Atlas 300 AI Accelerator Card
1.0.0

DDK Installation Guide (Ubuntu, ARM) (Model 3000)

Issue 02

Date 2020-03-30

HUAWEI

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 installation method and FAQs for the device development kit (DDK).

It provides users with an NPU-based digital developer kit, and 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 Environment Preparation

To ensure that the DDK can be used normally, preparations must be made by referring to this section.

2.1 Preparing the Server Environment

The server provides an installation environment for the device development kit (DDK). The following requirements on the hardware and operating system (OS) must be met.

Table 2-1 Ubuntu system version information

Category	Version Restriction	Obtaining Method	Precautions
OS	18.04.1	Download the software of the corresponding version from **http://old-releases.ubuntu.com/releases/** and install the software. You can download the server version **ubuntu-18.04.1-server-arm64.iso**.	-
Python	Python 2: 2.7+ Python 3: 3.5+	For details, see **2.4 Installing Dependency**.	-

2.2 Setting a Common User

Background

It is recommended that installation be performed by the common user.

Procedure

If no common user exists, create one as the **root** user as follows:

1. Run the following command to create a common user and set the **$HOME** directory of the common user:
   ```
   useradd -d /home/username -m username
   ```
2. Run the following command to set the password:
   ```
   passwd username
   ```
3. Run the following command to access the **/home** directory:
   ```
   chmod 750 /home/username
   ```

NOTE

- Replace ***username*** with the common user who installs Mind Studio.
- The **umask** value of a common user cannot be greater than **0027**.
  - You can view the **umask** value by running the **umask** command.
  - You can change the **umask** value by running the **umask** ***New value*** command.

2.3 Setting the Source

Prerequisites

The server with the DDK installed is normally connected to the network.

Procedure

Step 1 Switch to the **root** user and set the **sudo apt-get** permission for the common user.

1. Run the following commands to open the **/etc/sudoers** file:
   ```
   su root
   chmod u+w /etc/sudoers
   vi /etc/sudoers
   ```
2. Add the following content below **# User privilege specification** of the file:
   ```
   username ALL=(ALL:ALL)  NOPASSWD:SETENV:/usr/bin/apt-get
   ```
   Replace **username** with the name of the common user who executes the installation script.

   NOTE

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

Step 2 Configure the Ubuntu18.04 source, which is in the **/etc/apt/sources.list** file.

Step 3 Run the **su** *username* command to switch to the common user.

Step 4 Run the **sudo apt-get update** command to check whether the source is available. If the dependency cannot be downloaded based on the source in the **/etc/apt/sources.list** file, you are advised to configure an available proxy or search for an available source.

----End

2.4 Installing Dependency

Switch to the common user and perform the following operations:

Step 1 Run the following command to install the DDK dependency:

```
sudo -E apt-get install gcc g++ cmake make python-pip python3-pip python3 python
```

Step 2 Install the Python development environment.

1. Install the Python 2 environment.
   - Environment deployment depends on pip. Run the following commands to install the dependencies (numpy and decorator) of the TE software package respectively:
     ```
     $ pip2 install numpy==1.16.0 --user
     $ pip2 install decorator --user
     ```
   - Check the Python 2 environment.

     Run the following commands to check Python and pip, respectively:
     ```
     $ python2 --version
     $ pip2 --version
     ```
     If the following information is displayed respectively for the preceding three commands, the installation is successful:
     ```
     Python 2.7.5
     pip 9.0.1 from /usr/lib/python2.7/dist-packages (python 2.7)
     ```
2. Install the Python 3 (3.5+) environment.
   - Environment deployment depends on pip3. Run the following commands to install the dependencies (numpy and decorator) of the TE software package respectively:
     ```
     $ pip3 install numpy==1.16.0 --user
     $ pip3 install decorator --user
     ```
   - Check the Python 3 (3.5+) environment.

     Run the following commands to check the Python 3 and pip3, respectively:
     ```
     $ python3 --version
     $ pip3 --version
     ```
     If the following information is displayed respectively for the preceding three commands, the installation is successful:
     ```
     Python 3.5.2
     pip 9.0.1 from /usr/lib/python3/dist-packages (python 3.5)
     ```
3. Switch to the **root** user and run the following commands to install the setuptools dependency of Python 2 and Python 3:


```
$ pip2 install setuptools==41.0.1
$ pip3 install setuptools==41.0.1
```

----End

NOTE

- If the pip installation is abnormal during the dependency installation, rectify the fault by referring to **6.1 What Do I Do If a Message Is Displayed Indicating pip2 or pip Unavailability During Mind Studio or DDK Installation?**.
- The version numbers in the command outputs are only examples, which are subject to the actual situations.

3 Preparing Software Packages

Before installing the tool, prepare the following software packages:

DDK installation package: Visit **https://support.huawei.com/enterprise**, enter 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 3-1 describes the software packages.

Table 3-1 Overview of the software packages

Installation Package	Content	Application Scenario	Integrity Verification File for the Software Package
MSpore_DDK-{version}-<uihost arch.os>-<host arch.os>-<device arch.os>.tar.gz For details about the parameter description, see **Table 3-2**.	DDK installation package In this example, the DDK installation package contains the following packages: DDK installation package for non-DK scenarios (such as Atlas 300): **MSpore_DDK-{version}-aarch64.ubuntu18.04-aarch64.ubuntu18.04-aarch64.miniOS.tar.gz**		**MSpore_DDK-{version}-<uihost arch.os>-<host arch.os>-<device arch.os>.tar.gz.asc**
	ddk.tar.gz	DDK installation package	
	install.sh	Installation script	

Installation Package	Content	Application Scenario	Integrity Verification File for the Software Package
	check_sha.sh	Used to verify the integrity of the preceding two files. This script is automatically called to perform integrity check during the execution of **install.sh**.	

Table 3-2 describes the naming rules of the DDK installation package.

Table 3-2 Naming rules of the DDK installation package

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

4 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**.

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 Procedure

Prerequisites

The operations required in **2 Environment Preparation** and **3 Preparing Software Packages** are completed.

Procedure

Switch to the common user and perform the following operations:

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

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

NOTE

In actual operations, *MSpore_DDK****.tar.gz.asc* must be replaced with the actual installation package. For details about the decompressed files, see **Table 3-1**.

Step 2 Run the following command to install the DDK:

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

NOTE

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 installation is successful.

----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 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

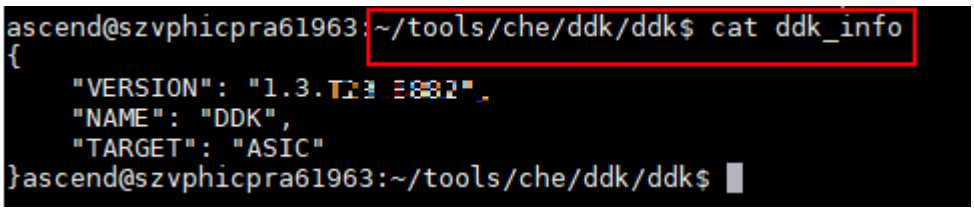
After the DDK is installed, you can view the DDK version number in the DDK installation directory.

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 a Message Is Displayed Indicating pip2 or pip Unavailability During Mind Studio or DDK Installation?

Symptom

During the installation of Mind Studio or a DDK, the system displays a message indicating that the pip2 or pip is unavailable and exits the installation, as shown in **Figure 6-1** and **Figure 6-2**.

Figure 6-1 Message indicating pip2 unavailability

```
Python 2.7.12
Traceback (most recent call last):
  File "/usr/bin/pip2", line 9, in <module>
    load_entry_point('pip==8.1.1', 'console_scripts', 'pip2')()
  File "/usr/local/lib/python2.7/dist-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 318, in load_entry_point
  File "/usr/local/lib/python2.7/dist-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 2221, in load_entry_point
  File "/usr/local/lib/python2.7/dist-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 1954, in load
  File "/usr/lib/python2.7/dist-packages/pip/__init__.py", line 18, in <module>
    from pip.commands import get_summaries, get_similar_commands
  File "/usr/lib/python2.7/dist-packages/pip/commands/__init__.py", line 8, in <module>
    from pip.commands.freeze import FreezeCommand
  File "/usr/lib/python2.7/dist-packages/pip/commands/freeze.py", line 8, in <module>
    from pip.operations.freeze import freeze
  File "/usr/lib/python2.7/dist-packages/pip/operations/freeze.py", line 11, in <module>
    from pip._vendor.pkg_resources import RequirementParseError
ImportError: cannot import name RequirementParseError
### Your pip2 has problem, please solve it before install MindStudio, Exit 1
~
~
```

Figure 6-2 Message indicating pip unavailability

```
Traceback (most recent call last):
  File "/usr/bin/pip", line 9, in <module>
    from pip import main
ImportError: cannot import name main
```


Possible Cause

pip2 is not updated during pip re-installation.

Solution 1

Step 1 Run the **su root** command to switch to the root user and run the **pip list** command. If no error message is displayed, the pip is available. If an error message is displayed after the **pip2 list** command is executed, the pip2 is unavailable.

Step 2 Run the **rm /usr/bin/pip2** command as the root user to delete the pip2.

Step 3 Run the **ln -s pip pip2** command to soft link the pip2 to pip.

Step 4 Run the **pip2 list** command again. If no error message is displayed, it indicates that the fault has been rectified.

If the pip and pip2 are still unavailable, see **Solution 2**.

----End

Solution 2

If the pip installation is abnormal during the dependency installation, run the following commands in sequence:

```
sudo apt-get remove python-pip python3-pip
wget https://bootstrap.pypa.io/get-pip.py
python get-pip.py --user
python3 get-pip.py  --user
```


A Change History

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

