

PLEORA TECHNOLOGIES INC.



## eBUS SDK 6.x - Raspberry PI 4/5 Quick Start Guide



# Table of Contents

<b>1</b>	<b>About this Guide.....</b>	<b>3</b>
1.1	About the eBUS SDK for Raspberry PI 4/5.....	3
1.2	What this Guide Provides.....	3
1.3	Documented Product Version.....	4
1.4	Related Documents .....	4
1.5	System Requirements .....	5
<b>2</b>	<b>Installing eBUS SDK on Raspberry PI 4/5 .....</b>	<b>6</b>
<b>3</b>	<b>Installing the eBUS Runtime for Raspberry PI 4/5 .....</b>	<b>8</b>
3.1	Uninstalling the eBUS SDK (and eBUS Runtime) for Raspberry PI 4/5. ....	9
<b>4</b>	<b>Activating eBUS Edge License on Raspberry PI 4/5 .....</b>	<b>10</b>
4.1	Understanding Licensing .....	10
4.2	Activating an eBUS SDK License .....	10
<b>5</b>	<b>Optimizing Operation with GigE Vision Devices .....</b>	<b>12</b>
5.1	Using the eBUS Universal Pro for Ethernet Filter Driver .....	12
5.2	Additional optimization steps.....	13
<b>6</b>	<b>Providing Access to Third-Party USB3 Vision Transmitter Devices .....</b>	<b>14</b>
<b>7</b>	<b>Using eBUS Player to Configure Devices and Stream Images .....</b>	<b>15</b>
<b>8</b>	<b>Compiling and Running Sample Applications.....</b>	<b>16</b>
<b>9</b>	<b>Troubleshooting.....</b>	<b>18</b>

## Document Number and Revisions

Document ID: QSG-0019

Revision	Reason for Revision	Date
2.1	Minor Release with Title modification to include 6.x.	March 2026
2.0	Final Release with eBUS SDK 6.6	November 2025
1.0	Initial release for eBUS SDK 6.5.0	May 2024

# 1 About this Guide

This chapter describes the purpose and scope of this guide, and provides a list of complementary guides.

## 1.1 About the eBUS SDK for Raspberry PI 4/5

eBUS SDK is built to receive video over GigE, 10 GigE, and USB 3.0 that is portable across Windows, Linux and macOS operating systems. With an eBUS SDK Seat License, designers can develop production-ready software applications in the same environment as their end-users, and quickly and easily modify applications for different media, while avoiding supporting multiple APIs from various vendors. Compared to camera vendor provided SDKs, eBUS frees developers from being tied to a specific camera, and instead they can choose the device that is best for the application.



### **eBUS Edge for Sensor Devices**

eBUS Edge is a software implementation of a full device level GigE Vision transmitter, without requiring any additional hardware. Adding eBUS Edge to a CPU's software stack turns it into a fully compliant GigE Vision device that supports image transmission and enables the device to respond to control requests from a host controller. eBUS Edge is GigE Vision and GenICam compliant, meaning end-users can use any standards-compliant third-party image processing system. eBUS Edge currently supports the GigE Vision standard.

### **eBUS Receive for Host Applications**

eBUS Receive manages high-speed reception of images or data into buffers for hand-off to the end application for further analysis. Developers can write applications that run on a host computer to seamlessly control and configure an unlimited number of GigE Vision or USB3 Vision and GenICam compliant sensors.

The eBUS Universal Pro driver reduces CPU usage when receiving images or data, leaving more processing power for analysis and inspection applications while helping meet latency and throughput requirements for real-time applications. The eBUS Universal Pro driver is easily integrated into third-party processing software to bring performance advantages to end-user applications.

## 1.2 What this Guide Provides

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.

## 1.3 Documented Product Version

This guide covers Release 6.5 of the eBUS SDK. The features and functionality documented in this guide may vary if you are using a later version of eBUS SDK and any previous versions ( < 6.5 ) are not supported.

## 1.4 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](https://supportcenter.pleora.com)):

- [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 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 7.0) 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 SDK 7.0 \(and later\) 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.
- [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.
- [.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.
- [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.
- [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 7.0 and later.

## 1.5 System Requirements

**Ensure the computer on which you install the eBUS SDK meets the following recommended requirements:**

- Supported operating systems on Raspberry PI4/5:
  - Raspberry Pi OS (64 bits) ( Debian aarch64 GNU/Linux 12 bookworm )



Ensure that you uninstall any previous versions of eBUS SDK from your machine before installing eBUS SDK 6.x.

## 2 Installing eBUS SDK on Raspberry PI 4/5



**Qt5** is required before running eBUS Player for the first time.

Use the installation package to install the eBUS SDK and eBUS Runtime for Raspberry PI 4/5.

### To install the eBUS SDK

1. Install Qt5 on your system, ensure that you have the correct version.
  - Execute the command:
    - `sudo apt-get install qtbase5-dev qt5-qmake`
2. Copy the eBUS SDK installation package to the Raspberry PI. The installation package is available for download at [supportcenter.pleora.com](https://supportcenter.pleora.com).
3. From the terminal, execute the following command.
  - `sudo dpkg -i eBUS_SDK_Raspberry_Pi4_Pi5_linux-aarch64-arm-<6.x.x>.deb`
4. We recommend that you reboot the Raspberry PI to ensure that the correct environment variables are set at The eBUS SDK is installed in the following directory:  
`/opt/pleora/ebus_sdk/linux-aarch64-arm/`



The following steps are only needed if the eBUS SDK application requires MP4 video saving functionality.

5. On Raspberry PI:
  - `sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev`
6. Enabling MP4 Video Saving in eBUS Player sample
  - Navigate to `/opt/pleora/ebus_sdk/linux-aarch64-arm/samples`
  - Make a copy of the following directory, as a backup for the original source files:  
`/opt/pleora/ebus_sdk/linux-aarch64-arm/share/samples`
  - Navigate to the directory that contains the copy of the sample code that you have "write"
  - Edit Makefile by adding the following highlighted code in the locations shown below:

```

28 endif
29 CFLAGS += -D_UNIX_ -D_LINUX_ -fPIC -std=c++11
30 CPPFLAGS += -D_UNIX_ -D_LINUX_ -DQT_GUI_LIB -fPIC -std=c++11
31
32 LDFLAGS += -L$(PUREGEV_ROOT)/lib \
33           -lPvAppUtils \
34           -lPtConvertersLib \
35           -lPvBase \
36           -lPvBuffer \
37           -lPvGenICam \
38           -lPvSystem \
39           -lPvStream \
40           -lPvDevice \
41           -lPvTransmitter \
42           -lPvVirtualDevice \
43           -lPvPersistence \
44           -lPvSerial \
45           -lPvCameraBridge
46
47 LDFLAGS += -lswscale \
48           -lavcodec \
49           -lavformat \
50           -lavutil
51
52 CPPFLAGS += -DPV_ENABLE_MP4
53

```

- Navigate to <your sample folder>/eBUSPlayer/
- Type “make” to recompile.



### 3 Installing the eBUS Runtime for Raspberry PI 4/5

#### To install the eBUS Runtime packages

- From the terminal, execute the following command according to the different packages.
  - If all runtime packages are needed:
    - `sudo dpkg -i eBUS_Runtime_Raspberry_Pi4_Pi5_linux-aarch64-arm-<6.x.x>.deb`
  - If only eBUS Edge is needed, both eBUS Base Runtime and eBUS Edge Runtime will need to be installed:
    - `sudo dpkg -i eBUS_Base_Runtime_Raspberry_Pi4_Pi5_linux-aarch64-arm-<6.x.x>.deb`
    - `sudo dpkg -i eBUS_Edge_Runtime_Raspberry_Pi4_Pi5_linux-aarch64-arm-<6.x.x>.deb`
  - If only eBUS Receive is needed, both eBUS Base Runtime and eBUS Receive Runtime will need to be installed:
    - `sudo dpkg -i eBUS_Base_Runtime_Raspberry_Pi4_Pi5_linux-aarch64-arm-<6.x.x>.deb`
    - `sudo dpkg -i eBUS_Receive_Runtime_Raspberry_Pi4_Pi5_linux-aarch64-arm-<6.x.x>.deb`
  - If eBUS Python is needed, install eBUS Base Runtime, eBUS Receive Runtime, eBUS Edge Runtime and eBUS Python Runtime.
    - `sudo dpkg -i eBUS_Base_Runtime_Raspberry_Pi4_Pi5_linux-aarch64-arm-<6.x.x>.deb`
    - `sudo dpkg -i eBUS_Receive_Runtime_Raspberry_Pi4_Pi5_linux-aarch64-arm-<6.x.x>.deb`
    - `sudo dpkg -i eBUS_Edge_Runtime_Raspberry_Pi4_Pi5_linux-aarch64-arm-<6.x.x>.deb`
    - `sudo dpkg -i eBUS_Python_Runtime_Raspberry_Pi4_Pi5_linux-aarch64-arm-<6.x.x>.deb`
    - OR**
    - `sudo dpkg -i eBUS_Runtime_Raspberry_Pi4_Pi5_linux-aarch64-arm-<6.x.x>.deb`
- We recommend that you reboot the Raspberry PI to ensure that the correct environment variables are set at The eBUS SDK is installed in the following directory:  
`/opt/pleora/ebus_sdk/linux-aarch64-arm/`



If there is a failure to compile and install the eBUS Universal Pro driver due to permissions or missing dependencies, you can manually compile and install the driver. For more information, see ["To manually install the eBUS Universal Pro for Ethernet Filter Driver"](#).



If the eBUS SDK installation is not successful, your system may be missing the required Linux kernel headers.

For more information, see ["Error message: Cannot find the files to build kernel module in this PC"](#).

### 3.1 Uninstalling the eBUS SDK (and eBUS Runtime) for Raspberry PI 4/5.

You can use the dpkg command to uninstall the eBUS SDK for Raspberry PI 4/5.

#### To uninstall the eBUS SDK

From the terminal, execute command.

- `sudo dpkg -r ebus-sdk`



To see installed packages, you can execute one of the following commands:

- `dpkg -l | grep ebus`

#### To uninstall eBUS Runtime packages, Release 6.5 and higher

Execute the following commands according to the case:

- if eBUS\_Python\_Runtime is installed:
  - `sudo dpkg -r ebus-python-runtime`
- if eBUS\_Receive\_Runtime is installed:
  - `sudo dpkg -r ebus-receive-runtime`
- if eBUS\_Edge\_Runtime is installed:
  - `sudo dpkg -r ebus-edge-runtime`
- if eBUS\_Base\_Runtime is installed:
  - `sudo dpkg -r ebus-base-runtime`
- if eBUS\_Runtime is installed:
  - `sudo dpkg -r ebus-runtime`

## 4 Activating eBUS Edge License on Raspberry PI 4/5

A license is required to take full advantage of the eBUS Edge's transmit and eBUS SDK's receive third party GigE and USB3 vision devices. When a license is activated, the embossed watermark that appears on transmitted and received images is no longer applied, and restrictions for transmitting and receiving images are removed.

If you use the eBUS SDK without a license, the following limitations apply:

- Received images (received from third-party GigE Vision or USB3 Vision transmitter devices) have an embossed watermark.
- The raw data payload type cannot be received.
- Connections to a software-based GigE Vision device (developed with the eBUS Edge portion of the Pleora eBUS SDK) will disconnect after 15 minutes.
- Certain device information strings cannot be customized by a software developer when creating a software-based GigE Vision device, such as the device's model name.

### 4.1 Understanding Licensing

eBUS Receive and eBUS Edge licenses are associated to the MAC address of a network interface card (NIC). Depending on the type of license you are purchasing, you will need to provide the following information:

- For an **eBUS Receive** or **eBUS SDK Seat** license, provide the MAC address of a NIC in the workstation.
- For an **eBUS Edge** license, provide the MAC address of a network interface in the embedded computer that will run your software-based GigE Vision device.

Pleora includes the MAC address in the license file that you purchase, which allows the eBUS SDK to accept the license. The MAC address is used to identify the workstation or embedded computer.



A pre-programmed USB license dongle for GigE Vision and USB3 Vision devices on Linux operating systems on x86 platforms is supported in eBUS SDK version 6.2 and later. The license dongle allows you to quickly license a system by inserting the dongle in the PC. The license dongle eliminates the need to find a MAC address and deploy a runtime license on a new system when you are using eBUS Receive, thereby minimizing system down time.

The pre-programmed USB license dongle is not supported on Linux (ARM platforms). If you need to use Linux (ARM platforms), install the license file, which is available for purchase online.

### 4.2 Activating an eBUS SDK License

When you activate a license on your workstation or embedded computer, the restrictions are removed.



Please take note of the following important points:

- DO NOT rename the license file provided by your Pleora
- DO NOT disable or remove the NIC (or WiFi adapter) that is associated with the

#### To activate an eBUS SDK license

1. On the Raspberry PI, copy the license file to:  
/opt/pleora/ebus\_sdk/linux-aarch64-arm/licenses
2. Restart any eBUS applications that are currently running (for example, eBUS Player), to ensure that the license takes effect.  
You have completed the license activation.

#### 4.2.1 For More Information

For detailed information about licensing, including troubleshooting tips or using the pre-programmed USB dongle, see the *eBUS SDK Licensing Overview Knowledge Base Article*, available on the Pleora Technologies Support Center at [supportcenter.pleora.com](https://supportcenter.pleora.com).

## 5 Optimizing Operation with GigE Vision Devices

This chapter provides some steps you can take to optimize operation when using GigE Vision devices with the eBUS SDK.

### 5.1 Using the eBUS Universal Pro for Ethernet Filter Driver

The eBUS Universal Pro for Ethernet Filter driver is automatically installed and loaded on your workstation or embedded computer during the installation of the eBUS SDK. This driver optimizes operation with GigE Vision devices. It also enhances the performance of your system by allowing GigE Vision Stream Protocol (GVSP) data to bypass some (or all) of the operating system's network stack, delivering the data directly to the eBUS SDK.

When installing the eBUS SDK, if the eBUS Universal Pro for Ethernet Filter Driver fails to compile and install the eBUS Universal Pro driver due to permissions or missing dependencies, you can manually compile and install the driver.

If you would like to uninstall the filter driver, you can do so. Keep in mind that you can still work with GigE Vision devices after you uninstall the eBUS Universal Pro for Ethernet Filter driver. However, the operation of these devices is not optimized.

#### To manually install the eBUS Universal Pro for Ethernet Filter Driver

1. Navigate to the following directory:  
`/opt/pleora/ebus_sdk/linux-aarch64-arm/module`
2. Execute the following command:  
`sudo ./build.sh`
3. After you compile the driver, install it by executing the following command:  
`sudo ./install_driver.sh --install`

#### To uninstall the eBUS Universal Pro for Ethernet Filter Driver

- Execute the following script and follow the prompts:  
`sudo /opt/pleora/ebus_sdk/linux-aarch64-arm/module/install_driver.sh --uninstall`

#### To start, stop, or check the status of the eBUS Universal Pro for Ethernet Filter Driver

- Execute the following command:  
`sudo /opt/pleora/ebus_sdk/linux-aarch64-arm/module/ebdriverlauncher.sh <command>` (where <command> can be start, stop, or status)



You can also check the status of the driver by executing the following command:  
`lsmod | grep ebUniversalProForEthernet`

## 5.2 Additional optimization steps

For GigE Vision devices, when a driver is installed, further optimization can be achieved by running the `set_socket_buffer_size.sh` script located in `/opt/pleora/ebus_sdk/linux-aarch64-arm/bin`, as `sudo` with the following command:

```
sudo ./set_socket_buffer_size.sh.
```

For further information, see the *eBUS SDK C++ API Quick Start Guide* or the *eBUS SDK Python API Quick Start Guide* for `PvStreamGEV::SetUserModeSocketRxBufferSize` details for Linux virtual NIC support.

## 6 Providing Access to Third-Party USB3 Vision Transmitter Devices

To access third-party, non-Pleora USB3 Vision transmitter devices, you must add the device's vendor ID to the eBUS SDK.



If you are not sure if your device is a Pleora transmitter, observe the USB GUID that appears on the device's label or in your software application. If it begins with the Pleora vendor ID (28b7), it uses Pleora's transmitter technology.

### To set up access for USB3 Vision devices that are not enabled with Pleora's technology

1. Execute the following command (this command works on any Linux platform):  
`./opt/pleora/ebus_sdk/linux-aarch64-arm/bin/set_udev_rules.sh`
2. When prompted enter 0 and/or click enter to add a new rule for a device,
3. When prompted, enter the vendor ID assigned to the device's This number often appears on the device's label.

## 7 Using eBUS Player to Configure Devices and Stream Images

You can use eBUS Player (which is precompiled and installed with the release in the bin folder) to connect to, configure, and stream images from GigE Vision (this includes eBUS Edge) USB3 Vision devices.

You must disable the firewall before using GigE Vision devices. Some optimization may be desirable before using GigE Vision or USB3 Vision devices with the eBUS SDK in a production environment, as was discussed in [“Optimizing Operation with GigE Vision Devices”](#).

**i** Qt is required to run eBUS Player. Relevant Qt version must be installed for your particular OS prior to launching/running eBUS Player.

**i** In the `opt/pleora/linux-aarch64-arm/ebus_sdk/bin` folder, you will find both an eBUSPlayer script and an eBUSPlayer.bin executable. You should always execute the eBUSPlayer script, which sets the proper environment variables, and then automatically executes the eBUSPlayer.bin executable.





## 8 Compiling and Running Sample Applications

We strongly recommend that you copy the sample applications from the `/opt/pleora/ebus_sdk/linux-aarch64-arm/share/` directory to your personal development workspace before modifying or compiling them.

A list of C++ samples, with a description of each, can be found in the `index.html` file in the `/opt/pleora/ebus_sdk/linux-aarch64-arm/share/samples` directory.



Compilation will fail for GUI-based samples if the Qt development package is not installed. For Qt version information, see [“System Requirements”](#).

A list of Python samples can be found in the `/opt/pleora/ebus_sdk/linux-aarch64-arm/share/samples/python/ebus` directory.

For information about building the sample applications and creating your own applications, see the *eBUS SDK C++ API Quick Start Guide*, and the *eBUS SDK Python API Quick Start Guide*.

### To compile the eBUS SDK C++ sample applications

1. If you have not restarted your computer since installing the eBUS SDK, we recommend that you do so. This ensures that the correct environment variables are set at startup.
2. Make a copy of the following directory, as a backup for the original source files:  
`/opt/pleora/ebus_sdk/linux-aarch64-arm/share/samples`
3. Navigate to the directory that contains the copy of the sample code (that was created in Step 2 above).
4. Do one of the following:
  - **To compile all sample applications at one time:** Execute the sh script by typing `./build.sh`
  - **To compile a specific sample application:** Navigate to the directory of the sample (for example, `MulticastMaster`) and execute the make command.

### To use the eBUS SDK Python sample applications

1. If you have not restarted your computer since installing the eBUS SDK, we recommend that you do so. This ensures that the correct environment variables are set at startup.
2. Make a copy of the following directory, as a backup for the original source files:  
`/opt/pleora/ebus_sdk/linux-aarch64-arm/share/samples/python/ebus`
3. Navigate to the directory that contains the copy of the sample code (that was created in Step 2).
4. Launch the Python application by executing the following command:
  - `python3 ./<python_sample>.py`.



If you encounter issues with running the samples due to the environment variables not being set, you can restart your computer or set the environment variables manually, as outlined in [“The sample applications compiled successfully, but they will not run.”](#).

## 9 Troubleshooting

### Cannot detect or connect to GigE Vision devices.

In the **Device Selection** dialog box, select the **Show unreachable Network Devices** check box. If the device still does not appear, do one of the following:

- Disable the firewall.  
- And/or -
- Execute the following script and then choose option **0** (to disable strict Reverse Path Forwarding filtering):  
`sudo /opt/pleora/ebus_sdk/linux-aarch64-arm/bin/set_rp_filter.sh`

### Cannot detect or connect to USB3 Vision devices.

If the device does not use Pleora's transmitter technology and the device's vendor ID has not been added to the eBUS SDK, you will be unable to detect or connect to it. To see if your device uses Pleora transmitter technology, observe the USB GUID that appears on the device's label or in your software application. If it begins with the Pleora vendor ID (28b7), it uses Pleora transmitter technology. For information about accessing the camera, see ["Providing Access to Third-Party USB3 Vision Transmitter Devices"](#).

### A high number of GTK errors or warnings appear when running an application.

It is likely that you are running the sample application as superuser, but are logged on as a standard user.

### The sample applications compiled successfully, but they will not run.

As superuser, execute the `/opt/pleora/ebus_sdk/linux-aarch64-arm/bin/install_libraries.sh` script to ensure that the required libraries have been added to the system. Also, ensure that you source the `bin/set_puregev_env.sh` script to set the environment variables (`source set_puregev_env.sh`).



You can also launch the sample applications using the `RunFromEnv.sh` script. For example:  
`source ./RunFromEnv.sh && ~/samples/PvStreamSample/PvStreamSample`



In this example, it is assumed that a copy of the sample applications is available in your Home directory and that you have "write" access.

### The following error appears: **GENICAM\_V3\_4 is not set.**

Restart the computer. This issue can occur if you did not restart the computer after installing the eBUS SDK. If the issue persists, follow the steps to set the environment variables, as outlined above in ["The sample applications compiled successfully, but they will not run."](#)

### A watermark appears on received images.

You require a receive license. For more information, see ["Activating eBUS SDK Licenses"](#) or the *eBUS SDK Licensing Overview Knowledge Base Article*, available on the Pleora Support Center ([supportcenter.pleora.com](http://supportcenter.pleora.com)).

### Error message: **Cannot find the files to build kernel module in this PC**

This error, which can occur when you are installing the eBUS SDK or eBUS Universal Pro for Ethernet Filter Driver, indicates that the kernel headers are not present on your system. The kernel headers are required to compile one of the Linux modules and they must match the system kernel version.

To install the Linux kernel headers, execute one of the following commands:

- `sudo apt-get install linux-headers-$(uname -r)`

# 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](https://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 © 2026 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: QSG-0019