

PLEORA TECHNOLOGIES INC.



Docker Support for eBUS SDK User Guide



Table of Contents

1	About this Guide.....	3
1.1	Documented Product Version.....	3
2	Installing the eBUS SDK.....	5
2.1	System Requirements.....	5
3	Installation and Run Procedure (Linux host – x86_64 and aarch64).....	6

Document Number and Revisions

Document ID: MN-0064

Revision	Reason for Revision	Date
2.0	Final release for eBUS SDK 6.5.1	June 2024
1.0	Initial release for eBUS SDK 6.5.0	May 2024

1 About this Guide

This guide provides instruction on how to get started with using eBUS SDK in a Docker environment. It also provides basic instruction on how to set up Docker on a Ubuntu-based system.

1.1 Documented Product Version

This guide covers Release 6.5.1 of the eBUS SDK. The features and functionality documented in this guide may vary if you are using an earlier or later version of the eBUS SDK.

Related Documents

eBUS SDK Related Documents

Guides are complemented by the following Pleora Technologies documents, which are available on the Pleora Technologies Support Center (supportcenter.pleora.com):

- [eBUS SDK C++ API Quick Start Guide](#)
This guide provides you with the information you need to install the eBUS SDK (which lets you use the eBUS SDK C++ API) and an overview of the system requirements.
- [Docker Support for eBUS SDK User Guide](#)
This guide provides instruction on how to get started with using eBUS SDK in a Docker environment. It also provides basic instruction on how to set up Docker on a Ubuntu-based system.
- [Getting Started with eBUS Edge](#)
The eBUS Edge API (introduced in eBUS SDK 6.0) allows developers to create a software-based GigE Vision transmitter device. eBUS Edge is fully compliant with GigE Vision and GenICam and will work with any GigE Vision and GenICam compliant third-party image processing systems or software.
- [eBUS Edge on OpenSTLinux \(ST Micro MP2 Platform\) Quick Start Guide](#)
This guide provides you with the steps to use the eBUS Edge on the ST Micro MP2 Platform running OpenSTLinux. This guide is intended for novice Linux users, although advanced Linux users may be interested in some of the eBUS SDK-specific elements of this guide for cross-compiling eBUS Edge application using eBUS SDK.
- [eBUS SDK Licensing](#)
This knowledge base article explains the eBUS SDK license structure, explains how to obtain a license, and provides activation instructions. If you are experiencing difficulty activating your license, please review the troubleshooting steps at the end of this publication.
- [eBUS SDK for Linux Quick Start Guide](#)
This guide provides you with the steps to use the eBUS SDK on the Linux operating system, on a Linux x86_64, or ARM platform. This guide is intended for novice Linux users, although advanced Linux users may be interested in some of the eBUS SDK-specific elements of this guide.

- [eBUS SDK .NET API Quick Start Guide](#)

This guide provides you with instructions for compiling and using the sample code are provided, along with an overview of the basic calls that can be used to build custom applications using the eBUS SDK .NET API.

- [eBUS Player User Guide](#)

This guide provides in-depth details about setting up and using the eBUS Player software application to control your GigE Vision or USB3 Vision compliant video transmitters (cameras) and receivers.

- [eBUS Player Quick Start Guide](#)

This quick start guide provides you with the information you need to start using the eBUS Player application, which lets you control the parameters of your GigE Vision or USB3 Vision compliant device and lets you view imaging video and data.

- [eBUS SDK Python API Quick Start Guide](#)

This guide provides you with the information you need to install the eBUS SDK (which lets you use the eBUS SDK Python API) and an overview of the system requirements.

- [eBUS SDK on Raspberry PI 4/5 Quick Start Guide](#)

This guide provides you with the steps to use the eBUS SDK on a Raspberry PI 4/5. This guide is intended for novice Raspberry PI 4/5 users, although advanced Raspberry PI 4/5 users may be interested in some of the eBUS SDK-specific elements of this guide.

- [Supported Software Ecosystem](#)

This support summary provides an overview of the supported protocols, operating systems, ARM platforms, development environments, and drivers. Pleora is currently supporting eBUS SDK 5.0 and later.

2 Installing the eBUS SDK

2.1 System Requirements

The following SW packages are required (please visit Downloads (<https://supportcenter.pleora.com/s/downloads>):


- eBUS SDK/Runtime package (see Table 1 below to help select the correct one)
- unpack-ebus-docker_<ver>.sh file

Table 1: Host OS to eBUS Packages Mapping

Host OS	Supported eBUS 6.5.0 Package(s) to be installed in Docker container	eBUS Host Driver to Install
Ubuntu 22.04 x86 64-bit	eBUS SDK package for Ubuntu 22.04 x86 64-bit eBUS runtime packages for Ubuntu 22.04 x86 64-bit	eBUS_Docker_Driver for Ubuntu 22.04 x86 64-bit
Ubuntu 20.04 x86 64-bit		eBUS_Docker_Driver for Ubuntu 20.04 x86 64-bit
Ubuntu 18.04 x86 64-bit		eBUS_Docker_Driver for Ubuntu 18.04 x86 64-bit
Jetpack 5.1.1 (Ubuntu 20.04 arm64)	eBUS SDK package for Jetpack 5.1 arm 64-bit eBUS runtime packages for Jetpack 5.1 arm 64-bit	eBUS_Docker_Driver for Jetpack 5.1 arm 64-bit
Jetpack 4.6 (Ubuntu 18.04 arm64)		eBUS_Docker_Driver for Jetpack 4.6 arm 64-bit
Jetpack 6.0.0 (Ubuntu 22.04 arm64)		eBUS_Docker_Driver for Jetpack 6.0.0 arm 64-bit
Raspberry PI 4B (Running Raspberry Pi OS (64-bit))	eBUS SDK package for Raspberry PI 4B/5B arm 64-bit eBUS runtime packages for Raspberry PI 4B/5B arm 64-bit	eBUS_Docker_Driver for Raspberry Pi 4_Pi5 arm 64-bit
Raspberry PI 5B (Running Raspberry Pi OS (64-bit))		eBUS_Docker_Driver for Raspberry Pi 4_Pi5 arm 64-bit

3 Installation and Run Procedure (Linux host – x86_64 and aarch64)

Setup Process

 Before you can install Docker Engine, you need to uninstall any conflicting packages. Instruction can be found below in "Uninstall old versions of Docker Engine".

1. The first step is to install the **Docker Engine** on the host machine. Instruction can vary slightly between different linux-based platforms.

The following instructions are for Linux distributions enumerated in Table 1.

- a. Install Docker on Ubuntu 22.04/20.04/18.04 x86 64-bit
 - i. Add the Docker repos

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /
etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/
keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

- ii. Install Docker apps

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-
buildx-plugin docker-compose-plugin
```

- iii. Add user to docker group

```
sudo gpasswd -a $USER docker
```

- iv. Restart the machine
- v. (Optional) Verify Docker is working after the restart

```
docker run hello-world
```

```
[...]

Hello from Docker!
This message shows that your installation appears to be working
correctly
[...]
```

- b. Install Docker on Jetpack 6.0 (Ubuntu 22.04 arm64), Jetpack 5.1.x (Ubuntu 20.04 arm64) and Jetpack 4.6 (Ubuntu 18.04 arm64)
 - i. Add user to docker group

```
sudo gpasswd -a $USER docker
```

- ii. Restart the machine
 - iii. (Optional) Verify Docker is working after the restart

```
docker run hello-world
[...]

Hello from Docker!
This message shows that your installation appears to be working
correctly
[...]
```

- c. Install Docker on Raspberry Pi OS (64-bit) on Raspberry Pi 4b and 5b
 - i. Add the Docker Repos

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /
etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/
keyrings/docker.asc] https://download.docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

- ii. Install the Docker Apps

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-
buildx-plugin docker-compose-plugin
```

- iii. Add user to the Docker Group



```
sudo gpasswd -a $USER docker
```

- iv. Restart the system
- v. (Optional) Verify Docker is working after the restart

```
docker run hello-world
[...]

Hello from Docker!
This message shows that your installation appears to be working
correctly
[...]
```

2. Goto Pleora's support center and download the following packages (onto the Host machine):
 - a. **eBUS Docker Driver** Debian Package for your host (see Table 1 above for the correct install package)
 - b. Appropriate **eBUS 6.5.0 and later** Debian Package (see Table 1 above for the correct install package)
 - c. `unpack_ebus_docker.sh` shell script
3. Install the **eBUS Docker Driver** Debian Package
 - a. `sudo dpkg -i <path to eBUS Docker Driver>.deb`
4. Run the script `unpack_ebus_docker_<ver>.sh`. The first time it runs you will be prompted to put files in the `preparation` folder.
 - a. Put the **eBUS 6.5.0 and later** Debian Package in the `preparation /install/` folder.
 - b. Optionally, place your **License File** (ending in `.lic`) in the `preparation /install/` folder.
 - c. If you have any precompiled eBUS apps or python scripts, place them in `preparation /app`
5. Re-run the script `unpack_ebus_docker_<ver>.sh` after placing the files in Step 4. If the required files were found, the script will create a folder `ebus-sdk-docker` and copy the files into the container automatically.
 - When the process completes, you are put in a root shell. You can run your apps in the `/app` directory, or launch eBUS Player from `/opt/pleora/.../bin/eBUSPlayer`
 - You can also build and test the samples that come with eBUS SDK.
 - When running a python script that depends on eBUS import, launch it via `/opt/pleora/<...>/bin/RunFromEnv.sh /usr/bin/python3 <your_py_script.py>`
6. If you need to relaunch the container, navigate to the `ebus-sdk-docker` created in Step #5 and run `setup.sh`

 More information on installing Docker Engine can be found at: <https://docs.docker.com/engine/install/ubuntu/>

Uninstall old versions of Docker Engine

Distro maintainers provide unofficial distributions of Docker packages in APT.

You must uninstall these packages before you can install the official version of Docker Engine.

The unofficial packages to uninstall are:

- `docker.io`
- `docker-compose`
- `docker-compose-v2`
- `docker-doc`
- `podman-docker`

Moreover, Docker Engine depends on `containerd` and `runc`. Docker Engine bundles these dependencies as one bundle: `containerd.io`¹.

If you have installed the `containerd` or `runc` previously, uninstall them to avoid conflicts with the versions bundled with Docker Engine.

Run the following command to uninstall all conflicting packages:

```
$ for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker
containerd runc; do sudo apt-get remove $pkg; done
```

`apt-get` might report that you have none of these packages installed.

Images, containers, volumes, and networks stored in `/var/lib/docker/` aren't automatically removed when you uninstall Docker.

If you want to start with a clean installation, and prefer to clean up any existing data preform the following steps

1. Uninstall the Docker Engine, CLI, containerd, and Docker Compose packages:

```
$ sudo apt-get purge docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin docker-ce-rootless-extras
```

2. Images, containers, volumes, or custom configuration files on your host aren't automatically removed. To delete all images, containers, and volumes:

```
$ sudo rm -rf /var/lib/docker
$ sudo rm -rf /var/lib/containerd
```

You have to delete any edited configuration files manually.

¹ <http://containerd.io>

Technical Support

On the Pleora Support Center, you can:

- Download the latest software and firmware.
- Log a support issue.
- View documentation for current and past releases.
- Browse for solutions to problems other customers have encountered.
- Read knowledge base articles for information about common tasks.

To visit the Pleora Support Center:

- Go to supportcenter.pleora.com.
- Most material is available without logging in to a Support Center account.
- To access software and firmware downloads, in addition to other content, log in to the Support Center.
- If you do not have an account, click Request Account.
- Accounts are usually validated within one business day.

Copyright Information

Copyright © 2024 Pleora Technologies Inc.

These products are not intended for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Pleora Technologies Inc. (Pleora) customers using or selling these products for use in such applications do so at their own risk and agree to indemnify Pleora for any damages resulting from such improper use or sale.

Trademarks

VIVOEPlayer, PureGEV, eBUS, iPORT, vDisplay, AutoGEV, AutoGen, AI Gateway, eBUS Studio, Vaira and all product logos are trademarks of Pleora Technologies. Third party copyrights and trademarks are the property of their respective owners.

Notice of Rights

All information provided in this manual is believed to be accurate and reliable. No responsibility is assumed by Pleora for its use. Pleora reserves the right to make changes to this information without notice. Redistribution of this manual in whole or in part, by any means, is prohibited without obtaining prior permission from Pleora.

Document ID: MN-0064