main.cpp:

```
1 // DetectBlobs.cpp : Defines the entry point for the console application.
2 //
3 #include <cv.h>
4 #include <cxcore.h>
5 #include <highgui.h>
6
7 int main()
8 {
9     // Initialise
10    //std::string filepath = "spots.bmp";
11    std::string filepath = "spots2.jpg";
12    int num_blobs = 0;
13
14    // Load grayscale version of coloured input image
15    IplImage* original = cvLoadImage( filepath.c_str() );
```

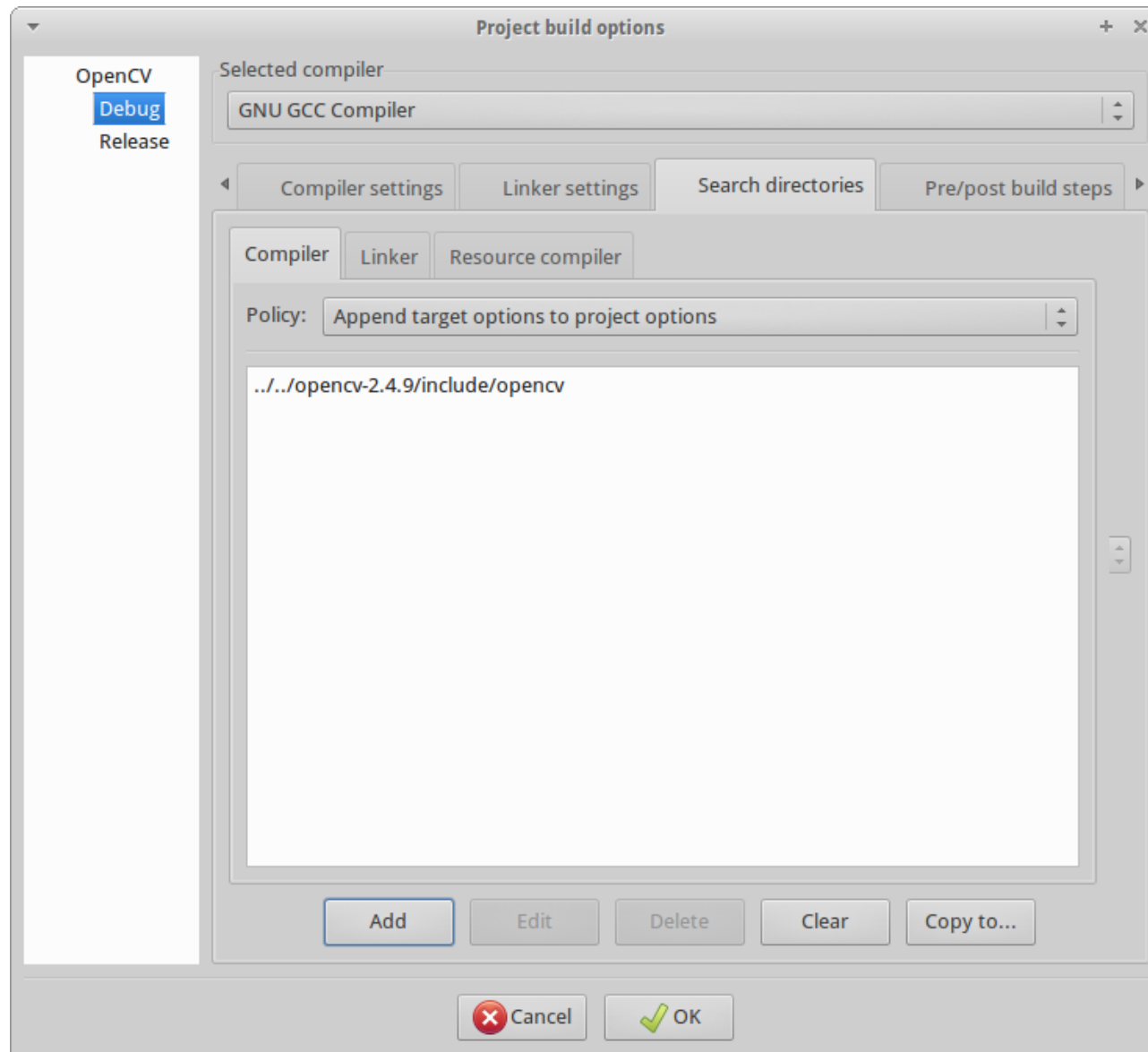
```
16 IplImage* grayscale = cvLoadImage( filepath.c_str(),
17                                     CV_LOAD_IMAGE_GRAYSCALE );
18
19 // Check bitmap image exists
20 assert( grayscale );
21
22 // Create IplImage struct for a black and
23 // white (binary) image
24 IplImage* img_bw = cvCreateImage( cvGetSize( grayscale ),
25                                   IPL_DEPTH_8U,
26                                   1 );
27
28 // Use thresholding to convert grayscale image
29 // into binary
30 cvThreshold( grayscale,           // source image
31             img_bw,              // destination image
32             40,                  // threshold val.
33             255,                 // max. val
34             CV_THRESH_BINARY );  // binary
35
36 // Create IplImage struct for inverted black
37 // and white image
38 IplImage* img_bw_inv = cvCloneImage( img_bw );
39 IplImage* img_bw_cpy = cvCloneImage( img_bw );
40
41 // Find connected components using OpenCV
42 CvSeq* seq;
43 CvMemStorage* storage = cvCreateMemStorage( 0 );
44 cvClearMemStorage( storage );
45
46 // cvFindContours the 12 + 1 extra object for
47 // white backgrounds and black spots, hence
48 // subtract 1
49 num_blobs = cvFindContours( img_bw,
50                             storage,
51                             &seq,
52                             sizeof( CvContour ),
53                             CV_RETR_LIST,
54                             CV_CHAIN_APPROX_NONE,
55                             cvPoint( 0, 0 ) ) - 1;
56
57 // Display the input / output windows and images
58 cvNamedWindow( "original" );
59 cvShowImage( "original", original );
60
61 cvNamedWindow( "grayscale" );
62 cvShowImage( "grayscale", grayscale );
63
64 cvNamedWindow( "black_and_white" );
65 cvShowImage( "black_and_white",
66             img_bw_cpy );
```

```
67
68 // Wait for user key press and then tidy up
69 cvWaitKey(0);
70
71 cvReleaseImage( &original );
72 cvReleaseImage( &grayscale );
73 cvReleaseImage( &img_bw );
74 cvReleaseImage( &img_bw_inv );
75 cvReleaseImage( &img_bw_cpy );
76
77 cvDestroyWindow( "greyscale" );
78 cvDestroyWindow( "black_and_white" );
79 cvDestroyWindow( "inverted" );
80
81 return 0;
82 }
```

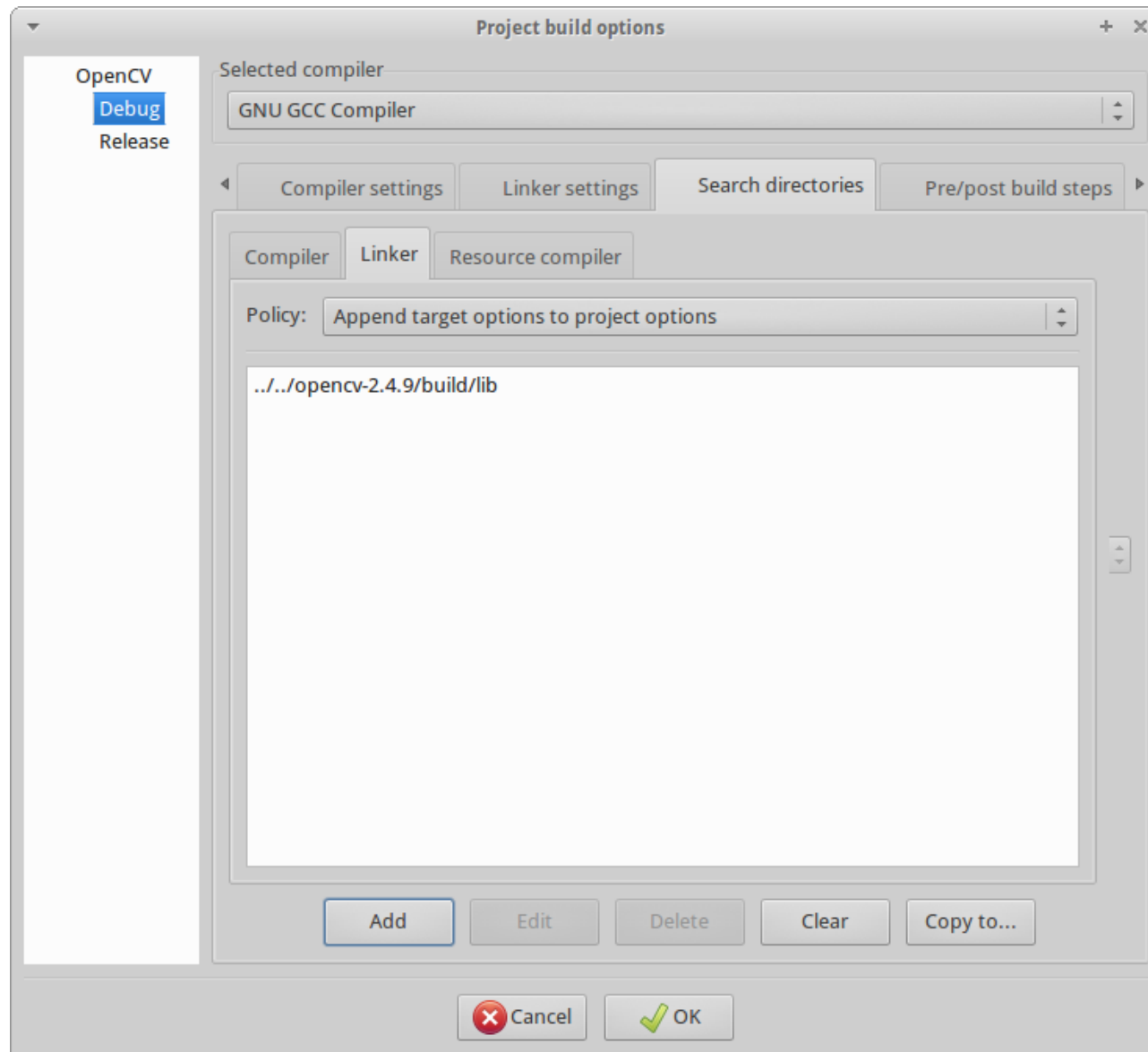
Step 5: Configure project settings

Right click the project folder and select Build Options...

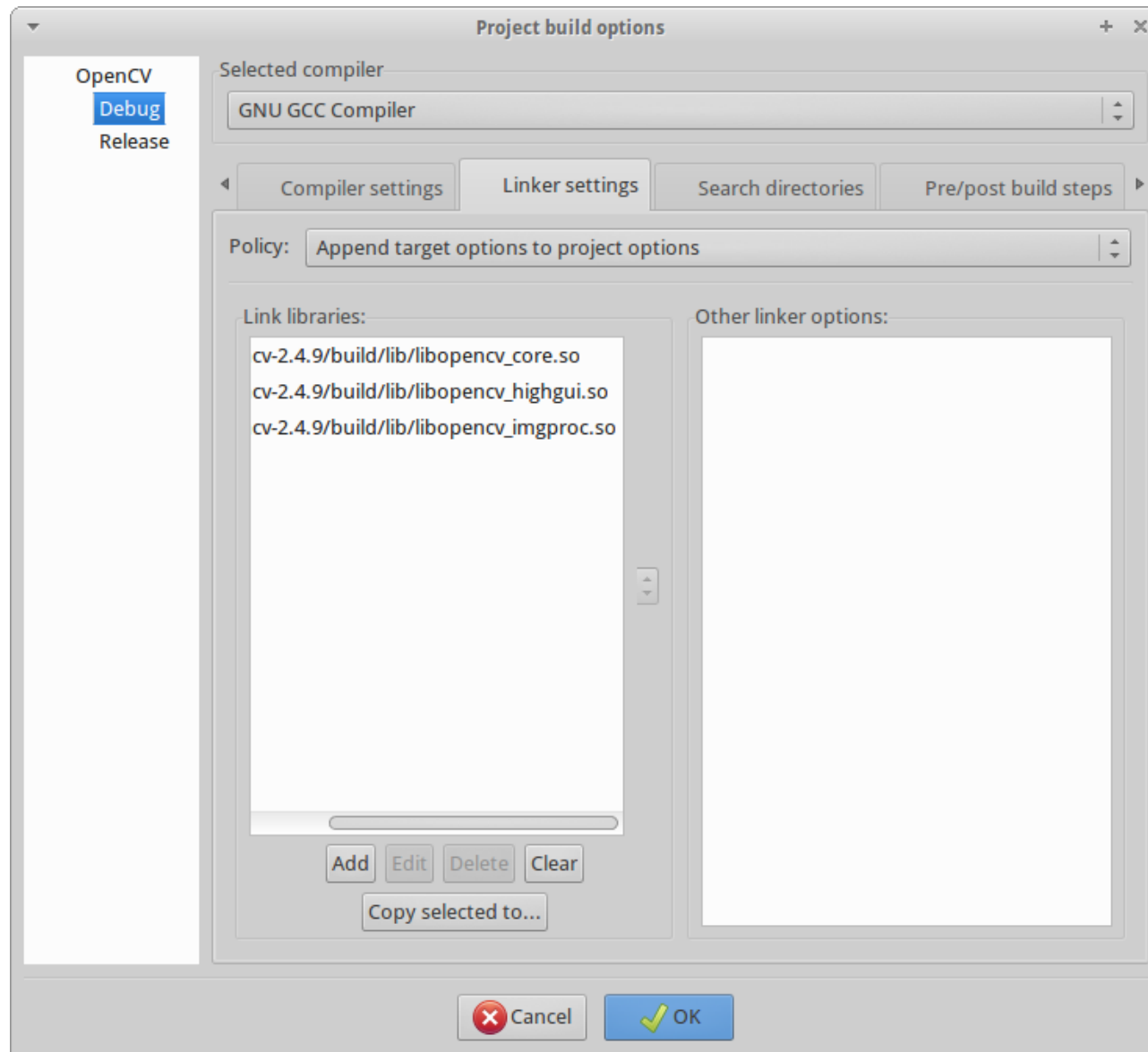
Select the Search Directories tab and then select the Compiler tab. Select the Add button in order to set the location of the include files:



Then select the Linker tab and then Add button in order to set the location of the OpenCV libraries, in the build/lib folder that was created using CMake earlier:



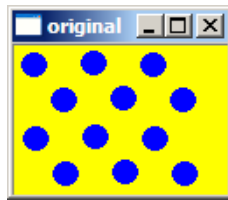
Select the Linker Settings tab and set the libraries that you will need to include, in this example these are core, highgui and imgproc:



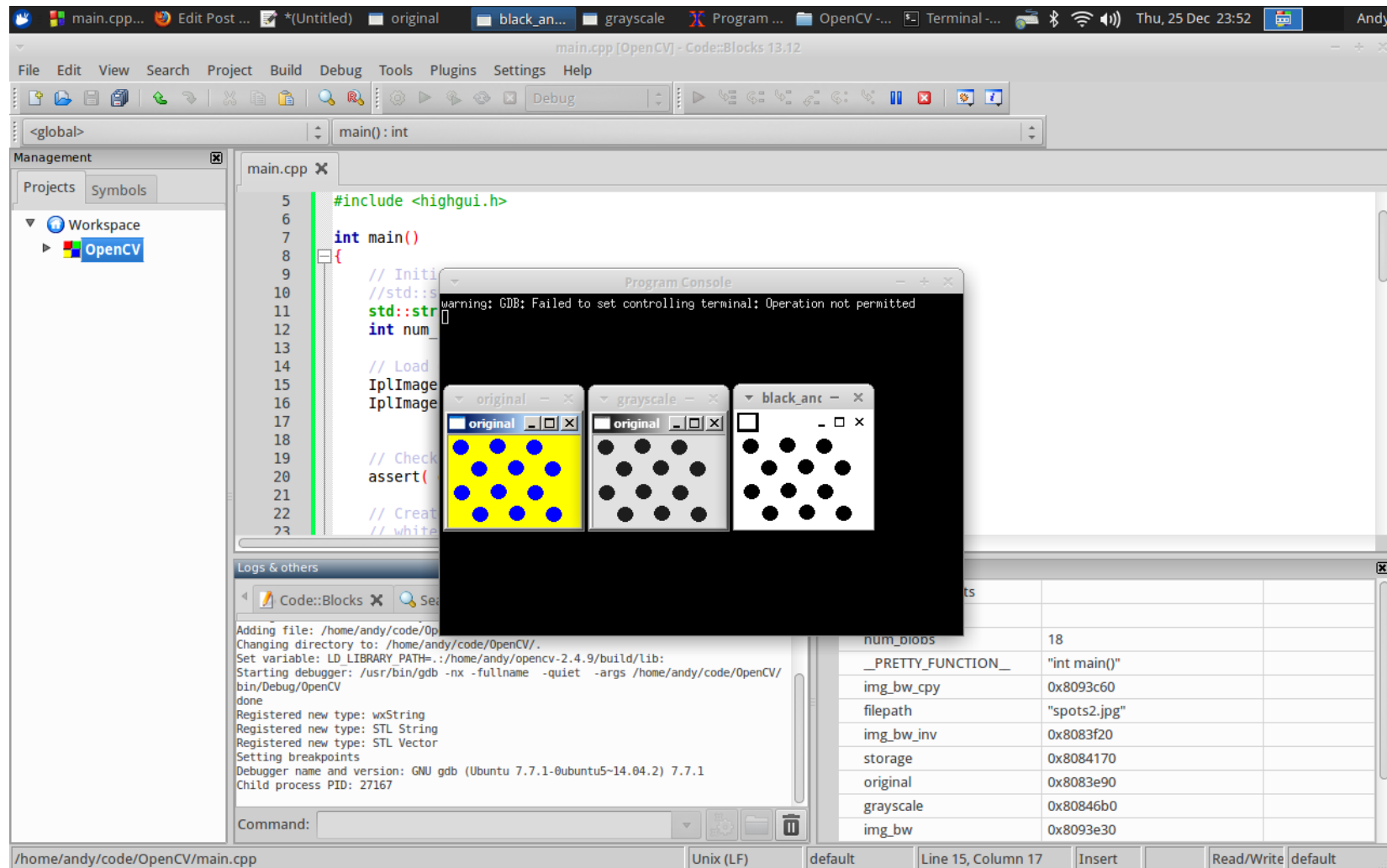
That should be sufficient to enable the project to build correctly.

Step 6: Try it!

This is the “spots2.jpg” image used in this test program (obviously, save the file in a place where your Code::Blocks project can find it):



And this screenshot shows the results of running the program, giving us the original colour image, the gray-scale equivalent image and the thresholded black and white image (and hence detect the number of spots in the jpg image):



Related Posts

No Preview

Using Boost regular expressions as word finders

No Preview

Sorting Objects Using STL

No Preview

Maintaining Recent File Lists in MFC Applications

About The Author



Andy

Latest Comments



Adarsh

13 NOVEMBER 2015

Works like a charm..thanks a lot 😊

[Reply](#)

Vishal



26 FEBRUARY 2016

Perfect tutorial! Thanks a lot 😊

[Reply](#)**Akarsh Seggemu**

13 SEPTEMBER 2016

Thanks for the tutorial

But is there any way that we could copy & paste project linker settings; as it looks tedious to link every library & CMakeLists.txt seems to be easy.

[Reply](#)**daniel**

22 SEPTEMBER 2016

Thanks a lot, I think it worth mentioning that 'make' process can be accelerated using 'make -jN' where N is a number of cores of your CPU.

[Reply](#)**Baltazar_le**

3 NOVEMBER 2016

Code:Block is telling me currently this, when i run the code: "Starting the debuggee failed: No executable file specified. Debugger finished with status 0"

And with this, I have 3 questions: Where can I get the spots2.jpg file from (I already saved the one from this webpage by clicking on webpage image by selecting save Image as) ? and Where do I save my image to (specified folder, maybe within code:blocks, within OpenCV) ?

Thank you

[Reply](#)

Pingback: [هسته های باز 30 - OpenCv در Ubuntu نحوه ی نصب](#) NOVEMBER 2016



Himj

4 OCTOBER 2017

showing error -undefined refernce to 'cvLoadImage'.
How can i fix this error?

[Reply](#)



Demir

10 MARCH 2018

I have the same Problem, and i don't find any solution, do you resolved it?

[Reply](#)



Demir

10 MARCH 2018

Resolved! I was using OpenCV 3.4.1, and in that release 'cvLoadImage' is in the library "imgcodecs" that has to be added at the linker settings.

[Reply](#)

**Andy**

10 MARCH 2018

Author

Thanks for the feedback this is very useful for others.

[Reply](#)**slim**

30 MARCH 2018

hello every one ! When I have save my image !

[Reply](#)

Leave a Reply

Comment

Comment Text*

Name*

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)



Categories

- 3rd Party Tools (21)
- Algorithms (2)
- Boost (13)
- C# / .Net / WPF (98)
- C++ / MFC / STL (109)
- Databases (2)
- Design Patterns (13)
- Development Environments (9)
- Encryption and Security (1)
- Genetic Algorithms (8)
- Graph Algorithms (11)
- Image Detection (9)

- [Installers \(25\)](#)
- [Java \(15\)](#)
- [Linux \(11\)](#)
- [OpenCV \(8\)](#)
- [Optimization \(15\)](#)
- [Python \(3\)](#)
- [Serialization \(3\)](#)
- [Smart Pointers \(7\)](#)
- [Version Control \(6\)](#)
- [Web development \(23\)](#)
- [Windows \(30\)](#)