

# Altera SDK for OpenCL

## Getting Started Guide



Subscribe



Send Feedback

**OCL001-15.0.0**  
2015.05.04

101 Innovation Drive  
San Jose, CA 95134  
[www.altera.com](http://www.altera.com)



# Contents

<b>Altera SDK for OpenCL Getting Started Guide.....</b>	<b>1-1</b>
Prerequisites for the Altera SDK for OpenCL.....	1-1
Contents of the Altera SDK for OpenCL.....	1-2
Overview of the Altera SDK for OpenCL Setup Process.....	1-4
<b>Getting Started with the Altera SDK for OpenCL for Windows.....</b>	<b>2-1</b>
Downloading the Altera SDK for OpenCL.....	2-2
Installing the Altera SDK for OpenCL.....	2-2
Setting the Environment Variables for Windows.....	2-3
Verifying the AOCL Installation.....	2-4
Licensing the Software.....	2-5
Installing an FPGA Board.....	2-6
Creating the FPGA Hardware Configuration File of an OpenCL Kernel.....	2-7
Updating the Hardware Image on the FPGA.....	2-8
Querying the Device Name of Your FPGA Board.....	2-9
Programming the Flash Memory of an FPGA on Windows.....	2-9
Executing an OpenCL Kernel on an FPGA.....	2-10
Building the Host Application.....	2-11
Running the Host Application.....	2-11
Output from Successful Kernel Execution.....	2-11
Uninstalling the Software.....	2-12
Uninstalling the FPGA Board.....	2-12
<b>Getting Started with the Altera SDK for OpenCL for Linux.....</b>	<b>3-1</b>
Downloading the Altera SDK for OpenCL.....	3-2
Installing the Altera SDK for OpenCL.....	3-2
Setting the Environment Variables for Linux.....	3-3
Verifying the AOCL Installation.....	3-4
Licensing the Software.....	3-5
Installing an FPGA Board.....	3-5
Creating the FPGA Hardware Configuration File of an OpenCL Kernel.....	3-6
Updating the Hardware Image on the FPGA.....	3-8
Querying the Device Name of Your FPGA Board.....	3-8
Programming the Flash Memory of an FPGA on Linux.....	3-8
Executing an OpenCL Kernel on an FPGA.....	3-9
Building the Host Application.....	3-10
Running the Host Application.....	3-10
Output from Successful Kernel Execution.....	3-10
Uninstalling the Software.....	3-11
Uninstalling the FPGA Board.....	3-11

**Document Revision History..... A-1**

2015.05.04

OCL001-15.0.0



Subscribe



Send Feedback

The *Altera SDK for OpenCL Getting Started Guide* describes the procedures you follow to install the Altera® Software Development Kit (SDK) for OpenCL™ (AOCL)<sup>(1)</sup>. This document also contains instructions on how to compile an example OpenCL<sup>(2)</sup> application with the AOCL.

OpenCL is a C-based open standard for the programming of heterogeneous parallel devices. For more information on the OpenCL Specification version 1.0, refer to the OpenCL Reference Pages. For detailed information on the OpenCL application programming interface (API) and programming language, refer to the *OpenCL Specification version 1.0*.

The AOCL provides a compiler and tools for you to build and run OpenCL applications that target Altera FPGAs. The AOCL supports the embedded profile of the OpenCL Specification version 1.0.

**Attention:** If you only require the AOCL kernel deployment functionality, download and install the Altera Runtime Environment (RTE) for OpenCL. Refer to the *Altera RTE for OpenCL Getting Started Guide* for more information.

Do not install the AOCL and the RTE on the same host system.

**Attention:** If you want to use the AOCL with the Cyclone® V SoC Development Kit, refer to the *Altera SDK for OpenCL Cyclone V SoC Getting Started Guide* for more information.

## Related Information

- [OpenCL Reference Pages](#)
- [OpenCL Specification version 1.0](#)
- [Altera RTE for OpenCL Getting Started Guide](#)
- [Altera SDK for OpenCL Cyclone V SoC Getting Started Guide](#)

## Prerequisites for the Altera SDK for OpenCL

To install the AOCL and create an OpenCL application for an Altera preferred accelerator board, your system must meet certain hardware, target platform, and software requirements.

<sup>(1)</sup> The Altera SDK for OpenCL is based on a published Khronos Specification, and has passed the Khronos Conformance Testing Process. Current conformance status can be found at [www.khronos.org/conformance](http://www.khronos.org/conformance).

<sup>(2)</sup> OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission of the Khronos Group™.

## Hardware Requirements

### 1. Accelerator boards requirements:

- Acquire a Reference Platform from Altera, or a Custom Platform from an Altera preferred board vendor.

For more information, refer to the OpenCL Reference Platforms page on the Altera website.

### 2. Development system requirements:

- You must have administrator privileges on the development system to install the necessary packages and drivers.
- The development system has at least 24 gigabytes (GB) of RAM.

**Tip:** Refer to board vendor's documentation on the recommended system storage size.

- For PCI Express® (PCIe®) accelerator boards, the host machine motherboard must have an available PCIe port slot that is at least the same width (that is, the same number of PCIe lanes) as the board.

### 3. The host system must be running one of the following supported target platforms:

- Microsoft 64-bit Windows 7 on the x86-64 architecture.
- Red Hat Enterprise 64-bit Linux (RHEL) version 6 on the x86-64 architecture.

**Important:** For x86\_64 Linux systems, install the Linux OS kernel source and headers (for example, **kernel-devel.x86\_64** and **kernel-headers.x86\_64**), and the GNU Compiler Collection (GCC) (**gcc.x86\_64**).

To install the Linux kernel source or header package, invoke the `yum install <kernel_package_name>` command.

You must have administrator privileges on the host system to install the necessary packages and drivers.

## Software Prerequisites

### 1. Develop your host application using one of the following AOCL- and Quartus® II software-compatible C compiler or software development environment:

- For Windows systems, use Microsoft Visual Studio version 2010 Professional.
- For Linux systems, use the C compiler included with the GCC.

### 2. Linux systems require the Perl command version 5 or later. Ensure that your *PATH* environment variable setting includes the path to the Perl command.

## Related Information

[OpenCL Platforms page on the Altera website](#)

# Contents of the Altera SDK for OpenCL

The AOCL provides logic components, drivers, and AOCL-specific libraries and files.



## Logic Components

- The *Altera Offline Compiler* (AOC) translates your OpenCL device code into a hardware configuration file that the system loads onto an Altera FPGA.
- The *AOCL utility* includes a set of commands you can invoke to perform high-level tasks such as running diagnostic tests.
- The *host runtime* provides the OpenCL host platform application programming interface (API) and runtime API for your OpenCL host application.

The host runtime consists of libraries that provide OpenCL APIs, hardware abstractions, and helper libraries.

## Drivers, Libraries and Files

The software installation process installs the software into a directory that you own. The `ALTERAOCLSDKROOT` environment variable references the path to the software installation directory.

**Table 1-1: Select Contents of the AOCL Installation Directory**

Windows Folder	Linux Directory	Description
<b>bin</b>	<b>bin</b>	User commands in the AOCL. Include this directory in your <i>PATH</i> environment variable setting.
<b>board</b>	<b>board</b>	The AOCL Custom Platform Toolkit and Reference Platforms available with the software. <ul style="list-style-type: none"> <li>• The path to the Custom Platform Toolkit is <b><code>ALTERAOCLSDKROOT/board/custom_platform_toolkit</code></b></li> <li>• The path to the s5_ref Reference Platform is <b><code>ALTERAOCLSDKROOT/board/s5_ref</code></b></li> </ul>
<b>ip</b>	<b>ip</b>	Intellectual property (IP) cores used to compile device kernels.
<b>host</b>	<b>host</b>	Files necessary for compiling and running your host application.
<b>host\include</b>	<b>host/include</b>	OpenCL Specification version 1.0 header files and software interface files necessary for compiling and linking your host application.  The <b>host/include/CL</b> subdirectory also includes the C++ header file <b>cl.hpp</b> . The file contains an OpenCL version 1.1 C++ wrapper API. These C++ bindings enable a C++ host program to access the OpenCL runtime APIs using native C++ classes and methods.  <b>Important:</b> The OpenCL version 1.1 C++ bindings are compatible with OpenCL Specification versions 1.0 and 1.1.  Add this path to the <b>include</b> file search path in your development environment.
<b>host\windows64\lib</b>	<b>host/linux64/lib</b>	OpenCL host runtime libraries that provide the OpenCL platform and runtime APIs. These libraries are necessary for linking your host application.  To run an OpenCL application on Linux, include this directory in the <code>LD_LIBRARY_PATH</code> environment variable setting.

Windows Folder	Linux Directory	Description
<b>host\ windows64\bin</b>	<b>host/linux64/bin</b>	Runtime commands and libraries necessary for running your host application, wherever applicable. For 64-bit Windows system, include this directory in your <i>PATH</i> environment variable setting.  For Windows system, this folder only contains runtime libraries.  For Linux system, this directory contains platform-specific binary for the <code>aocl</code> utility command.
<b>share</b>	<b>share</b>	Architecture-independent support files.

### Example OpenCL Applications

You can download example OpenCL applications from the OpenCL Design Examples page on the Altera website.

#### Related Information

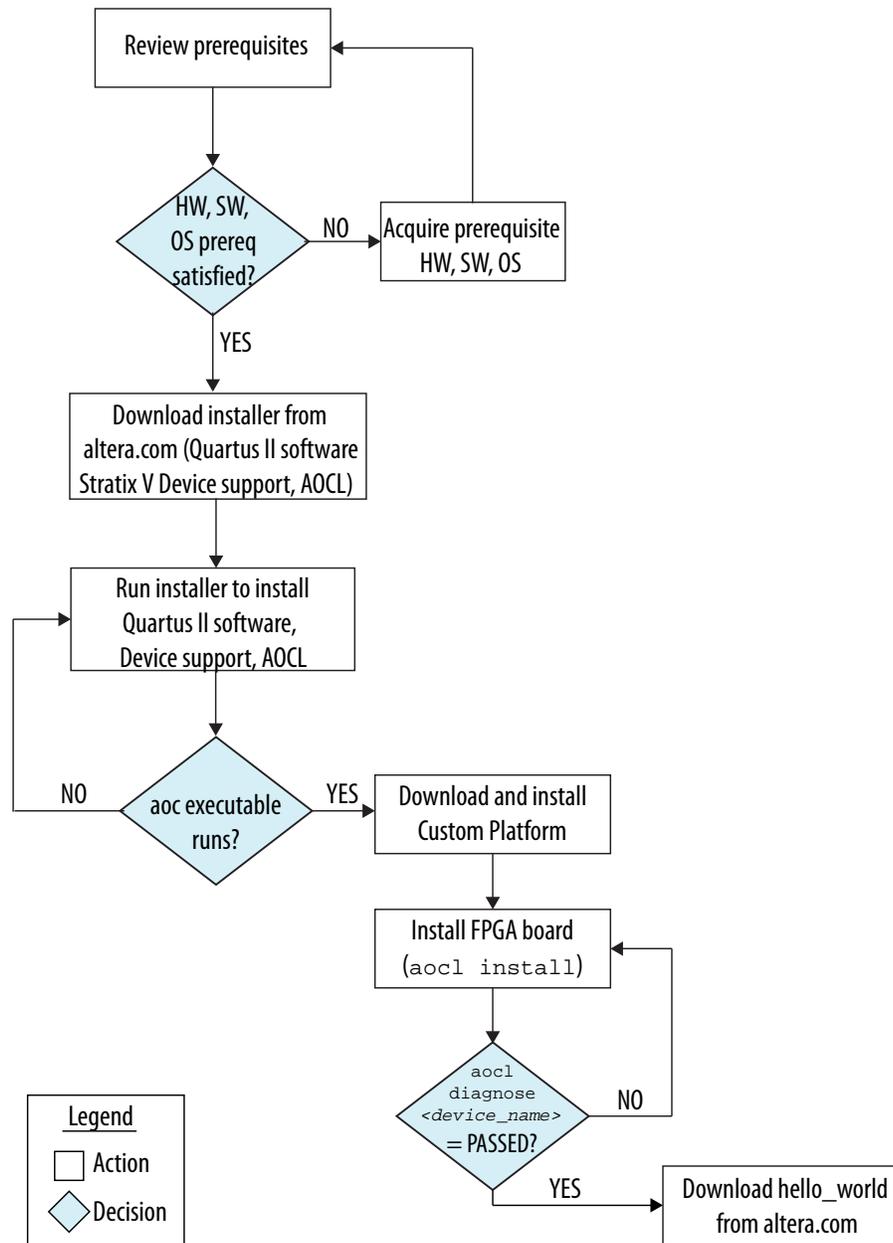
[OpenCL Design Examples page on the Altera website](#)

## Overview of the Altera SDK for OpenCL Setup Process

The *Altera SDK for OpenCL Getting Started Guide* outlines the procedures for installing the AOCL and programming your FPGA.

The figure below summarizes the steps for setting up the necessary software and installing the FPGA board.

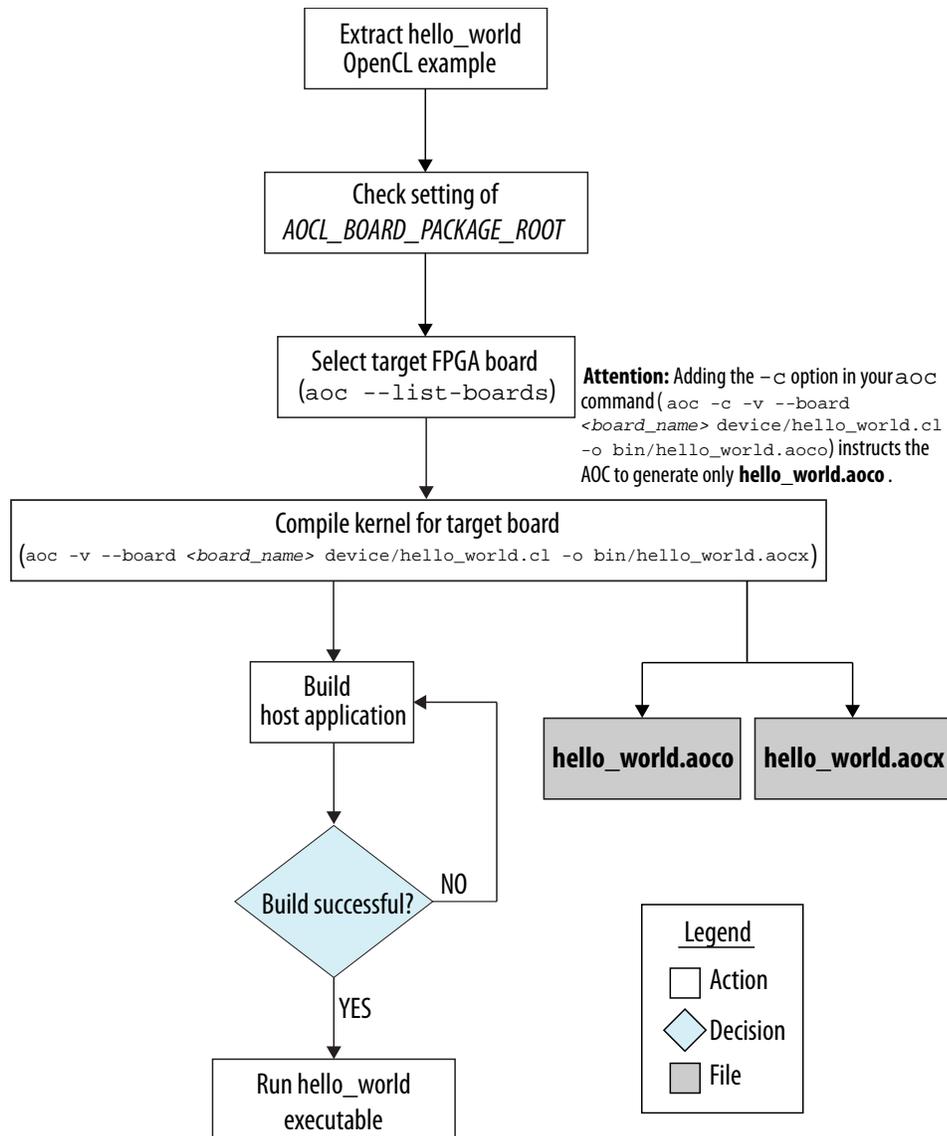
Figure 1-1: AOCL Installation Process Overview



After you complete the initial software and hardware setup successfully, you can create a hardware image from the *hello\_world* OpenCL design example. Before you program your FPGA with the hardware image, ensure that your FPGA contains an image created using a current version of the AOCL.

The figure below summarizes the steps you perform to program your FPGA.

Figure 1-2: FPGA Programming Overview



# Getting Started with the Altera SDK for OpenCL for Windows

# 2

2015.05.04

OCL001-15.0.0



Subscribe



Send Feedback

The AOCL setup process includes downloading and installing the software, installing the FPGA board, and then executing an OpenCL kernel on the FPGA.

- 1. Downloading the Altera SDK for OpenCL** on page 2-2  
Download the AOCL and all related software for Windows from the Download Center on the Altera website.
- 2. Installing the Altera SDK for OpenCL** on page 2-2  
Install the AOCL in a folder that you own.
- 3. Setting the Environment Variables for Windows** on page 2-3  
You have the option to set the AOCL environment variables permanently or transiently.
- 4. Verifying the AOCL Installation** on page 2-4  
Invoke the `version` utility command and verify that the correct version of the AOCL is installed.
- 5. Licensing the Software** on page 2-5  
Obtain the AOCL license from the Altera Self Service Licensing Center.
- 6. Installing an FPGA Board** on page 2-6  
To install your board into the host system, invoke the `install` utility command.
- 7. Creating the FPGA Hardware Configuration File of an OpenCL Kernel** on page 2-7  
To create the `.aocx` file for the `hello_world` example OpenCL application, you must download the example design for Windows from the OpenCL Design Examples page on the Altera website. Extract the downloaded source files and then compile the kernel using the Altera Offline Compiler (AOC).
- 8. Updating the Hardware Image on the FPGA** on page 2-8  
If applicable, before you execute an OpenCL kernel program on the FPGA, ensure that the flash memory of the FPGA contains a hardware image created using a current version of the OpenCL software.
- 9. Executing an OpenCL Kernel on an FPGA** on page 2-10  
Build your OpenCL host application in Microsoft Visual Studio version 2010 Professional, and run the application by invoking the `hello_world.exe` executable.
- 10. Uninstalling the Software** on page 2-12  
To uninstall the AOCL for Windows, run the uninstaller, and restore all modified environment variables to their previous settings.
- 11. Uninstalling the FPGA Board** on page 2-12  
To uninstall an FPGA board for Windows, invoke the `uninstall` utility command, uninstall the Custom Platform, and unset the relevant environment variables.

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at [www.altera.com/common/legal.html](http://www.altera.com/common/legal.html). Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO  
9001:2008  
Registered

## Downloading the Altera SDK for OpenCL

Download the AOCL and all related software for Windows from the Download Center on the Altera website. Download the AOCL and all related software into a single installation package. Alternatively, if you already have the current version of the Quartus II software in your system, download the AOCL as a stand-alone software.

Download the following software and files:

- Quartus II software
- Device support (for example, Stratix® V device support)
- AOCL

1. In the main page of the Altera website, click **MYALTERA** and log into your account.  
If you do not have a myAltera account, register for a new account.
2. Click **DOWNLOADS** to enter the Download Center.
3. Click **Altera SDK for OpenCL** to enter the download page for the subscription edition of the AOCL.
4. Select the software version. The default selection is the current version.
5. Select the **Akamai DLM3 Download Manager** or **Direct Download** as the download method.

By default, when you select **Akamai DLM3 Download Manager**, all the available software and device support will be selected for download. Choose this download method if you need to install all the files.

If you do not require all the files, select **Direct Download** and download only the files that you need. Choose this download method if you already have the current version of the Quartus II software and device support installed on your machine.

6. Click the **Windows SDK** tab and then click **More** beside **Download and install instructions** to view the download and installation procedure.
7. Perform the steps outlined in the download and installation instructions on the download page.

### Related Information

[Altera website](#)

## Installing the Altera SDK for OpenCL

If you download the AOCL, the Quartus II software, and the device support into a single installation package, run the Quartus II software installer for Windows to install all the software and files. If you download the AOCL as a stand-alone software, run the AOCL installer. Install the AOCL in a folder that you own.

### Before you begin

You must have administrator privileges.

- To install the AOCL, Quartus II software, and device support files simultaneously, perform the following tasks:
  1. Run the **QuartusSetup-<version>-windows.exe** file to launch the installation wizard. Perform the tasks outlined in the wizard.
  2. The installer sets the user environment variable *ALTERAOCLSDKROOT* to point to the path of the software installation. To verify the *ALTERAOCLSDKROOT* setting, from the Windows main menu, click **Control Panel > System and Security > System > Advanced system settings**. In the **System Properties** pop-up window, click the **Environment Variables** button to open the **Environment Variables** pop-up window. The *ALTERAOCLSDKROOT* environment variable and its setting appears under **User variables for <user\_name>**. Edit the value if the path does not point to the AOCL installation.
  3. In the **Environment Variables** pop-up window, verify that the *PATH* environment variable setting includes the **bin** folder in the Quartus II system folder (for example, *<home\_directory>\altera\<version>\quartus\bin*). Edit the value as needed.
- To install the AOCL as a stand-alone software, perform the following tasks:
  1. Ensure that your system already has the current version of the Quartus II software and applicable device support files.
  2. Ensure that the *PATH* environment variable includes the **bin** folder in the Quartus II system folder (for example, *<home\_directory>\altera\<version>\quartus\bin*).
  3. Run the **AOCLSetup-<version>-windows.exe** installer. Direct the installer to extract the software to an empty folder that you own (that is, not a system folder).  
The installation path must not contain any spaces (for example, *<home\_directory>\altera\<version>\hld*).
  4. The installer sets the environment variable *ALTERAOCLSDKROOT*. Ensure that *ALTERAOCLSDKROOT* points to the current version of the software.

## Setting the Environment Variables for Windows

You have the option to set the AOCL environment variables permanently or transiently. The environment variable settings describe the FPGA board and the host runtime to the software.

**Attention:** If you set the environment variables permanently, you apply the settings once during installation. If you set the environment variables transiently, you must apply the settings during installation and during every subsequent session you run.

- To apply permanent environment variable settings, perform the following tasks:
  1. In the Windows **Start** menu, click **Control Panel > System and Security > System**.
  2. In the **System** window, click **Advanced system settings**. In the **Advanced** tab of the **System Properties** dialog box, click **Environment Variables**.
  3. In the **Environment Variables** dialog box, include the following paths in the corresponding environment variable settings:

Environment Variable	Path to Include
PATH	<ol style="list-style-type: none"> <li>1. %ALTERAOCLSDKROOT%\bin</li> <li>2. %ALTERAOCLSDKROOT%\host\windows64\bin</li> </ol> <p>where <i>ALTERAOCLSDKROOT</i> points to the path of the software installation</p>

- To apply transient environment variable settings, open a command window and run the **%ALTERAOCLSDKROOT%\init\_opencl.bat** script.

Example script output:

```
AOCL_BOARD_PACKAGE_ROOT path is not set in environment
Setting to default s5_ref board.
If you want to target another board, do
  set AOCL_BOARD_PACKAGE_ROOT=board_pkg_dir
and re-run this script
Adding %ALTERAOCLSDKROOT%\bin to PATH
Adding %ALTERAOCLSDKROOT%\host\windows64\bin to PATH
Adding %AOCL_BOARD_PACKAGE_ROOT%\windows64\bin to PATH
```

where *AOCL\_BOARD\_PACKAGE\_ROOT* points to the path of the Custom or Reference Platform.

Running the **init\_opencl.bat** script only affects the current command window. The script performs the following tasks:

- Finds the Microsoft Visual Studio installation
- Imports the Microsoft Visual Studio environment to properly set the *LIB* environment variable
- Ensures that the *PATH* environment variable includes the path to the Microsoft **LINK.EXE** file

## Verifying the AOCL Installation

Invoke the `version` utility command and verify that the correct version of the AOCL is installed.

- At a command prompt, invoke the `aocl version` utility command.  
An output similar to the one below notifies you of a successful installation:

```
aocl <version>.<build> (Altera SDK for OpenCL, Version <version> Build <build>,
Copyright (C) <year> Altera Corporation)
```

- If installation was unsuccessful, reinstall the software. You can also refer to the *Altera Software Installation and Licensing* manual and the Knowledge Center on the Altera website for more information.

#### Related Information

- [Altera Software Installation and Licensing](#)
- [Altera Knowledge Base](#)

## Licensing the Software

Obtain the AOCL license from the Altera Self Service Licensing Center. The AOCL license allows the software to access relevant Quartus II software functionalities and device support without an additional Quartus II software license.

### Before you begin

For information on the licensing options and requirements, refer to the *Licensing Altera Software* section of the *Altera Software Installation and Licensing* manual, the **LICENSE.txt** file that accompanies each software, and the Altera Licensing page on the Altera website.

1. In the main page on the Altera website, click **MYALTERA** and log into your account.
2. Click the link to the Self-Service Licensing Center.
3. Perform the steps outlined in the *Using the Self-Service Licensing Center* section of the *Altera Software Installation and Licensing* manual to obtain and activate the license.
4. If you have a fixed license, append the `<path_to_license_file>/<license_filename>` file to the `LM_LICENSE_FILE` environment variable in the following manner:
  - a. Create a backup copy of the provided license file.
  - b. Save the new license file on your local hard drive.
  - c. Append the `<path_to_license_file>/<license_filename>` file to the `LM_LICENSE_FILE` environment variable.
5. If you have a floating licence, append the `<path_to_license_file>/<license_filename>` file to the `LM_LICENSE_FILE` environment variable in the following manner:
  - a. Obtain the port number and host name from the network or system administrator. Alternatively, the information is in the license file line `SERVER <hostname> <8 to 12 character host or NIC ID> <port>`.  
The license location for the user is `<port>@<hostname>`. If a port is not listed in the license file, specify `@<hostname>`.
  - b. Modify the license file to update the port number and host name.
  - c. Append the `<path_to_license_file>/<license_filename>` file to the `LM_LICENSE_FILE` environment variable setting.

**Related Information**

- [Altera Licensing page on the Altera website](#)
- [Using the Self-Service Licensing Center](#)
- [Altera website](#)

## Installing an FPGA Board

Before creating an OpenCL application for an FPGA boards, you must first download and install the Custom Platform from your board vendor. Most Custom Platform installers require administrator privileges. To install your board into the host system, invoke the `install` utility command.

The steps below outline the board installation procedure. Some Custom Platforms require additional installation tasks. Consult your board vendor's documentation for further information on board installation.

**Attention:** If you are installing the Cyclone V SoC Development Kit for use with the Cyclone V SoC Development Kit Reference Platform, refer to *Installing the Cyclone V SoC Development Kit in the Altera SDK for OpenCL Cyclone V SoC Getting Started Guide* for more information.

1. Follow your board vendor's instructions to connect the FPGA board to your system.
2. Download the Custom Platform for your FPGA board from your board vendor's website.  
For more information, refer to the OpenCL Reference Platforms page on the Altera website.
3. Install the Custom Platform in a folder that you own (that is, not a system folder).
4. Set the environment variable `AOCL_BOARD_PACKAGE_ROOT` to point to the location of the Custom Platform subfolder containing the `board_env.xml` file.

For example, for the Stratix V Network Reference Platform (`s5_net`), set `AOCL_BOARD_PACKAGE_ROOT` to point to the `<path_to_s5_net>\s5_net` folder.

5. Add the Custom Platform library paths to the `PATH` environment variable setting. You may apply permanent settings manually by adding the path to the memory-mapped (MMD) library within the Custom Platform. Alternatively, you may apply transient settings to the current session by running the `%ALTERAOCLSDKROOT%\init_opencl.bat` script.

For example, if you use `s5_net`, the Windows `PATH` environment variable setting is `%AOCL_BOARD_PACKAGE_ROOT%\windows64\bin`.

For information on `init_opencl.bat`, refer to the *Setting the Environment Variables for Windows* section.

6. Invoke the command `aocl install` at a command prompt.  
Invoking `aocl install` also installs a board driver that allows communication between host applications and hardware kernel programs.
7. To query a list of FPGA devices installed in your machine, invoke the `aocl diagnose` command. The software generates an output that includes the `<device_name>`, which is an `acl` number that ranges from `acl0` to `acl15`.

For more information on querying the `<device_name>` of your accelerator board, refer to the *Querying the Device Name of Your FPGA Board* section.

8. To verify the successful installation of the FPGA board, invoke the command `aocl diagnose <device_name>` to run any board vendor-recommended diagnostic test.

#### Related Information

- [Installing the Cyclone V SoC Development Kit](#)
- [Querying the Device Name of Your FPGA Board](#) on page 2-9
- [OpenCL Platforms page on the Altera website](#)
- [Setting the Environment Variables for Windows](#) on page 2-3

## Creating the FPGA Hardware Configuration File of an OpenCL Kernel

The FPGA hardware configuration file of an OpenCL kernel is called an *Altera Offline Compiler Executable file (.aocx)*. To create the `.aocx` file for the `hello_world` example OpenCL application, you must download the example design for Windows from the OpenCL Design Examples page on the Altera website. Extract the downloaded source files and then compile the kernel using the Altera Offline Compiler (AOC).

After you successfully install your FPGA board, you can create a `.aocx` file that executes on the device. The steps below describe the process of creating a `.aocx` file from the `hello_world` example design. For more information on the OpenCL design examples, refer to the OpenCL Design Examples page on the Altera website.

1. In the OpenCL Design Examples page, under Basic Examples, click **Hello World**.
2. In the Hello World Design Example page, under Downloads, click **Windows package (.zip)** to download the compressed file for your platform.
3. Unzip the `exm_openc1_hello_world_x64_windows_<version>.zip` file and save it in a folder to which you have write access.
4. Ensure that you have installed the Custom Platform provided by your board vendor successfully. Refer to *Installing an FPGA Board (install)* for more information.
5. Ensure that the environment variable `AOCL_BOARD_PACKAGE_ROOT` points to a local subfolder within your Custom Platform that contains the `board_env.xml` file.
6. Select your target FPGA board. To list the FPGA boards available in your Custom Platform, invoke the command `aoc --list-boards` at a command prompt.

For more information on the `--list-boards` option of the `aoc` command, refer to the *Listing the Available FPGA Boards in Your Custom Platform (--list-boards)* section of the *Altera SDK for OpenCL Programming Guide*.

7. At a command prompt, navigate to the `hello_world` design.
8. To compile the kernel for your target FPGA board, invoke the following command:  

```
aoc -v --board <board_name> device/hello_world.cl -o bin/hello_world.aocx
```

This command performs the following tasks:

- Generates the Quartus II design project files from the OpenCL source code.
- Checks for initial syntax errors.
- Performs basic optimizations.
- Creates a **hello\_world** folder containing necessary intermediate files.
- Creates the *Altera Offline Compiler Object file (.aoco)*.
- Creates the **.aocx** hardware configuration file.

**Attention:** The **.aocx** file might take hours to build, depending on the complexity of the kernel. To view the progress of the compilation on-screen, include the `-v` flag in your `aoc` command. An example output is shown below.

```
aoc: Environment checks are completed successfully.
aoc: You are now compiling the full flow!!
aoc: Selected target board <board_name>
aoc: Running OpenCL parser....
aoc: OpenCL parser completed successfully.
aoc: Compiling....
aoc: Linking with IP library ...
aoc: First stage compilation completed successfully.
aoc: Setting up project for CvP revision flow....
aoc: Hardware generation completed successfully.
```

The AOC displays the line `aoc: Hardware generation completed successfully.` to signify the completion of the compilation process.

For more information on the `--board <board_name>` option of the `aoc` command, refer to the *Compiling a Kernel for a Specific FPGA Board (--board <board\_name>)* section of the *Altera SDK for OpenCL Programming Guide*.

For more information on the `-v` option of the `aoc` command, refer to the *Generating Compilation Progress Report (-v)* section of the *Altera SDK for OpenCL Programming Guide*.

For more information on the `-o <filename>` option of the `aoc` command, refer to the *Specifying the Name of an AOC Output File (-o <filename>)* section of the *Altera SDK for OpenCL Programming Guide*.

#### Related Information

- [OpenCL Design Examples page on the Altera website](#)
- [Installing an FPGA Board](#) on page 2-6
- [Listing the Available FPGA Boards in Your Custom Platform \(--list-boards\)](#)
- [Compiling a Kernel for a Specific FPGA Board \(--board <board\\_name>\)](#)
- [Generating Compilation Progress Report \(-v\)](#)
- [Specifying the Name of an AOC Output File \(-o <filename>\)](#)

## Updating the Hardware Image on the FPGA

If applicable, before you execute an OpenCL kernel program on the FPGA, ensure that the flash memory of the FPGA contains a hardware image created using a current version of the OpenCL software.

**Remember:** If your Custom Platform requires that you preload a valid OpenCL image into the flash memory, for every major release of the Altera Complete Design Suite, program the flash

memory of the FPGA with a hardware image compatible with the current version of the software.

## Querying the Device Name of Your FPGA Board

Some AOCL utility commands require you to specify the device name (*<device\_name>*). The *<device\_name>* refers to the acl number (e.g. acl0 to acl15) that corresponds to the FPGA device. When you query a list of accelerator boards, the AOCL produces a list of installed devices on your machine in the order of their device names.

- To query a list of installed devices on your machine, type `aocl diagnose` at a command prompt. The software generates an output that resembles the example shown below:

```
aocl diagnose: Running diagnostic from ALTERAOCLSDKROOT/board/<board_name>/
<platform>/libexec

Verified that the kernel mode driver is installed on the host machine.

Using board package from vendor: <board_vendor_name>
Querying information for all supported devices that are installed on the host
machine ...

device_name  Status  Information

acl0         Passed  <descriptive_board_name>
           PCIe dev_id = <device_ID>, bus:slot.func = 02:00.00,
           at Gen 2 with 8 lanes.
           FPGA temperature=43.0 degrees C.

acl1         Passed  <descriptive_board_name>
           PCIe dev_id = <device_ID>, bus:slot.func = 03:00.00,
           at Gen 2 with 8 lanes.
           FPGA temperature = 35.0 degrees C.

Found 2 active device(s) installed on the host machine, to perform a full
diagnostic on a specific device, please run aocl diagnose <device_name>

DIAGNOSTIC_PASSED
```

## Programming the Flash Memory of an FPGA on Windows

By default, you configure an FPGA using the hardware image stored in the flash memory of the device. When there is no power, the FPGA retains the hardware configuration file in the flash memory. When you power up the system, it configures the FPGA circuitry based on this hardware image in the flash memory. Therefore, it is imperative that an OpenCL-compatible hardware configuration file is loaded into the flash memory of your FPGA.

Preloading an OpenCL image into the flash memory is necessary for the proper functioning of many Custom Platforms. For example, most PCI Express (PCIe)-based boards require a valid OpenCL image in flash memory so that hardware on the board can use the image to configure the FPGA device when the host system powers up for the first time. If the FPGA is not configured with a valid OpenCL image, the system will fail to enumerate the PCIe endpoint, or the driver will not function.

Before running any designs, ensure that the flash memory of your board has an valid OpenCL image that is compatible with the current OpenCL software version. Consult your board vendor's documentation for board-specific requirements.

**Caution:** When you load the hardware configuration file into the flash memory of the FPGA, maintain system power for the entire loading process, which might take a few minutes. Also, do not

launch any host code that calls OpenCL kernels or might otherwise communicate with the FPGA board.

To load your hardware configuration file into the flash memory of your FPGA board, perform the following tasks:

1. Install any drivers or utilities that your Custom Platform requires.

For example, some Custom Platforms require you to install the USB-Blaster™ driver to load your hardware configuration file into the flash memory. For installation instructions, refer to the *USB-Blaster II Download Cable User Guide*.

2. Ensure that you set the `AOCL_BOARD_PACKAGE_ROOT` environment variable to point to the subfolder in your Custom Platform that contains the `board_env.xml` file.
3. Download an example design for your Custom Platform.

**Remember:** You can download example designs from the OpenCL Design Examples page, and extract the example to a location to which you have write access.

4. To load the hardware configuration file into the flash memory, invoke the `aocl flash <device_name> <example_design_filename>.aocx` command, where `<device_name>` refers to the acl number (e.g. `acl0` to `acl15`) that corresponds to your FPGA device, and `<example_design_filename>.aocx` is the precompiled hardware configuration file you can find in the example design package.
5. Power down your device or computer and then power it up again.

Power cycling ensures that the FPGA configuration device retrieves the hardware configuration file from the flash memory and configures it into the FPGA.

**Warning:** Some Custom Platforms require you to power cycle the entire host system after programming the flash memory. For example, PCIe-based Custom Platforms might require a host system reboot to reenumerate the PCIe endpoint. Altera recommends that you power cycle the complete host system after programming the flash memory.

#### Related Information

- [USB-Blaster II Download Cable User Guide](#)
- [OpenCL Design Examples page on the Altera website](#)

## Executing an OpenCL Kernel on an FPGA

Build your OpenCL host application in Microsoft Visual Studio version 2010 Professional, and run the application by invoking the `hello_world.exe` executable. The AOCL is compatible with 64-bit host binaries only.

## Building the Host Application

The `<path_to_exm_openc1_hello_world_x64_windows_<version>>\hello_world\hello_world.sln` file contains the host solution. After you open this `.sln` file in Microsoft Visual Studio, you can build the host application in the `main.cpp` file.

To build the host application, perform the following tasks:

1. Open the `<path_to_exm_openc1_hello_world_x64_windows_<version>>\hello_world\hello_world.sln` file in Microsoft Visual Studio.
2. Verify that the build configuration is correct. The default build configuration is **Debug**, but you can use **Release**. You must select the appropriate option as the solution platform (for example, for x64 architecture, select **x64**).
3. Build the solution by selecting the **Build > Build Solution** menu option, or by pressing the F7 key. The `hello_world.exe` executable will be in the `<path_to_exm_openc1_hello_world_x64_windows_<version>>\hello_world\bin` folder.
4. Verify that the build is correct. An output ending with a message similar to the one shown below notifies you of a successful build:

```
1> Build succeeded.
1>
1> Time Elapsed 00:00:03:29
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

**Attention:** You can ignore the LNK4009: PDB 'vc90.pdb' was not found with... warnings because they have no effect on the build. The compiler might issue this type of warning messages if you have built your Windows libraries using a previous version of Microsoft Visual Studio.

## Running the Host Application

To execute the OpenCL kernel on the FPGA, run the Windows host application that you built from the `.sln` file.

1. Add the path `%ALTERAOCLSDKROOT%\host\windows64\bin` to the `PATH` environment variable.
2. At a command prompt, navigate to the host executable within the `<path_to_exm_openc1_hello_world_x64_windows_<version>>\hello_world\bin` folder.
3. Invoke the `hello_world.exe` executable.  
The `hello_world` executable executes the kernel code on the FPGA.

## Output from Successful Kernel Execution

When you run the host application to execute your OpenCL kernel on the target FPGA, the AOCL notifies you of a successful kernel execution.

Example output:

```
Found 1 OpenCL platforms.

Querying platform for info:
=====
CL_PLATFORM_NAME           = Altera SDK for OpenCL
CL_PLATFORM_VENDOR        = Altera Corporation
CL_PLATFORM_VERSION       = OpenCL 1.0 Altera SDK for OpenCL, Version <version>

Querying device for info:
```

```

=====
CL_DEVICE_NAME                = <board_name> : <descriptive_board_name>
CL_DEVICE_VENDOR              = <board_vendor_name>
CL_DEVICE_VENDOR_ID          = <board_vendor_ID>
CL_DEVICE_VERSION             = OpenCL 1.0 Altera SDK for OpenCL, Version <version>
CL_DRIVER_VERSION             = <version>
CL_DEVICE_ADDRESS_BITS       = 64
CL_DEVICE_AVAILABLE          = true
CL_DEVICE_ENDIAN_LITTLE      = true
CL_DEVICE_GLOBAL_MEM_CACHE_SIZE = 32768
CL_DEVICE_GLOBAL_MEM_CACHELINE_SIZE = 0
CL_DEVICE_GLOBAL_MEM_SIZE    = 0
CL_DEVICE_IMAGE_SUPPORT      = false
CL_DEVICE_LOCAL_MEM_SIZE     = 16384
CL_DEVICE_MAX_CLOCK_FREQUENCY = 1000
CL_DEVICE_MAX_COMPUTE_UNITS  = 1
CL_DEVICE_MAX_CONSTANT_ARGS  = 8
CL_DEVICE_MAX_CONSTANT_BUFFER_SIZE = 1073741824
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS = 3
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS = 1024
CL_DEVICE_MIN_DATA_TYPE_ALIGN_SIZE = 128
CL_DEVICE_PREFERRED_VECTOR_WIDTH_CHAR = 4
CL_DEVICE_PREFERRED_VECTOR_WIDTH_SHORT = 2
CL_DEVICE_PREFERRED_VECTOR_WIDTH_INT = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_LONG = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_FLOAT = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_DOUBLE = 0
Command queue out of order?    = false
Command queue profiling enabled? = true

Kernel initialization is complete.
Launching the kernel...

Thread #2: Hello from Altera's OpenCL Compiler!

Kernel execution is complete.

```

## Uninstalling the Software

To uninstall the AOCL for Windows, run the uninstaller, and restore all modified environment variables to their previous settings.

1. From the Windows Start Menu shortcut, navigate to the **Altera <version>\Altera SDK for OpenCL <version> (64-bit)** folder.
2. Run **Uninstall Altera SDK for OpenCL <version>**.
3. Remove the following paths from the *PATH* environment variable:
  - a. **%ALTERAOCLSDKROOT%\bin**
  - b. **%ALTERAOCLSDKROOT%\host\windows64\bin**
4. Remove the *ALTERAOCLSDKROOT* environment variable.

## Uninstalling the FPGA Board

To uninstall an FPGA board for Windows, invoke the `uninstall` utility command, uninstall the Custom Platform, and unset the relevant environment variables. You must uninstall the existing FPGA board if you migrate your OpenCL application to another FPGA board from a different Custom Platform.

To uninstall your FPGA board, perform the following tasks:

1. Following your board vendor's instructions to disconnect the board from your machine.
2. Invoke the `aocl uninstall` utility command to remove the current host computer drivers (for example, PCI Express (PCIe) drivers). The AOCL uses these drivers to communicate with the FPGA board.
3. Uninstall the Custom Platform.
4. Unset the `PATH` environment variable.
5. Unset the `AOCL_BOARD_PACKAGE_ROOT` environment variable.

2015.05.04

OCL001-15.0.0



Subscribe



Send Feedback

The AOCL setup process includes downloading and installing the software, installing the FPGA board, and then executing an OpenCL kernel on the FPGA.

1. **Downloading the Altera SDK for OpenCL** on page 3-2  
Download the AOCL and all related software for Linux from the Download Center on the Altera website.
2. **Installing the Altera SDK for OpenCL** on page 3-2  
Install the AOCL in a directory that you own.
3. **Setting the Environment Variables for Linux** on page 3-3  
You have the option to set the AOCL environment variables permanently or transiently.
4. **Verifying the AOCL Installation** on page 3-4  
Invoke the `version` utility command and verify that the correct version of the AOCL is installed.
5. **Licensing the Software** on page 3-5  
Obtain the AOCL license from the Altera Self Service Licensing Center.
6. **Installing an FPGA Board** on page 3-5  
To install your board into the host system, invoke the `install` utility command.
7. **Creating the FPGA Hardware Configuration File of an OpenCL Kernel** on page 3-6  
To create the `.aocx` file for the `hello_world` example OpenCL application, you must download the example design for Linux from the OpenCL Design Examples page on the Altera website. Extract the downloaded source files and then compile the kernel using the Altera Offline Compiler (AOC).
8. **Updating the Hardware Image on the FPGA** on page 3-8  
If applicable, before you execute an OpenCL kernel program on the FPGA, ensure that the flash memory of the FPGA contains a hardware image created using a current version of the OpenCL software.
9. **Executing an OpenCL Kernel on an FPGA** on page 3-9  
You must build your OpenCL host application with the `Makefile` file, and run the application by invoking the `hello_world` executable.
10. **Uninstalling the Software** on page 3-11  
To uninstall the AOCL for Linux, remove the software package via the Red Hat Package Manager (RPM) or GUI uninstaller, then delete the software directory and restore all modified environment variables to their previous settings.
11. **Uninstalling the FPGA Board** on page 3-11  
To uninstall an FPGA board for Linux, invoke the `uninstall` utility command, uninstall the Custom Platform, and unset the relevant environment variables.

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at [www.altera.com/common/legal.html](http://www.altera.com/common/legal.html). Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO  
9001:2008  
Registered



## Downloading the Altera SDK for OpenCL

Download the AOCL and all related software for Linux from the Download Center on the Altera website. Download the AOCL and all related software into a single installation package. Alternatively, if you already have the current version of the Quartus II software in your system, download the AOCL as a stand-alone software.

Download the following software and files:

- Quartus II software
- Device support (for example, Stratix V device support)
- AOCL

1. In the main page of the Altera website, click **MYALTERA** and log into your account.  
If you do not have a myAltera account, register for a new account.
2. Click **DOWNLOADS** to enter the Download Center.
3. Click **Altera SDK for OpenCL** to enter the download page for the subscription edition of the AOCL.
4. Select the software version. The default selection is the current version.
5. Select **Direct Download** as the download method.
6. Click the **Linux SDK** tab and then click **More** beside **Download and install instructions** to view the download and installation procedure.
7. Perform the steps outlined in the download and installation instructions on the download page.

### Related Information

[Altera website](#)

## Installing the Altera SDK for OpenCL

If you download the AOCL, the Quartus II software, and the device support into a single installation package, run the Quartus II software installer for Linux to install all the software and files. If you download the AOCL as a stand-alone software, run the AOCL installer. Install the AOCL in a directory that you own.

### Before you begin

1. You must have sudo or root privileges.
2. You must install the Linux OS kernel source and headers (for example, **kernel-devel.x86\_64** and **kernel-headers.x86\_64**), and the GNU Compiler Collection (GCC) (**gcc.x86\_64**).

**Attention:** If you install the software on a system that does not contain any C Shell Run Commands file (**.cshrc**) or Bash Run Commands file (**.bashrc**) in your directory, you must set the environment variables **ALTERAOCLSDKROOT** and **PATH** manually. Alternatively, you may create the **.cshrc** and **.bashrc** files, and then append the environment variables to them. To ensure that the updates take effect, restart your terminal after you set the environment variables.

- To install the AOCL, Quartus II software, and device support files simultaneously, perform the following tasks:
  1. Run the **QuartusSetup-<version>-linux.run** installer, and follow the installation instructions in the *Altera Software Installation and Licensing* manual.
  2. The installer sets the environment variable `ALTERAOCLSDKROOT` to point to the path of the software installation. Ensure that `ALTERAOCLSDKROOT` points to the current version of the software.
  3. Ensure that the `PATH` environment variable includes the `bin` directory in the Quartus II system directory (for example, `<home_directory>/altera/<version>/quartus/bin`).
- To install the AOCL as a stand-alone software, perform the following tasks:
  1. Ensure that your system already has the current version of the Quartus II software and applicable device support files.
  2. Ensure that the `PATH` environment variable includes the `bin` directory in the Quartus II system folder (for example, `<home_directory>/altera/<version>/quartus/bin`).
  3. To install the software using the Red Hat Package Manager (RPM), at the command prompt, invoke one of the following commands:
    - At a command prompt, type `rpm -i aocl-sdk-<version>.x86_64.rpm` to install the software in the default location (for example, `opt/altera/aocl-sdk`).
    - To install the software in the default location with verbose progress reporting, type `rpm -ivh aocl-sdk-<version>.x86_64.rpm`
    - To install the software in an alternate directory that you own (that is, not a system directory), type the `rpm -i --prefix <aocl_destination_directory> aocl-sdk-<version>.x86_64.rpm` command.
  4. To install the software using the GUI installer, run the **AOCLSetup-<version>-linux.run** installer. Direct the installer to extract the software to an empty folder that you own (that is, not a system directory). The installation path must not contain any spaces (for example, `usr/altera/<version>/hld`).
  5. The installer sets the environment variable `ALTERAOCLSDKROOT` to the path of the software installation. Ensure that `ALTERAOCLSDKROOT` points to the current version of the software.

#### Related Information

[Altera Software Installation and Licensing](#)

## Setting the Environment Variables for Linux

You have the option to set the AOCL environment variables permanently or transiently. The environment variable settings describe the FPGA board and the host runtime to the software.

**Attention:** If you set the environment variables permanently, you apply the settings once during installation. If you set the environment variables transiently, you must apply the settings during installation and during every subsequent session you run.

- To apply permanent environment variable settings, manually include the following paths in the corresponding environment variable settings:

Environment Variable	Path to Include
<code>PATH</code>	<p><code>\$ALTERAOCLSDKROOT/bin</code></p> <p>where <code>ALTERAOCLSDKROOT</code> points to the path of the software installation</p>
<code>LD_LIBRARY_PATH</code>	<p><code>\$ALTERAOCLSDKROOT/host/linux64/lib</code></p> <p><code>\$AOCL_BOARD_PACKAGE_ROOT/linux64/lib</code></p> <p>where <code>AOCL_BOARD_PACKAGE_ROOT</code> points to the path of the Custom or Reference Platform</p>

- To apply transient environment variable settings, open a command-line terminal and run the `source $ALTERAOCLSDKROOT/init_openc1.sh` command.

Example script output:

```
AOCL_BOARD_PACKAGE_ROOT path is not set in environment
Setting to default s5_ref board.
If you want to target another board, do
  set AOCL_BOARD_PACKAGE_ROOT=board_pkg_dir
Adding $ALTERAOCLSDKROOT/bin to PATH
Adding $ALTERAOCLSDKROOT/host/linux64/lib to LD_LIBRARY_PATH
Adding $AOCL_BOARD_PACKAGE_ROOT/linux64/lib to LD_LIBRARY_PATH
```

## Verifying the AOCL Installation

Invoke the `version` utility command and verify that the correct version of the AOCL is installed.

- At a command prompt, invoke the `aocl version` utility command. An output similar to the one below notifies you of a successful installation:

```
aocl <version>.<build> (Altera SDK for OpenCL, Version <version> Build <build>,
Copyright (C) <year> Altera Corporation)
```

- If installation was unsuccessful, reinstall the software. You can also refer to the *Altera Software Installation and Licensing* manual and the Knowledge Center on the Altera website for more information.

### Related Information

- [Altera Software Installation and Licensing](#)
- [Altera Knowledge Base](#)

## Licensing the Software

Obtain the AOCL license from the Altera Self Service Licensing Center. The AOCL license allows the software to access relevant Quartus II software functionalities and device support without an additional Quartus II software license.

### Before you begin

For information on the licensing options and requirements, refer to the *Licensing Altera Software* section of the *Altera Software Installation and Licensing* manual, the **LICENSE.txt** file that accompanies each software, and the Altera Licensing page on the Altera website.

1. In the main page on the Altera website, click **MYALTERA** and log into your account.
2. Click the link to the Self-Service Licensing Center.
3. Perform the steps outlined in the *Using the Self-Service Licensing Center* section of the *Altera Software Installation and Licensing* manual to obtain and activate the license.
4. If you have a fixed license, append the `<path_to_license_file>/<license_filename>` file to the `LM_LICENSE_FILE` environment variable in the following manner:
  - a. Create a backup copy of the provided license file.
  - b. Save the new license file on your local hard drive.
  - c. Append the `<path_to_license_file>/<license_filename>` file to the `LM_LICENSE_FILE` environment variable.
5. If you have a floating licence, append the `<path_to_license_file>/<license_filename>` file to the `LM_LICENSE_FILE` environment variable in the following manner:
  - a. Obtain the port number and host name from the network or system administrator. Alternatively, the information is in the license file line `SERVER <hostname> <8 to 12 character host or NIC ID> <port>`.  
The license location for the user is `<port>@<hostname>`. If a port is not listed in the license file, specify `@<hostname>`.
  - b. Modify the license file to update the port number and host name.
  - c. Append the `<path_to_license_file>/<license_filename>` file to the `LM_LICENSE_FILE` environment variable setting.

### Related Information

- [Altera Licensing page on the Altera website](#)
- [Using the Self-Service Licensing Center](#)
- [Altera website](#)

## Installing an FPGA Board

Before creating an OpenCL application for an FPGA board on Linux, you must first download and install the Custom Platform from your board vendor. Most Custom Platform installers require administrator privileges. To install your board into the host system, invoke the `install` utility command.

The steps below outline the board installation procedure. Some Custom Platforms require additional installation tasks. Consult your board vendor's documentation for further information on board installation.

**Attention:** If you are installing the Cyclone V SoC Development Kit for use with the Cyclone V SoC Development Kit Reference Platform, refer to *Installing the Cyclone V SoC Development Kit* in the *Altera SDK for OpenCL Cyclone V SoC Getting Started Guide* for more information.

1. Follow your board vendor's instructions to connect the FPGA board to your system.
2. Download the Custom Platform for your FPGA board from your board vendor's website.  
For more information, refer to the OpenCL Reference Platforms page within the Altera website.
3. Install the Custom Platform in a directory that you own (that is, not a system directory).
4. Set the environment variable `AOCL_BOARD_PACKAGE_ROOT` to point to the location of the Custom Platform subdirectory containing the `board_env.xml` file.  
For example, for the Stratix V Network Reference Platform (s5\_net), set `AOCL_BOARD_PACKAGE_ROOT` to point to the `<path_to_s5_net>/s5_net` directory.
5. Add the Custom Platform library paths to the `LD_LIBRARY_PATH` environment variable. You may apply permanent settings manually by adding the path to the memory-mapped (MMD) library within the Custom Platform. Alternatively, you may apply transient settings to the current command window by running the `$ALTERAOCLSDKROOT/init_opencl.sh` script.  
For example, if you use s5\_net, the Linux `LD_LIBRARY_PATH` setting is `$AOCL_BOARD_PACKAGE_ROOT/linux64/lib`.  
For more information on the `init_opencl.sh` script, refer to the *Setting the Environment Variables for Linux* section.
6. Invoke the command `aocl install` at a command prompt.  
Invoking `aocl install` also installs a board driver that allows communication between host applications and hardware kernel programs.
7. To query a list of FPGA devices installed in your machine, invoke the `aocl diagnose` command. The software generates an output that includes the `<device_name>`, which is an acl number that ranges from `acl0` to `acl15`.  
For more information on querying the `<device_name>` of your accelerator board, refer to the *Querying the Device Name of Your FPGA Board* section.
8. To verify the successful installation of the FPGA board, invoke the command `aocl diagnose <device_name>` to run any board vendor-recommended diagnostic test.

## Creating the FPGA Hardware Configuration File of an OpenCL Kernel

The FPGA hardware configuration file of an OpenCL kernel is called an *Altera Offline Compiler Executable file (.aocx)*. To create the `.aocx` file for the `hello_world` example OpenCL application, you must download the example design for Linux from the OpenCL Design Examples page on the Altera website. Extract the downloaded source files and then compile the kernel using the Altera Offline Compiler (AOC).

After you successfully install your FPGA board, you can create a `.aocx` file that executes on the device. The steps below describe the process of creating a `.aocx` file from the `hello_world` example design. For more

information on the OpenCL design examples, refer to the OpenCL Design Examples page on the Altera website.

1. In the OpenCL Design Examples page, under Basic Examples, click **Hello World**.
2. In the Hello World Design Example page, under Downloads, click **Linux package (.tgz)** to download the compressed file for your platform.
3. Decompress the **exm\_openccl\_hello\_world\_x64\_linux\_<version>.tgz** file and save it in a directory to which you have write access.
4. Ensure that you have installed the Custom Platform provided by your board vendor successfully.
5. Ensure that the environment variable `AOCL_BOARD_PACKAGE_ROOT` points to a local subdirectory within your Custom Platform that contains the **board\_env.xml** file.
6. Select your target FPGA board. To list the FPGA boards available in your Custom Platform, invoke the command `aoc --list-boards` at a command prompt.

For more information on the `--list-boards` option of the `aoc` command, refer to the *Listing the Available FPGA Boards in Your Custom Platform (--list-boards)* section of the *Altera SDK for OpenCL Programming Guide*.

7. At a command prompt, navigate to the `hello_world` design.
8. To compile the kernel for your target FPGA board, invoke the following command:  

```
aoc -v --board <board_name> device/hello_world.cl -o bin/hello_world.aocx
```

This command performs the following tasks:

- Generates the Quartus II design project files from the OpenCL source code.
- Checks for initial syntax errors.
- Performs basic optimizations.
- Creates a **hello\_world** directory containing necessary intermediate files.
- Creates the *Altera Offline Compiler Object file (.aoco)*.
- Creates the **.aocx** hardware configuration file.

**Attention:** The **.aocx** file might take hours to build, depending on the complexity of the kernel. To view the progress of the compilation on-screen, include the `-v` flag in your `aoc` command. An example output is shown below.

```
aoc: Environment checks are completed successfully.  
You are now compiling the full flow!!  
aoc: Selected target board <board_name>  
aoc: Running OpenCL parser....  
aoc: OpenCL parser completed successfully.  
aoc: Compiling....  
aoc: Linking with IP library ...  
aoc: First stage compilation completed successfully.  
aoc: Setting up project for CvP revision flow....  
aoc: Hardware generation completed successfully.
```

The AOC displays the line `aoc: Hardware generation completed successfully.` to signify the completion of the compilation process.

For more information on the `--board <board_name>` option of the `aoc` command, refer to the *Compiling a Kernel for a Specific FPGA Board (--board <board\_name>)* section of the *Altera SDK for OpenCL Programming Guide*.

For more information on the `-v` option of the `aoc` command, refer to the *Generating Compilation Progress Report (-v)* section of the *Altera SDK for OpenCL Programming Guide*.

For more information on the `-o <filename>` option of the `aoc` command, refer to the *Specifying the Name of an AOC Output File (-o <filename>)* section of the *Altera SDK for OpenCL Programming Guide*.

## Updating the Hardware Image on the FPGA

If applicable, before you execute an OpenCL kernel program on the FPGA, ensure that the flash memory of the FPGA contains a hardware image created using a current version of the OpenCL software.

**Remember:** If your Custom Platform requires that you preload a valid OpenCL image into the flash memory, for every major release of the Altera Complete Design Suite, program the flash memory of the FPGA with a hardware image compatible with the current version of the software.

## Querying the Device Name of Your FPGA Board

Some AOCL utility commands require you to specify the device name (`<device_name>`). The `<device_name>` refers to the `acl` number (e.g. `acl0` to `acl15`) that corresponds to the FPGA device. When you query a list of accelerator boards, the AOCL produces a list of installed devices on your machine in the order of their device names.

- To query a list of installed devices on your machine, type `aocl diagnose` at a command prompt. The software generates an output that resembles the example shown below:

```
aocl diagnose: Running diagnostic from ALTERAOCLSDKROOT/board/<board_name>/
<platform>/libexec

Verified that the kernel mode driver is installed on the host machine.

Using board package from vendor: <board_vendor_name>
Querying information for all supported devices that are installed on the host
machine ...

device_name  Status  Information
acl0         Passed  <descriptive_board_name>
           PCIe dev_id = <device_ID>, bus:slot.func = 02:00.00,
           at Gen 2 with 8 lanes.
           FPGA temperature=43.0 degrees C.
acl1         Passed  <descriptive_board_name>
           PCIe dev_id = <device_ID>, bus:slot.func = 03:00.00,
           at Gen 2 with 8 lanes.
           FPGA temperature = 35.0 degrees C.

Found 2 active device(s) installed on the host machine, to perform a full
diagnostic on a specific device, please run aocl diagnose <device_name>

DIAGNOSTIC_PASSED
```

## Programming the Flash Memory of an FPGA on Linux

By default, you configure an FPGA using the hardware image stored in the flash memory of the device. When there is no power, the FPGA retains the hardware configuration file in the flash memory. When you power up the system, it configures the FPGA circuitry based on this hardware image in the flash

memory. Therefore, it is imperative that an OpenCL-compatible hardware configuration file is loaded into the flash memory of your FPGA.

Preloading an OpenCL image into the flash memory is necessary for the proper functioning of many Custom Platforms. For example, most PCI Express (PCIe)-based boards require a valid OpenCL image in flash memory so that hardware on the board can use the image to configure the FPGA device when the host system powers up for the first time. If the FPGA is not configured with a valid OpenCL image, the system will fail to enumerate the PCIe endpoint, or the driver will not function.

Before running any designs, ensure that the flash memory of your board has an valid OpenCL image that is compatible with the current OpenCL software version. Consult your board vendor's documentation for board-specific requirements.

**Caution:** When you load the hardware configuration file into the flash memory of the FPGA, maintain system power for the entire loading process, which might take a few minutes. Also, do not launch any host code that calls OpenCL kernels or might otherwise communicate with the FPGA board.

To load your hardware configuration file into the flash memory of your FPGA board, perform the following tasks:

1. Install any drivers or utilities that your Custom Platform requires.
2. Ensure that you set the `AOCL_BOARD_PACKAGE_ROOT` environment variable to point to the subdirectory in your Custom Platform that contains the `board_env.xml` file.
3. Download an example design for your Custom Platform.

**Remember:** You can download example designs from the OpenCL Design Examples page, and extract the example to a location to which you have write access.

4. To load the hardware configuration file into the flash memory, invoke the `aocl flash <device_name> <example_design_filename>.aocx` command, where `<device_name>` refers to the acl number (e.g. `acl0` to `acl15`) that corresponds to your FPGA device, and `<example_design_filename>.aocx` is the precompiled hardware configuration file you can find in the example design package.
5. Power down your device or computer and then power it up again.

Power cycling ensures that the FPGA configuration device retrieves the hardware configuration file from the flash memory and configures it into the FPGA.

**Warning:** Some Custom Platforms require you to power cycle the entire host system after programming the flash memory. For example, PCIe-based Custom Platforms might require a host system reboot to reenumerate the PCIe endpoint. Altera recommends that you power cycle the complete host system after programming the flash memory.

#### Related Information

[OpenCL Design Examples page on the Altera website](#)

## Executing an OpenCL Kernel on an FPGA

You must build your OpenCL host application with the `Makefile` file, and run the application by invoking the `hello_world` executable. You need GNU development tools such as `gcc` and `make` to build the OpenCL application.

## Building the Host Application

Build the host executable with the `<path_to_exm_openc1_hello_world_x64_linux_<version>>/hello_world/Makefile` file.

To build the host application, perform the following tasks:

1. Navigate to the **hello\_world** directory.
2. Invoke the `$ make -f Makefile` command. Alternatively, you can simply invoke the `make` command.

The **hello\_world** executable will be in the `<path_to_exm_openc1_hello_world_x64_linux_<version>>/hello_world/bin` directory.

## Running the Host Application

To execute the OpenCL kernel on the FPGA, run the Linux host application that you built from the **Makefile**.

1. Add the path `$ALTERAOCLSDKROOT/host/linux64/lib` to the `LD_LIBRARY_PATH` environment variable.
2. At a command prompt, navigate to the host executable within the `<path_to_exm_openc1_hello_world_x64_linux_<version>>/hello_world/bin` directory.
3. Invoke the **hello\_world** executable.

The **hello\_world** executable executes the kernel code on the FPGA.

## Output from Successful Kernel Execution

When you run the host application to execute your OpenCL kernel on the target FPGA, the AOCL notifies you of a successful kernel execution.

Example output:

```
Found 1 OpenCL platforms.

Querying platform for info:
=====
CL_PLATFORM_NAME           = Altera SDK for OpenCL
CL_PLATFORM_VENDOR         = Altera Corporation
CL_PLATFORM_VERSION        = OpenCL 1.0 Altera SDK for OpenCL, Version <version>

Querying device for info:
=====
CL_DEVICE_NAME              = <board_name> : <descriptive_board_name>
CL_DEVICE_VENDOR           = <board_vendor_name>
CL_DEVICE_VENDOR_ID        = <board_vendor_ID>
CL_DEVICE_VERSION          = OpenCL 1.0 Altera SDK for OpenCL, Version <version>
CL_DRIVER_VERSION          = <version>
CL_DEVICE_ADDRESS_BITS     = 64
CL_DEVICE_AVAILABLE       = true
CL_DEVICE_ENDIAN_LITTLE    = true
CL_DEVICE_GLOBAL_MEM_CACHE_SIZE = 32768
CL_DEVICE_GLOBAL_MEM_CACHELINE_SIZE = 0
CL_DEVICE_GLOBAL_MEM_SIZE  = 0
CL_DEVICE_IMAGE_SUPPORT    = false
CL_DEVICE_LOCAL_MEM_SIZE   = 16384
CL_DEVICE_MAX_CLOCK_FREQUENCY = 1000
CL_DEVICE_MAX_COMPUTE_UNITS = 1
CL_DEVICE_MAX_CONSTANT_ARGS = 8
CL_DEVICE_MAX_CONSTANT_BUFFER_SIZE = 1073741824
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS = 3
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS = 1024
```

```
CL_DEVICE_MIN_DATA_TYPE_ALIGN_SIZE      = 128
CL_DEVICE_PREFERRED_VECTOR_WIDTH_CHAR   = 4
CL_DEVICE_PREFERRED_VECTOR_WIDTH_SHORT  = 2
CL_DEVICE_PREFERRED_VECTOR_WIDTH_INT    = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_LONG   = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_FLOAT  = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_DOUBLE = 0
Command queue out of order?             = false
Command queue profiling enabled?         = true
```

```
Kernel initialization is complete.
Launching the kernel...
```

```
Thread #2: Hello from Altera's OpenCL Compiler!
```

```
Kernel execution is complete.
```

## Uninstalling the Software

To uninstall the AOCL for Linux, remove the software package via the Red Hat Package Manager (RPM) or GUI uninstaller, then delete the software directory and restore all modified environment variables to their previous settings.

1. Remove the software package by performing one of the following tasks:
  - a. To uninstall the AOCL via RPM at the command line, type the `rpm -e aocl-sdk` command.
  - b. To uninstall the AOCL via the GUI uninstaller, run the `aocl-<version>-uninstall.run` program located in the `<install directory>/uninstall` directory.
2. Remove `$ALTERAOCLSDKROOT/bin` from the `PATH` environment variable.
3. Remove `$ALTERAOCLSDKROOT/host/linux64/lib` from the `LD_LIBRARY_PATH` environment variable.
4. Remove the `ALTERAOCLSDKROOT` environment variable.

## Uninstalling the FPGA Board

To uninstall an FPGA board for Linux, invoke the `uninstall` utility command, uninstall the Custom Platform, and unset the relevant environment variables. You must uninstall the existing FPGA board if you migrate your OpenCL application to another FPGA board from a different Custom Platform.

To uninstall your FPGA board, perform the following tasks:

1. Following your board vendor's instructions to disconnect the board from your machine.
2. Invoke the `aocl uninstall` utility command to remove the current host computer drivers (for example, PCI Express (PCIe) drivers). The AOCL uses these drivers to communicate with the FPGA board.
3. Uninstall the Custom Platform.
4. Unset the `LD_LIBRARY_PATH` environment variable.
5. Unset the `AOCL_BOARD_PACKAGE_ROOT` environment variable.

# Document Revision History



2015.05.04

OCL001-15.0.0



Subscribe



Send Feedback

**Table A-1: Document Revision History of the Altera SDK for OpenCL Getting Started Guide**

Date	Version	Changes
May 2015	15.0.0	<ul style="list-style-type: none"><li>Reorganized instructions into the following sections:<ul style="list-style-type: none"><li>Getting Started with the AOCL on Windows</li><li>Getting Started with the AOCL on Linux</li></ul></li></ul>
December 2014	14.1.0	<ul style="list-style-type: none"><li>Reorganized information flow.</li><li>Updated Red Hat Enterprise Linux (RHEL) version support.</li><li>Included the <i>Contents of the AOCL</i> section.</li><li>Updated licensing instructions for the new Altera Software Development Kit (SDK) for OpenCL (AOCL) single license.</li><li>Updated board uninstallation instructions to include the <code>aocl uninstall</code> utility command.</li><li>Included information on the <code>init_opencil</code> script for setting environment variables.</li><li>Grouped software and board uninstallation instructions under <i>Uninstalling the Software and the FPGA Board</i>.</li></ul>

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at [www.altera.com/common/legal.html](http://www.altera.com/common/legal.html). Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO  
9001:2008  
Registered



Date	Version	Changes
June 2014	14.0.0	<ul style="list-style-type: none"> <li>• Updated the <i>Prerequisites</i> section.</li> <li>• Updated the figure <i>AOCL Installation Process Overview</i>.</li> <li>• Updated software download instructions.</li> <li>• Updated AOCL installation and uninstallation instructions for Windows.</li> <li>• For Linux systems: <ul style="list-style-type: none"> <li>• Added the sections <i>Installing the AOCL on Linux Using RPM</i> and <i>Uninstalling the AOCL on Linux Using RPM</i>.</li> <li>• Updated the section <i>Installing the AOCL on Linux Using the GUI Installer</i>.</li> </ul> </li> <li>• Added the section <i>Licensing the Software</i>.</li> <li>• Updated the section <i>Installing an FPGA Board</i> with updated instructions on querying your devices and running diagnostic tests.</li> <li>• Updated the section <i>Creating the FPGA Hardware Configuration File of an OpenCL Kernel</i>: <ul style="list-style-type: none"> <li>• Updated path that you have to set for <code>AOCL_BOARD_PACKAGE_ROOT</code>.</li> <li>• Updated example AOC output for compiling <code>hello_world.cl</code> with the <code>-v</code> option.</li> <li>• Updated AOC output.</li> </ul> </li> <li>• Added the section <i>Identifying the Device Name of Your FPGA Board</i>.</li> <li>• Modified instructions for building and running the host application with updated <code>hello_world</code> directory and file names.</li> <li>• Added the section <i>Uninstalling an FPGA Board</i>.</li> </ul>
December 2013	13.1.1	<ul style="list-style-type: none"> <li>• Updated the <i>Prerequisites</i> section to include a reminder to install Linux kernel source, headers, and GCC.</li> </ul>

Date	Version	Changes
November 2013	13.1.0	<ul style="list-style-type: none"> <li>• Reorganized information flow.</li> <li>• Updated the <i>Prerequisites</i> section.</li> <li>• Updated board installation instructions.</li> <li>• Updated software download instructions.</li> <li>• Inserted the section <i>Installing the AOCL and the Quartus II Software Together</i>.</li> <li>• Updated software installation and uninstallation instructions.</li> <li>• Inserted the following figures:               <ul style="list-style-type: none"> <li>• <i>AOCL Installation Process Overview</i></li> <li>• <i>FPGA Programming Overview</i></li> </ul> </li> <li>• Removed the <i>Licensing</i> section.</li> <li>• Removed all board-specific installation and configuration instructions.</li> <li>• Changed example OpenCL application used to demonstrate kernel configuration and FPGA programming from <code>moving_average</code> to <code>hello_world</code>.</li> <li>• Inserted the section <i>Updating the Hardware Image on the FPGA</i>, which contained the updated flash programming instructions.</li> <li>• Removed the section <i>Installing the USB-Blaster Driver on Windows</i>.</li> <li>• Updated output from successful execution of <code>hello_world</code> kernel on FPGA for Windows and Linux systems.</li> <li>• Removed the figure <i>Contents of the Moving Average Example</i>.</li> <li>• Removed the figure <i>Opening host.sln in Visual Studio</i>.</li> </ul>
June 2013	13.0 SP1.0	<ul style="list-style-type: none"> <li>• Updated requisite Quartus II and AOCL software versions from 13.0 to 13.0 SP1.</li> <li>• Inserted the figure <i>A Correct Windows Device Manager After Complete Board Driver Installation for a BittWare Board</i>.</li> <li>• Updated the vendor and device IDs in the <i>Verifying the Functionality of the BittWare Board</i> section for Windows.</li> <li>• Updated the AOCL installation instructions for Linux systems that do not contain a <code>.cshrc</code> or a <code>.bashrc</code> file in the directory.</li> <li>• Updated path to the AOCL design examples.</li> <li>• Updated the figure <i>Contents of the Moving Average Example</i>.</li> <li>• Updated flash programming instructions.</li> </ul>

Date	Version	Changes
May 2013	13.0.1	<ul style="list-style-type: none"> <li>• Renamed the <b>OpenCL_SDK</b> folder or directory to <b>AOCL</b>.</li> <li>• Inserted warning about the AOCL installation dependency on <b>.cshrc</b> and <b>.bashrc</b> files for Linux systems.</li> <li>• Included reminder to BittWare board users about installing the BittWare development software.</li> <li>• Inserted warning about potential Jungo WinDriver installation failure for systems running on Windows 7. Included reinstallation instructions.</li> <li>• Inserted warnings about error messages displayed for <code>aocl</code> commands that have not been implemented for the BittWare FPGA board.</li> <li>• Inserted caution message about setting the environment variable <code>AOCL_BOARD_PACKAGE_ROOT</code>.</li> <li>• Updated board driver installation procedures for Windows and Linux systems.</li> <li>• Modified the path to the default location of the AOCL for Windows and Linux systems.</li> <li>• Modified the path name added to the <code>PATH</code> environment variable when installing the AOCL on Linux systems. The path name should be <code>\$QUARTUS_ROOTDIR/bin</code> instead of <code>\$QUARTUS_ROOTDIR/bin64</code>.</li> </ul>
May 2013	13.0.0	<ul style="list-style-type: none"> <li>• Updated installation and compilation procedures.</li> <li>• Incorporated licensing procedure.</li> <li>• Updated flash programming procedure and moved it to Appendix A.</li> <li>• Updated links to software and documentation download pages.</li> </ul>
November 2012	12.1.0	Initial release.