amdgpu Graphics Stack Documentation

Release unknown-build

Advanced Micro Devices, Inc.

May 05, 2021

Contents

1 Table of Contents **1**
- 1.1 Preamble . . . 1
- 1.2 Overview . . . 1
- 1.3 Prerequisites . . . 3
- 1.4 Using the `amdgpu-install` Script . . . 4
- 1.5 Installing or Uninstalling AMDGPU . . . 6
- 1.6 Reporting Bugs . . . 10

CHAPTER 1

Table of Contents

1.1 Preamble

© 2018 Advanced Micro Devices, Inc. The information contained herein is for informational purposes only and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and assumes no liability of any kind, including the implied warranties of non-infringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. This notice does not change the terms and limitations applicable to the purchase or use of AMD's products that may be set forth in a separate signed agreement between you and AMD.

AMD, the AMD Arrow logo, AMD Radeon, AMD FirePro and combinations thereof are trademarks of Advanced Micro Devices, Inc.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Vulkan and the Vulkan logo are registered trademarks of the Khronos Group Inc.

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

OpenGL® and the oval logo are trademarks or registered trademarks of Hewlett Packard Enterprise in the United States and/or other countries worldwide.

1.2 Overview

Table of Contents

- *Overview*

– *Stack Variants*

This guide provides installation and uninstallation procedures for the AMDGPU graphics and compute stack.

Note: The rest of this document will refer to Radeon™ Software for Linux® as AMDGPU.

1.2.1 Stack Variants

There are two major stack variants available for installation:

- **Pro**: recommended for use with **Radeon Pro** graphics products.
- **All-Open**: recommended for use with consumer products.

Installing Option	Components
All-Open Stack	• Base kernel drivers • Base accelerated graphics drivers • Mesa OpenGL • Mesa multimedia
Pro Stack	• Base kernel drivers • Base accelerated graphics drivers • Mesa multimedia • Pro OpenGL • **Pro OpenCL** – ROCr OpenCL stack (supports Vega 10 and later products) – Legacy OpenCL stack (supports legacy products older than Vega 10) • Pro Vulkan

You can install a combination of stack components using the `amdgpu-install` script, as listed in the following:

- Base kernel and accelerated graphics drivers + Pro OpenGL + Pro Vulkan
- Base kernel and accelerated graphics drivers + Pro OpenGL + Pro Vulkan + Pro OpenCL
- Base kernel and accelerated graphics drivers + Pro OpenCL
- Base kernel drivers (**no** accelerated graphics drivers) + Pro OpenCL (headless mode)

Note:

- When installing the All-Open stack using `amdgpu-install` script, every component from the All-Open stack will be installed. There is no supported way to install arbitrary combinations of these components when using `amdgpu-install` but you can use the system package manager to do so.
- 32-bit libraries are automatically installed with Pro OpenGL component when installation is performed using `amdgpu-install` script.

1.3 Prerequisites

Table of Contents

- *Prerequisites*
  - *Downloading the Package Archive*
  - *Extracting the Package Archive*
  - *Configuring Access to the Distribution Repository (RHEL and SLE only)*

1.3.1 Downloading the Package Archive

The AMDGPU graphics stack is delivered as an archive of native packages. Prior to installation, you must first obtain the archive from your AMD Customer Engagement Representative or from the AMD web site.

Download the AMDGPU-Pro tar archive (for example, to `~/Downloads`), which contains the installation script.

1.3.2 Extracting the Package Archive

Extract the tar archive to a known location. In the following commands, replace `YY.XX` with the actual release number (e.g. 18.30) and `NNNNNN` six digit build number of the downloaded file:

```
$ cd ~/Downloads
$ tar -Jxvf amdgpu-pro-YY.XX-NNNNNN.tar.xz
$ cd ~/Downloads/amdgpu-pro-YY.XX-NNNNNN
```

1.3.3 Configuring Access to the Distribution Repository (RHEL and SLE only)

AMDGPU stack depends on packages provided by the Linux distribution vendors.

The AMDGPU-Pro driver requires access to specific RPMs from Red Hat Enterprise Linux (RHEL) or SUSE Linux Enterprise (SLE) installation media for the purpose of dependency resolution. You must ensure **one** of the following:

- Have a valid subscription and be connected to Internet during installation.
- Mount an installation media (for example, DVD, USB key or ISO file). Media mounting instructions for **Red Hat** systems are provided at https://access.redhat.com/solutions/1355683. For **SLE**, use YaST to add the installation media as a new repository, see the SUSE documentation for details.

Note: In previous releases (before 18.30), a script named `amdgpu-pro-preinstall.sh` was provided to carry out this step and to perform other pre-installation checks and configuration. Starting with 18.30, all external dependencies (excluding dependencies on packages provided by the Linux distribution vendor) have been eliminated and the checks have been integrated into `amdgpu-install`. Consequently, `amdgpu-pro-preinstall.sh` has been eliminated to streamline installation. However, access to distribution media or online package repositories is still required to satisfy base operating system dependencies. It is assumed most customers have a subscription with access to

online repositories or are using a Linux distribution which does not require a subscription and has online repositories enabled by default.

1.4 Using the `amdgpu-install` Script

Table of Contents

- *Using the `amdgpu-install` Script*
  - *Script Types*
  - *Purpose*
  - *Invoking the `amdgpu-install` Script*
  - *Scenarios*
    - *Specifying an OpenCL Implementation*
    - *Viewing Help or Usage Output*
    - *Installing Non-Interactively*
    - *Adding Advanced Package Management Functionality*

1.4.1 Script Types

The AMDGPU graphics stack offers the following scripts to help you install a coherent set of stack components:

- `amdgpu-install`
- `amdgpu-pro-install`

Each of these scripts support the same set of arguments that are explained in the following sections of this page.

Note:

- Although examples on this page show `amdgpu-install` script being invoked, you can use same arguments with `amdgpu-pro-install` script.
- Running the script `amdgpu-pro-install` is synonymous to running the script as `amdgpu-install --pro`.

1.4.2 Purpose

The `amdgpu-install` script does the following:

- Simplifies the installation of the AMDGPU graphics and compute stack by encapsulating the distribution specific package installation logic and by using command line options that allow to specify the:
  - Variant of the AMDGPU stack to be installed (All-Open or Pro)

  - Combination of components (in case of the Pro stack)
- Performs post-install checks to verify whether the installation was performed successfully.
- Installs the uninstallation script to allow you to remove the whole AMDGPU stack from the system by using a single command.

The script is packaged into the AMDGPU-Pro tar archive.

1.4.3 Invoking the `amdgpu-install` Script

Once you have downloaded and extracted the AMDGPU-Pro tar archive, access the directory where you extracted it. Invoke the script as `./amdgpu-install`.

Note: Use of the `-y` option is recommended to avoid prompts from the package manager.

1.4.4 Scenarios

Specifying an OpenCL Implementation

There are two OpenCL implementations available:

- **ROCr**: Provides support for *Vega 10* and newer hardware.
- **Legacy**: Provides support for hardware older than *Vega 10*.

You can either install one or both implementations.

Note: In previous releases, the "PAL" implementation was used for Vega 10 and newer, but this has been removed in favour for the ROCr backend, which is derived from the ROCm open source stack

---

Note: To use the ROCr implementation of OpenCL, the running user might need additional permissions. Usually the user must be added to the "render" group or to the "video" group. See the notes in *OpenCL (Optional Component)* for more details.

OpenCL is installed using the `--opencl` option, as follows:

Option	Description
`--opencl=rocr`	Installs ROCr OpenCL.
`--opencl=legacy`	Installs legacy OpenCL.
`--opencl=rocr,legacy`	Installs both ROCr and legacy OpenCL.

Applicable examples on this page demonstrate installation of both OpenCL implementations.

Viewing Help or Usage Output

Executing the script with `-h` or `--help` argument provides the help or usage output.

```
$ ./amdgpu-install -h
```

Installing Non-Interactively

Using the argument `-y` informs the package management system to assume the answer *yes* for any interactive question. This option allows the install script to embed itself inside other automated scripts.

Example:

```
$ ./amdgpu-install -y
```

Adding Advanced Package Management Functionality

More advanced functionality can be obtained by providing the installation script with arguments to pass through to the package management software of the distribution you are using.

- On Ubuntu, refer to the apt man page for more information.
- On Red Hat, refer to yum man page for more information.

1.5 Installing or Uninstalling AMDGPU

Table of Contents

- *Installing or Uninstalling AMDGPU*
  - *Installing the All-Open Variant*
  - *Installing the Pro Variant*
    - *OpenGL (Default Component)*
    - *OpenCL (Optional Component)*
    - *Vulkan (Default Component)*
    - *PX Platform Support (Ubuntu only)*
  - *Secure Boot Support*
    - *Ubuntu and Debian Based Systems*
    - *RHEL/CentOS Systems*
    - *SLE Systems*
    - *Using the MOK Manager to Enroll the MOK Certificate*
  - *Uninstalling the AMDGPU Graphics Stack*

1.5.1 Installing the All-Open Variant

Run the following command to install the All-Open variant:

```
$ ./amdgpu-install -y
```

Note: The `-y` option installs non-interactively. See *Using the amdgpu-install script* for more information.

Reboot the target system after running `amdgpu-install`.

1.5.2 Installing the Pro Variant

Refer to the following table to understand how to install a combination of Pro components:

Command	Installed Components
`$ ./amdgpu-pro-install -y`	• Base kernel • Accelerated graphics • Mesa multimedia • Pro OpenGL • Pro Vulkan
`$ ./amdgpu-pro-install -y --opencl=rocr,legacy`	• Base kernel • Accelerated graphics • Mesa multimedia • Pro OpenGL • Pro Vulkan • Pro OpenCL
`$ ./amdgpu-pro-install -y --opencl=rocr,legacy --headless`	• Only base kernel • Pro OpenCL (headless mode)

Reboot the target system after running `amdgpu-pro-install`.

For more information, refer to the following sections to understand how to install individual components.

OpenGL (Default Component)

OpenGL is a default component of the Pro variant.

Note: Unless a headless installation is requested, OpenGL is installed.

OpenCL (Optional Component)

OpenCL is an optional component of the Pro variant and is installed only if it is specifically requested.

Two different implementations of OpenCL (legacy and ROCr) are provided. Either of these implementations or both of them can be installed on the target system.

To use the ROCr implementation of OpenCL, the running user might need additional permissions depending on OS policy. If `clinfo` or any openCL application does not work, check ownership and permissions of the render nodes:

```
ls -l /dev/dri/render*
```

If the render nodes are owned by group render but not readable and writable by all users, consider adding the current user to the render group:

```
sudo usermod -a -G render $LOGNAME
```

If you are an administrator, `$LOGNAME` can be replaced by any valid username.

Alternatively, if the render nodes are owned by the video group but not readable and writable by all users, consider adding the current user to the video group:

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

Different Linux distributions have different ownership and permission policies for render nodes. In addition, your organization may have its own policies that override the distribution defaults. If unsure, consult the documentation for your distribution or your organization.

In some scenarios, it may be desirable to install **only** the OpenCL portion of the Pro variant (omitting the OpenGL portion), which can be accomplished by adding the `--headless` option. The typical use case is headless compute.

Vulkan (Default Component)

Vulkan is a default component of the Pro variant.

Note: Unless a headless installation is requested, Vulkan is installed.

PX Platform Support (Ubuntu only)

Note: PX is deprecated in the latest version of the AMDGPU Graphics Stack. The All-Open variant with PRIME GPU offloading is recommend for all mobile variants with hybrid graphics, which allows GPU workloads to be offloaded to a discrete GPU on demand.

For PX (*PowerExpress*) platform support, use the `--px` option as shown in the following command:

```
$ ./amdgpu--pro-install --px
```

1.5.3 Secure Boot Support

The kernel modules used by the AMDGPU Graphics Stack rely on DKMS. DKMS will compile the modules during installation and re-compile the modules when kernel is updated. To use secure boot with DKMS, a machine owned key (MOK), which is unique to the system, will be generated to sign the kernel modules.

Please see the section below that applies to the distro of the target system.

Note: If you already have a MOK enrolled on your system, the driver package expects the private key for this MOK to be installed in a specific location (see table below). Prior to installing the driver, either copy the private key to this location or create a symbolic link.

DISTRO	MOK PRIVATE KEY
Ubuntu/Debian	/var/lib/shim-signed/mok/MOK.priv
RHEL/CentOS	/root/mok.priv

Note: The MOK directory contains the private key used for signing kernel modules. It is highly suggested to use disk encryption to avoid leaking this key to a malicious attacker. Please be aware of the risk.

Ubuntu and Debian Based Systems

Prior to installing the driver, enable secure boot on the target system. If secure boot is enabled during driver installation, a MOK will be automatically generated during install, if it doesn't already exist in the MOK directory, and the user will be prompted to enter a temporary password to import the MOK certificate into the system. This prompt uses a text mode tool, which can be navigated using TAB to highlight, and ENTER to select.

Reboot the system and the MOK manager will start automatically.

See *Using the MOK Manager to Enroll the MOK Certificate* for further instructions.

RHEL/CentOS Systems

To use secure boot with this driver stack, you must manually import the MOK certificate onto your system. The driver installation will automatically generate a MOK if it doesn't already exist in the MOK directory.

To import the MOK certificate, first install the "mokutil" package using your package manager:

```
sudo yum install mokutil
```

Then run mokutil to import the certificate:

```
sudo mokutil --import /root/mok.der
```

Mokutil will require you to provide a temporary password for importing the certificate. Reboot the system and the MOK manager will start automatically.

See *Using the MOK Manager to Enroll the MOK Certificate* for further instructions.

SLE Systems

Signing support for dkms requires dkms version 2.8, which is unavailable in SLE. If DKMS is updated manually by the user, signing can be used in the same way as RHEL, but this is untested and might not work as intended. Please use at your own risk.

Using the MOK Manager to Enroll the MOK Certificate

After requesting to import the MOK certificate and entering a temporary password, the MOK manager will start on next boot.

Hit any key to avoid continuing boot. Do no select "Continue Boot", but rather select "Enroll MOK" and "View key 0" to confirm that the key is correct.

Select "Continue", "Yes", and enter the temporary password to finish enrolling the key certificate. Select "Reboot" to exit the MOK manager.

The signed driver should now load successfully.

1.5.4 Uninstalling the AMDGPU Graphics Stack

To remove all components of the stack, run the uninstall script from anywhere in the system.

1. Based on whether you installed the All-Open or Pro components, run one of the following commands:

```
# Use this for All-Open components
$ amdgpu-uninstall

# Use this for Pro components
$ amdgpu-pro-uninstall
```

2. After uninstalling the components, reboot the target system.

1.6 Reporting Bugs

Bugs should be reported through your AMD Customer Engagement Representative.

If you downloaded a release archive directly from the AMD web site and do not have an AMD Customer Engagement Representative, you may report bugs using the freedesktop.org Bugzilla. File bugs according to the following table:

Variant	Bugzilla Product	Bugzilla Component
All-Open	DRI	DRM/AMDgpu
Pro	DRI	DRM/AMDgpu-pro

1.6.1 Generating a Bug Report

In order to facilitate efficient investigation, it's important to include as much detail as possible with the bug report. Perform the following steps before you file a bug report:

1. Create a destination folder for the report.

Tip: If you gather several reports, identify them with the date and if needed add a number at the end to differentiate between your reports, as shown in the following example.

```
$ LOGDIR=~/LOGS/20170913-1
$ mkdir -p $LOGDIR
$ cd $LOGDIR
```

2. Gather basic configuration details.

```
$ sudo lshw -c cpu | grep product >> basic.conf
$ lspci -nn | grep "VGA\|Display" >> basic.conf
$ sudo dmidecode -t BIOS | grep Version >> basic.conf
$ lsb_release -sd >> basic.conf
$ uname -a >> basic.conf
$ dkms status | grep "amdgpu\|radeon" >> basic.conf
```

3. Create detailed log files by running the following commands:

```
$ dkms status > dkms.status.log
$ lsmod | grep amdgpu > lsmod.amdgpu.log
$ dmesg > dmesg.log
$ cp /var/log/Xorg.0.log .
$ type -p dpkg && dpkg -l > package.log
$ type -p rpm && rpm -qa > package.log
```

4. (Optional) To facilitate in-depth analysis, you can run the following commands:

```
$ lspci -vnn > lspci.vnn.log
$ lspci -nn > lspci.nn.log
$ sudo dmidecode > dmidecode.log
$ uname -a > uname.a.log
$ lsinitramfs /boot/initrd.img-`uname -r` > lsinitramfs.log
$ sudo lshw > lshw.log
$ modinfo amdgpu > modinfo.amdgpu.log
$ glxinfo > glxinfo.log
```

Capturing the Output of Your Commands to a File

1. Start a script session to capture your commands to a log file.

```
$ script /tmp/install-log.txt
```

2. **Run all commands you want to capture in the script (for example, install** All-Open base driver).

```
$ ./amdgpu-install -y
```

3. End the capture session.

```
$ exit
```