Intel® CPU Runtime for OpenCL™ Applications 18.1 for Intel Core™ and Intel Xeon® Processors Release Notes

October 25, 2018

Contents

1 Introduction

The Intel® CPU Runtime for OpenCL™ Applications 18.1 (previously known as Intel® SDK for OpenCL™ - CPU only Runtime Package) provides OpenCL™ support for Intel® 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® CPU Runtime for OpenCL™ Applications 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 contains system requirements, installation instructions, information about known 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.

Refer to https://software.intel.com/en-us/iocl_tec_opg for OpenCL™ Developer Guide for Intel® Core™ and Intel® Xeon® Processor.

2 What's New

The 18.1 release includes:

- Support of Intel® Advanced Vector Extensions 512 (Intel® AVX-512) ISA on Intel® Xeon® Platinum processor (formerly code name Skylake).
- Enabled features of OpenCL™ 2.1. The product is based on a published Khronos* Specification and has passed the Khronos Conformance Process. The conformance record can be found at https://www.khronos.org/conformance/adopters/conformant-products/opencl. Refer to submission #322 recorded on October 7, 2018.

- Support for vectorization width 16 for the environment and configuration file variable `CL_CONFIG_CPU_VECTORIZER_MODE`, as well as for OpenCL™ C optional kernel attribute `intel_vec_len_hint`

- Support for OpenCL™ Kernel debugging on Linux* OS with GDB*

- Improved coexistence support with Intel® Graphics Compute Runtime for OpenCL™ Driver when both are installed

- Changed the platform name returned via `clGetPlatformInfo` OpenCL™ API call with `CL_PLATFORM_NAME` bitflag to "Intel(R) CPU Runtime for OpenCL(TM) Applications"

- New environment variable `CL_CONFIG_CPU_TARGET_ARCH`

  It generates code exclusively for a given target CPU architecture. Allows only lowering the instruction set level supported by CPU

  Allowed values are:

`skx`	Generates code for processors that support Intel® Advanced Vector Extensions 512 (Intel® AVX-512) Foundation instructions, Intel® AVX-512 Conflict Detection instructions, Intel® AVX-512 Doubleword and Quadword instructions, Intel® AVX-512 Byte and Word instructions and Intel® AVX-512 Vector Length Extensions for Intel® processors, and the instructions enabled with `core-avx2`.
`core-avx2`	Generates code for processors that support Intel® Advanced Vector Extensions 2 (Intel® AVX2), Intel® AVX, SSE4.2 SSE4.1, SSE3, SSE2, SSE, and SSSE3 instructions.
`corei7-avx`	Generates code for processors that support Intel® Advanced Vector Extensions (Intel® AVX), Intel® SSE4.2, SSE4.1, SSE3, SSE2, SSE, and SSSE3 instructions.
`corei7`	Generates code for processors that support Intel® SSE4.2 Efficient Accelerated String and Text Processing instructions. May also generate code for Intel® SSE4 Vectorizing Compiler and Media Accelerator, Intel® SSE3, SSE2, SSE, and SSSE3 instructions.

- Fixed an issue with user functions not being inlined in programs created using `clCreateProgramWithIL` OpenCL™ API call

- Fixed incorrectly reported `CL_DEVICE_MAX_COMPUTE_UNITS` for multi-socket Intel® Xeon™ systems (reported on forum https://software.intel.com/en-us/forums/opencl/topic/702240)
- Fixed incompatibility with Intel® Threading Building Blocks (Intel® TBB) max_allowed_parallelism parameter
- Fixed an issue with `CL_DRIVER_VERSION` returning incorrect driver version
- Improved OpenCL™ C compiler diagnostics
- Minor bug fixes
- Updated the compiler infrastructure to LLVM* version 6.0

NOTE: Intel® CPU Runtime for OpenCL™ Applications 18.1 supports CPU only. For Intel Xeon Phi™ coprocessor support, use the version 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/.

3.2.1 Processor Requirements

The Intel® CPU Runtime for OpenCL™ Applications 18.1 provides CPU device support on the following processors:

- Intel Core™ processor family 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 Xeon® Scalable processors Platinum, Gold, Silver, Bronze families with Intel® SSE4.2 support or higher

Intel® CPU Runtime for OpenCL™ Applications 18.1 provides optimizations for processors that support the following instruction sets:

- Intel® Advanced Vector Extensions 512 (Intel® AVX-512) Foundation instructions (Intel® AVX-512F), Intel® AVX-512 Conflict Detection instructions (AVX-512CD), Intel® AVX-512 Doubleword and Quadword instructions (AVX-512DQ), Intel® AVX-512 Byte and

Word instructions (AVX-512BW) and Intel® AVX-512 Vector Length Extensions (AVX-512VL)

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

To enable Intel® Graphics Technology device support on the aforementioned processors on Windows*, install the Intel® Graphics Driver package. It includes the Intel® CPU Runtime for OpenCL™ Applications. For Intel® Graphics Technology support on Linux* operating systems, refer to Intel® Graphics Compute Runtime for OpenCL™ (https://01.org/compute-runtime).

***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.

3.2.2 Software Requirements

Supported Operating Systems

The following is the list of supported operating systems:

Linux* operating systems (64-bit only):

- Red Hat Enterprise Linux* OS 7.4
- SUSE Linux Enterprise Server* 12SP3
- Ubuntu* 16.04
- Cent OS 7.4

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

- Microsoft Windows* Server 2016
- Microsoft Windows* 10

Higher versions are expected to be functional.

On Linux* OS libnuma1, zlib1g, libxml2 packages are required.

For better OpenCL™ Kernel CPU Debugging experience on Linux* OS, GDB* 7.12 or higher is required. In cases where an older GDB* version is used, some issues related to JIT code debugging may occur (such as no automatic breakpoints reset between runs, fail to stop on a breakpoint for second and other runs).

4 Installation Notes

4.1 Installation on Microsoft Windows*

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

To remove the Intel® CPU Runtime for OpenCL™ Applications, go to **Control Panel** > **Programs and Features** > **OpenCL™ Runtime** > **Uninstall**.

4.2.3 Installation on Linux*

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

4.2.4 Installing the Product

To install the product, do the following:

1. Extract the TGZ archive contents:

   ```
   # tar xzf opencl_runtime_18.1_x64_<OS>_<VERSION>.tgz

   # cd opencl_runtime_18.1_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
   ```

4.2.5 Uninstalling the Product

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

1. Go to the folder with the Intel® CPU Runtime for OpenCL™ Applications 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 “intel-openclrt-18.1”. To do so, run the following commands:

- For Red Hat Enterprise Linux OS:

```
# sudo yum remove "intel-openclrt-18.1.*"
```

- For SUSE Linux Enterprise OS:

```
# sudo zypper remove "intel-openclrt-18.1.*"
```

- For Ubuntu OS:

```
# sudo apt-get purge "intel-openclrt-18.1.*"
```

5 Installation and Configuration Issues

PATH Environment Variable

The Intel® CPU Runtime for OpenCL™ Applications 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 Windows* system, you will not be able to install the Intel® CPU Runtime for OpenCL™ Applications.

However, to update OpenCL™ for CPU support, you can perform a manual installation of the package by performing the actions specified in section 5.1 (Windows*) or 5.2 (Linux*) of this document. Note that these steps performed on a system with Intel® HD Graphics Driver installed do not guarantee OpenCL™ for Intel® HD Graphics functionality to remain stable.

5.1 Manual Installation on Windows* OS

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

```
msiexec /a opencl_runtime_18.1_x64_setup.msi /qn TARGETDIR=<TMP_DIR>
```

4. Copy the content of `<TMP_DIR>\Intel\OpenCL` to `<INST_DIR>` of your choice.

5. Replace the `OpenCL.dll` files in `C:\Windows\SysWOW64` and `C:\Windows\System32` with files from `<TMP_DIR>\SystemFolder` and `<TMP_DIR>\System64Folder` directories correspondingly.
   If you want to restore the initial files, preserve them in a temporary location.
6. Add `<INST_DIR>\windows\compiler\lib\intel64_win` and `<INST_DIR>\windows\compiler\lib\ia32_win` 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]
   "<INST_DIR>\windows\compiler\lib\intel64_win\Intelocl64.dll"=dword:00000000
   [HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Khronos\OpenCL\Vendors]
   "<INST_DIR>\windows\compiler\lib\ia32_win\Intelocl32.dll"=dword:00000000

5.2 Manual Installation on Linux* OS

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

```
tar xzvf opencl_runtime_*.tgz
rpm2cpio opencl_runtime_*/rpm/intel-openclrt-18.1*.rpm | cpio –idmv
```

5. Copy content of temporary directory to <TARGET_DIR>
   ```
   cp -R <TMP_DIR>/opt/intel/opencl_*/linux/compiler/lib/.
   <TARGET_DIR>
   ```
6. Replace `libOpenCL.so*` files available in the system with the ones from `<TMP_DIR>/opt/intel/opencl_*/linux/compiler/lib/intel64_lin`
7. If you would like to restore the initial files, preserve them in a temporary location.
8. Add `<TARGET_DIR>/intel64_lin` to the LD_LIBRARY_PATH environment variable.
9. Rename any Intel-related ICD files in the directory by adding suffix `.hide`
10. Create a new ICD file, for example:

```
echo <TARGET_DIR>/intel64_lin/libintelocl.so >
/etc/OpenCL/vendors/intel_manual.icd
```

NOTE: Make sure that `libOenCL.so*` libraries are available on the machine.

6 Known Issues

The following are known issues:

- Intel® CPU Runtime for OpenCL™ Applications has a dependency on Intel® Threading Building Blocks (Intel® TBB) version 2017 Update 8 on Linux* and 2018 Update 2 on Windows* OS. The library is included in the Intel® CPU Runtime installation directory.
    - Linux* OS case:

      Make sure there is no other Threading Building Blocks (TBB) library in your OpenCL™ host application library search path on Linux* OS. Intel® CPU Runtime for OpenCL™ Applications was tested only with Intel® Threading Building Blocks (Intel® TBB) libraries included in the package.

      In case OpenCL™ host application intentionally uses features of a standalone Threading Building Blocks (TBB) make sure that it is of a higher version than the library version in the package and is found earlier in the shared library search procedure. If standalone Threading Building Blocks (TBB) libraries are loaded functionality and performance may vary.
    - Windows* OS case:

      In Windows* OS product distribution Intel® Threading Building Blocks (Intel® TBB) libraries included in the package have different names rather than in standard shipment. That ensures that no load conflict takes place with pre-installed libraries.

      In case OpenCL™ host code uses features of a standalone Threading Building Blocks (TBB) libraries application performance may vary (e.g. due to oversubscription issues in case of both standalone library pool and OpenCL thread pool being fully loaded).
- After OpenCL™ Contexts are created for both “Experimental OpenCL 2.1 CPU Only Platform” and “Intel(R) CPU Runtime for OpenCL(TM) Applications” OpenCL™ Kernel compilation for any of the two may destabilize the application. Use “Intel(R) CPU Runtime for OpenCL(TM) Applications” platform for features of “Experimental OpenCL 2.1 CPU Only Platform” instead.
- Compilation of kernels with empty loop bodies may not terminate. Disable vectorization or all compiler optimizations in that case.
- Configuration file parameter CL_CONFIG_CPU_TARGET_ARCH cannot be set from cl.cfg, only as environment variable.

- The ‘finish’ GDB* or LLDB* command return to the same line as a function call. The ‘finish’ command should advance instruction pointer to the next instruction after the call. It is expected that it will point to the next line following the line of calling, except when a callee function returns a value for using. In the case of OpenCL™ kernel code it points to the same line as the call, even the callee function is a void one.

- Setting `__local` variable values within a debugger session (‘set’ GDB* command) has no effect on program execution. An attempt to set variables in a local address space has no effect. The new value will be discarded just after the next step.

- Conditional breakpoints on work-items may take long time to evaluate. It may take time to evaluate conditional breakpoints depending on a work item number for heavy NDRange kernels. To avoid such breakpoints, the kernel source code can be modified using the ‘get_global_id’ function.

- Installing Intel® CPU Runtime for OpenCL™ Applications 18.1 over an existing 16.1.2 deployment may result in an extra, non-usable CPU OpenCL™ device still detected. Manually assisting uninstallation of 16.1.2 should remove the unintended platform and device detection:

    1. Remove following registry keys (see the system regedit utility):

        - On 32-bit OS:

            - HKEY_LOCAL_MACHINE\SOFTWARE\Khronos\OpenCL\Vendors \intelopencl32.dll

        - On 64-bit OS:

            - HKEY_LOCAL_MACHINE\SOFTWARE\Khronos\OpenCL\Vendors \intelopencl64.dll

            - HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Khronos\ OpenCL\Vendors\intelopencl32.dll

    2. (optional) Remove following files if they exist:

        - On 32-bit OS:

            - C:\Windows\System32\intelopencl32.dll

        - On 64-bit OS:

- C:\Windows\System32\intelopencl64.dll
- C:\Windows\SysWOW64\intelopencl32.dll

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 Corporation 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 has passed the Khronos Conformance Process. The conformance record can be found at https://www.khronos.org/conformance/adopters/conformant-products/opencl. Refer to submission #322 recorded on October 7, 2018.