Atlas 300 AI Accelerator Card
1.0.0

DDK Installation Guide (EulerOS) (Model 3000)

Issue	01
Date	2020-03-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Introduction

This document describes the method of separately installing the device development kit (DDK) and troubleshooting of common problems. It is applicable to development in command line interface (CLI) mode.

The DDK is an NPU-based digital developer kit that can be used to build a compilation environment for projects. Different release packages integrate the DDKs of different NPU forms. The DDK of the current version contains the dependency libraries and header files of components such as the TE, DVPP, and process orchestration. Users can compile the project files by using **makefile**.

2 Obtaining Software Packages

Before installing the tool, prepare the following software package:

DDK installation package: Visit **https://support.huawei.com/enterprise**, input the product name in the search box, and press **Enter** or click the search icon. In the displayed window, click the **Software Download** tab to obtain the desired installation package.

Table 2-1 describes the required software package.

Table 2-1 Overview of the software package

Installation Package	Content	Integrity Verification File for the Software Package
MSpore_DDK-{version}-aarch64.eulerOS-aarch64.eulerOS-aarch64.miniOS.tar.gz	DDK installation package	MSpore_DDK-{version}-aarch64.eulerOS-aarch64.eulerOS-aarch64.miniOS.tar.gz.asc

3 Verifying the Software Package Integrity

To prevent a software package from being maliciously tampered with during transmission or storage, download the corresponding digital signature file for integrity verification when downloading the software package.

After the software package is downloaded, verify its PGP digital signature according to the *OpenPGP Signature Verification Guide*. If the software package fails the verification, do not use the software package, and contact Huawei technical support.

Before a software package is used in installation or upgrade, its digital signature also needs to be verified according to *OpenPGP Signature Verification Guide* to ensure that the software package is not tampered with.

For carrier users, visit **https://support.huawei.com/carrier/digitalSignatureAction**.

For enterprise users, visit **https://support.huawei.com/enterprise/en/tool/pgp-verify-TL1000000054**.

4 Environment Preparation

Environment Requirements

The DDK installation must meet the following environment requirements on the hardware and operating system (OS).

Table 4-1 EulerOS version information

Category	Version	Obtaining Method	Precautions
OS	EulerOS-V2.0SP8	Contact Huawei technical support to obtain the following recommended version: **EulerOS-V2.0SP8-aarch64-dvd.iso**	You are advised to use EulerOS whose kernel version is **4.19.36-vhulk1907.1.0.h410.eulerosv2r8.aarch64**.
Python	Python 2: 2.7+ Python 3: 3.5+	For details, see **Installing Dependency**.	-

(Optional) Preparing the DDK Installation User

You can install the DDK as any user (**root** or non-**root**).

- If the **root** user is used for installation, skip this section.
- If an existing non-**root** user is used for installation, ensure that the user has the read, write, and execute permissions on the **$HOME** directory.
- If you want to install the DDK as a new non-**root** user, perform the following steps as the **root** user: This document describes how to install the DDK as a new non-**root** user.

a. Run the following commands to create a DDK installation user and set the **$HOME** directory of the user. Ensure that the user has the read, write, and execute permissions on the **$HOME** directory.
```
useradd -d /home/username -m username
```

b. Run the following command to set the password:
```
passwd username
```

> NOTE
>
> *username* indicates the user name for installing the DDK. The **umask** value of the user cannot be greater than **0027**.
>
> - To view the **umask** value, run the **umask** command.
> - You can change the **umask** value by running the **umask *NewValue*** command.

If the created non-**root** user is **HwHiAiUser**, the DDK is deployed on the host server, and **HwHiAiUser** is used for offline model conversion, logs are stored in the **var/dlog** directory on the host by default. To enable the output of logs to the screen by default, you can set the following environment variable:
```
export  SLOG_PRINT_TO_STDOUT=1
```

- Change the size of the file descriptor.

a. Open the system configuration file **/etc/security/limits.conf** as the **root** user by running the following command:
```
vi /etc/security/limits.conf
```

b. Add the following content to the end of the **limits.conf** file:
```
* soft nproc 65535
* hard nproc 65535
* soft nofile 65535
* hard nofile 65535
```

> NOTE
>
> The asterisk (*) indicates all users. You are advised to replace it with the DDK installation user name.

c. Run the **:wq!** command to save the file and exit.

(Optional) Configuring Permissions of the DDK Installation User

When a non-**root** user is used for installation, perform the following operations. Skip the following operations if the **root** user is used for installation.

Before installing the DDK, you need to download the dependencies, which requires permissions such as **sudo yum**. Perform the following operations as the **root** user:

1. Run the following commands to open the **/etc/sudoers** file:
```
chmod u+w /etc/sudoers
vi /etc/sudoers
```
2. Under **root ALL=(ALL:ALL) ALL** in the file, add the following content:
```
username ALL=(ALL:ALL) NOPASSWD:SETENV: /usr/bin/yum
```
Replace **username** with the name of the common user who executes the installation script.
3. Run **:wq!** to save the file.
4. Run the following command to revoke the write permission on the **/etc/sudoers** file:
```
chmod u-w /etc/sudoers
```


> **NOTE**
>
> Ensure that the last line of the **/etc/sudoers** file is **#includedir /etc/sudoers.d**. Otherwise, add it manually.

Configuring the Local Source

The EulerOS image has minor version differences. To avoid compatibility differences, you are advised to use the local image source for installation.

Perform the following steps as the **root** user:

Step 1 Upload **EulerOS-V2.0SP8-aarch64-dvd.iso** to the server and run the following command to mount the image:

Command: **mount -o loop** *Euler image.iso Image mount directory*

Example: **mount -o loop EulerOS-V2.0SP8-aarch64-dvd.iso /mnt**

Step 2 Go to the **/etc/yum.repos.d** directory and back up the repo file.

Command: **cp** *Source management file.repo Source management file.repo.bak*

Example: **cp euleros_aarch64.repo euleros_aarch64.repo.bak**

Step 3 Replace the source management file.

Command: **vim** *Source management file .repo*

Example: **vim euleros_aarch64.repo**

```
[base]
name=EulerOS-2.0SP8 base
baseurl=file:///mnt
enabled=1
gpgcheck=0
```

> **NOTE**
>
> If the image mount directory is not **/mnt**, replace the value of **baseurl**.

Step 4 Run the following commands to update the software information:

```
yum clean all
yum makecache
```

----End

Installing Dependency

Run the **su -** ***username*** command to switch to the DDK installation user and perform the following operations to install components such as Python on which the DDK depends:

Step 1 Check whether Python 2 and Python 3 are installed.

The DDK installation depends on Python 2 and Python 3. Run the **python -V** and **python3 -V** commands to check whether Python 2 and Python 3 are installed.

If the following information is displayed, the software has been installed: Otherwise, go to **Step 2** to install Python 2 and Python 3.


```
python 2.7.15
python 3.7.0
```

Step 2 Install Python 2 and Python 3.

1. Check whether Python 2 and Python 3 are available in the source list.
   ```
   sudo yum list "*python2*"
   sudo yum list "*python3*"
   ```
2. If the source list contains the **python2.aarch64** and **python3.aarch64** images (the image names may vary depending on the image sources), run the following command to install Python 2 and Python 3:

   Command: **sudo yum install** *Image name*

   Example: **sudo yum install python3.aarch64**

Step 3 Install the dependencies of Python 2 and Python 3.

1. Run the following commands to search for installed NumPy and decorator libraries:
   ```
   rpm -qa | grep numpy
   rpm -qa | grep decorator
   ```
   If the following information is displayed, the NumPy and decorator libraries for Python 2 and Python 3 have been installed: If the NumPy and Decorator libraries are not installed, perform **Step 3.2** to **Step 3.3** to install the required NumPy and decorator libraries.
   ```
   python3-decorator-4.3.0-1.eulerosv2r8.noarch
   python3-numpy-1.15.1-1.eulerosv2r8.aarch64
   python2-decorator-4.3.0-1.eulerosv2r8.noarch
   python2-numpy-1.15.1-1.eulerosv2r8.aarch64
   ```
2. Search for available NumPy and decorator libraries.
   ```
   sudo yum list "*numpy*"
   sudo yum list "*decorator*"
   ```
   If the following information is displayed, the NumPy and decorator libraries are available:
   ```
   python2-numpy.aarch64
   python2-decorator.noarch
   python3-numpy.aarch64
   python3-decorator.noarch
   ```

   NOTE

   If the NumPy library for Python 3 cannot be found, contact Huawei technical support to obtain the EulerOS image package whose kernel version is 4.19.36-vhulk1907.1.0.h529.eulerosv2r8.aarch64 or later. Perform operations in **Configuring the Local Source** again.
3. Run the following commands to install the required NumPy and decorator libraries for Python 2 and Python 3:

   Command: **sudo yum install** *NumPy dependency library*

   Example: **sudo yum install python3-numpy.aarch64**

   Command: **sudo yum install** *decorator dependency library*

   Example: **sudo yum install python3-decorator.noarch**

----End

(Optional) Restoring the Source Configuration

If you do not need to use the local source, perform the following steps to restore the source management file:

Step 1 Go to the **/etc/yum.repos.d** directory and restore the repo file.

cp -f euleros_aarch64.repo.bak euleros_aarch64.repo

Step 2 Unmount the image file.

Command: **umount** *Mount directory*

Example: **umount /mnt**

----End

5 Installing the DDK

This section describes the installation method of the DDK and common operations.

This section describes how to install and use the DDK.

5.1 Installation

Prerequisites

The operations required in **4 Environment Preparation** and **3 Verifying the Software Package Integrity** have been completed.

Procedure

Switch to the DDK installation user and perform the following operations:

Step 1 Run the following command to decompress the installation package:

```
tar -zxvf MSpore_DDK****tar.gz
```

In actual operations, replace *MSpore_DDK****.tar.gz.asc* with the actual installation package. For details about the decompressed files, see **Table A-1**.

Step 2 Run the following command to install the DDK:

```
bash install.sh DDK installation directory
```

The DDK installation directory is automatically created during the installation. For example, if the installation directory is set to **$HOME/tools/che/ddk**, the **tools/che/ddk** directory is automatically created during the installation. Alternatively, you can specify an installation path.

If the message "Successfully installed the DDK!" is displayed, the DDK has been installed successfully.

NOTE

- If the message "Warning: DDK functions except TE have been successfully installed." is displayed during the installation, rectify the fault by referring to **6.1 What Do I Do If the Message "Warning: DDK functions except TE have been successfully installed." Is Displayed During DDK Installation?**.
- After the DDK is installed, the profiling function under the installation path **$HOME/tools/che/ddk/ddk/toolchains/profiler** is unavailable. If you need to use the profiling function, install Mind Studio and the DDK by referring to *Ascend 310 Mind Studio Installation Guide* (select an installation guide based on the Mind Studio server).

----End

5.2 Common Operations

This section describes how to install and use the DDK.

5.2.1 Uninstalling the DDK

To update the DDK version, uninstall the current version and then install a new version according to the installation procedure described in 5.1. This section describes how to uninstall the DDK.

Go to the DDK installation path, for example, *$HOME/tools/che/ddk*/**ddk/scripts**. Run the following command as the DDK installation user to uninstall the DDK:

```
./uninstall.sh
```

If the following information is displayed, the DDK is uninstalled successfully:

```
Info: ide_daemon pem uninstall succ
Starting to remove ddk dir.
```

5.2.2 Querying the DDK Version

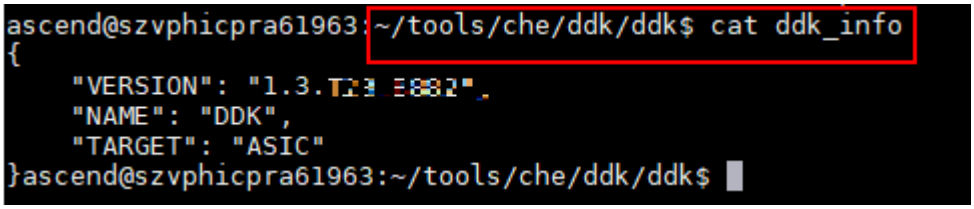
After the DDK is installed, you can view the DDK version number in the DDK installation directory.

For example, if the installation directory of the DDK is *$HOME/tools/che/ddk*, run the following command in the *$HOME/tools/che/ddk* directory to view the DDK version number:

```
cat ddk_info
```

Figure 5-1 shows the return result.

Figure 5-1 Querying the DDK version



In **Figure 5-1**:

- **VERSION**: Indicates the version number of the DDK. The query result varies according to the actual situations.
- **TARGET**: Indicates the running environment of the DDK, which is **ASIC** or **Atlas DK**.

6 FAQs

6.1 What Do I Do If the Message "Warning: DDK functions except TE have been successfully installed." Is Displayed During DDK Installation?

Symptom

After the DDK installation script is executed, messages such as "error: install te-0.4.0.egg failed" and "error: install topi-0.4.0.egg failed" are displayed during the installation. However, the message "Warning: DDK functions except TE have been successfully installed." is displayed when the installation is complete. The DDK has been successfully installed, but TE functions are unavailable.

Possible Cause

The **root** user adopts an earlier version (0.9.8) of setuptools. When the **root** user is switched to the DDK installation user to install the DDK, the system still reads the setuptools path of the **root** user. As a result, setuptools used by the DDK installation user does not take effect.

Solution

Run the following command as the DDK installation user to set the environment variable for the setuptools run by the DDK installation user to take effect, and then reinstall the DDK:

```
export PATH=$HOME/bin:$HOME/.local/bin:$PATH
```


A Overview of Software Packages

For details about the decompressed DDK installation package, see **Table A-1**.

Table A-1 Overview of the software packages

Installation Package	Content	Description
MSpore_DDK-{version}-<uihost arch.os>-<host arch.os>-<device arch.os>.tar.gz For details about the parameter description, see **Table A-2**.	ddk.tar.gz	DDK installation package
	install.sh	Installation script
	check_sha.sh	Script used to verify the integrity of the preceding files. This script is automatically called to perform integrity check during the execution of **install.sh**.

Table A-2 Naming rules of the DDK installation package

Parameter	Description
{version}	Version number
<uihost arch.os>	CPU architecture, OS, and version on the UI host side, for example, x86_64.ubuntu16.04 and x86_64.centOS7.4
<host arch.os>	CPU architecture, OS, and version on the host side, for example, x86_64.ubuntu16.04 and x86_64.centOS7.4
<device arch.os>	CPU architecture, OS, and version on the device side, for example, aarch64.ubuntu16.04 and aarch64.miniOS • Device-side architecture of the DK form (Atlas 200 DK), for example, aarch64.ubuntu16.04 • Device-side architecture of the non-DK form (Atlas 300), for example, aarch64.miniOS

B Open Source Third-Party Libraries

cereal

cereal is a header-only BSD-licensed C++11 serialization library. It is designed to be fast, light-weight, and easy to extend. cereal takes arbitrary data types and reversibly turns them into different representations, such as compact binary encodings, XML, or JSON. The version currently used is 1.2.2.

For details, visit the cereal official website at http://uscilab.github.io/cereal/.

gflags

Google GFlags (GFlags), the Global Flags Editor, contains a Python-friendly C++ library that implements command-line flags processing, replacing systems like **getopt()**. The version currently used is 2.2.1.

For details, visit the GFlags official website at https://github.com/gflags/gflags.

glog

Google glog is a library that implements application-level logging. This library provides logging APIs based on C++-style streams and various helper macros. Similar to assert defined in the standard C library, it provides more output information and flexibility.

For details, visit the glog official website at https://github.com/google/glog.

opencv

OpenCV (short for Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. This cross-platform library sets its focus on real-time image processing, computer vision, and pattern recognition programs. The version currently used is 3.4.2.

For details, visit the OpenCV official website at https://opencv.org/.

Protobuf

Protobuf (short for Protocol buffers) are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data. It is useful in

developing programs to communicate with each other over a wire or for storing data. The version currently used is 3.5.1.

For details, visit the official website at https://developers.google.com/protocol-buffers/.

C Change History

Release Date	Description
2020-05-30	This issue is the first official release.

