

Intel® SDK for OpenCL™ - CPU only Runtime Package 16.1.2 for Intel Core™ and Intel Xeon® Processors Release Notes

November 17, 2017

Contents

- 1 Introduction..... 2
 - Customer support 2
- 2 What's New..... 2
- 3 System Requirements..... 5
 - Processor Requirements 5
 - Supported Operating Systems 6
- 4 Installation Notes 7
 - 4.1 Installation on Microsoft Windows* 7
 - 4.2 Installation on Linux* 7
 - 4.2.1 Installing the Product 7
 - 4.2.2 Uninstalling the Product 7
- 5 Installation and Configuration Issues 8
 - PATH Environment Variable 8
 - Installation Conflict 8
 - 5.1 Manual Installation on Windows* 8
 - 5.2 Manual Installation on Linux* 9
- 6 Known Issues..... 9
- 7 Disclaimer and Legal Information.....10

1 Introduction

The Intel® SDK for OpenCL™ - CPU only runtime package 16.1.2 adds OpenCL™ support for CPU devices.

More runtimes and drivers with support for Intel® Graphics Technology and the Intel Xeon Phi™ coprocessor are available at <https://software.intel.com/en-us/articles/opencl-drivers>.

The Intel® SDK for OpenCL™ - CPU only runtime package is fully compatible with Intel® SDK for OpenCL™ Applications – a complete development suite for developing, debugging and profiling OpenCL™ applications. For more information, see <https://software.intel.com/en-us/intel-opencl>.

This document provides system requirements, installation instructions, issues and limitations, and legal information.

Customer support

To learn more about this product, see documentation, FAQ, code samples, and other support information at this site: <https://software.intel.com/en-us/intel-opencl-support>.

For technical support, including answers to questions not addressed in the installed product, go to the OpenCL™ forum at this site: <https://software.intel.com/en-us/forums/opencl>.

2 What's New

The 16.1.2 release update includes:

- New optional `__attribute__((intel_vec_len_hint(<uint>)))`

This attribute can be used to provide a hint to the compiler that the kernel will perform best if vectorized to the specified vector length.

You can specify one of the following lengths for this attribute:

0	The compiler uses heuristics to decide whether to vectorize the kernel, and if so, which vector length to use. This is the default behavior.
1	No vectorization is performed by the compiler. Explicit vector data types in kernels are left intact.

4	Disables heuristics and vectorizes to the length of 4 respectively.
8	Disables heuristics and vectorizes to the length of 8 respectively.

- New OpenCL™ C predefined macro `__INTEL_OPENCL_CPU_<CPUSIGN>`

This macro can be used to fine tune the kernel for a specific CPU device microarchitecture. <CPUSIGN> is the CPU signature of a device.

You can specify one of the following values for this macro:

Macro	Intel Microarchitectures
<code>__INTEL_OPENCL_CPU_SKL__</code>	Intel® microarchitecture code name Skylake
<code>__INTEL_OPENCL_CPU_SKX__</code>	Intel® microarchitecture code name Skylake on Intel Xeon® processor family
<code>__INTEL_OPENCL_CPU_BDW__</code>	Intel® microarchitecture code name Broadwell
<code>__INTEL_OPENCL_CPU_BDW_XEON__</code>	Intel® microarchitecture code name Broadwell on Intel Xeon® processor family
<code>__INTEL_OPENCL_CPU_HSW__</code>	Haswell microarchitecture
<code>__INTEL_OPENCL_CPU_HSW_XEON__</code>	Haswell microarchitecture on Intel Xeon® processor family
<code>__INTEL_OPENCL_CPU_IVB__</code>	Intel® microarchitecture code name Ivy Bridge
<code>__INTEL_OPENCL_CPU_IVB_XEON__</code>	Intel® microarchitecture code name Ivy Bridge on Intel Xeon® processor family
<code>__INTEL_OPENCL_CPU_SNB__</code>	Intel® microarchitecture code name Sandy Bridge
<code>__INTEL_OPENCL_CPU_SNB_XEON__</code>	Intel® microarchitecture code name Sandy Bridge on Intel Xeon® processor family

__INTEL_OPENCL_CPU_WST__	Intel® microarchitecture code name Westmere
__INTEL_OPENCL_CPU_WST_XEON__	Intel® microarchitecture code name Westmere on Intel Xeon® processor family
__INTEL_OPENCL_CPU_UNKNOWN__	Unknown microarchitecture

- Improved heuristics for choosing local size when ndrange is enqueued to the command queue that was created with CL_QUEUE_THREAD_LOCAL_EXEC_ENABLE_INTEL property (extension https://www.khronos.org/registry/OpenCL/extensions/intel/cl_intel_thread_local_exec.txt).
- A fix for a previous issue where an incorrect library was loaded when running on Intel® microarchitecture code name Skylake.

The 16.1.1 release update includes:

- A fix for the known incompatibility issue with the CPU Kernel Debugger from the Intel® SDK for OpenCL™ Applications 2016 R2 and the CPU only runtime package version.
- The following performance optimizations:
 - Compiler vectorizer heuristic tuning for a set of workloads
 - Workgroup fusion optimization improvements
 - Performance enhancements of the vload()/vstore() built-in functions
- A fix for the issue reported on the forum (<https://software.intel.com/en-us/comment/1844607#comment-1844607>) where the vectorizer produces incorrect code on Intel processors that support Intel® Streaming SIMD Extensions 4.2 (Intel® SSE 4.2) instructions when using the samplerless read_imagef() built-in function with image2d_t and int2 coordinates as arguments.
- The cl_khr_gl_sharing extension was disabled because of incompatibility with the Microsoft* Basic Display Adapter. To use this extension, please install the OpenCL™ Driver for Iris® graphics and Intel® HD Graphics for Windows* from <https://software.intel.com/en-us/articles/opencl-drivers#iris>. The driver package includes the OpenCL™ Runtime package for CPUs.

Because of a performance issue, the Intel® Threading Building Blocks (Intel® TBB) library was downgraded from 4.2, Interface version 7001, Oct 2 2013" to 4.2, Interface version 7005 , Jun 1 2014.

The 16.1 release includes:

- Support for Intel® Core™ 6th generation and Intel Xeon® v4 processors (former Intel® microarchitecture code name Broadwell)
- Support for the OpenCL™ 2.0 specification
- Improved cross-CPU support of a pre-compiled kernel binary in runtime:
 - Enables loading pre-generated kernel binaries that save OpenCL™ program build time. For more information, see <https://software.intel.com/en-us/node/540584>
 - Enables generating a JIT binary for the target CPU model by the Intel® SDK for OpenCL™ - offline compiler. For more information, see <https://software.intel.com/en-us/node/539388>
- Bug and memory leak fixes.
- The compiler infrastructure was updated to LLVM* version 3.6.2.

NOTE: OpenCL™ Runtime 16.1.2 supports CPU only. For Intel Xeon Phi™ coprocessor support, use the OpenCL™ runtime 14.2. For more information, see OpenCL™ runtime entry and release notes on the OpenCL™ driver page at: <https://software.intel.com/en-us/articles/opencl-drivers> .

3 System Requirements

For an explanation of architecture names, see <http://software.intel.com/en-us/articles/intel-architecture-platform-terminology/>.

Processor Requirements

The Intel® SDK for OpenCL™ - CPU only runtime package 16.1.2 provides CPU device support on the following processors:

- Intel Core™ Processors with Intel® Streaming SIMD Extensions 4.2 (Intel® SSE4.2) support or higher
- Intel Xeon® processor E3, E5, and E7 families with Intel® SSE4.2 support or higher

Intel® SDK for OpenCL™ - CPU only runtime package 16.1.2 provides optimizations for processors that support following instruction sets:

- Intel® Streaming SIMD Extensions 4.2 (Intel® SSE4.2)
- Intel® Advanced Vector Extensions (Intel® AVX)
- Intel® Advanced Vector Extensions 2 (Intel® AVX2)

Intel® SDK for OpenCL™ - CPU only runtime package 16.1.2 optimizes kernels for the Intel® AVX2 instruction set on Intel® microarchitecture code name Skylake.

To enable GPU device support on the aforementioned processors, install the Intel® Graphics Driver. The graphics driver includes the CPU runtime as well.

NOTE: Incompatible or proprietary instructions in non-Intel processors may cause the analysis capabilities of this product to function incorrectly. Any attempt to analyze code not supported by Intel processors may lead to failures in this product.

Supported Operating Systems

The following is the list of supported operating systems:

Linux* operating systems:

- Red Hat Enterprise Linux* OS 6.5 or higher (64-bit version)
- SUSE Linux Enterprise Server* 11.3 or higher (64-bit version)
- Ubuntu* 14.04
- Cent OS 7.0 or higher (64-bit version)

Windows* operating systems (32- and 64-bit):

- Microsoft Windows* 7 SP1
- Microsoft Windows* 8 / 8.1
- Microsoft Windows* Server 2008 R2
- Microsoft Windows* Server 2012
- Microsoft Windows* 10

Because of possible incompatibility of Intel® Advanced Vector Extensions (Intel® AVX) issues with the default glibc 2.11.1 implementation, the product libraries require glibc-2.12-1.47 or higher. Refer to the operating system documentation for more information.

4 Installation Notes

4.1 Installation on Microsoft Windows*

To install the Intel® SDK for OpenCL™ - CPU only runtime package on Windows* systems, download the Runtime package and follow the installer prompts.

To remove the Intel® SDK for OpenCL™ - CPU only runtime package, go to **Control Panel > Programs and Features > OpenCL™ Runtime > Uninstall**.

The uninstaller removes all originally installed files, leaving any temporary or newly created files. To ensure a clean uninstallation, verify that the INTELOCLSDKROOT, INTELOCLSAMPLESROOT, and PATH environment variables are in their preinstall state.

4.2 Installation on Linux*

The following sections describe installing and uninstalling this product on Linux*.

4.2.1 Installing the Product

To install the product, do the following:

1. Extract the TGZ archive contents

```
# tar xzf openc1_runtime_16.1.2_x64_<OS>_<VERSION>.tgz
# cd openc1_runtime_16.1.2_x64_<OS>_<VERSION>
```

2. Run the following command (for command-line interface) and follow the installer prompts:

```
# ./install.sh
```

Alternatively (for installation with graphical user interface), run the following command:

```
# ./install_GUI.sh
```

4.2.2 Uninstalling the Product

To uninstall the product using the uninstallation script, do the following:

1. Go to the folder with the Intel® SDK for OpenCL™ - CPU only runtime package installation (for example, /opt/intel/opencl).
2. Run the uninstall.sh script.

You can use an operating system-specific command to remove all the packages, starting with "opencl-1.2-". To do so, run the following commands:

For Red Hat Enterprise Linux OS:

```
# sudo yum remove "opencl-1.2-*
```

For SUSE Linux Enterprise OS:

```
# sudo zypper remove "opencl-1.2-*
```

5 Installation and Configuration Issues

PATH Environment Variable

The Intel® SDK for OpenCL™ - CPU only runtime package installer adds the target installation folder of the CPU runtime to the end of the system PATH environment variable. If the variable is too long, the application might not be able to load the CPU runtime DLL files. To solve this problem, move the folders to the beginning of the PATH variable or delete unnecessary folders from the PATH.

Installation Conflict

If the Intel® HD Graphics Driver is installed on your system, you will not be able to install the Intel® SDK for OpenCL™ - CPU only runtime package.

However, to update OpenCL™ support without impacting the display driver, you can perform a manual installation of the package by performing the actions specified in section 5.1 (Windows*) or 5.2 (Linux*).

5.1 Manual Installation on Windows*

1. Open a command prompt.
2. Download the Intel® SDK for OpenCL™ - CPU only runtime package (MSI version)
3. Extract binaries from the MSI package:

```
msiexec /a opencl_runtime_16.1.2_x64_setup.msi /qn TARGETDIR=<TMP_DIR>
```

4. Copy the content of <TMP_DIR>\ProgramFilesFolder\Common Files\Intel\OpenCL to <TARGET_DIR>
5. Replace the OpenCL.dll files in C:\Windows\System32 and C:\Windows\SysWOW64 with files from <TMP_DIR>\SystemFolder and <TMP_DIR>\System64 directories, correspondingly.
If you want to restore the initial files, preserve them in a temporary location.
6. Add <TARGET_DIR>\bin\x64 and <TARGET_DIR>\bin\x86 to the PATH environment variable.
7. Disable all Intel related keys in the branches below by setting them to 1:
[HKEY_LOCAL_MACHINE\SOFTWARE\Khronos\OpenCL\Vendors]
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Khronos\OpenCL\Vendors]
8. Add registry key as follows:
[HKEY_LOCAL_MACHINE\SOFTWARE\Khronos\OpenCL\Vendors]
"<TARGET_DIR>\bin\x64\Intelocl64.dll"=dword:00000000
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Khronos\OpenCL\Vendors]
"<TARGET_DIR>\bin\x86\Intelocl32.dll"=dword:00000000

5.2 Manual Installation on Linux*

1. Open a command prompt
2. Download the Intel® SDK for OpenCL™ - CPU only runtime package (Linux version)
3. Create <TMP_DIR> and change the current directory to it
4. Extract binaries from the package:

```
tar xzvf openc1_runtime_*.tgz
rpm2cpio openc1_runtime_*/rpm/openc1-1.2-intel-cpu-*.rpm | cpio -idmv
rpm2cpio openc1_runtime_*/rpm/openc1-1.2-base-*.rpm | cpio -idmv
```

5. Copy content of temporary directory to <TARGET_DIR>
cp -R <TMP_DIR>/opt/intel/openc1-*/lib64/. <TARGET_DIR>
6. Replace libOpenCL.so* files available in the system with the ones from <TMP_DIR>/opt/intel/openc1-*/lib64/
If you would like to restore the initial files, preserve them in a temporary location.
7. Add <TARGET_DIR> to the LD_LIBRARY_PATH environment variable.
8. Rename any Intel-related ICD files in the directory by adding suffix .hide
9. Create a new ICD file; for example:

```
echo <TARGET_DIR>/libintelocl.so > /etc/OpenCL/vendors/intel_manual.icd
```

Note: Make sure that libOpenCL.so* libraries are available on the machine.

6 Known Issues

The following are known issues:

- Intel® SDK for OpenCL™ - CPU only runtime requires Intel® Threading Building Blocks (Intel® TBB) version 4.2.1, which is included in the OpenCL Runtime package installation folder.

Make sure there is no Intel® TBB version conflict in your system upon runtime installation:

- Any standalone Intel® TBB package loaded by the OpenCL™ host code should be of a higher version than the OpenCL™ version of Intel® TBB.
 - The standalone Intel® TBB package must use the default Intel® TBB configuration, which is also used by the OpenCL™ runtime.
 - Make sure you use and load the correct Intel® TBB libraries. For example, if you plan to use new features of a standalone Intel® TBB version higher than the OpenCL version, ensure that the corresponding standalone Intel® TBB libraries are correctly loaded (LD_LIBRARY_PATH in Linux* or PATH in Windows*).
- Device fission extension is not supported. Only the device fission core feature is supported.
 - If you use CL_DRIVER_VERSION to find the version of Intel® SDK for OpenCL™ - CPU only runtime package for Linux*, it incorrectly returns 1.2.0.10037. The correct version is 6.4.0.19.
On Windows*, the platform version is reported correctly.

7 Disclaimer and Legal Information

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, Intel Atom, Intel Atom Inside, Intel Core, Intel vPro, Intel Xeon Phi, Itanium, Pentium, Ultrabook, VTune, Xeon, are trademarks of Intel Corporation in the U.S. and/or other countries.

* Other names and brands may be claimed as the property of others.

Microsoft, Windows, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission from Khronos.

The product is based on a published Khronos Specification and is expected to pass the Khronos Conformance Process. Current conformance status can be found at <http://www.khronos.org/conformance> .

Copyright © 2017, Intel Corporation. All rights reserved.