AMD ROCm™ Installation Guide for Linux v5.0

Publication # **1.0** Revision: **0209**
Issue Date: **February 2022**

Table of Contents

Chapter 1 Overview of ROCm Installation Methods

1.1 What is ROCm

ROCm is a brand name for ROCm™ open software platform (for software) or the ROCm™ open platform ecosystem (includes hardware like FPGAs or other CPU architectures).

NOTE: ROCm no longer functions as an acronym.

In addition to the installation method using the native Package Manager, AMD ROCm v5.0 introduces added methods to install and upgrade ROCm. Users can now upgrade the existing ROCm installation to a specific or latest ROCm release using an upgrade mechanism. In this release, the ROCm installation uses *amdgpu-install* and *amdgpu-uninstall* scripts.

The *amdgpu-install* script streamlines the installation process by:

- Abstracting the distribution-specific package installation logic.
- Performing the repository setup.
- Allowing a user to specify the use case and automating the installation of all the required packages.
- Installing multiple ROCm releases simultaneously on a system.
- Enhanced functionality of amdgpu-install script automates updating local repository information before proceeding to ROCm installation.
- Performing post-install checks to verify whether the installation was completed successfully.
- Uninstalling the installed single version or multi-version ROCm releases.

The *amdgpu-uninstall* script allows the removal of the entire ROCm stack by using a single command.

Some of the ROCm-specific use cases that the installer currently supports are:

- OpenCL (ROCr/KFD based) runtime
- HIP runtimes
- ROCm libraries and applications
- ROCm Compiler and device libraries
- ROCr runtime and thunk
- Kernel-mode driver

For more information, refer to the *Installation Methods* section in this guide.

1.2 Notice for Open-source and Closed-source ROCm Repositories in Future Releases

To make a distinction between open-source and closed-source components, all ROCm repositories will consist of sub-folders in future releases.

- All open-source components will be placed in the *base-url/<rocm-ver>/main* sub-folder
- All closed-source components will reside in the *base-url/<rocm-ver>/ proprietary* sub-folder

1.3 About This Document

This document is intended for users familiar with the Linux environments and discusses the installation/uninstallation of ROCm programming models on the various flavors of Linux.

This document also refers to Radeon™ Software for Linux® as AMDGPU stack, including the kernel-mode driver *amdgpu-dkms*.

The guide provides the instructions for the following:

- Kernel mode Driver Installation
- ROCm single and multi-version Installation
- ROCm release and Kernel Mode Driver Upgrade
- ROCm single and multi-version Uninstallation
- Kernel Mode Driver Uninstallation

1.4 System Requirements

The system requirements for the ROCm v5.0 installation are as follows:

Linux Distribution Support

OS-Version (64-bit)	Kernel Versions
CentOS 8.3	4.18.0-193.el8
CentOS 7.9	3.10.0-1127
RHEL 8.5	4.18.0-348.7.1.el8_5.x86_64
RHEL 8.4	4.18.0-305.el8.x86_64
RHEL 7.9	3.10.0-1160.6.1.el7
SLES 15 SP3	5.3.18-59.16-default
Ubuntu 20.04.3	5.8.0 LTS / 5.11 HWE
Ubuntu 18.04.5 [5.4 HWE kernel]	5.4.0-71-generic

NOTE: Installing ROCm on Linux will require superuser privileges. For systems that have enabled *sudo* packages, ensure you use the *sudo* prefix for all required commands.

1.5 Package Licensing

IMPORTANT: AOCC CPU OPTIMIZATIONS BINARY IS SUBJECT TO THE LICENSE AGREEMENT ENCLOSED IN THE DIRECTORY FOR THE BINARY AND IS AVAILABLE HERE: /opt/rocm-5.0.0/share/doc/rocm-llvm-alt/EULA. BY USING, INSTALLING, COPYING, OR DISTRIBUTING THE AOCC CPU OPTIMIZATIONS, YOU AGREE TO THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT INSTALL, COPY, OR USE THE AOCC CPU OPTIMIZATIONS.

For rest of the ROCm packages, user can find the licensing information at the following location:

```
/opt/rocm-<version>/share/<component-name>/
```

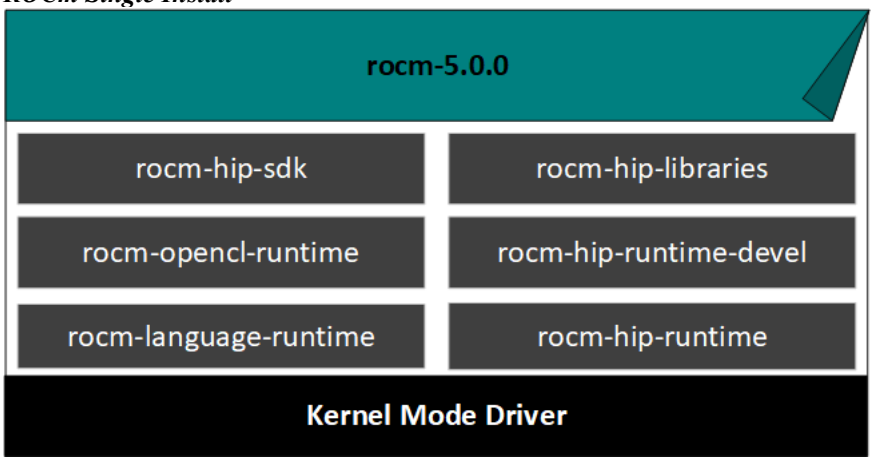
For example, users can fetch the licensing information of *amd_comgr* component from the path */opt/rocm-5.0.0/share/amd_comgr*. A file named *LICENSE.txt,* contains the license details.

1.6 Types of Installation

1.6.1 Single Install

The single version ROCm installation refers to the installation of a single instance of the ROCm release on a system. Single install uses non-versioned ROCm meta-packages, as shown below.

ROCm Single Install

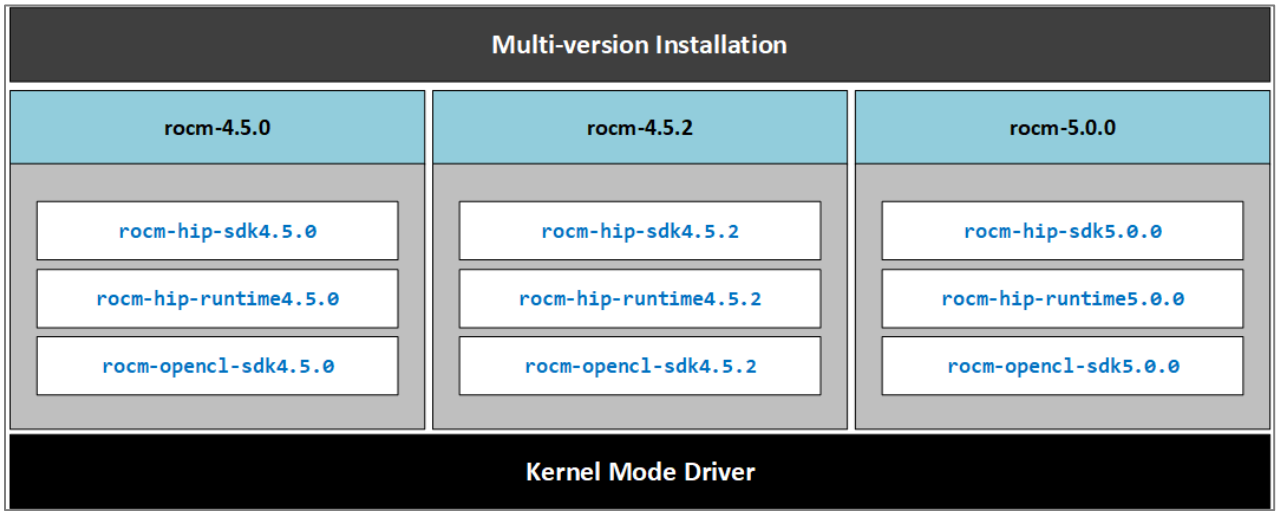


1.6.2 Multi-version Install

The multi-version installation refers to installing multiple versions of the ROCm stack simultaneously on a system. The ability to install multiple versions of packages simultaneously is added by extending the package name and its dependencies with the release version.

NOTE: This feature is not available for the AMDGPU stack.

ROCm Multiversion Installation



NOTE: Refer to *ROCm Package Naming Conventions* for more details.

Chapter 2 Prerequisite Actions

You must perform the following steps before installing ROCm programming models and check if the system meets all the requirements to proceed with the installation.

- Confirm the system has a supported Linux distribution version
- Confirm the system has a ROCm-capable GPU
- Confirm the system has required tools and packages installed

2.1 Confirm You Have a Supported Linux Distribution Version

The ROCm installation is supported only on specific Linux distributions and their kernel versions.

NOTE: The ROCm installation is not supported on 32-bit operating systems.

2.1.1 How to Check Linux Distribution and Kernel Versions on Your System

2.1.1.1 Linux Distribution Information

Ensure you obtain the distribution information of the system by using the following command on your system from the Command Line Interface (CLI),

```
$ uname -m && cat /etc/*release
```

For example, running the command above on an Ubuntu system results in the following output:

```
x86_64

DISTRIB_ID=Ubuntu

DISTRIB_RELEASE=18.04

DISTRIB_CODENAME=bionic

DISTRIB_DESCRIPTION="Ubuntu 18.04.5 LTS"
```

2.1.1.2 Kernel Information

Type the following command to check the kernel version of your Linux system.

```
$ uname -srmv
```

The output of the command above lists the kernel version in the following format:

```
Linux 5.4.0-77-generic #86~18.04.5-Ubuntu SMP Fri Jun 18 01:23:22 UTC 2021
x86_64
```

2.1.1.3 OS and Kernel Version Match

Confirm that the obtained Linux distribution and kernel versions match with *System Requirements.*

2.2 Confirm You Have a ROCm-Capable GPU

The ROCm platform is designed to support the following list of GPUs:

GPU support for ROCm programming models

Classification	GPU Name	Product Id
GFX9 GPUs	AMD Radeon Instinct™ MI50	Vega 20
	AMD Radeon Instinct™ MI60	
	AMD Radeon™ Pro VII	
RDNA GPUs	AMD Radeon™ Pro W6800	73a3 OR Navi 21 GL-XL
	AMD Radeon™ Pro V620	Device 73a1
CDNA GPUs	AMD Instinct™ MI100	Arcturus
	AMD Instinct™ MI200	Aldebaran

2.2.1 How to Verify Your System Has A ROCm-Capable GPU

To verify that your system has a ROCm-capable GPU, enter the following command from the Command Line Interface (CLI):

```
lspci | grep -i display
```

The command displays the details of detected GPUs on the system in the following format in case of AMD Instinct™ MI200:

```
c1:00.0 Display controller: Advanced Micro Devices, Inc. [AMD/ATI] Aldebaran
c5:00.0 Display controller: Advanced Micro Devices, Inc. [AMD/ATI] Aldebaran
```

NOTE: Verify from the output that the listed product names match with the *Product Id* given in in the table above.

2.3 Confirm the System Has Required Tools and Packages Installed

You must install and configure *Devtoolset-7* to use RHEL/CentOS 7.9

2.3.1 How to Install and Configure Devtoolset-7

Refer to the *RHEL/CentOS Installation* section for more information on the steps necessary for installing and setting up *Devtoolset-7*.

2.3.2 Enable the Additional Repositories

Enable the required repositories using the following commands:

```
#For CentOS 7&8:

sudo yum install -y epel-release

#For CentOS 8.3 Only:

$ sudo yum config-manager --set-enabled powertools

#For RHEL 8.3/8.4 only:

$ sudo subscription-manager repos --enable codeready-builder-for-rhel-8-x86_64-
rpms
```

2.3.3 Required Packages

Verify if the *wget package* for downloading files from server, is installed on your system using command below:

UBUNTU/DEBIAN

```
$ sudo apt list --installed | grep 'wget\|gnupg2'
```

RHEL/CentOS

```
$ sudo yum list installed | grep wget
```

SLES/OPENSUSE

```
$ sudo zypper search --installed-only | grep wget
```

If it's not installed, execute the following command to install it:

UBUNTU/DEBIAN

```
$ sudo apt-get update

$ sudo apt-get install wget gnupg2
```

RHEL/CentOS

```
$ sudo yum clean all

$ sudo yum install wget
```

SLES/OPENSUSE

```
$ zypper install wget
```

2.3.4 Setting Permissions for Groups

This section provides steps to add any current user to a video group to access GPU resources.

Issue the following command to check the groups in your system:

```
$ groups
```

Add yourself to the video group using the following instruction:

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

For all ROCm supported operating systems, continue to use video group. By default, you can add any future users to the video and render groups.

NOTE: render group is required only for Ubuntu v20.04.

To add future users to the video and render groups, run the following command:

```
$ echo 'ADD_EXTRA_GROUPS=1' | sudo tee -a /etc/adduser.conf

$ echo 'EXTRA_GROUPS=video' | sudo tee -a /etc/adduser.conf

$ echo 'EXTRA_GROUPS=render' | sudo tee -a /etc/adduser.conf
```

Chapter 3 Meta-packages in ROCm Programming Models

This section provides information about the required meta-packages for the following AMD ROCm™ programming models:

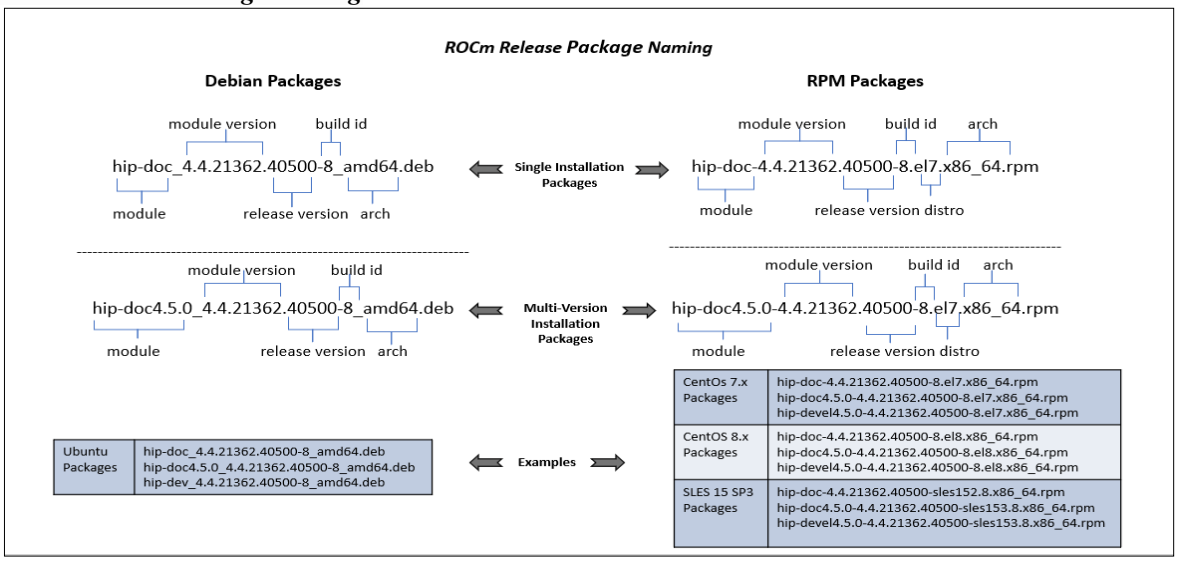
- Heterogeneous-Computing Interface for Portability (HIP)
- OpenCL™

3.1 ROCm Package Naming Conventions

A meta-package is a grouping of related packages and dependencies used to support a specific use-case, for example, running HIP applications. All meta-packages exist in both versioned and non-versioned forms.

- Non-versioned packages – For a single installation of the latest version of ROCm
- Versioned packages – For multiple installations of ROCm

ROCm Release Package Naming



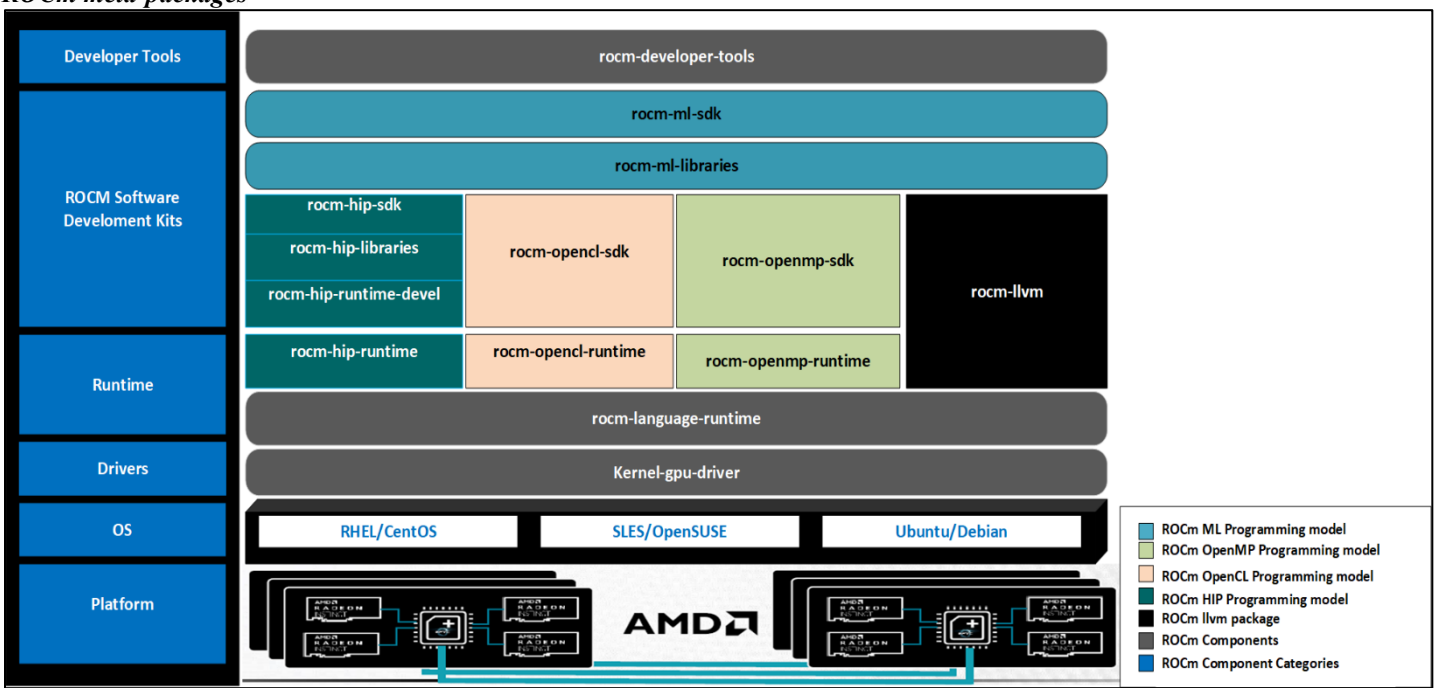
The image above demonstrates the single and multi-version ROCm packages' naming structure, including examples for various Linux distributions.

3.2 Components of ROCm Programming Models

The following image demonstrates the high-level layered architecture of ROCm programming models and their meta-packages. All meta-packages are a combination of required packages and libraries. For example,

- `rocm-hip-runtime` is used to deploy on supported machines to execute HIP applications.
- `rocm-hip-sdk` contains runtime components to deploy and execute HIP applications and tools to develop the applications.

ROCm meta-packages



NOTE: *rocm-llvm* is a single package that installs the required ROCm compiler files.

Meta-packages and Their Descriptions

Meta-packages	Description
rocm-language-runtime	*rocm-language-runtime* meta-package is intended to install ROCr, also known as ROCm runtime.
rocm-hip-runtime	*rocm-hip-runtime* is intended to install packages necessary to run an application written in HIP for the AMD platform.
rocm-opencl-runtime	*rocm-opencl-runtime* installs packages required to run OpenCL based applications on the AMD platform.
rocm-hip-runtime-devel	*rocm-hip-runtime-devel* meta-package contains packages to develop an application on HIP or port it from CUDA.
rocm-opencl-sdk	*rocm-opencl-sdk* installs packages required to develop applications in OpenCL for the AMD platform.
rocm-hip-libraries	*rocm-hip-libraries* installs HIP libraries optimized for AMD platforms.
rocm-hip-sdk	rocm-hip-sdk installs packages necessary to develop/port applications using HIP and libraries for AMD platforms**.**
rocm-developer-tools	*rocm-hip-developer-tools* installs packages required to debug and profile HIP-based

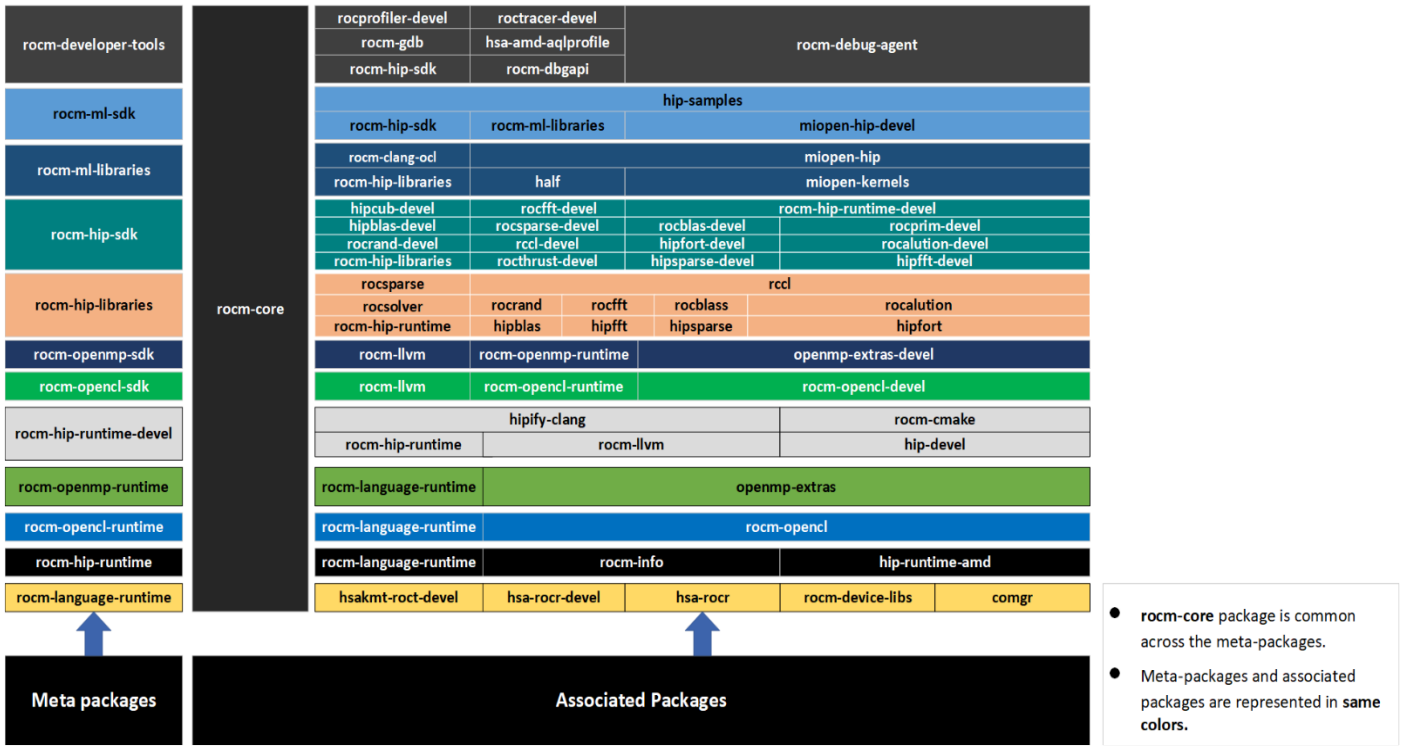
	applications.
rocm-ml-sdk	*rocm-ml-sdk* installs packages necessary to develop and run Machine Learning applications with Machine Learning primitives optimized for AMD platforms.
rocm-ml-libraries	*rocm-ml-libraries* install packages for key Machine Learning libraries, specifically the MIOpen and MIGraphX.
rocm-openmp-sdk	*rocm-openmp-sdk* installs packages necessary to develop OpenMP-based applications for AMD platforms.
rocm-openmp-runtime	*rocm-openmp-runtime* installs packages necessary to run OpenMP based applications for AMD platforms.

3.3 Packages in ROCm Programming Models

This section discusses the available meta-packages and their packages. In a ROCm programming model, packages refer to a collection of scripts, libraries, text files, a manifest, license, and other associated files that enable you to install a meta-package.

The following image visualizes the meta-packages and their associated packages in a ROCm programming model.

Associated Packages



NOTE: The image above is for informational purposes only as the individual packages in a meta-package are subject to change. Users should install meta-packages, and not individual packages, to avoid conflicts.

Chapter 4 How To Install ROCm

You may use the following installation methods to install ROCm:

- Installer Script Method
- Package Manager Method

4.1 Installer Script Method

The Installer script method automates the installation process for the AMDGPU and ROCm stack. The Installer script handles the complete installation process for ROCm, including setting up the repository, cleaning the system, updating, and installing the desired drivers and meta-packages. With this approach, the system has more control over the ROCm installation process. Thus, users who are less familiar with the Linux standard commands can choose this method for ROCm installation.

For a fresh AMDGPU and ROCm installation using the Installer script method on Linux distribution, you must:

- Meet Prerequisites – Ensure the *Prerequisite Actions* are met before downloading and installing the installer using the Installer Script method.
- Download and Install the Installer – Ensure you download and install the installer script from the recommended URL. Note, the installer package is updated periodically to resolve known issues and add new features. The links for each Linux distribution always point to the latest available build.
- Use the Installer Script on Linux Distributions – Ensure you execute the script for installing use cases.

4.1.1 Download and Install the Installer

4.1.1.1 Downloading and Installing the Installer Script on Ubuntu

Ubuntu 18.04

Download and install the installer using the following command:

```
$ sudo apt-get update

$ wget https://repo.radeon.com/amdgpu-install/21.50/ubuntu/bionic/amdgpu-install_21.50.50000-1_all.deb

$ sudo apt-get install ./amdgpu-install_21.50.50000-1_all.deb
```

Ubuntu 20.04

Download and install the installer.

```
$ sudo apt-get update

$ wget https://repo.radeon.com/amdgpu-install/21.50/ubuntu/focal/amdgpu-install_21.50.50000-1_all.deb

$ sudo apt-get install ./amdgpu-install_21.50.50000-1_all.deb
```

4.1.1.2 Downloading and Installing the Installer Script on RHEL/CentOS

RHEL/CentOS 7.9

Use the following command to download and install the installer on RHEL/CentOS 7.9.

```
$ sudo yum install https://repo.radeon.com/amdgpu-install/21.50/rhel/7.9/amdgpu-install-21.50.50000-1.el7.noarch.rpm
```

RHEL 8.4/CentOS 8.3

Use the following command to download and install the installer on RHEL 8.4 / CentOS 8.3.

```
sudo yum install https://repo.radeon.com/amdgpu-install/21.50/rhel/8.4/amdgpu-install-21.50.50000-1.el8.noarch.rpm
```

RHEL 8.5

Type the following command to download and install the installer for RHEL 8.5 distribution.

```
sudo yum install https://repo.radeon.com/amdgpu-install/21.50/rhel/8.5/amdgpu-install-21.50.50000-1.el8.noarch.rpm
```

4.1.1.3 Downloading and Installing the Installer Script on SLES 15

SLES 15 Service Pack 3

Use the following command to download and install the installer on SLES

```
$ sudo zypper --no-gpg-checks install https://repo.radeon.com/amdgpu-install/.21.50/sle/15/amdgpu-install-21.50.50000-1.noarch.rpm
```

4.1.2 Using the Installer Script for Single-Version ROCm Installation

To install use cases specific to your requirements, use the installer *amdgpu-install* as follows:

```
# To install a single use case

$ sudo amdgpu-install --usecase=rocm

# To install kernel mode driver

$ sudo amdgpu-install --usecase=dkms

# To install multiple use-cases

$ sudo amdgpu-install --usecase=hiplibsdk,rocm


 # To clean the repo amdgpu-build.list and rocm-build.list

 sudo amdgpu-repo --clean

# To display a list of available use cases. Note, the list in this section
represents only a sample of available use cases for ROCm.

$ sudo amdgpu-install --list-usecase

If --usecase option is not present, the default selection is
"graphics,opencl,hip"


Available use cases:

rocm(for users and developers requiring full ROCm stack)

- OpenCL (ROCr/KFD based) runtime
- HIP runtimes
- Machine learning framework
- All ROCm libraries and applications
- ROCm Compiler and device libraries
- ROCr runtime and thunk

lrt(for users of applications requiring ROCm runtime)
- ROCm Compiler and device libraries
- ROCr runtime and thunk

opencl(for users of applications requiring OpenCL on Vega or

later products)
- ROCr based OpenCL
- ROCm Language runtime


openclsdk (for application developers requiring ROCr based OpenCL)
- ROCr based OpenCL
- ROCm Language runtime
```

```
- development and SDK files for ROCr based OpenCL


hip(for users of HIP runtime on AMD products)
- HIP runtimes

hiplibsdk (for application developers requiring HIP on AMD products)
- HIP runtimes
- ROCm math libraries
- HIP development libraries
```

NOTE: Adding *-y* as a parameter to *amdgpu-install* will skip user prompts (for automation). For example,

```
amdgpu-install -y --usecase=rocm
```

Using Installer Script in Docker

When the installation is initiated in Docker, the installer tries to install the use case along with the kernel-mode driver. However, the kernel-mode driver cannot be installed in a Docker container. To skip the installation of the kernel-mode driver, proceed with the *--no-dkms* option, as shown below.

```
$ sudo amdgpu-install --usecase=rocm --no-dkms
```

4.1.3 Using Installer Script for Multi-Version ROCm Installation

The multi-version ROCm installation requires users to download and install the *latest ROCm release installer* from the list of ROCm releases they want to install simultaneously on their system.

For example, if users want to install ROCm releases ***4.5.0, 4.5.1, and 5.0.0*** simultaneously, users are required to download the installer from the latest v ***5.0.0*** ROCm release.

To download and install the installer, refer to *Download And Install the Installer* section.

NOTE: If the existing ROCm release contains non-versioned ROCm packages, users must uninstall those packages before proceeding with the multi-version installation to avoid conflicts.

4.1.3.1 Add Required ROCm Repositories

NOTE: Add the AMDGPU and ROCm repositories manually for all ROCm releases users want to install except the latest one from the wish list. The required repositories for the latest release are added automatically by *amdgpu-install* script.

Adding ROCm Repositories on Ubuntu

Add the ROCm repositrory using the following step. Choose `<release specific rocm baseurl>` from given *Base URLs for AMDGPU and ROCm Stack Repositories.*

```
$ echo 'deb [arch=amd64] <Release-1 specific rocm baseurl> ubuntu main' | sudo
tee /etc/apt/sources.list.d/rocm.list
```

```
$ echo 'deb [arch=amd64] <Release-2 specific rocm baseurl> ubuntu main' | sudo
tee /etc/apt/sources.list.d/rocm.list

$ sudo apt update
```

Adding ROCm Repositories on RHEL/CentOS

1. Create a /etc/yum.repos.d/rocm.repo file with the following content.

   NOTE: Provide the `<Release specific rocm baseurl>` baseurl from the given *Base URLs for AMDGPU and ROCm Stack Repositories.*

   ```
   [ROCm-1]

   Name=ROCm<Release version 1>

   baseurl=<Release-1 specific rocm baseurl>

   enabled=1

   gpgcheck=1

   gpgkey=https://repo.radeon.com/rocm/rocm.gpg.key

   [ROCm-2]

   Name=ROCm<Release version 2>

   baseurl=<Release-2 specific rocm baseurl>

   enabled=1

   gpgcheck=1

   gpgkey=https://repo.radeon.com/rocm/rocm.gpg.key
   ```

2. Execute the command below to clean the cached files from enabled repositories:

   ```
   $ sudo yum clean all
   ```

Adding ROCm Repositories on SLES/OpenSUSE

- Add new ROCm repositories for the release you want to install simultaneously.

  NOTE: Select the `<Release specific rocm baseurl>` from *Base URLs for AMDGPU and ROCm Stack Repositories.*

  ```
  $ sudo zypper addrepo <Release-1 specific rocm baseurl> ROCm

  $ sudo zypper addrepo <Release-2 specific rocm baseurl> ROCm
  ```

- Use the following command to update the added repositories.

  ```
  $ sudo zypper ref
  ```

4.1.3.2 Use the Installer to Install Multi-version ROCm Meta-packages

Use the installer script as given below.

```
$ sudo amdgpu-install --usecase=rocm --rocmrelease=<release-number-1>

$ sudo amdgpu-install --usecase=rocm --rocmrelease=<release-number-2>

$ sudo amdgpu-install --usecase=rocm --rocmrelease=<release-number-3>
```

NOTE: If the kernel-mode driver is already present on the system and users do not want to upgrade it, use *--no-dkms* option to skip the installation of kernel mode driver, as shown in the following samples.

```
$ sudo amdgpu-install --usecase=rocm --rocmrelease=4.5.0 --no-dkms
```

```
$ sudo amdgpu-install --usecase=rocm --rocmrelease=4.5.2 --no-dkms
```

The following are examples of ROCm multi-version Installation. The *kernel-mode driver*, associated with the ROCm release 5.0.0, will be installed as its latest release in the wish list.

```
$ sudo amdgpu-install --usecase=rocm --rocmrelease=4.5.0
```

```
$ sudo amdgpu-install --usecase=rocm --rocmrelease=4.5.2
```

```
$ sudo amdgpu-install --usecase=rocm --rocmrelease=5.0.0
```

4.2 Package Manager Method

The Package Manager method involves a manual set up of the repository, which includes cleaning up the system, updating and installing/uninstalling meta-packages using standard commands such as *yum, apt*, and others respective to the Linux distribution.

NOTE: Users must enter the desired meta-package as the `<package-name>` in the command. To utilize the newly installed packages, users must install the relevant drivers and restart the system after the installation.

The typical functions of a package manager installation system include:

- Working with file archivers to extract package archives.
- Ensuring the integrity and authenticity of the package by verifying them checksums and digital certificates, respectively.
- Looking up, downloading, installing, or updating existing packages from an online repository.
- Grouping packages by function to reduce user confusion.
- Managing dependencies to ensure a package is installed with all packages it requires, thus avoiding dependency.

NOTE: Users may consult the documentation for their package manager for more details.

4.2.1 Installing ROCm on Linux Distributions

For a fresh ROCm installation using the Package Manager method on a Linux distribution, follow the steps below:

1. Meet prerequisites – Ensure the *Prerequisite Actions* are met before the ROCm installation.
2. Install kernel headers and development packages – Ensure kernel headers and development packages are installed on the system.
3. Select the base URLs for AMDGPU and ROCm stack repository – Ensure the base URLs for AMDGPU, and ROCm stack repositories are selected.
4. Add AMDGPU stack repository – Ensure AMDGPU stack repository is added.
5. Install the kernel-mode driver and reboot the system – Ensure the kernel-mode driver is installed and the system is rebooted.
6. Add ROCm stack repository – Ensure the ROCm stack repository is added.
7. Install single version or multi-version ROCm meta-packages – Users may install the desired meta-packages.
8. Verify installation for the applicable distributions – Verify if the installation is successful.

NOTE: The kernel-mode driver cannot be installed in a docker container. Refer to the sections below for specific commands to install each Linux distribution's ROCm and AMDGPU stack.

4.2.2 Understanding AMDGPU and ROCm Stack Repositories on Linux Distributions

The AMDGPU and ROCm stack repositories are divided into two categories:

- Repositories with latest release packages
- Repositories for specific releases

4.2.2.1 Repositories with Latest Packages

These repositories contain the latest AMDGPU and ROCm packages available at the time. Based on the operating system's configuration, choosing this repository updates the packages automatically.

4.2.2.2 Repositories for Specific Releases

The release-specific repositories consist of packages from a specific release of the AMDGPU stack and ROCm stack. The repositories are not updated for the latest packages with subsequent releases. When a new ROCm release is available, the new repository, specific to that release, is added. Users can select a specific release to install, update the previously installed single version to the later available release, or add the latest version of ROCm and currently installed by using the multi-version ROCm packages.

NOTE: Users installing multiple versions of the ROCm stack must use the release-specific base URL.

4.2.3 Using Package Manager on Ubuntu

4.2.3.1 Installation Of Kernel Headers and Development Packages on Ubuntu

The following instructions to install kernel headers and development packages apply to all versions and kernels of Ubuntu. The ROCm installation requires the Linux-headers and Linux-modules-extra package to be installed with the correct version corresponding to the kernel's version. For example, if the system is running the Linux kernel version 5.4.0-77, the identical versions of Linux-headers and development packages must be installed. You may refer to the *Kernel Information* section to check the system's kernel version.

For the Ubuntu/Debian environment, execute the following command to verify the kernel headers and development packages are installed with the respective versions.

```
$ sudo dpkg -l | grep linux-headers
```

The command indicates if there are Linux headers installed as shown below:

```
linux-headers-5.4.0-77-generic 5.4.0-77.86~18.04.1
amd64 Linux kernel headers for version 5.4.0 on 64 bit x86 SMP
```

Execute the following command to check whether the development packages are installed,

```
$ sudo dpkg -l | grep linux-modules-extra
```

When run, the command mentioned above lists the installed linux-modules-extra packages like the output below:

```
linux-modules-extra-5.4.0-77-generic 5.4.0-77.86~18.04.1
amd64  Linux kernel extra modules for version 5.4.0 on 64-bit x86 SMP
```

If the supported version installation of Linux headers and development packages are not installed on the system, execute the following command to install the packages:

```
$ sudo apt install linux-headers-`uname -r` linux-modules-extra-`uname -r`
```

4.2.3.2 Base URLs For AMDGPU and ROCm Stack Repositories

Ubuntu 18.04 / 20.04

Repositories with Latest Packages

```
amdgpu baseurl= https://repo.radeon.com/amdgpu/latest/ubuntu

rocm baseurl= https://repo.radeon.com/rocm/apt/debian/
```

Repositories for Specific Releases

```
amdgpu baseurl= https://repo.radeon.com/amdgpu/21.50/ubuntu

rocm baseurl= https://repo.radeon.com/rocm/apt/5.0
```

4.2.3.3 Adding AMDGPU Stack Repository

Add GPG Key for AMDGPU and ROCm Stack

Add the *gpg* key for AMDGPU and ROCm repositories. For Debian-based systems like Ubuntu, configure the Debian ROCm repository as follows:

```
$ wget -q -O - https://repo.radeon.com/rocm/rocm.gpg.key | sudo apt-key add -
```

NOTE: The *gpg* key may change. Ensure it is updated when installing a new release. If the key signature verification fails while updating, re-add the key from the ROCm *apt* repository as mentioned above. The current *rocm.gpg.key* is not available in a standard key ring distribution.

```
73f5d8100de6048aa38a8b84cd9a87f05177d208 rocm.gpg.key
```

Add the AMDGPU Stack Repository

You may skip this section if you have a version of the kernel-mode driver installed. If you do not have a version of the kernel-mode driver installed, follow the commands below to add the AMDGPU stack repository.

For `<amdgpu baseurl>` in the command below, refer to the AMDGPU base URLs as documented in *Base URLs for AMDGPU and ROCm Stack Repositories*

```
Ubuntu 18.04

$ echo 'deb [arch=amd64] <amdgpu baseurl> bionic main' | sudo tee
/etc/apt/sources.list.d/amdgpu.list

Ubuntu 20.04

$ echo 'deb [arch=amd64] <amdgpu baseurl> focal main' | sudo tee
/etc/apt/sources.list.d/amdgpu.list
```

Execute the command below to update the package list.

```
$ sudo apt-get update
```

4.2.3.4 Install the Kernel Mode Driver and Reboot System

You may skip this section if you have the kernel-mode driver installed on your system. If you do not have the kernel-mode driver on your system, follow the instructions below.

Ensure the system is rebooted after the kernel-mode driver is installed.

```
$ sudo apt install amdgpu-dkms

$ sudo reboot
```

4.2.3.5 Add ROCm Stack Repository

Add the ROCm repository.

For `<rocm baseurl>` in the command below, refer to the ROCm base URLs as documented in *Base URLs for AMDGPU and ROCm Stack Repositories*

```
$ echo 'deb [arch=amd64] <rocm baseurl> ubuntu main' | sudo tee
/etc/apt/sources.list.d/rocm.list

$ sudo apt-get update
```

4.2.3.6 Install ROCm Meta-packages

Single Version ROCm Packages Installation

Install ROCm meta-packages. Specify the name of the meta-package you want to install as `<package-name>`, as shown below:

```
$ sudo apt install <package-name>
```

For example:

```
$ sudo apt install rocm-hip-sdk
```

```
$ sudo apt install rocm-hip-sdk rocm-opencl-sdk
```

Multi-version ROCm Packages Installation

NOTE: If the existing ROCm release contains non-versioned ROCm packages, you must uninstall those packages before proceeding to the multi-version installation to avoid conflicts.

Adding New ROCm Repositories

Add the ROCm repository using the following steps and choose `<release specific rocm baseurl>` from given *Base URLs for AMDGPU and ROCm Stack Repositories.*

```
$ echo 'deb [arch=amd64] <Release-1 specific rocm baseurl> ubuntu main' | sudo
tee /etc/apt/sources.list.d/rocm.list

$ echo 'deb [arch=amd64] <Release-2 specific rocm baseurl> ubuntu main' | sudo
tee /etc/apt/sources.list.d/rocm.list

$ sudo apt update
```

NOTE: The above step to add a repository must be repeated for each release that users prefer to install.

Installing Multi-version ROCm Package

Install the desired packages using the command below:

```
$ sudo apt install <package-names with release version>
```

For example:

```
$ sudo apt install rocm-hip-sdk4.5.0 rocm-hip-sdk5.0.0
```

Refer to the *Post Install Actions and Verification Process* section to verify if the installation is successful.

4.2.4 Using Package Manager on RHEL/CentOS

4.2.4.1 Installation Of Kernel Headers and Development Packages on RHEL/CentOS

The ROCm installation requires the linux-headers and linux-modules-extra package to be installed with the correct version corresponding to the kernel's version. For example, if the system is running Linux kernel version 5.4.0-77, the identical versions of linux-headers and development packages must be installed.

Refer to the *Kernel Information* section to check the kernel version on your system.

To verify you have the supported version of the installed linux-headers and linux-modules-extra package, type the following on the command line:

```
$ sudo yum list installed | grep linux-headers
```

The command mentioned above displays the list of linux headers versions currently present on your system. Verify if the listed linux headers have the same versions as the kernel.

The following command lists the development packages on your system. Verify if the listed development package's version number matches the kernel version number.

```
$ sudo yum list installed | grep linux-modules-extra
```

If the supported version installation of linux headers and development packages does not exist on the system, execute the commands below to install:

```
$ sudo yum install kernel-headers-`uname -r` kernel-devel-`uname -r`
```

Preparing RHEL 7.9 for Installation

You must enable the external repositories to install on the devtoolset-7 environment and the support files.

NOTE: Devtoolset is not required for CentOS v8.3/RHEL v8.4

NOTE: The subscription for RHEL must be enabled and attached to a pool ID. See the Obtaining an RHEL image and license page for instructions on registering your system with the RHEL subscription server and linking to a pool id.
Enable the following repositories for RHEL v7.9:

```
$ sudo subscription-manager repos --enable rhel-server-rhscl-7-rpms

$ sudo subscription-manager repos --enable rhel-7-server-optional-rpms

$ sudo subscription-manager repos --enable rhel-7-server-extras-rpms

$ sudo subscription-manager repos --enable=rhel-7-server-devtools-rpms
```

Preparing CentOS for Installation

The following steps help users prepare the CentOS system for the ROCm installation.

Extra Packages for Enterprise Linux (EPEL) provides additional packages for CentOS that are not available in their standard repositories. Install the EPEL repository configuration package using the following command.

```
#For CentOS 7&8:

$ sudo yum install epel-release

#Only for CentOS 7.9

$ sudo yum install -y centos-release-scl
```

Installing Devtoolset-7 for RHEL 7.9 /CentOS 7.9

Use the following command to install Devtoolset-7:

```
$ sudo yum install devtoolset-7

$ source scl_source enable devtoolset-7
```

4.2.4.2 Base URLs For AMDGPU and ROCm Stack Repositories

CentOS/RHEL 7.9

Repositories with Latest Packages

```
amdgpu baseurl= https://repo.radeon.com/amdgpu/latest/rhel/7.9/main/x86_64

rocm baseurl= https://repo.radeon.com/rocm/yum/rpm
```

Repositories for Specific Releases

```
amdgpu baseurl= https://repo.radeon.com/amdgpu/21.50/rhel/7.9/main/x86_64

rocm baseurl= https://repo.radeon.com/rocm/yum/5.0
```

CentOS 8.3 /RHEL 8.4

Repositories with Latest Packages

```
amdgpu baseurl= https://repo.radeon.com/amdgpu/latest/rhel/8.4/main/x86_64

rocm baseurl= https://repo.radeon.com/rocm/centos8/rpm
```

Repositories for Specific Releases

```
amdgpu baseurl=https://repo.radeon.com/amdgpu/21.50/rhel/8.4/main/x86_64

rocm baseurl=https://repo.radeon.com/rocm/centos8/5.0/
```

RHEL 8.5

Repositories with Latest Packages

```
amdgpu baseurl= https://repo.radeon.com/amdgpu/latest/rhel/8.5/main/x86_64

rocm baseurl= https://repo.radeon.com/rocm/centos8/rpm
```

Repositories for Specific Releases

```
amdgpu baseurl= https://repo.radeon.com/amdgpu/21.50/rhel/8.5/main/x86_64

rocm baseurl= https://repo.radeon.com/rocm/centos8/5.0/
```

4.2.4.3 Adding AMDGPU Stack Repository

You may skip this section if you have a version of the kernel-mode driver installed. If you do not have a version of the kernel-mode driver installed, follow the commands below to add the AMDGPU stack repository.

Add the AMDGPU Stack Repository

Create a */etc/yum.repos.d/amdgpu.repo* file with the following contents with *amdgpu base URL*. For `<amdgpu baseurl>` in the command below, refer to the AMDGPU base URLs as documented in *Base URLs for AMDGPU and ROCm Stack Repositories*.

```
[amdgpu]

name=amdgpu

baseurl=<amdgpu baseurl>

enabled=1

gpgcheck=1

gpgkey=https://repo.radeon.com/rocm/rocm.gpg.key
```

NOTE: The *gpg* key may change; ensure it is updated when installing a new release. If the key signature verification fails while updating, re-add the key from the ROCm to the *yum* repository as mentioned above. The current *rocm.gpg.key* is not available in a standard key ring distribution but has the following sha1sum hash:

```
73f5d8100de6048aa38a8b84cd9a87f05177d208 rocm.gpg.key
```

Execute the command below to clean the cached files from enabled repositories:

```
$ sudo yum clean all
```

4.2.4.4 Install the Kernel Mode Driver and Reboot System

You may skip this section if the kernel-mode driver is already installed on your system. If you do not have a version of the kernel-mode driver installed, follow the commands below to install the kernel-mode driver:

```
$ sudo yum install amdgpu-dkms
```

Reboot the system after the completion of driver installation.

```
$ sudo reboot
```

4.2.4.5 Add the ROCm Stack Repository

Create a */etc/yum.repos.d/rocm.repo* file with the following content.

For `<rocm baseurl>` in the command below, refer to the ROCm base URLs documented in *Base URLs for AMDGPU and ROCm Stack Repositories*.

```
[rocm]

name=rocm

baseurl=<rocm baseurl>

enabled=1

gpgcheck=1

gpgkey=https://repo.radeon.com/rocm/rocm.gpg.key
```

NOTE: The *gpg* key may change; ensure it is updated when installing a new release. If the key signature verification fails while updating, re-add the key from the ROCm *yum* repository as mentioned above. The current *rocm.gpg.key* is not available in a standard key ring distribution, but has the following sha1sum hash:

```
73f5d8100de6048aa38a8b84cd9a87f05177d208 rocm.gpg.key
```

Execute the command below to clean the cached files from enabled repositories:

```
$ sudo yum clean all
```

4.2.4.6 Install ROCm Meta-packages

Single Version ROCm Packages Installation

Use the following command to install the ROCm packages.

```
$ sudo yum install <package-name>
```

Specify the meta-package name as `<package-name>`, which you want to install, in the command given above.

For example,

```
$ sudo yum install rocm-hip-sdk
```

```
$ sudo yum install rocm-hip-sdk rocm-opencl-sdk
```

Multi-Version ROCm Meta-Packages Installation

NOTE: If the existing ROCm release contains non-versioned ROCm packages, you must uninstall those packages before proceeding to the multi-version installation to avoid conflicts.

Adding New ROCm repositories

1. Create a /etc/yum.repos.d/rocm.repo file with the following content.

   NOTE: Provide the `<Release specific rocm baseurl>` baseurl from given *Base URLs for AMDGPU and ROCm Stack Repositories.*

   ```
   [ROCm-1]

   Name=ROCm<Release version 1>

   baseurl=<Release-1 specific rocm baseurl>

   enabled=1

   gpgcheck=1

   gpgkey=https://repo.radeon.com/rocm/rocm.gpg.key

   [ROCm-2]

   Name=ROCm<Release version 2>

   baseurl=<Release-2 specific rocm baseurl>

   enabled=1

   gpgcheck=1

   gpgkey=https://repo.radeon.com/rocm/rocm.gpg.key
   ```

   NOTE: The step above to add the repository must be repeated for each release users prefer to install.

2. Execute the command below to clean the cached files from enabled repositories:

   ```
   $ sudo yum clean all
   ```

Installing Multi-version ROCm Meta-Package

Use the following commands to install the required meta-packages

```
$ sudo yum install <package-name with release version>
```

For example:

```
$ sudo yum install rocm-hip-sdk4.5.0 rocm-hip-sdk5.0.0
```

Refer to the *Post Install Actions and Verification Process* section to verify if the installation is successful.

4.2.5 Using Package Manager on SLES/OpenSUSE

This section introduces the ROCm installation process on SLES/OpenSUSE.

4.2.5.1 Installation of Kernel Headers and Development Packages

ROCm installation requires linux-headers and linux-modules-extra package to be installed with the correct version corresponding to the kernel's version. For example, if the system is running the Linux kernel version 5.4.0-77, the same versions of linux-headers and development packages must be installed.

Refer to the *Kernel Information* section to check the kernel version on your system.

Ensure that the correct version of the latest *kernel-default-devel* and *kernel-default* packages are installed. The following command lists the installed *kernel-default-devel* and *kernel-default package*.

```
$ sudo zypper info kernel-default-devel or kernel-default
```

NOTE: This next step is only required if you find from the above command that the "kernel-default-devel" and "kernel-default" versions of the package, corresponding to the kernel release version, do not exist on your system.

If the required version of packages does not exist on the system, install with the command below:

```
$ sudo zypper install kernel-default-devel or kernel-default
```

4.2.5.2 Base URLs For AMDGPU And ROCm Stack Repositories

SLES 15 Service Pack 3

Repositories with Latest Packages

```
amdgpu baseurl= https://repo.radeon.com/amdgpu/latest/sle/15/main/x86_64

rocm baseurl=  https://repo.radeon.com/rocm/zyp/zypper/
```

Repositories for Specific Releases

```
amdgpu baseurl= https://repo.radeon.com/amdgpu/21.50/sle/15/main/x86_64

rocm baseurl= https://repo.radeon.com/rocm/zyp/5.0/
```

4.2.5.3 Adding AMDGPU Stack Repository

You may skip this section if you have a version of the kernel-mode driver installed. If you do not have a version of the kernel-mode driver installed, follow the commands below to add the AMDGPU stack repository.

Add the AMDGPU Stack Repository

Create a /etc/zypp/repos.d/amdgpu.repo file with the following content.

For `<amdgpu baseurl>` in the command below, refer to the AMDGPU base URLs as documented in *Base URLs for AMDGPU and ROCm Stack Repositories*.

```
[amdgpu]

name=amdgpu

baseurl=<amdgpu baseurl>

enabled=1

gpgcheck=1

gpgkey=https://repo.radeon.com/rocm/rocm.gpg.key
```

NOTE: The *gpg* key may change; ensure it is updated when installing a new release. If the key signature verification fails while updating, re-add the key from the ROCm *zypp* repository as mentioned above. The current *rocm.gpg.key* is not available in a standard key ring distribution but has the following sha1sum hash:

```
73f5d8100de6048aa38a8b84cd9a87f05177d208 rocm.gpg.key
```

Use the following commands to update the added repository and add perl repository:

```
$ sudo zypper ref

$ sudo zypper clean --all

$ sudo zypper addrepo
https://download.opensuse.org/repositories/devel:languages:perl/SLE_15/devel:languages:perl.repo

$ sudo SUSEConnect -p sle-module-desktop-applications/15.3/x86_64

$ sudo SUSEConnect --product sle-module-development-tools/15.3/x86_64

$ sudo SUSEConnect --product PackageHub/15.3/x86_64

$ sudo zypper ref
```

4.2.5.4 Install the Kernel Mode Driver and Reboot System

Install the kernel-mode driver. If you already have a version of the kernel-mode driver installed, you may skip this section. If you do not have a version of the kernel-mode driver installed, follow the commands below to install and reboot the system.

```
$ sudo zypper --gpg-auto-import-keys install amdgpu-dkms

$ sudo reboot
```

4.2.5.5 Add the ROCm Stack Repository

Add the ROCm repository by executing the following commands,

Create a /etc/zypp/repos.d/rocm.repo file with the following content.

For `<rocm baseurl>` in the command below, refer to the ROCm base URLs documented in *Base URLs for AMDGPU and ROCm Stack Repositories.*

```
[rocm]

name=rocm

baseurl=<rocm baseurl>

enabled=1

gpgcheck=1

gpgkey=https://repo.radeon.com/rocm/rocm.gpg.key
```

NOTE: The *gpg* key may change. Ensure it is updated when installing a new release. If the key signature verification fails while updating, re-add the key from the ROCm *zypp* repository as mentioned above. The current *rocm.gpg.key* is not available in a standard key ring distribution but has the following sha1sum hash:

```
73f5d8100de6048aa38a8b84cd9a87f05177d208 rocm.gpg.key
```

Use the following command to update the added repository.

```
$ sudo zypper ref
```

4.2.5.6 Install ROCm Meta-packages

Single Version ROCm Packages Installation

Install the ROCm package by typing the command below:

```
$ sudo zypper --gpg-auto-import-keys install <package-name>
```

Specify the name of the meta-package name as `<package-name>`, which you want to install, in the command given above.

For example,

```
$  sudo zypper --gpg-auto-import-keys install rocm-hip-sdk
```

```
$  sudo zypper --gpg-auto-import-keys install rocm-hip-sdk rocm-opencl-sdk
```

Multi-version ROCm Packages Installation

NOTE: If the existing ROCm release contains non-versioned ROCm packages, you must uninstall those packages before proceeding to the multi-version installation to avoid conflicts.

Adding new ROCm repositories

Add new ROCm repositories for the release, you want to install simultaneously. Select the `<Release specific rocm baseurl>` from *Base URLs for AMDGPU and ROCm Stack Repositories.*

```
$ sudo zypper addrepo <Release-1 specific rocm baseurl> ROCm

$ sudo zypper addrepo <Release-2 specific rocm baseurl> ROCm
```

NOTE: The step to add a repository must be repeated for each release that users want to install.

Use the following command to update the added repositories.

```
$ sudo zypper ref
```

Installing Multi-version ROCm Package

Install ROCm meta-packages using the command below.

```
$ sudo zypper --gpg-auto-import-keys install <package-name with release
version>
```

For example:

```
$ sudo zypper --gpg-auto-import-keys install rocm-hip-sdk4.5.0 rocm-hip-sdk5.0.0
```

Refer to the *Post Install Actions and Verification Process* section to verify if the installation is successful.

4.3 Post Install Actions and Verification Process

4.3.1 Post Install Actions

Users might set LD_LIBRARY_PATH to load the ROCm library version of choice.

```
$ export LD_LIBRARY_PATH=/opt/rocm-<version>/lib;/opt/rocm-<version>/lib64
```

NOTE: For convenience, users may add the ROCm binaries in your PATH, as shown in the example below.

```
$ echo 'export PATH=$PATH:/opt/rocm-<version>/bin:/opt/rocm-
<version>/opencl/bin'
```

4.3.2 Verifying Kernel-Mode Driver Installation

Users can check the installation of the kernel-mode driver by typing the command given below:

```
$ sudo dkms status
```

4.3.3 Verifying ROCm Installation

After completing the ROCm installation, users can execute the following commands on the system to verify if the installation is successful. If you see your GPUs listed by both commands, the installation is considered successful.

```
/opt/rocm-<version>/bin/rocminfo

OR

/opt/rocm-<version>/opencl/bin/clinfo
```

4.3.4 Verifying Package Installation

Users can use the following commands to ensure the packages are installed successfully.

Installed Packages

Linux Distro	Command
Ubuntu/Debian	`$ sudo apt list --installed`
RHEL/CentOS	`$ sudo yum list installed`
OpenSUSE / SLES	`$ sudo zypper search --installed-only`

Chapter 5 How To Upgrade ROCm

This section explains how to upgrade the existing kernel-mode driver and ROCm packages to the latest version.

NOTE: The assumption is that users already have a version of the kernel-mode driver and the ROCm software stack is installed on the system.

NOTE: Package upgrade is applicable to single version packages only. If the preference is to install an updated version of the ROCm stack along with the currently installed version, refer to *Installing Multiple Versions of ROCm*.

To upgrade the system with the desired ROCm release, follow the steps below:

1. Update the AMDGPU Stack Repository – Ensure the AMDGPU repository is updated.
2. Upgrade the kernel-mode driver and reboot the system – Ensure the kernel-mode driver is upgraded, and the system is rebooted.
3. Update the ROCm Repository – Ensure the ROCm repository is updated with the desired ROCm release.
4. Upgrade the ROCm meta-packages – Upgrade the ROCm meta-packages.
5. Verify upgrade for the applicable distributions- Verify if the upgrade is successful.

Refer to the sections below for specific commands to upgrade ROCm on different Linux distributions.

5.1.1 Upgrading ROCm on Ubuntu

Follow the steps below to upgrade the ROCm release for Ubuntu.

5.1.1.1 Update AMDGPU Stack Repository

If users already have an *amdgpu baserul* pointing to *Repositories with Latest Packages*, skip to the next step . Otherwise, either

- Update `<amdgpu baseurl>` pointing to *Repositories with Latest Packages* OR
- Update `<amdgpu baseurl>` pointing to *Repositories for Specific Releases,* you want to upgrade.

Ubuntu 18.04

```
$ echo 'deb [arch=amd64] <amdgpu baseurl> bionic main' | sudo tee
/etc/apt/sources.list.d/amdgpu.list
```

Ubuntu 20.04

```
$ echo 'deb [arch=amd64] <amdgpu baseurl> focal main' | sudo tee
/etc/apt/sources.list.d/amdgpu.list
```

Execute the command given below to update the package list.

```
$ sudo apt update
```

5.1.1.2 Upgrade The Kernel Mode Driver and Reboot System

Install the kernel-mode driver using the following commands.

```
$ sudo apt install amdgpu-dkms

$ sudo reboot
```

5.1.1.3 Update The ROCm Stack Repository

If users already have a *rocm baserul* pointing to the *Repositories with Latest Packages,* skip to the next step. Otherwise you can either

- Update `<rocm baseurl>` pointing to *Repositories with Latest Packages* OR
- Update `<rocm baseurl>` pointing to *Repositories for Specific Releases,* you want to upgrade.

```
$ echo 'deb [arch=amd64] <rocm baseurl> ubuntu main' | sudo tee
/etc/apt/sources.list.d/rocm.list
```

5.1.1.4 Upgrade ROCm Meta-Packages

Execute the command below to update the package list.

```
$ sudo apt update
```

5.1.2 Upgrading ROCm on RHEL/CentOS

Users can upgrade the ROCm release on the RHEL/CentOS system by following the steps given below.

5.1.2.1 Update AMDGPU Stack Repository

If users already have an *amdgpu baserul* in /etc/yum.repos.d/amgdgpu.repo file pointing to the *Repositories with Latest Packages*, skip to the next step . Otherwise,

- Update `<amdgpu baseurl>` in /etc/yum.repos.d/amgdgpu.repo file pointing to *Repositories with Latest Packages*

  OR

- Update `<amdgpu baseurl>` in /etc/yum.repos.d/amgdgpu.repo file pointing to *Repositories for Specific Releases,* you want to upgrade.

```
[amdgpu]

Name=amdgpu

baseurl=<amdgpu baseurl>

enabled=1

gpgcheck=1

gpgkey=https://repo.radeon.com/rocm/rocm.gpg.key
```

Execute the command below to clean the cached files from enabled repositories:

```
$ sudo yum clean all
```

5.1.2.2 Upgrading Kernel Mode Driver and Reboot System

Install the kernel-mode driver and reboot the machine using the following commands.

```
$ sudo yum install amdgpu-dkms

$ sudo reboot
```

5.1.2.3 Updating ROCm Repositories

If users already have a *rocm baserul* in the file /etc/yum.repos.d/rocm.repo pointing to the *Repositories with Latest Packages,* skip to next step. Otherwise, you can either

- Update `<rocm baseurl>` in the file /etc/yum.repos.d/rocm.repo pointing to *Repositories with Latest Packages*

  OR

- Update `<rocm baseurl>` in the file /etc/yum.repos.d/rocm.repo pointing to *Repositories for Specific Releases,* you want to upgrade.

```
[ROCm]

Name=ROCm

baseurl=<rocm baseurl>

enabled=1

gpgcheck=1

gpgkey=https://repo.radeon.com/rocm/rocm.gpg.key
```

Execute the command below to clean the cached files from enabled repositories:

```
$ sudo yum clean all
```

5.1.2.4 Upgrade ROCm Meta-Packages

Use the following commands to upgrade the meta-packages.

```
$ sudo yum update
```

5.1.3 Upgrading ROCm on SLES/OpenSUSE

Upgrade ROCm release on SLES/OpenSUSE distribution by executing the steps mentioned in the following section.

5.1.3.1 Updating The AMDGPU Stack Repository

If users already have an *amdgpu baserul* pointing to the *Repositories with Latest Packages*, skip to the next step. Otherwise, either

- Update `<amdgpu baseurl>` pointing to *Repositories with Latest Packages* OR
- Update `<amdgpu baseurl>` pointing to *Repositories for Specific Releases,* you want to upgrade.

```
$ sudo zypper addrepo <amdgpu baseurl> amdgpu

$ sudo zypper ref
```

5.1.3.2 Upgrade Kernel-Mode Driver and Reboot System

Upgrade the kernel-mode driver using the following command and reboot the system.

```
$ sudo zypper --gpg-auto-import-keys install amdgpu-dkms

$ sudo reboot
```

5.1.3.3 Update ROCm Stack Repositories

If users already have a *rocm baserul* pointing to the *Repositories with Latest Packages,* skip to the next step. Otherwise, you can either

- Update `<rocm baseurl>` pointing to *Repositories with Latest Packages* OR
- Update `<rocm baseurl>` pointing to *Repositories for Specific Releases,* you want to upgrade.

```
$ sudo zypper addrepo <rocm baseurl> ROCm

$ sudo zypper ref
```

5.1.3.1 Upgrade ROCm Meta-Packages

Type the following command to upgrade the meta-packages.

```
$ sudo zypper update
```

5.1.4 Verification Process

Refer to the *Post Install Actions and Verification Process* section to verify if the upgrade is successful.

Chapter 6 ROCm Stack Uninstallation

Uninstallation of ROCm entails removing ROCm packages, tools, and libraries from the system.

6.1 Uninstalling ROCm Stack

6.1.1 Removing ROCm Toolkit and Driver

This section describes the uninstallation process in detail. The following methods remove the ROCm stack from the system.

6.1.2 Choosing an Uninstallation Method

You can uninstall using the following methods:

- Uninstallation using the Uninstall Script
- Package Manager Uninstallation

NOTE: If ROCm is installed using package manager commands, it can be uninstalled using package manager uninstallation method only. Otherwise, users can choose any method from given above.

6.1.2.1 Uninstallation Using Uninstall Script

The following commands uninstall all installed ROCm packages:

Uninstalling Single Version Install

```
$ sudo amdgpu-uninstall
```

NOTE*: amdgpu-uninstall* ignores all parameters/arguments and uninstalls all ROCm packages associated with the installed ROCm release along with *kernel mode driver*.

Uninstalling A Specific ROCm Release

If a system has multiple ROCm release installed, users have option to uninstall specific ROCm release by using the following command:

```
$ sudo amdgpu-uninstall --rocmrelease=<release-number>
```

This command will uninstall all the ROCm specific packages only, from the mentioned release number.

Uninstalling All ROCm Releases

All the installed ROCm releases can be uninstalled with uninstall script:

```
$ sudo amdgpu-uninstall --rocmrelease=all
```

NOTE: This command will not uninstall kernel mode driver. If you wish to uninstall kernel mode driver, execute the following command:

```
$ sudo amdgpu-uninstall
```

6.1.2.2 Uninstallation Using Package Manager

The Package Manager uninstallation offers a method for a clean uninstallation process for ROCm. This section describes how to uninstall the ROCm for various Linux distributions.

Use the following commands to remove the specific meta-packages from the system.

Uninstalling Specific Meta-packages

Use the following command to uninstall specific meta-packages. You may specify the name of the meta-package name as `<package-name>` you want to uninstall in the command given below.

UBUNTU/DEBIAN

```
#Uninstall single version ROCm packages

$ sudo apt autoremove <package-name>

#Uninstall multi-version ROCm packages

$ sudo apt autoremove <package-name with release version>
```

RHEL/CentOS

```
#Uninstall single version ROCm packages

$ sudo yum remove <package-name>

#Uninstall multi-version ROCm packages

$ sudo yum remove <package-name with release version>
```

SLES/OPENSUSE

```
#Uninstall single version ROCm packages

$ sudo zypper remove <package-name>

#Uninstall multi-version ROCm packages

$ sudo zypper remove <package-name with release version>
```

Complete Uninstallation of ROCm Packages

If you want to uninstall all installed ROCm packages from a ROCm release, use the following command.

Note, the uninstallation of the `rocm-core` package removes all ROCm specific packages from the system.

UBUNTU/DEBIAN

```
#Uninstall all single version ROCm packages

$ sudo apt autoremove rocm-core

#Uninstall all multi-version ROCm packages

$ sudo apt autoremove rocm-core<release version>
```

RHEL/CentOS

```
#Uninstall all single version ROCm packages

$ sudo yum remove rocm-core

#Uninstall all multi-version ROCm packages

$ sudo yum remove rocm-core<release version>
```

SLES/OPENSUSE

```
#Uninstall all single version ROCm packages

$ sudo zypper remove rocm-core

#Uninstall all multi-version ROCm packages

$ sudo zypper remove rocm-core<release version>
```

NOTE: The above command removes all installed ROCm-specific packages.

Refer to the *Uninstall Kernel Mode Driver* section below to uninstall the kernel-mode driver uninstallation.

Uninstall Kernel Mode Driver

Users can uninstall the kernel-mode driver by entering the following command on the system.

UBUNTU/DEBIAN

```
$ sudo apt autoremove amdgpu-dkms
```

RHEL/CentOS

```
$ sudo yum remove amdgpu-dkms
```

SLES/OPENSUSE

```
$ sudo zypper remove amdgpu-dkms
```

Remove ROCm and AMDGPU Repositories

UBUNTU/DEBIAN

Use the following commands to remove the AMDGPU and ROCm repository from the Ubuntu/Debian system:

```
$ sudo rm /etc/apt/sources.list.d/<rocm_repository-name>.list

$ sudo rm /etc/apt/sources.list.d/<amdgpu_repository-name>.list
```

Clear cache and clean the system.

```
$ sudo rm -rf /var/cache/apt/*

$ sudo apt-get clean all
```

Reboot the system.

```
$ sudo reboot
```

RHEL/CentOS

This section describes the process of removing AMDGPU *and ROCm repositories* from the RHEL/CentOS environment.

Remove the reference to the AMDGPU and ROCm repository from the system using the following instructions

```
$ sudo rm -rf /etc/yum.repos.d/<rocm_repository-name> # Remove only rocm repo

$ sudo rm -rf /etc/yum.repos.d/<amdgpu_repository-name> # Remove only amdgpu
repo
```

Clear cache and clean the system.

```
$ sudo rm -rf /var/cache/yum   #Remove the cache

$ sudo yum clean all
```

Restart the system.

```
$ sudo reboot
```

SLES/OPENSUSE

This section describes the process of removing AMDGPU and ROCm repositories from the SLES/OPENSUSE environment.

Remove the reference to the amdgpu and ROCm repository from the system with the commands below.

```
$ sudo zypper removerepo <rocm_repository-name>

$ sudo zypper removerepo <amdgpu_repository-name>
```

Clear cache and clean the system.

```
$ sudo zypper clean --all
```

Restart the system.

```
$ sudo reboot
```

Chapter 7 Frequently Asked Questions

Q. Can users install multiple packages at the same time with the installer script?

Ans: Yes, users can install multiple packages at the same time with the installer script. Provide package names in the *--usecase* parameter, separated by a comma, as shown below.

```
$ sudo amdgpu-install --usecase=hiplibsdk,rocm
```

Q. How to list all the possible inputs for the `--usecase` parameter in the `amdgpu-install` script?

Ans: The following command lists all the possible options for `–usecase`

```
amdgpu-install --list-usecase
```

Q. What are the available options other than the `--usecase` in the `amdgpu-install` script?

Ans: The following command lists all possible options users can provide in the *amdgpu-install* script.

```
$ sudo amdgpu-install --help
```

Q. How to check if the kernel module is installed successfully?

Ans: Type the following command on the system.

```
$ sudo dkms status
```

The command displays the output in the following format if the installation of the kernel module is successful.

```
amdgpu, 4.3-52.el7, 3.10.0-1160.11.1.el7.x86_64, x86_64: installed
(original_module exists)
```

Q. Does docker container supports command - `$ sudo SUSEConnect --product PackageHub/15.2/x86_64?`

Ans: Users do not need to execute the command `$ sudo SUSEConnect --product PackageHub/15.2/x86_64` in docker container.
