

Altera SDK for OpenCL

Getting Started Guide



Subscribe



Send Feedback

OCL001-14.1.0
2014.12.15

101 Innovation Drive
San Jose, CA 95134
www.altera.com



Contents

Altera SDK for OpenCL Getting Started Guide.....	1-1
Prerequisites.....	1-1
Contents of the AOCL.....	1-2
Overview of the AOCL Setup Process.....	1-4
Downloading the Software.....	1-6
Installing the Software.....	1-7
Installing the Software on 64-Bit Windows.....	1-7
Installing the Software on x86_64 Linux.....	1-9
Verifying the Installation.....	1-11
Licensing the Software.....	1-11
Installing an FPGA Board (install).....	1-12
Creating the FPGA Hardware Configuration File of an OpenCL Kernel.....	1-13
Updating the Hardware Image on the FPGA.....	1-15
Querying the Device Name of Your FPGA Board (diagnose).....	1-15
Programming the Flash Memory of an FPGA.....	1-16
Setting Up the Host Application.....	1-17
Building and Running the Host Application on Windows.....	1-17
Building and Running the Host Application on Linux.....	1-18
Output from Successful Kernel Execution.....	1-19
Uninstalling the Software and the FPGA Board.....	1-20
Uninstalling the Software on 64-Bit Windows.....	1-20
Uninstalling the Software on x86_64 Linux.....	1-20
Uninstalling the FPGA Board (uninstall).....	1-21
Conclusion.....	1-21
Document Revision History.....	1-21

2014.12.15

OCL001-14.1.0



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)⁽¹⁾. This document also contains instructions on how to compile an example OpenCL⁽²⁾ 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

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

-
- ⁽¹⁾ 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.
- ⁽²⁾ OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission of the Khronos Group™.

© 2014 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. 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



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 of 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 a PCIe port 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

- 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
- 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 Reference Platforms page](#)

Contents of the AOCL

The Altera Software Development Kit (SDK) for OpenCL (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.
- 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 folder or directory that you own. The path to the software installation folder or directory is referenced by the *ALTERAOCLSDKROOT* environment variable. The table below highlights some of the contents of this folder or directory.

Windows Folder	Linux Directory	Description
bin	bin	User commands in the AOCL. Include this folder or directory in your <i>PATH</i> environment variable.
board	board	The AOCL Custom Platform Toolkit and Reference Platforms available with the software. <ul style="list-style-type: none"> • The path to the Custom Platform Toolkit is <i>ALTERAOCLSDKROOT/board/custom_platform_toolkit</i> • The path to the s5_ref Reference Platform is <i>ALTERAOCLSDKROOT/board/s5_ref</i>
ip	ip	Intellectual property (IP) core used to compile device kernels.
host	host	Files necessary for compiling and running your host application.
host\include	host/include	OpenCL Specification version 1.0 header files and software interface files necessary for compiling and linking your host application. The host/include/CL subfolder or subdirectory also includes the C++ header file cl.hpp . 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. Important: The OpenCL version 1.1 C++ bindings are compatible with OpenCL Specification versions 1.0 and 1.1. Add this path to the include file search path in your development environment.
host\windows64\lib	host/linux64/lib	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 <i>LD_LIBRARY_PATH</i> environment variable.

Windows Folder	Linux Directory	Description
host\windows64\bin	host/linux64/bin	Runtime commands and libraries necessary for running your host application, wherever applicable. For 64-bit Windows system, include this folder in your <i>PATH</i> environment variable. 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.
share	share	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

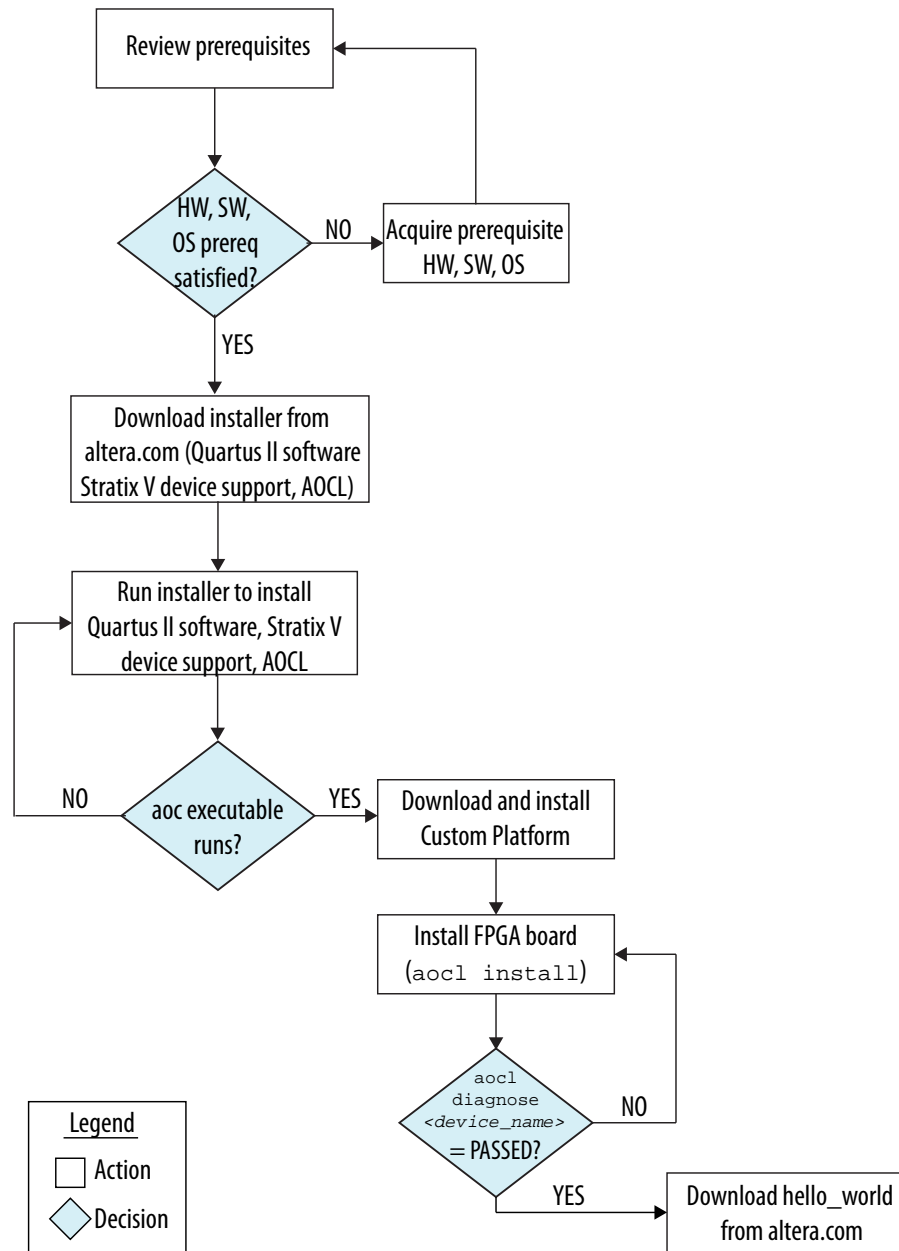
[OpenCL Design Examples](#)

Overview of the AOCL Setup Process

The *Altera SDK for OpenCL Getting Started Guide* outlines the procedures for installing the Altera Software Development Kit (SDK) for OpenCL (AOCL) and programming your FPGA.

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

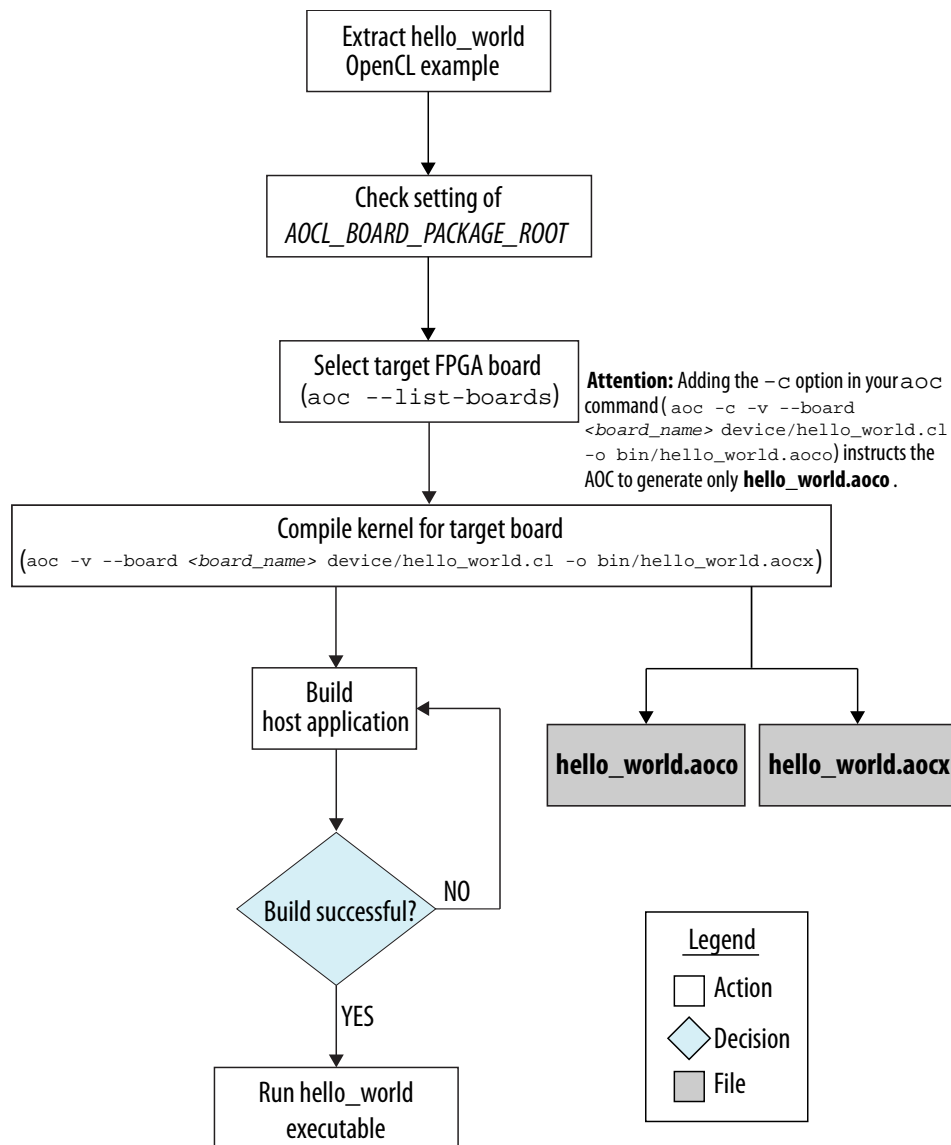
Figure 1-1: AOCL Installation Process Overview



After you complete the initial software and hardware setup successfully, you can create a hardware image from the *hello_world* example OpenCL application. 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



Downloading the Software

Download the Altera Software Development Kit (SDK) for OpenCL (AOCL) and all related software from the Download Center of 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, you can 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. On the main page of the [Altera website](#), click **myAltera Account** to log on to your myAltera account.
If you do not have a myAltera account, follow the instructions on the page to create one.
2. Click **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 operating system (OS) (that is, Windows or Linux).
6. Select the download method.
 - For Windows systems, select the **Akamai DLM3 Download Manager** or **Direct Download**.
 - For Linux systems, select **Direct Download**.
7. To download the AOCL and all related software into a single Quartus II software installer, click the SDK tab and perform the following steps:
 - a. Select **Quartus II software**.
 - b. Select the relevant device support options.
 - c. Select **Altera SDK for OpenCL (Standalone Installer)**.
For Linux systems, to download the AOCL Red Hat Package Manager (RPM) installation package, select **Altera SDK for OpenCL (RPM Package)**.
 - d. Proceed to Step 9.
8. If you already have the current version of the Quartus II software and the device support file(s) in your system, perform steps 7c and 7d only.
The AOCL installer or RPM package installs the AOCL without the Quartus II software or the device support file(s).
9. Click **Download Selected Files**.
10. Perform the steps outlined in the download and installation instructions on the download page.

Installing the Software

The installation instructions for the Altera Software Development Kit (SDK) for OpenCL, the Quartus II software and the device support are platform-specific.

If you download the AOCL, the Quartus II software, and the device support into a single installation package, run the Quartus II software installer to install all the software and files. If you download the AOCL as a stand-alone software, run the AOCL installer to install the AOCL only.

Installing the Software on 64-Bit Windows on page 1-7

Install the software in a folder that you own, and set all the necessary environment variables.

Installing the Software on x86_64 Linux on page 1-9

Install the software in a directory that you own, and set all the necessary environment variables.

Verifying the Installation on page 1-11

To verify the successful installation of the software, invoke the `version` Altera Software Development Kit (SDK) for OpenCL (AOCL) utility command.

Installing the Software on 64-Bit Windows

Install the software in a folder that you own, and set all the necessary environment variables.

Before you begin

You must have administrator privileges.

Running the Installer on Windows

- To install the Altera Software Development Kit (SDK) for OpenCL (AOCL), Quartus II software, and device support files simultaneously, perform the following tasks:
 1. Run the **QuartusSetup-<version>-windows.exe** 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 AOCL.
 3. Ensure that the *PATH* environment variable includes the **bin** folder in the Quartus II system folder (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 the 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.

Related Information

[Altera Software Installation and Licensing](#)

Setting the Environment Variables on Windows

You have the option to apply environment variables permanently or transiently.

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

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

- To apply transient environment variable settings, open a command window and run the `%ALTERAOCLSDKROOT%\init_openc1.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_openc1.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.

Installing the Software on x86_64 Linux

Install the software in a directory that you own, and set all the necessary environment variables.

Before you begin

- You must have `sudo` or root privileges.
- 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`).

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

Running the Installer on Linux

- To install the Altera Software Development Kit (SDK) for OpenCL (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 the 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 on Linux

You have the option to apply environment variables permanently or transiently.

Before you begin

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 apply permanent environment variable settings, manually include the following paths to the corresponding environment variables:

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

- To apply transient environment variable settings, open a command window and run the `source $ALTERAOCLSDKROOT/init_opencl.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 Installation

To verify the successful installation of the software, invoke the `version` Altera Software Development Kit (SDK) for OpenCL (AOCL) utility command.

- 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 Center](#)

Licensing the Software

Obtain the Altera Software Development Kit (SDK) for OpenCL (AOCL) license from the Altera Self Service Licensing Center. The AOCL license allows the AOCL 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 *Altera Software Installation and Licensing* manual, the **LICENSE.txt** file that accompanies each software, and the Altera Licensing page of the Altera website.

1. Log on to your myAltera account from the Altera website.
2. Click the link to the Self-Service Licensing Center.
3. Perform the steps outlined in 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.

Related Information

- [Altera Licensing](#)
- [Altera Software Installation and Licensing](#)
- [Altera website](#)
- [Self-Service Licensing Center](#)

Installing an FPGA Board (install)

To target an FPGA board, you must first download and install the Custom Platform from your board vendor. Most Custom Platform installers require administrator privileges.

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 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 of the Altera website.

3. Install the Custom Platform to a folder or directory.
4. Set the environment variable `AOCL_BOARD_PACKAGE_ROOT` to the subfolder or subdirectory within the Custom Platform that contains 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 or directory.

5. Add the Custom Platform library paths to the `PATH` (Windows) or `LD_LIBRARY_PATH` (Linux) environment variable. You may apply permanent settings manually or apply transient settings to the current command window by running the `init_opencil` script.
 - a. Apply permanent settings by adding the path to the memory-mapped (MMD) library within the Custom Platform.
 - b. Apply transient settings by opening a new command window and running the `%ALTERAOCLSDKROOT%\init_opencil.bat` (Windows) or `$ALTERAOCLSDKROOT/init_opencil.sh` (Linux) script.

For example, if you use `s5_net`, the Windows `PATH` environment variable setting is `%AOCL_BOARD_PACKAGE_ROOT%\windows64\bin`. The Linux `LD_LIBRARY_PATH` setting is `$AOCL_BOARD_PACKAGE_ROOT/linux64/lib`.

For more information on the `init_opencil` script, refer to the *Setting the Environment Variables on Windows* and *Setting the Environment Variables on Linux* sections of the *Altera SDK for OpenCL Getting Started Guide*.

6. Invoke the command `aocl install` at a command prompt.

Invoking the `aocl install` command 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

- [Altera SDK for OpenCL Cyclone V SoC Getting Started Guide](#)
- [OpenCL Reference Platforms page](#)
- [Querying the Device Name of Your FPGA Board \(diagnose\)](#) on page 1-15

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 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. On the [OpenCL Design Examples](#) page, click **Hello World**.
2. Under the *Using This Design Example* section, click to download the compressed file for your platform.
3. Extract the `hello_world` example to a location 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 subfolder or 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** folder or subdirectory 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.
```

When the line `aoc: Setting up project for CvP revision flow...` appears, it signifies the beginning of the hardware build, which is the time-consuming part of the compilation process. The AOC displays the line `aoc: Hardware generation completed successfully.` to signify the completion of the compilation process.

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*.

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

- [Installing an FPGA Board \(install\)](#) on page 1-12
- [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 (diagnose)

Some Altera Software Development Kit (SDK) for OpenCL 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 software 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, perform the following task:

- 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
-----
aocl0        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.

aocl1        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

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 ahead of time so that hardware on the board can use the image to configure the FPGA device when the host system powers up. 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 Driver for Windows 7 and Windows Vista](#) page on the Altera website.

2. Ensure that you set the `AOCL_BOARD_PACKAGE_ROOT` environment variable to point to the subfolder or 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.

Setting Up the Host Application

The procedures to build and run the host application that executes the OpenCL kernel on your device are operating system (OS)-specific.

[Building and Running the Host Application on Windows](#) on page 1-17

Build your host application in Microsoft Visual Studio version 2010 Professional, and run the application by invoking the `hello_world.exe` executable.

[Building and Running the Host Application on Linux](#) on page 1-18

You must build your host application with the `Makefile` file, and run the application by invoking the `hello_world` executable.

[Output from Successful Kernel Execution](#) on page 1-19

When you run the host application to execute your OpenCL kernel on the target FPGA, the Altera Software Development Kit (SDK) for OpenCL (AOCL) notifies you of a successful kernel execution.

Building and Running the Host Application on Windows

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

Building the Host Application

The `hello_world.sln` file in the `<path_to_exm_openc1_hello_world_x64_windows_<version>>\hello_world` folder 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 `.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.

Related Information

[Output from Successful Kernel Execution](#) on page 1-19

Building and Running the Host Application on Linux

You must build your 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 **Makefile** file, located in the `<path_to_exm_opencl_hello_world_x64_linux_<version>>/hello_world` directory.

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_opencl_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_opencl_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.

Related Information

[Output from Successful Kernel Execution](#) on page 1-19

Output from Successful Kernel Execution

When you run the host application to execute your OpenCL kernel on the target FPGA, the Altera Software Development Kit (SDK) for OpenCL (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_CACHESIZE = 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 and the FPGA Board

[Uninstalling the Software on 64-Bit Windows](#) on page 1-20

To uninstall the Altera Software Development Kit (SDK) for OpenCL (AOCL) on Windows, run the uninstaller, and restore all modified environment variables to their previous settings.

[Uninstalling the Software on x86_64 Linux](#) on page 1-20

To uninstall the Altera Software Development Kit (SDK) for OpenCL (AOCL), 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.

[Uninstalling the FPGA Board \(uninstall\)](#) on page 1-21

To uninstall an FPGA board, invoke the `uninstall` utility command, uninstall the Custom Platform, and unset the relevant environment variables.

Uninstalling the Software on 64-Bit Windows

To uninstall the Altera Software Development Kit (SDK) for OpenCL (AOCL) on 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 Software on x86_64 Linux

To uninstall the Altera Software Development Kit (SDK) for OpenCL (AOCL), 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 (uninstall)

To uninstall an FPGA board, 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. The Altera Software Development Kit (SDK) for OpenCL (AOCL) uses these drivers to communicate with the FPGA board.
For example, PCI Express (PCIe) drivers
3. Uninstall the Custom Platform.
4. Unset the environment variable `LD_LIBRARY_PATH` (for Linux) or `PATH` (for Windows).
5. Unset the environment variable `AOCL_BOARD_PACKAGE_ROOT`.

Conclusion

You have performed successfully all the steps necessary to execute an OpenCL application on an Altera preferred FPGA-based accelerator board. You may follow the same steps to execute a different OpenCL example application using the Altera Software Development Kit (SDK) for OpenCL (AOCL).

For more in-depth explanation on the features of the AOCL, refer to the *Altera SDK for OpenCL Programming Guide*. The programming guide also serves as a general guide to writing OpenCL applications with the AOCL. The *Altera SDK for OpenCL Best Practices Guide* helps you master optimization techniques for AOCL applications. Both guides are available on the AOCL documentation page of the Altera website.

Related Information

- [Altera SDK for OpenCL Programming Guide](#)
- [Altera SDK for OpenCL Best Practices Guide](#)

Document Revision History

Date	Version	Changes
December 2014	14.1.0	<ul style="list-style-type: none"> • Reorganized information flow. • Updated Red Hat Enterprise Linux (RHEL) version support. • Included the <i>Contents of the AOCL</i> section. • Updated licensing instructions for the new Altera Software Development Kit (SDK) for OpenCL (AOCL) single license. • Updated board uninstallation instructions to include the <code>aocl uninstall</code> utility command. • Included information on the <code>init_opencl</code> script for setting environment variables. • Grouped software and board uninstallation instructions under <i>Uninstalling the Software and the FPGA Board</i>.
June 2014	14.0.0	<ul style="list-style-type: none"> • Updated the <i>Prerequisites</i> section. • Updated the figure <i>AOCL Installation Process Overview</i>. • Updated software download instructions. • Updated AOCL installation and uninstallation instructions for Windows. • For Linux systems: <ul style="list-style-type: none"> • Added the sections <i>Installing the AOCL on Linux Using RPM</i> and <i>Uninstalling the AOCL on Linux Using RPM</i>. • Updated the section <i>Installing the AOCL on Linux Using the GUI Installer</i>. • Added the section <i>Licensing the Software</i>. • Updated the section <i>Installing an FPGA Board</i> with updated instructions on querying your devices and running diagnostic tests. • Updated the section <i>Creating the FPGA Hardware Configuration File of an OpenCL Kernel</i>: <ul style="list-style-type: none"> • Updated path that you have to set for <code>AOCL_BOARD_PACKAGE_ROOT</code>. • Updated example AOC output for compiling <code>hello_world.cl</code> with the <code>-v</code> option. • Updated AOC output. • Added the section <i>Identifying the Device Name of Your FPGA Board</i>. • Modified instructions for building and running the host application with updated <code>hello_world</code> directory and file names. • Added the section <i>Uninstalling an FPGA Board</i>.
December 2013	13.1.1	<ul style="list-style-type: none"> • Updated the <i>Prerequisites</i> section to include a reminder to install Linux kernel source, headers, and GCC.

Date	Version	Changes
November 2013	13.1.0	<ul style="list-style-type: none"> • Reorganized information flow. • Updated the <i>Prerequisites</i> section. • Updated board installation instructions. • Updated software download instructions. • Inserted the section <i>Installing the AOCL and the Quartus II Software Together</i>. • Updated software installation and uninstallation instructions. • Inserted the following figures: <ul style="list-style-type: none"> • <i>AOCL Installation Process Overview</i> • <i>FPGA Programming Overview</i> • Removed the <i>Licensing</i> section. • Removed all board-specific installation and configuration instructions. • Changed example OpenCL application used to demonstrate kernel configuration and FPGA programming from <code>moving_average</code> to <code>hello_world</code>. • Inserted the section <i>Updating the Hardware Image on the FPGA</i>, which contained the updated flash programming instructions. • Removed the section <i>Installing the USB-Blaster Driver on Windows</i>. • Updated output from successful execution of <code>hello_world</code> kernel on FPGA for Windows and Linux systems. • Removed the figure <i>Contents of the Moving Average Example</i>. • Removed the figure <i>Opening host.sln in Visual Studio</i>.
June 2013	13.0 SP1.0	<ul style="list-style-type: none"> • Updated requisite Quartus II and AOCL software versions from 13.0 to 13.0 SP1. • Inserted the figure <i>A Correct Windows Device Manager After Complete Board Driver Installation for a BittWare Board</i>. • Updated the vendor and device IDs in the <i>Verifying the Functionality of the BittWare Board</i> section for Windows. • 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. • Updated path to the AOCL design examples. • Updated the figure <i>Contents of the Moving Average Example</i>. • Updated flash programming instructions.

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