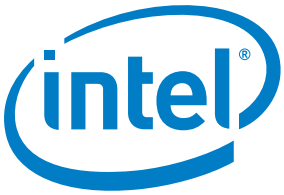

C-For-Media Color Copy Pipeline Installation Guide

March 2019

Document Number: XXXXXX-x.x

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein

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

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate 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: http://www.intel.com/design/literature.htm

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at http://www.intel.com/ or from the OEM or retailer.

[Delete this paragrah if your document does not make any security claims.] No computer system can be absolutely secure.

Intel and the Intel logo 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.

Copyright © 2019, Intel Corporation. All rights reserved.

Contents

1.0 Installation Guide 5

1.1 Download C for Media Package 5
1.2 Yocto/Ubuntu 5
1.2.1 Update CMake 5
1.2.2 Install libva 6
1.2.3 Install GMMLib 6
1.2.4 Install Media Driver 7
1.2.5 Install Intel Graphic Compiler (IGC) 7
1.2.6 Environment setting for Yocto 7
1.2.7 Environment setting for Ubuntu 8
1.3 Installing gflags library 8
1.4 C for Media Samples and Documents 8
1.4.1 C for Media Documents 8
1.5 Color Copy Pipeline Samples 9
1.5.1 Download Copy Pipeline Samples 9
1.5.2 Build Copy Pipeline Samples 9

Revision History

Date	Revision	Description
March 2019	0.5	Initial release.

§

1.0 Installation Guide

The Intel® C for Media development package is a software development package for Intel® Graphics Technology. It includes the Intel® C for Media Compiler, the Intel® C for Media Runtime, Intel® Media Driver for VAAPI, and reference examples, which can be used to develop applications accelerated by Intel® Graphics Media Accelerator. A typical application contains two kinds of source code, kernel and host. The kernel is written in Intel® C for Media language, compiled to GPU ISA binary by the Intel® C for Media Compiler, and executed on the GPU. Host manages workloads through the Intel® C for Media Runtime and user mode media driver.

Official website for C for Media
https://01.org/c-for-media-development-package

Current, the downloadable package officially supports Ubuntu 16.04 and above. For non-supported distros, Intel has released source code for all required components to build the C for Media development package.

1.1 Download C for Media Package

The latest version of the C for Media package is 20190221

https://01.org/sites/default/files/downloads//linuxcformediadevelopmentpackage20190221.zip

1) Download and unzip to a folder and will use C_for_Media for as folder name for subsequent instruction.

1.2 Yocto/Ubuntu

This section provides installation guide to build GMMLib and Intel Media Driver from sources and install Intel Graphic Compiler included in C for Media package.

To ensure consistent the source code use to generate 20190221 package, all the sources checkout from the various repositories are using the hashes match with the package.

Install and update all the packages as follow.

1.2.1 Update CMake

APL Yocto recipe is using older CMake. Some of the C for Media components require newer version of CMake like 3.5.1. As a result, a newer CMake is required. Skip this

step for Ubuntu 16.04 if CMake version is >= 3.5.1

1) git clone https://github.com/Kitware/CMake.git
2) cd CMake
3) ./configure
4) make -j4
5) sudo make install

1.2.2 Install libva

1) Extra libraries are need if the Linux system doesn't have it
    a. libtool -> sudo apt-get install libtool
    b. libpciaccess -> https://01.org/linuxgraphics/downloads/2018q1-intel-graphics-stack-recipe
    c. libdrm -> https://01.org/linuxgraphics/downloads/2018q1-intel-graphics-stack-recipe
2) git clone https://github.com/intel/libva.git
3) cd libva
4) git checkout 25b330768
5) ./autogen.sh
6) make –j4
7) sudo make install

1.2.3 Install GMMLib

1) git clone https://github.com/intel/gmmlib.git
2) cd gmmlib
3) git checkout 8bee050d28
4) apply this patch by

    git am 0001-Temporary-workaround-performance-issue-in-APL.patch

0001-Temporary-wo
rkaround-performar

5) mkdir build
6) cd build
7) cmake –DCMAKE_INSTALL_PREFIX=/usr/local ../
8) make –j4
9) sudo make install

1.2.4 Install Media Driver

1) git clone https://github.com/intel/media-driver.git
2) cd media-driver
3) git checkout 8f9771eb577
4) mkdir build
5) cd build
6) cmake –DCMAKE_INSTALL_PREFIX=/usr/local ../
7) make –j4
8) sudo make install

1.2.5 Install Intel Graphic Compiler (IGC)

The IGC can be directly reuse from the downloaded C for Media package. Go to the unzip C for Media folder and extract the IGC.

1) cd C_for_Media
2) cd drivers/IGC
3) ar x IGC-20190221Linux.deb
4) tar –vxf data.tar.xz
5) cp usr/lib/libig* /usr/local/lib

1.2.6 Environment setting for Yocto

Before running C for media samples, export several environment variables

1) export LIBVA_DRIVER_NAME=iHD
2) export LIBVA_DRIVERS_PATH=/usr/lib /dri

3) export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH

1.2.7 Environment setting for Ubuntu

1) export LIBVA_DRIVER_NAME=iHD

4) export LIBVA_DRIVERS_PATH=/usr/lib/x86_64-linux-gnu/dri

1.3 Install gflags library

1) Ubuntu 16.04
    a. sudo apt-get install libgflags-dev
2) Yocto
    a. git clone https://github.com/gflags/gflags.git
    b. cd gflags
    c. mkdir build; cd build
    d. cmake .. -DBUILD_SHARED_LIBS=ON -DBUILD_STATIC_LIBS=ON -DBUILD_gflags_LIB=ON -DCMAKE_INSTALL_PREFIX=/usr
    e. make -j8
    f. sudo make install

1.4 C for Media Samples and Documents

C for Media samples are stored in the C_for_Media/examples folder. The package provides a run_all.sh script which will run all the samples and can use to verify all the required components are install correctly.

1.4.1 C for Media Documents

1) More details of the samples are documented in this folder.

   C_for_Media/documents/compiler/html/cmtutorial/cmtut.html

2) C for Media Language specification located in

   C_for_Media/documents/compiler/html/cmlangspec/cmlangspec.html

3) C for Media Runtime API located in

   C_for_Media/documents/cmrtlib/html/index.html

1.5 Color Copy Pipeline Samples

Color Copy Pipeline Sample implements the graph from RGB to CMYK similar to OpenVX version in https://software.intel.com/en-us/sample-color-copy-color-copy-pipeline-the-rgb-to-cmyk-openvx-graph.

There are some differences between both versions

1) OpenVX version utilizes heterogeneous system and distributes kernels to CPU, GPGPU and IPU.
2) C for Media version only computes the full graph in GPGPU.
3) OpenVX version supports both halftone and error diffusion, while C for Media version only supports halftone.
4) OpenVX version supports various configurations like High1, High3 and High6 while C for Media version only supports High6.
5) As C for Media kernels don't need to share buffer among various compute units, all intermediate surfaces are 2D surface, while OpenVX use linear buffers for all intermediate surfaces.

1.5.1 Download Copy Pipeline Samples

Color copy pipeline sample doesn't distribute from C for Media package, but resides in C for Media compiler GitHub.

To download

1) git clone https://github.com/intel/cm-compiler.git
2) The path for the sample is cm-compiler/test/external_contribution/CopyPipelineHigh6/

1.5.2 Build Copy Pipeline Samples

1) Copy color copy pipeline into C for Media package

   cp -r cm-compiler/test/external_contribution/CopyPipelineHigh6/ C_for_Media/examples

2) Edit Makefile.linux in CopyPipelineHigh6. Comment the first CM_ROOT and uncomment the second CM_ROOT, and the output as below.

   #CM_ROOT := /home/gangche1/Downloads/linuxcformediadevelopmentpackage20180202

```
#CMC := ../../../build.64.linux/bin/cmc
CM_ROOT := ../..
```

3) In the same Makefile.linux, replace igfxcmrt64.so to libigfxcmrt.so, and output as below.

```
(HW_X64_APP): $(HW_X64_APP_OBJS)
$(CXX) $^ $(HW_LDFLAGS) -rdynamic
$(CM_ROOT)/runtime/lib/x64/libigfxcmrt.so -o $@
```

4) Compile using the Makefile.linux in CopyPipelineHigh6

```
make –f Makefile.linux
```

5) To run, make sure the environment has set correctly either describe in section 1.2.6 or 1.3.1.

```
./hw_x64.CopyPipelineHigh6 nest.bmp
```