
1 Overview

The AMD APP SDK is provided to the developer community to accelerate the programming in a heterogeneous environment by enabling AMD GPUs to work in concert with the system's x86 CPU cores. The SDK provides samples, documentation, and other materials to quickly get you started leveraging accelerated compute using OpenCL™, Bolt, OpenCV, or C++ AMP for your C/C++ application.

This document provides instructions on using the AMD APP SDK. The necessary prerequisite installations, environment settings, build and execute instructions for the samples are provided.

Review the following quick links to the important sections:

- [Section 2, “AMD APP SDK on Windows”](#)
 - [Section 2.1, “Installation”](#)
 - [Section 2.2, “General Prerequisites”](#)
 - [Section 2.3, “OpenCL”](#)
 - [Section 2.4, “BOLT”](#)
 - [Section 2.5, “C++ AMP”](#)
 - [Section 2.6, “OpenCV”](#)
- [Section 3, “AMD APP SDK on Linux”](#)
 - [Section 3.1, “Installation”](#)
 - [Section 3.2, “General prerequisites”](#)
 - [Section 3.3, “OpenCL”](#)
 - [Section 3.4, “BOLT”](#)
 - [Section 3.5, “OpenCV”](#)
- [Section Appendix A, “Important Notes”](#)
- [Section Appendix C, “CMAKE”](#)
- [Section Appendix D, “Building OpenCV from sources”](#)

2 AMD APP SDK on Windows

2.1 Installation

The AMD APP SDK 3.0 installer is delivered as a self-extracting installer for 32-bit and 64-bit systems on Windows. For details on how to install the AMD APP SDK on Windows, see the *AMD APP SDK Installation Notes* document. The default installation path is

```
C:\Users\<userName>\AMD APP SDK\<AMD APP SDK Version>\.
```

2.2 General Prerequisites

AMD APP SDK 3.0 is supported on the following Windows versions:

- Windows 10 (32-bit/64-bit)
- Windows 8.1 (32-bit/64-bit)
- Windows 7 (32-bit/64-bit)

The AMD APP SDK includes sample applications for OpenCL, Bolt, C++ AMP and and OpenCV-CL. To build and execute these samples, you must ensure that two sets of prerequisites are met: the common prerequisites that apply to all the samples, listed in this section; and the specific prerequisites required for the specific category of the samples, listed in the later sections of this document.

Before you build and execute the samples, ensure that you have installed the following:

- Microsoft Visual Studio 2012 redistributable package
This package is required for running the pre-built binaries of the samples. Installing the AMD Catalyst Driver installs the Visual Studio 2012 redistributable by default.
- Microsoft Visual Studio version 2010 and/or higher
This package is required for building and executing the samples.
- CMake version 2.8.0 or higher (optional)
CMake is used for generating the AMD APP SDK sample make files or Visual Studio projects. In addition to Visual Studio solution/project files for samples, the AMD APP SDK includes CMake files. CMake supports creating `make` files across different platforms and generating project files across different IDEs including Visual Studio.
For details on using CMake to generate make files or Visual Studio files for the AMD APP SDK samples, see [Appendix C, “CMAKE”](#).

Note: The AMD APP SDK 3.0 package for Windows includes Visual Studio 2010, 2012, and 2013 projects.

2.3 OpenCL

2.3.1 Prerequisites

In addition to the common prerequisites mentioned in [Section 2.2, “General Prerequisites”](#), to build and run OpenCL samples that use OpenGL and `zsd`, you must install one of the following:

- Windows SDK 8.0 or above

2.3.2 How to run the pre-built samples

The AMD APP SDK ships with pre-built binaries of OpenCL samples. To execute the samples, you must perform the following steps:

- Open a command prompt.
- Change the directory to the appropriate architecture directory (x86 or x86_64) within `<<AMDAPPSDKSamplesInstallPath>>\samples\opencl\bin`.
- Run the samples by typing the name of the executables.
To review the command line arguments for samples, see the respective sample documents.

2.3.3 How to rebuild the samples

Building with Visual Studio Solution files –

The AMD APP SDK installation includes a master Visual Studio Solution file for OpenCL samples. This solution file contains Visual Studio projects of all the OpenCL samples. In the current version of AMD APP SDK, master solution files for Microsoft Visual Studio 2010, 2012, and 2013 are provided. If during the installation of the SDK, custom installation of OpenCL 1.x/2.0 was chosen, then only the master solution files specific to the custom selection are installed. These files are located at `<<AMDAPPSDKSamplesInstallPath>>\samples\opencl\`.

To build a sample:

- Open the required master solution file.
- To build all the projects, select Build > Build Solution.
To build a specific project, select the project file in the Solutions Explorer and select Build to build the particular sample.

Building with Visual Studio Solution files by using the Intel Compiler (icl) –

- Right-click on a project file, and select Properties.
- Under Configuration Properties | General, change the Platform Toolset item to Intel C++ Compiler, and click OK.
- To build the sample, right-click on the project file, and select Build.

2.4 BOLT

2.4.1 Prerequisites

In addition to the common prerequisites mentioned in [Section 2.2, “General Prerequisites”](#), to build and run the BOLT samples you must ensure the following installations and environment variable settings:

- Microsoft Visual Studio 2012 or higher for Bolt C++AMP samples and Microsoft Visual Studio 2010 or higher for Bolt OpenCL samples
- The TBB library: <http://threadingbuildingblocks.org/download>.
 - Required for running the Bolt samples using the multi-core CPU path.

- Download and install the Bolt 1.3 library: <http://developer.amd.com/tools-and-sdks/opencv-zone/opencv-libraries/bolt-c-template-library/>. Bolt can also be built from the *github* sources: <https://github.com/HSA-Libraries/Bolt>.

Environment Variables –

Set the following environment variables:

- Set the `TBB_ROOT` environmental variable to the root directory of the installed TBB binaries.
- Set the `BOLT_LIB_DIR` environmental variable to the root directory to which Bolt is extracted. For example, if VS 2012 Bolt 1.3 is downloaded, the path will be: `<Bolt Install Path>\Bolt-1.3-VS2012\Bolt-1.3-VS2012\`.
- Append the `PATH` environment variable with the directory containing all the TBB binaries. For example, on a 64-bit machine with VS 2012, this path will be: `%TBB_ROOT%\bin\intel64\vc11\`.

2.4.2 How to run the pre-built samples

The AMD APP SDK ships with pre-built binaries of BOLT samples. To execute the samples, you must perform the following steps:

- Open a command prompt.
- Change the directory to the appropriate directory (x86 or x86_64) within `<<AMDAPPSDKSamplesInstallPath>>\samples\bolt\bin`.
- Run the samples by typing the name of their executables. To review the command line arguments for samples, see the respective sample documents.

2.4.3 How to rebuild the samples

The AMD APP SDK installation includes a master Visual Studio Solution file for Bolt samples. This solution file contains Visual Studio projects of all the Bolt samples. In the current version of AMD APP SDK, master solution files for Visual Studio 2010, 2012 and 2013 projects of Bolt samples are provided. These files are located at `$<AMDAPPSDKSamplesInstallPath>\samples\bolt\`.

To build a sample:

- Open the required master solution file.
- Select the appropriate build configuration, `Debug` or `Release`.
- To build TBB-enabled Bolt samples, select the corresponding build configuration, `Debug_TBB` or `Release_TBB`. To build all the projects, select `Build > Build Solution`. To build a specific project, select the project file in the Solutions Explorer and select `Build` to build the sample.

2.5 C++ AMP

2.5.1 Prerequisites

C++ AMP is supported only on Microsoft Visual Studio 2012 and higher versions. Also, C++ AMP samples do not work on Linux. Therefore, to build and run C++AMP samples, install Microsoft Visual Studio 2012 or higher.

2.5.2 How to run the pre-built samples

The AMD APP SDK ships with pre-built binaries of the C++AMP samples. To execute the samples, you must perform the following steps:

- Open a command prompt.
- Change the directory to the appropriate architecture directory (x86 or x86_64) within `<<AMDAPPSDKSamplesInstallPath>>\samples\C++Amp\bin`.
- Run the samples by typing the name of the executables.
To review the command line arguments for samples, see the respective sample documents.

2.5.3 How to rebuild the samples

The AMD APP SDK installation includes a master Visual Studio Solution file for the C++AMP samples. This solution file contains Visual Studio projects of all the C++AMP samples. In the current version of AMD APP SDK, master solution files for Visual Studio 2012 and 2013 projects of C++AMP samples are provided. These files are located at `$<AMDAPPSDKSamplesInstallPath>\samples\C++Amp\`.

To build a sample:

- Open the required master solution file.
- Select the appropriate build configuration, `Debug` or `Release`.
- To build all the projects, select `Build > Build Solution`.
To build a specific project, select the project file in the Solutions Explorer and select `Build` to build the sample.

2.6 OpenCV

2.6.1 Prerequisites

The AMD APP SDK 3.0 OpenCV-CL samples work with OpenCV 2.4.9. Download the OpenCV 2.4.9 binaries for Windows from: <http://opencv.org/>.

The Windows download includes the pre-built OpenCV binaries. However, you can also create binaries by downloading the source files and building them, as described in [Appendix D, "Building OpenCV from sources"](#). For answers to frequently asked questions on OpenCV, see the *AMD APP SDK FAQ* document.

Environment Variables –

Before building and running the OpenCV-CL samples, you must set the following environment variables:

- Create and set the environmental variable, `OPENCV_DIR`, to the root directory containing the OpenCV include and lib files that are extracted from the downloaded OpenCV package.
- Create and set the environmental variable, `OCVCL_VER`, to the OpenCV version used for AMD APP SDK 3.0 release, that is, set `OCVCL_VER = 249`.
- Append the `PATH` environment variable with the directory containing all the OpenCV .dll files. For example, for a 64-bit machine with VS 2012, this path will be `%OPENCV_DIR%\x64\vc11\bin`.

OpenNI Libraries –

The GestureRecognition AMD APP SDK OpenCV-CL sample makes use of OpenNI libraries to extract video frames. OpenNI framework is an open source SDK used for the development of 3D sensing middleware libraries and applications. The OpenNI SDK can be downloaded from <http://structure.io/openni>.

You must set the following environment variables:

1. For 32-bit builds, add `OPENNI2_REDIST` to the `PATH` environment variable.
For 64-bit builds, add `OPENNI2_REDIST64` to the `PATH` environment variable.
2. Header and library paths will be added by the OpenNI Windows installer. If those paths are missing from the system, then the user must set the following environment variables:
 - i. For 32-bit platforms:
 - Set `OPENNI2_INCLUDE` to `<<OPENNI-INSTALL_PATH>>\Include`
 - Set `OPENNI2_LIB` to `<<OPENNI-INSTALL_PATH>>\Redist`
 - Add `OPENNI2_LIB` to the `PATH` environment variable
 - ii. For 64-bit platforms:
 - Set `OPENNI2_INCLUDE64` to `<<OPENNI-INSTALL_PATH>>\Include`
 - Set `OPENNI2_LIB64` to `<<OPENNI-INSTALL_PATH>>\Redist`
 - Add `OPENNI2_LIB64` to the `PATH` environment variable

The GestureRecognition sample currently works on only Windows platforms.

2.6.2 How to run the pre-built samples

The AMD APP SDK ships with pre-built binaries of the OpenCV-CL samples. To execute the samples, you must perform the following steps:

- Open a command prompt.
- Change the directory to the appropriate architecture directory (x86 or x86_64) within `<<AMDAPPSDKSamplesInstallPath>>\samples\opencv\bin`.
- Run the samples by typing the name of the executables.
To review the command line arguments for samples, see the respective sample documents.

2.6.3 How to rebuild the samples

The AMD APP SDK installation includes a master Visual Studio Solution file for the OpenCV-CL samples. This solution file contains Visual Studio projects of all the OpenCV-CL samples. In the current version of AMD APP SDK, master solution files for Visual Studio 2010, 2012 and 2013

projects of OpenCV-CL samples are provided. These files are located at
`$(AMDAPPSDKSamplesInstallPath)\samples\opencv\.`

To build a sample:

- Open the required master solution file.
- Select the appropriate build configuration, `Debug` or `Release`.
- To build all the projects, select `Build > Build Solution`.
To build a specific project, select the project file in the Solutions Explorer and select `Build` to build the sample.

3 AMD APP SDK on Linux

3.1 Installation

The AMD APP SDK 3.0 installer is delivered as a self-extracting installer for 32-bit and 64-bit systems on Linux. For details on how to install AMD APP SDK on Linux, see the [AMD APP SDK Installation Notes](#) document.

3.2 General prerequisites

The AMD APP SDK 3.0 is supported on the following Linux flavors:

- Ubuntu[®] 14.04 (32-bit/64-bit)
- Red Hat[®] Enterprise Linux[®] 6.x (32-bit/64-bit)

The AMD APP SDK Linux package includes sample applications for OpenCL, Bolt and OpenCV-CL. To build and execute these samples, you must ensure that two sets of prerequisites are met: the common prerequisites that apply to all the samples, listed in this section; and the specific prerequisites required for the specific category of the samples, listed in the later sections of this document.

Before you build and execute the samples, ensure that you have installed the following:

- `gcc`
 - The AMD APP SDK samples have been compiled with `gcc 4.8.4`
- CMake version 2.8.0 or higher
 - For details on using CMake to generate make files for the AMD APP SDK samples, see [Appendix C, "CMAKE"](#).

3.3 OpenCL

3.3.1 Prerequisites

In addition to the common prerequisites mentioned in [Section 3.2, "General prerequisites"](#), to build and run the OpenCL samples you must ensure the following installation:

- OpenGL package `libglu1-mesa-dev`
 - Required to run AMD APP SDK samples that use OpenGL library
 - Ensure that the AMD Catalyst Driver is installed in order to run samples using OpenGL
 - Ensure that the `libGL.so.1` file is linked to `fglrx-libGL.so.1.2`, which is found in:
 - Ubuntu: `/usr/lib/fglrx/` and `/usr/lib32/fglrx/`
 - RHEL: `/usr/lib64/fglrx/`.
 The sample execution fails when linked to:
 - Ubuntu: `/usr/lib/x86_64-linux-gnu/mesa/libGL.so.1`, `/usr/lib/i386-linux-nu/mesa/libGL.so.1`
 - RHEL: `/usr/lib/libGL.so.1`, `/usr/lib64/libGL.so.1`.

3.3.2 How to run the pre-built samples

The AMD APP SDK ships with pre-built binaries of OpenCL samples. To execute the samples, you must perform the following steps:

- Open a terminal window.
- Change the directory to the appropriate architecture directory (x86 or x86_64) within `<<AMDAPPSDKSamplesInstallPath>>/samples/opengl/bin`.
- Run the samples by typing the name of the executables. You may need to prepend the executable name with `./`.
To review the command line arguments for samples, see the respective sample documents.

Note: The prebuilt samples on Linux have been compiled with `GCC 4.8.4`.

3.3.3 How to rebuild the samples

To compile the AMD APP SDK OpenCL samples, the CMake build tool is required. To build all the OpenCL samples, run the following commands:

```
$> cmake -G "Unix Makefiles"
"<AMDAPPSDKSamplesInstallPath>/samples/opengl/"
$> make
```

To build individual samples, run CMake from the respective sample source directory.

For more details on using CMake to generate make files for the AMD APP SDK samples, see [Appendix C, "CMAKE"](#).

3.4 BOLT

3.4.1 Prerequisites

In addition to the common prerequisites mentioned in [Section 3.2, "General prerequisites"](#), to build and run the BOLT samples you must ensure the following installations and environment variable settings:

- The TBB library: <http://threadingbuildingblocks.org/download>.

- Required for running the Bolt samples using the multi-core CPU path.
- Download and install the Bolt 1.3 library for Linux from: <http://developer.amd.com/tools-and-sdks/opencl-zone/opencl-libraries/bolt-c-template-library/>. Bolt can also be built from the *github* sources: <https://github.com/HSA-Libraries/Bolt>.

Environment Variables –

Set the following environment variables:

- Set and export the `TBB_ROOT` environmental variable to the root directory of the installed TBB binaries.
- Set and export the `BOLTLIB_DIR` environmental variable to the root directory to which Bolt is extracted.
- Append the `LD_LIBRARY_PATH` environment variable with the directory containing all the TBB binaries. For example, on a 64-bit machine, this path can be:
`/home/<user>/downloads/TBB/tbb42_xxxxxyzzoss/lib/intel64/gcc<majorVersion>.minorVersion>`.

3.4.2 How to run the pre-built samples

The AMD APP SDK ships with pre-built binaries of BOLT samples. To execute the samples, you must perform the following steps:

- Open a terminal window.
- Change the directory to the appropriate directory (x86 or x86_64) within `<<AMDAppSDKSamplesInstallPath>>/samples/bolt/bin`.
- Run the samples by typing the name of their executables. You may need to prepend the name of the executable with `./`.
To review the command line arguments for samples, see the respective sample documents.

3.4.3 How to rebuild the samples

To compile the AMD APP SDK Bolt samples, the CMake build tool is required. To build all the OpenCL samples, run the following commands:

```
$> cmake -G "Unix Makefiles"
"<AMDAppSDKSamplesInstallPath>/samples/bolt/"
$> make
```

To build individual samples, run CMake from the respective sample source directory.

To build a Bolt sample with the TBB multicore option enabled, use the `ENABLE_TBB` flag.

For example:

```
$> cmake -G "Unix Makefiles" -DENABLE_TBB=ON
"<AMDAppSDKSamplesInstallPath>samples/bolt/BoltSort/"
$> make
```

For more details on using CMake to generate make files for the AMD APP SDK samples, see [Appendix C, "CMAKE"](#).

3.5 OpenCV

3.5.1 Prerequisites

In addition to the common prerequisites mentioned in [Section 3.2, “General prerequisites”](#), to build and run the OpenCV-CL samples you must ensure the following installations and environment variable settings:

Installations –

- Download the OpenCV 2.4.9 source from: <http://opencv.org/>.
The AMD APP SDK 3.0 OpenCV-CL samples work with OpenCV 2.4.9.
- Compile the downloaded source files to generate the OpenCV binaries, as described in [Appendix D, “Building OpenCV from sources”](#). For answers to frequently asked questions on OpenCV, see the *AMD APP SDK FAQ* document.
- Install the `Libgtk2.0-dev` and `pkg-config` packages.

Environment Variables –

Before building and running the AMD APP SDK OpenCV-CL samples, you must set the following environment variables after generating the OpenCV binaries. Assuming that the generated OpenCV binaries are generated under the `=/usr/local/` directory:

- Set and export `OPENCV_DIR` to the root directory containing the OpenCV include and lib files. For example, `OPENCV_DIR=/usr/local/`.
- Set and export `OCVCL_VER` to the OpenCV version used for the AMD APP SDK 3.0 release, i.e., set `OCVCL_VER = 249`.
- Set `LD_LIBRARY_PATH` to the directory containing the OpenCV library files. For example, assign `LD_LIBRARY_PATH` to `/usr/local/lib`.

3.5.2 How to run the pre-built samples

The AMD APP SDK ships with pre-built binaries of the OpenCV-CL samples. To execute the samples, you must perform the following steps:

- Open a terminal window.
- Change the directory to the appropriate architecture directory (x86 or x86_64) within `<<AMDAppSDKSamplesInstallPath>>/samples/opencv/bin`.
- Run the samples by typing the name of the executables. You may need to prepend the name of the executable with `./`.
To review the command line arguments for samples, see the respective sample documents.

3.5.3 How to rebuild the samples

To compile the AMD APP SDK OpenCV-CL samples, the CMake build tool is required. To build all the OpenCL samples, run the following commands:

```
$> cmake -G "Unix Makefiles"  
"<AMDAppSDKSamplesInstallPath>/samples/opencv/"  
$> make
```

To build individual samples, run CMake from the respective sample source directory.

For more details on using CMake to generate make files for the AMD APP SDK samples, see [Appendix C, "CMAKE"](#).

Appendix A Important Notes

- Unless specifically recommended otherwise, developers must use the latest graphics drivers for their platform. These drivers can be downloaded from <http://support.amd.com/en-us/download>.

For current recommendations, see <http://developer.amd.com/AMDAPPSDK>.

Appendix B Supported Devices

AMD is continually qualifying devices. For an up-to-date list of supported devices, see <http://developer.amd.com/AMDAPPSDK>.

Appendix C CMAKE

Getting started with CMake:

CMake supports creating make files across different platforms and generating project files across different IDEs including Visual Studio and Eclipse. In order to use CMake with AMD APP SDK, the CMake binaries must be downloaded.

1. CMake binaries can be found at: <http://www.cmake.org/cmake/resources/software.html>.
2. The following page describes how to build a project for Windows and Linux using CMake: <http://www.cmake.org/cmake/help/runningcmake.html>.

Quick links to download:

1. For Windows: Download the latest CMake from: <http://www.cmake.org/cmake/resources/software.html>.
2. For Linux: The "sudo apt-get install cmake" command will install the cmake binaries.

Building an AMD APP SDK sample using CMake:

Before using CMake to generate Visual Studio projects or makefiles for any sample, ensure that all the environment variables required for the particular sample are set. For example, if CMake is to be used for any Bolt sample, all the environment variables corresponding to Bolt headers and libraries must be set. The later sections in this document specify which environment variables must be set for each category of samples.

a. Windows: Generating Visual Studio project files –

Developers can create Visual Studio project files for the sample individually or for all of them together. To create the project files individually, CMake must be run by specifying the corresponding sample directory. For example, if you want to create the Visual studio 2012 solution file for the AtomicCounters OpenCL sample, then the corresponding command is:

For 32 bit:

```
cmake.exe -G "Visual Studio 11" "<path to the AtomicCounters sample source>"
```

For 64 bit:

```
cmake.exe -G "Visual Studio 11 Win64" "<path the to the AtomicCounters sample source>"
```

It is recommended to create another directory under the sample and run CMake from there. CMake creates a few additional files and projects apart from the main sample project. Hence running CMake from a sub-directory will keep the project files separate from the sample source files. If run from a subdirectory, the command will change as follows:

```
cmake.exe .. -G "Visual Studio 11" "<path to the sub-directory inside the AtomicCounters sample source folder>"
```

To build all the OpenCL samples, run the following command

For 32 bit:

```
cmake -G "Visual Studio 11" "<AMDAPPSDKSamplesInstallPath>/samples/openc1/"
```

For 64 bit:

```
cmake -G "Visual Studio 11 Win64"  
"<AMDAPPSDKSamplesInstallPath>/samples/openc1/"
```

Alternatively, `cmake-gui` can also be used to build the samples.

b. Linux: Generating Makefiles –

To generate the Makefiles, the same commands mentioned in the Windows section can be used except that the "Generator" name must be changed from the Visual Studio variant to the Unix variant.

For example, to build all the OpenCL samples, run the following command:

```
cmake -G "Unix Makefiles" "<AMDAppSDKSamplesInstallPath>/samples/openc1/"
```

CMAKE AMD APP SDK-SPECIFIC OPTIONS:

1. `BUILD_OPENCL`: ON/OFF - Builds OpenCL samples. Default is ON.
2. `BUILD_OPENCV`: ON/OFF - Builds OpenCV samples. Default is ON.
3. `BUILD_AMP`: ON/OFF - Builds C++ AMP samples. Default is ON.
4. `BUILD_BOLT`: ON/OFF - Builds Bolt samples. Default is ON.
5. `ENABLE_TBB`: ON/OFF - Builds all Bolt samples with the `multicore` option enabled. Default is OFF.

For example, to build all the samples except OpenCV samples, use `-DBUILD_OPENCV=OFF` as an argument to `cmake` or deselect the corresponding box in `cmake-gui`.

```
cmake -G "Unix Makefiles" -DBUILD_OPENCV=OFF <AMDAPPSDKSamplesInstallPath>
```

Note: The AMD APP SDK samples have been tested with the following generators in CMake

1. Visual Studio 2010
2. Visual Studio 2012

3. Visual Studio 2013
5. Unix Makefiles

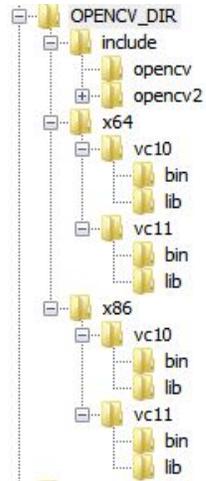
Appendix D Building OpenCV from sources

To build the OpenCV binaries from the sources, perform the following actions:

Building OpenCV from sources

To build the OpenCV binaries from the sources, perform the following actions:

1. Build the OpenCV library from the source files.
The OpenCV sources for the 2.4.9 build are available at <https://github.com/Itseez/opencv/tree/2.4.9>. To have OpenCL support for OpenCV, the `opencv_ocl` library must be built. During configuring with CMake, select the `WITH_OPENCL` option and provide the path of the OpenCL library (`libOpenCL.so` in Linux and `OpenCL.lib` in Windows). The following links from `opencv.org` are useful:
 - Linux: http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html
 - Windows:
http://docs.opencv.org/doc/tutorials/introduction/windows_install/windows_install.html
2. Set the correct directory structure.
On Windows, the directory structure of the compiled OpenCV binaries created in the preceding step must be restructured as per the OpenCV prebuilt directory structure. Restructure the OpenCV binaries created from sources as per the prebuilt directory structure for Windows as shown in the following figure. Place the created `include` directly under the root OpenCV directory and copy the `bin` and `lib` files (both debug and release versions) into the following locations:
 - `x86/vc10`, if your binaries and libraries are built with Microsoft Visual Studio 10 and the target is x86.
 - `x86/vc11`, if your binaries and libraries are built with Microsoft Visual Studio 12 and the target is x86.
 - `x86/vc12`, if your binaries and libraries are built with Microsoft Visual Studio 13 and the target is x86.
 - `x64/vc10` if your binaries and libraries are built with Microsoft Visual Studio 10 and the target is x64.
 - `x64/vc11` if your binaries and libraries are built with Microsoft Visual Studio 12 and the target is x64.
 - `x64/vc12` if your binaries and libraries are built with Microsoft Visual Studio 13 and the target is x64.



Alternatively, you may skip restructuring the directory as per the OpenCV pre-builts, in which case, the Visual Studio property sheets of the OpenCV-CL projects must be updated to point to the correct paths.

For Linux, retain the default directory structure of the OpenCV binaries created from the sources.

Contact

Advanced Micro Devices, Inc.
One AMD Place
P.O. Box 3453
Sunnyvale, CA, 94088-3453
Phone: +1.408.749.4000

For AMD APP SDK:
URL: developer.amd.com/appsdk
Developing: developer.amd.com/



The contents of this document are provided in connection with Advanced Micro Devices, Inc. ("AMD") products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. The information contained herein may be of a preliminary or advance nature and is subject to change without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD's Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Copyright and Trademarks

© 2015 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, ATI, the ATI logo, Radeon, FireStream, and combinations thereof are trademarks of Advanced Micro Devices, Inc. OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos. Other names are for informational purposes only and may be trademarks of their respective owners.