OpenCL™ Driver for Intel® HD, Iris™, and Iris™ Pro Graphics for Linux -- Installation[1]

Version Information

This document covers the Intel® OpenCL Linux graphics device driver version r5.0-BUILD_ID, hereafter referred to as the intel-opencl-r5.0 driver, where BUILD_ID refers to the build ID of the distributed files.

Intel® Media Server Studio 2017 Interoperability

PLEASE NOTE: This user-mode driver and 4.7 kernel patch set are focused on OpenCL compute use cases. Unless otherwise specified, interoperability with other drivers, operating systems, or platform features is not verified or supported.

Systems configured with Intel® Media Server Studio 2017 using its Getting Started Guide have completed the intel-opencl-r2.0 not intel-opencl-r5.0 driver procedure described in this installation document.

This release does not support Intel® Media Server Studio 2017 release.

Intel® Media Server Studio 2017 R2 Interoperability

Systems configured with Intel® Media Server Studio 2017 R2 using its Getting Started Guide have completed the intel-opencl-r4.1 not intel-opencl-r5.0 driver procedure described in this installation document.

This release does not support Intel® Media Server Studio R2 2017 release.

Intel® Media Server Studio 2017 R3 Interoperability

Systems configured with Intel® Media Server Studio 2017 R3 using its Getting Started Guide have completed the intel-opencl-r5.0 driver procedure described in this installation document.

This release supports Intel® Media Server Studio 2017 R3 release.

Overview

This document provides instructions on installing the intel-opencl-r5.0 driver for 64-bit Linux platforms. Some steps may involve time-consuming operations for certain platforms. Please review the full document to ensure a successful outcome.

Installation

Installation of the intel-opencl-r5.0 driver requires a 5th, 6th or 7th generation Intel® processor with Intel® Processor Graphics Technology not previously disabled by the BIOS or motherboard settings or Intel® Atom™ processor E3900 series, Intel Pentium N4200, J4000 Intel Celeron J3000, N3350. If unsure, consult the

---

hardware manufacturer's documentation or the Intel(R) CPU part information http://ark.intel.com. The CPU architecture can be verified using standard Linux operating system features. For example:

```
$ grep -m 1 name /proc/cpuinfo
model name  : Intel(R) Core(TM) i7-6999R CPU @ 3.20GHz
```

Intel® Processor Graphics availability to the OS can be tested like this:

```
$ lspci -nn | grep VGA
00:02.0 VGA compatible controller [0300]: Intel Corporation Device [8086:9999] (rev 08)
```

If no graphics controller is listed for a processor model with processor graphics capabilities, the root cause is "below" the OS. Please consult your OEM for more details, as next steps can be system specific. In some cases BIOS setting changes or BIOS updates can help.

Intel validates the intel-opencl-r5.0 driver on CentOS 7.2 and 7.3 when running the following 64-bit kernel configurations:

- Linux 4.7 kernel patched for intel-opencl-r5.0
- Linux 4.4 kernel patched for MSS 2017 R3

In addition to CentOS 7.2 and 7.3, the linux.org 4.7 kernel has the flexibility to be compiled and installed on a variety of Linux distributions. It is required by this release to patch the vanilla linux.org 4.7 kernel with the specific patch-set to add functionality not available in the official linux.org 4.7 kernel but needed for all the intel-opencl-r5.0 OpenCL features described in the release notes to function correctly.

If this driver is bundled into another product, such as Media Server Studio, it may include additional or alternative validated kernel configurations. Please refer to that product's documentation for more details.

The remainder of this document makes the following assumptions:

- the configured Linux operating system will have network access or the ability to install additional packages from the installation image,
- a non-privileged account exists (referred to as *USERNAME*), and
- unless explicitly specified, all commands are assumed to be executed by *USERNAME* and from *USERNAME*'s home directory.

Intel® Media Server Studio 2017, CentOS

Systems previously configured for Intel® Media Server Studio 2017 must be reconfigured with an updated 4.7 Linux kernel and an intel-opencl-r5.0 driver to install this release. Note: this is an upgrade path for OpenCL features only – there are no guarantees that media features will not be affected. Install of this package will disable Intel® Media Server Studio 2017 support.

Intel® OpenCL drivers installed during the installation of Intel® Media Server Studio 2017 using RPMs must be removed prior to installing the newer, intel-opencl-r5.0 driver.

1. Query the system for any existing Intel® OpenCL RPMs.

   ```
   $ rpm -qa | grep intel-opencl
   PACKAGE1.x86_64
   PACKAGE2.x86_64
   ```

2. Remove all RPMs listed in the previous step.

   ```
   $ sudo rpm -e --nodeps PACKAGE1 PACKAGE2
   ```

Intel® Media Server Studio 2017 R2, CentOS

Intel-opencl-r4.0 driver is part of the MSS 2017 R2 release.

To install intel-opencl-r5.0 driver on a MSS 2017 R2 preconfigured system you need to update the user mode drivers:

```
$ sudo rpm -Uvh intel-opencl-r5.0-BUILD_ID.x86_64.rpm
$ sudo rpm -Uvh intel-opencl-devel-r5.0-BUILD_ID.x86_64.rpm
$ sudo rpm -Uvh intel-opencl-cpu-r5.0-BUILD_ID.x86_64.rpm
```

Intel® Media Server Studio 2017 R2, Generic

Intel-opencl-r4.0 driver is part of the MSS 2017 R2 release.

To install intel-opencl-r5.0 driver on a MSS 2017 R2 preconfigured system you need to install the user mode drivers:

```
$ mkdir intel-opencl
$ tar -C intel-opencl -Jxf intel-opencl-r5.0-BUILD_ID.x86_64.tar.xz
$ tar -C intel-opencl -Jxf intel-opencl-devel-r5.0-BUILD_ID.x86_64.tar.xz
$ tar -C intel-opencl -Jxf intel-opencl-cpu-r5.0-BUILD_ID.x86_64.tar.xz
$ sudo cp -R intel-opencl/* /
$ sudo ldconfig
```

CentOS 7.2 and 7.3 with Linux 4.7 Kernel

CentOS Installation and Preliminary Setup

Although CentOS offers several base environments, this section assumes a *minimal* environment. Some of the packages listed in the following steps may already be available when using alternative environments.

1. Configure and install CentOS 7.2 or 7.3 using the minimal base environment or better. See https://www.centos.org for downloadable images and available installation methods.

2. Install necessary dependencies.

   ```
   $ sudo yum install tar libpciaccess numactl-libs
   ```

3. Once installed, configure all non-root accounts that will be using OpenCL 2.0 devices to be a member of the *video* group.

   ```
   $ sudo usermod -a -G video USERNAME
   ```

4. (optional) Verify the fingerprint and key ID, `13881D9C`, of the GPG public key provided with the intel-opencl-r5.0 driver.

   ```
   $ gpg --quiet --with-fingerprint vpg_ocl_linux_rpmdeb.public
     pub  2048R/13881D9C 2016-11-14 "CN = Intel(R) OpenCL GPU driver for Linux", O=Intel
   Corporation
     Key fingerprint = FE04 01BC 1EC7 A470 F042 4C64 DC5C DB54 1388 1D9C
   ```

5. (optional) If the GPG key fingerprint is correct, add the key to the system's RPM database.

   ```
   sudo rpm --import vpg_ocl_linux_rpmdeb.public
   ```

6. (optional) Verify Intel® provided the GPG key ID used to sign the intel-opencl-r5.0 driver.

   ```
   $ rpm -Kv intel-opencl-r5.0-BUILD_ID.x86_64.rpm | grep Signature
   ```

```
   Header V4 RSA/SHA1 Signature, key ID 13881d9c: OK
   V4 RSA/SHA1 Signature, key ID 13881d9c: OK

$ rpm -Kv intel-opencl-devel-r5.0-BUILD_ID.x86_64.rpm | grep Signature
   Header V4 RSA/SHA1 Signature, key ID 13881d9c: OK
   V4 RSA/SHA1 Signature, key ID 13881d9c: OK

$ rpm -Kv intel-opencl-cpu-r5.0-BUILD_ID.x86_64.rpm | grep Signature
   Header V4 RSA/SHA1 Signature, key ID 13881d9c: OK
```

7. Install the intel-opencl-r5.0 driver and the kernel patch set to the system.

```
$ sudo rpm -ivh intel-opencl-r5.0-BUILD_ID.x86_64.rpm
```

CentOS - Patch and Compile Linux Kernel 4.7

The following steps will patch and compile the Linux 4.7 kernel required for OpenCL applications.

1. Install required kernel build tools.

```
$ sudo yum groupinstall 'Development Tools'
$ sudo yum install rpmdevtools openssl openssl-devel bc
```

2. Download the Linux 4.7 kernel into a working directory.

```
$ mkdir ~/intel-opencl
$ cd ~/intel-opencl
$ curl -O https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.7.tar.xz
```

3. Create a build area to compile the Linux 4.7 kernel.

```
$ rpmdev-setuptree
```

4. Expand the Linux 4.7 kernel previously downloaded and enter its directory.

```
$ cd ~/intel-opencl
$ tar -Jxf linux-4.7.tar.xz
$ cd ~/intel-opencl/linux-4.7
```

5. Apply the intel-opencl-r5.0 driver kernel patch set to the 4.7 kernel.

```
$ patch -p1 < /opt/intel/opencl/kernel-4.7.patch
```

6. Use the currently running Linux kernel as a template to configure the 4.7 kernel.

```
$ sudo cat /boot/config-`uname -r` > .config
```

7. Tag the patched kernel to provide easy identification.

```
$ perl -pi -e 's/.*CONFIG_LOCALVERSION=.*/CONFIG_LOCALVERSION=".intel.r5.0"/' .config
```

8. Update the Linux `.config` file for the 4.7 kernel.

```
$ make olddefconfig
```

9. Compile the patched Linux kernel into a distributable package.

```
$ rm -f .version
$ make -j4 binrpm-pkg
```

10. Install the patched kernel and distribute to other similarly configured systems as required.

```
$ sudo rpm -ivh ~/rpmbuild/RPMS/x86_64/kernel-4.7.0.intel.r3.0-2.x86_64.rpm
```

The 4.7 linux kernel has preliminary hardware support for Intel® Atom™ processor E3900 series, Intel Pentium N4200, J4000 Intel Celeron J3000, N3350. To enable the OpenCL functionality for those platforms you need to add the parameter to the kernel command line:

```
i915.preliminary_hw_support=1
```

11. The system must be rebooted prior and the patched 4.7 kernel selected in order to support any OpenCL applications.

Uninstall

The patched 4.7 Linux kernel and intel-opencl-r5.0 driver can be removed, restoring the system to its previous condition.

1. Remove the intel-opencl-r5.0 driver and the patched 4.7 kernel RPMs:

```
$ sudo rpm -e intel-opencl-devel-r5.0
$ sudo rpm -e intel-opencl-r5.0
$ sudo rpm -e intel-opencl-cpu-r5.0
$ sudo rpm -e kernel-4.7.0.intel.r5.0-2.x86_64.rpm
$ sudo /usr/sbin/grub2-mkconfig --output=/etc/grub2.cfg
```

Generic OS with Linux 4.7 Kernel

Additional files have been included with the intel-opencl-r5.0 driver for users to apply the Intel® OpenCL patches to the Linux 4.7 kernel for installation on other distributions. Alternative distributions must support glibc 2.7 or better. Users are encouraged to adapt the steps in this section as needed for their particular distribution and situation.

Generic OS Installation and Preliminary Setup

Complete the following steps as necessary in order to install the intel-opencl-r5.0 driver to the target system. Some adjustments may be necessary for systems other than CentOS 7.2 and 7.3 or Ubuntu 14.04/16.04. For CentOS and other RPM-based systems, please first complete the steps in section CentOS Installation and Preliminary Setup to install the intel-opencl-r5.0 driver RPMs; once complete, continue with the steps in section Generic OS - Patch and Compile Linux Kernel 4.7

1. Install and configure the operating system.

2. Once installed, configure all non-root accounts that will be using OpenCL devices to be a member of the *video* group.

```
$ sudo usermod -a -G video USERNAME
```

3. (optional) Install the openssl package to verify the authenticity of the intel-opencl-r5.0 driver.

Ubuntu 14.04/16.04

```
$ sudo apt-get install openssl libnuma1 libpciaccess0
```

4. (optional) Save the Intel® certificate signing chain provided with the driver release as a .pem file.

```
$ openssl pkcs7 -print_certs -inform der -in \
  intel-opencl-r5.0-BUILD_ID.x86_64.tar.xz.sig > intel_signing_certs.pem
```

5. (optional) Verify the signing certificate serial number.

```
$ openssl x509 -in intel_signing_certs.pem -serial -noout
serial= 560000018AEFEC61E8918ADE7B00000000018A
```

6. (optional) Verify the signature of each `.tar` file against the certificate chain.

```
$ openssl smime -verify -in intel-opencl-r5.0-BUILD_ID.x86_64.tar.xz.sig \
  -inform der -content intel-opencl-r5.0-BUILD_ID.x86_64.tar.xz \
  -noverify intel_signing_certs.pem > /dev/null
Verification successful

$ openssl smime -verify -in intel-opencl-devel-r5.0-BUILD_ID.x86_64.tar.xz.sig \
  -inform der -content intel-opencl-devel-r5.0-BUILD_ID.x86_64.tar.xz \
  -noverify intel_signing_certs.pem > /dev/null
Verification successful

$ openssl smime -verify -in intel-opencl-cpu-r5.0-BUILD_ID.x86_64.tar.xz.sig \
  -inform der -content intel-opencl-r5.0-BUILD_ID.x86_64.tar.xz \
  -noverify intel_signing_certs.pem > /dev/null
Verification successful
```

7. Install the intel-opencl-r5.0 driver and kernel patches to the system.

Ubuntu 14.04/16.04

```
$ sudo apt-get install xz-utils

Common commands for Ubuntu
$ mkdir intel-opencl
$ tar -C intel-opencl -Jxf intel-opencl-r5.0-BUILD_ID.x86_64.tar.xz
$ tar -C intel-opencl -Jxf intel-opencl-devel-r5.0-BUILD_ID.x86_64.tar.xz
$ tar -C intel-opencl -Jxf intel-opencl-cpu-r5.0-BUILD_ID.x86_64.tar.xz
$ sudo cp -R intel-opencl/* /
$ sudo ldconfig
```

Generic OS - Patch and Compile Linux Kernel 4.7

The following steps will patch and compile the modified Linux 4.7 kernel required for OpenCL applications.

1. Install required kernel build tools.

Ubuntu 14.04/16.04 (tar method)

```
$ sudo apt-get update

$ sudo apt-get install build-essential openssl bc curl libssl-dev
```

2. Download the Linux 4.7 kernel.

```
$ cd ~/intel-opencl
$ curl -O https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.7.tar.xz
```

3. Expand the Linux 4.7 kernel previously downloaded and enter its directory.

```
$ tar -Jxf linux-4.7.tar.xz
$ cd ~/intel-opencl/linux-4.7
```

4. Apply the intel-opencl-r5.0 driver kernel patch set to the 4.7 kernel.

```
$ patch -p1 < /opt/intel/opencl/kernel-4.7.patch
```

5. Use the currently running Linux kernel as a template to configure the 4.7 kernel.

```
$ sudo cat /boot/config-`uname -r` > .config
```

6. Tag the patched kernel to provide easy identification.

```
$ perl -pi -e 's/.*CONFIG_LOCALVERSION=.*/CONFIG_LOCALVERSION=".intel.r5.0"/' .config
```

7. Update the Linux `.config` file for the 4.7 kernel.

```
$ make olddefconfig
```

8. Compile the patched Linux kernel either into a distributable package or directly on the system.

```
$ make -j4; make -j4 modules
```

9. Install the patched kernel and distribute to other similarly configured systems as required.

```
$ sudo make modules_install
$ sudo make install
```

   The 4.7 linux kernel has preliminary hardware support for Intel® Atom™ processor E3900 series, Intel Pentium N4200, J4000 Intel Celeron J3000, N3350. To enable the OpenCL functionality for those platforms you need to add the parameter to the kernel command line:

```
i915.preliminary_hw_support=1
```

10. The system must be rebooted prior and the patched 4.7 kernel selected in order to support any OpenCL applications.

Ubuntu 16.04.2 OS with vanilla 4.8 Kernel

Install the intel-opencl-r5.0 driver

```
$ sudo apt-get update
$ sudo apt-get install xz-utils
$ mkdir intel-opencl
$ tar -C intel-opencl -Jxf intel-opencl-r5.0-BUILD_ID.x86_64.tar.xz
$ tar -C intel-opencl -Jxf intel-opencl-devel-r5.0-BUILD_ID.x86_64.tar.xz
$ tar -C intel-opencl -Jxf intel-opencl-cpu-r5.0-BUILD_ID.x86_64.tar.xz
$ sudo cp -R intel-opencl/* /
$ sudo ldconfig
```

Intel® OpenCL Code Samples

Source code for several sample OpenCL programs is available from Intel that may be used to query the capabilities of the available OpenCL platform(s) and execute on OpenCL devices. Please visit https://software.intel.com/en-us/intel-opencl-code-builder-support/code-samples to download the samples.

In order to build and run the samples on CentOS 7.2 and 7.3, a few additional RPMs[2] may be required.

```
$ sudo rpm -ivh intel-opencl-r5.0-BUILD_ID.x86_64.rpm
$ sudo rpm -ivh intel-opencl-devel-r5.0-BUILD_ID.x86_64.rpm
$ sudo rpm -ivh intel-opencl-cpu-r5.0-BUILD_ID.x86_64.rpm
$ sudo yum install gcc-c++ make mesa-libGL-devel
```

The samples can then be built with

[2]On Ubuntu, the mesa-common-dev package is required.

```
$ make
```

Some sample and platform combinations may require a period of time exceeding the default i915 hang check timer. See the release notes, section Known Limitations, for more information on the hang check timer and how it can be disabled.

Feedback and Support

We welcome feedback to continue to make this product better. Please direct your feedback, including future feature requests, through your primary Intel product support channels.

Notice revision #20170629