Low Latency Displays Checkout Instructions

**Table of Contents**

[Repository Path](#h.w14a3f7ro08n)

[Additional Software](#h.qzfkerl0zedl)

[SVN Utilities](#h.zbou1px2uopk)

[Required Software & Libraries (Shared)](#h.c7ztfgg36wer)

[Required Software & Libraries (Not Shared)](#h.3wxi5d81rhzi)

[Recommended Software](#h.tu2jr2kvvqgs)

[Post Installation Steps](#h.tg4il664lza0)

[Additional Notes](#h.rhfcprqmcmbs)

[Building OpenCV 2.4.8 with CUDA Support](#h.7rko1kqnqbrp)

# Repository Path

All of the code is stored in a SVN repository. If SVN is not installed, then see [SVN Utilities](#h.zbou1px2uopk) for download instructions.

The main repository is accessible through the following URL:

<https://svn.cs.unc.edu/svn/LowLatencyDisplays>

When performing a checkout, make sure to include externals (via a checkbox or combobox in the checkout dialog box). The repository contains external references to another SVN repository ([MaskTracerLib](https://svn.cs.unc.edu/svn/MaskTracerLib)), and some projects will not compile without those external references being checked out.

# Additional Software

Compiling the code requires several external libraries, many of which I have put in a public (UNC CS authenticated user) downloads directory on my office computer

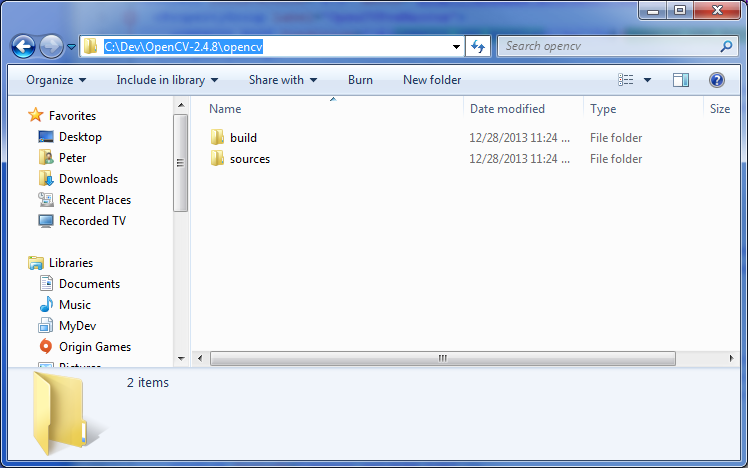
\\hires53.cs.unc.edu\Downloads

## SVN Utilities

* [TortoiseSVN](http://tortoisesvn.net/downloads.html) - Provides SVN support to Windows Explorer
  + \\hires53.cs.unc.edu\Downloads\SVN\TortoiseSVN
  + Install only x86 version on Windows x86. Install only x64 version on Windows x64.
  + Ensure that command line tools are installed
  + A reboot might be required after installing
* [AnkhSVN](https://ankhsvn.open.collab.net/downloads) - Provides SVN support to Visual Studio
  + \\hires53.cs.unc.edu\Downloads\SVN\VS Integration
  + Install this after installing all versions of Visual Studio you plan on using
  + After installing this, for each version of Visual Studio, verify that it is using AnkhSVN for source control (in Tools | Settings)

## Required Software & Libraries (Shared)

These items are available from the additional software folder listed above.

* [OpenCV](http://opencv.org/) 2.4.8
  + \\hires53\Downloads\OpenCV, though if you plan on using CUDA, you may want to download the source code from [here](http://opencv.org/opencv-2-4-8.html) and [build it yourself](#h.7rko1kqnqbrp)
  + Create environment variable OPENCV2\_ROOT248 : e.g. D:\Dev\OpenCV-2.4.8\opencv  
    

## Required Software & Libraries (Not Shared)

These items are available from other sources.

* Visual Studio 2010 Pro (or better), with all available service packs (SP1 for now) and updates
  + Installer: \\compsvc-cs.cs.unc.edu\CDs\Visual Studio 2010 (MSDNAA)
  + Also may be available on [DreamSpark](https://www.dreamspark.com/)
  + Service Packs/Updates are available via Microsoft Update, though if SP1 does not appear via Microsoft Update, then you can get it [here](http://www.microsoft.com/en-us/download/details.aspx?id=23691)
  + **Also apply the C++.NET Strong Name** [**workaround**](#id.3epddarfege8) **below**
* Visual Studio 2012 Pro (or better), with all available service packs and updates (Update 4 for now)
  + Installer is available via [DreamSpark](https://www.dreamspark.com/)
  + Update 4 is available [here](http://www.microsoft.com/en-us/download/details.aspx?id=39305)
  + Service Packs/Updates are available via Microsoft Update and through VS itself
  + [How to disable the ALL CAPS menus](http://stackoverflow.com/questions/10859173/how-to-disable-all-caps-menu-titles-in-visual-studio)
* [CUDA](https://developer.nvidia.com/cuda-zone) 6.0
  + Installer: <https://developer.nvidia.com/cuda-downloads>
  + Install after Visual Studio
  + Requires NVidia graphics card to run projects, but you can still install it on non-NVidia graphics computers (just skip installing the driver)
* Xilinx [ISE](http://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.htm) & [Vivado](http://www.xilinx.com/products/design-tools/vivado.html)
  + ISE Installer: <http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html>
  + Vivado Installer:  
    <http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools.html>
  + Look for “ISE Design Suite” and “Full Installer for Windows”
  + When installing, pick the maximal option

## Recommended Software

These items might not be required to build the projects, but are potentially useful.

* MATLAB 2011a
  + Old Installers Available Here: \\compsvc-cs\CDs
  + Newer Installers available via UNC ITS or the MathWorks website
  + The setup.exe installer will default to the same architecture as the host system (x86/x64).
    - If using Windows x64, ensure to install both the x86 and x64 versions.
    - The platform specific setup.exe’s are in subdirectories
    - Make sure to rename the shortcuts in all three places (desktop, start menu root, start menu subdirectory), after installing the first architecture, to avoid name collision when installing the second.
  + Also create two environment variables, e.g.:
    - MATLAB\_BIN32: C:\Program Files (x86)\MATLAB\R2011a\bin
    - MATLAB\_BIN64: C:\Program Files\MATLAB\R2011a\bin
  + To set up the Matlab build environment, some additional steps are required
    - Execute following command in each Matlab installed: mbuild -setup
    - Allow it to search for compilers.
    - When prompted for a compiler, select the Visual C++ 2012 (or version 11.0) compiler, and confirm it.
* [tangible T4 Editor](http://t4-editor.tangible-engineering.com/T4-Editor-Visual-T4-Editing.html)
  + Provides Intellisense for editing T4 Text-Template files within Visual Studio
  + Installers are here: [2010](http://visualstudiogallery.msdn.microsoft.com/60297607-5fd4-4da4-97e1-3715e90c1a23), [2012](http://visualstudiogallery.msdn.microsoft.com/b0e2dde6-5408-42c2-bc92-ac36942bbee9), [2013](http://visualstudiogallery.msdn.microsoft.com/6d1223ca-5e52-49d0-a489-910f9b76396e)
  + Install the versions corresponding to the version(s) of Visual Studio in use
* [NVIDIA Nsight](http://www.nvidia.com/object/nsight.html)
  + Provides OpenGL, DirectX, and CUDA debugging within Visual Studio
  + It is sometimes included with the CUDA installer, but it is also available here:
    - <https://developer.nvidia.com/nvidia-nsight-visual-studio-edition>
  + [Requires NVidia graphics card](https://developer.nvidia.com/nsight-visual-studio-edition-requirements)

# Post Installation Steps

* Visual Studio 2010 Strong Name Workaround
  + The ability to strongly name C++.NET assemblies is broken in RTM and SP1 versions of VS2010.
  + First, ensure that SP1 is installed
  + Second, apply the first workaround listed here (<http://blogs.msdn.com/b/vcblog/archive/2011/03/11/10140139.aspx>) to fix the typos in Microsoft.Cpp.Win32.targets. The edits to the project files have already been applied.
    - Alternatively, copy the fixed ...targets file from \\hires53.cs.unc.edu\Downloads\VS Service Packs\VS2010 to %ProgramFiles(x86)%\MSBuild\Microsoft.Cpp\v4.0\Platforms\Win32.

# Additional Notes

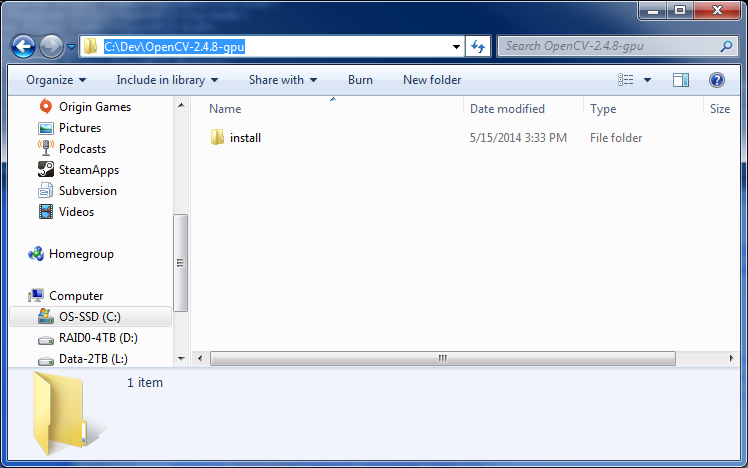
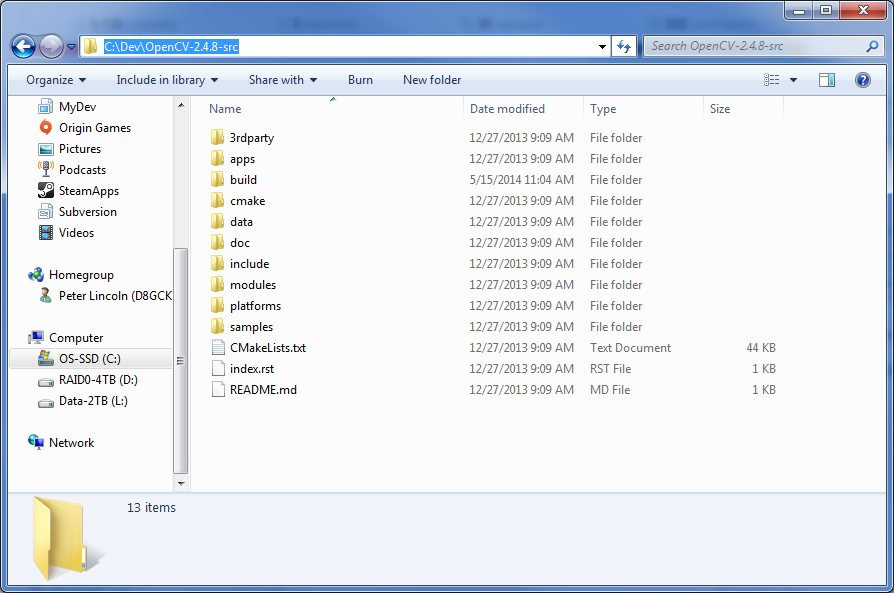
Note: the environment variables may need to be changed slightly depending on where you installed stuff. If you are using a 64-bit system install x64 versions instead of x86 versions where available. Not all parts of the codebase will compile to x64, though the x86 builds should still run on x64 platforms.

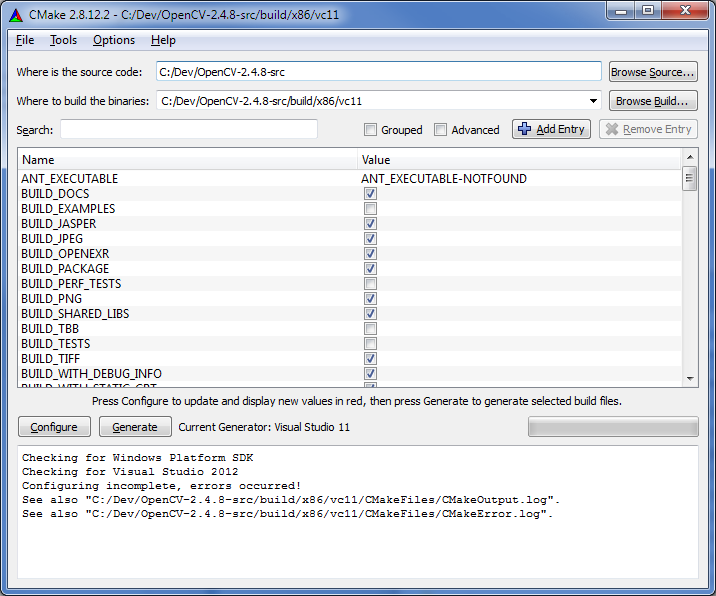
Similarly, on x64 systems, some (32-bit) tools will be installed to “C:\Program Files (x86)” instead of “C:\Program Files.”

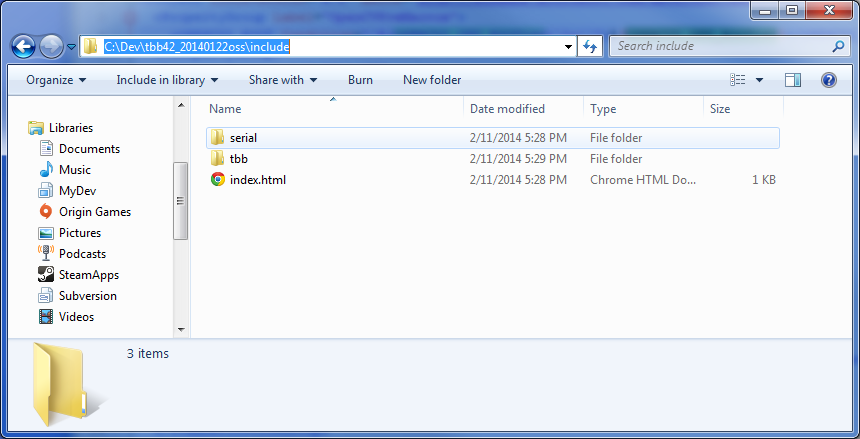
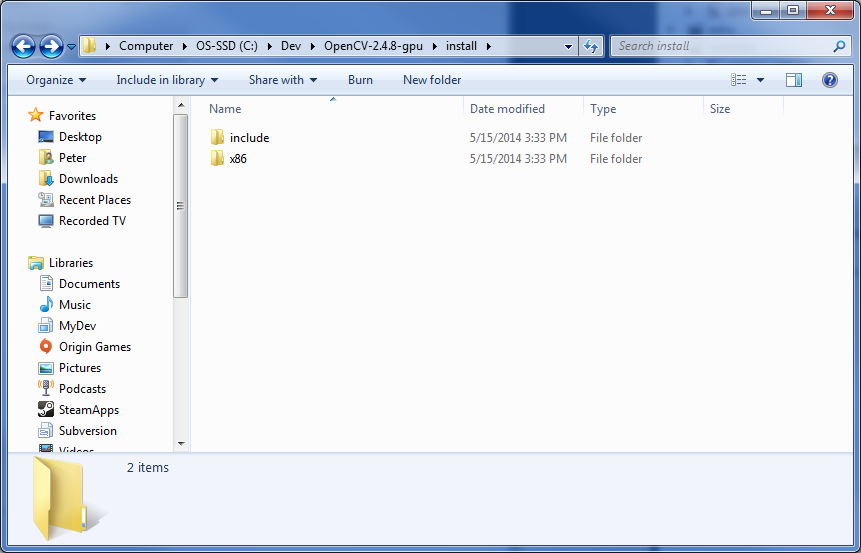
A reboot (or just a logoff/login cycle) may be required after creating environment variables for them to take effect.

# Building OpenCV 2.4.8 with CUDA Support

A copy of the source code can be downloaded by going [here](https://github.com/Itseez/opencv/tree/2.4.8), and then clicking on “Download ZIP” at the right side of the page. (The source code is also available in the sources subdirectory of %OPENCV2\_ROOT248%). You will need [CMake](http://www.cmake.org/) installed to generate the build files. You will also need [CUDA](#id.u5ut77snmvm1) installed and the [Intel Threading Building Blocks](https://www.threadingbuildingblocks.org/) extracted somewhere (e.g. C:\Dev\tbb42\_20140122oss).

1. Create a new empty directory to store all of the important build results (e.g. C:\Dev\OpenCV-2.4.8-gpu)
   1. Create environment variable OPENCV2\_GPU\_ROOT that points to this created directory
   2. Also create a subdirectory named install in this directory  
      
2. Extract the ZIP file to a new empty directory (e.g. C:\Dev\OpenCV-2.4.8-src)  
   
3. Run CMake
   1. Specify the directory to which you extracted the ZIP file contents as the location for “Where is the source code” (e.g. C:\Dev\OpenCV-2.4.8-src)
   2. Specify a subdirectory of that folder as “Where to build the binaries” using the following pattern: <root>/build/<PlatformName>/<VSVersion>, where
      1. <root> is “Where is the source code”
      2. <PlatformName> is either x86 or x64
      3. <VSVersion> is vc10 for 2010, vc11 for 2012, and vc12 for 2013



1. Click on Configure
   1. When prompted to pick a generator, select the appropriate version and platform of Visual Studio (10 is 2010, 11 is 2012, 12 is 2013) (<blank> is x86, x64 is x64)
2. After the progress completes, there will be a lot of red items in the list.
   1. Check the boxes of: WITH\_TBB, WITH\_CUDA, WITH\_OPENGL, INSTALL\_CREATE\_DISTRIB
   2. Uncheck the boxes of: BUILD\_PERF\_TESTS, BUILD\_TESTS, BUILD\_DOCS, WITH\_OPENCL
3. Click on Configure
4. After the progress completed, there will be one red item in the list at the top (TBB\_INCLUDE\_DIRS). Fill in the path to which you extracted the Intel Threading Building Blocks.  
   
5. Click on Configure
6. Click on Generate
7. Close CMake
8. In the folder that you specified as “Where to build the binaries,” there will be an OpenCV.sln file. Open it.
9. Build both the Release and Debug configurations of the “INSTALL” project. Each will take a lot of time (on the order of an hour+). For each configuration,
   1. First use “Build Solution”
   2. Then use “Build Project,” while the INSTALL project is selected
10. Close Visual Studio
11. Copy the install directory (located in the same place as OpenCV.sln) to your important build results directory (e.g. OPENCV2\_GPU\_ROOT or C:\Dev\OpenCV-2.4.8-gpu)  
    
12. [*Optional*] Delete the build directory, to free up disk space
13. Repeat steps 3-15 for each additional desired Visual Studio version and platform you require