eBUS SDK for Linux

eBUS SDK Version 6.2

Quick Start Guide

Pleora
Technologies

Table of Contents

Chapter 1

About this Guide

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

The following topics are covered in this chapter:

About the eBUS SDK for Linux

eBUS SDK is built on a single API 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.

What this Guide Provides

This guide provides you with the steps to use the eBUS SDK on the Linux operating system, on a Linux x86, 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.

Documented Product Version

This guide covers Release 6.2 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

The *eBUS SDK for Linux Quick Start Guide* is complemented by the following guides, available on the Pleora Support Center at supportcenter.pleora.com.

- *eBUS Player User Guide*
- *eBUS SDK C++ API Quick Start Guide*
- *Getting Started with eBUS Edge*
- *eBUS SDK Licensing Overview Knowledge Base Article*
- *Configuring Your Computer and Network Adapters for Best Performance Knowledge Base Article*
- eBUS SDK C++ API documentation. Navigate to
  `/opt/pleora/ebus_sdk/<distribution_targeted>/share/doc/sdk/index.html`

Chapter 2

Installing the eBUS SDK for Linux

This chapter provides system requirements and steps that you will use to install the eBUS SDK for Linux.

The following topics are covered in this chapter:

System Requirements

Ensure the workstation or embedded computer meets the following recommended requirements:

- At least one Gigabit Ethernet NIC (if you are using GigE Vision devices) or at least one USB 3 port (if you are using USB3 Vision devices).

  For supported USB 3 host controller chipsets, consult the *eBUS SDK Release Notes*, available on the Pleora Support Center.

- One of the following operating systems:
  - For the x86 Linux platform:
    - Red Hat Enterprise Linux 7 (RHEL), 64-bit
    - CentOS 7, 64-bit
    - Ubuntu 20.04 LTS (64-bit), 18.04 LTS (64-bit), 16.04 (32-bit or 64-bit), and 14.04 (32-bit or 64-bit)

      Note: Ensure you have the Linux kernel header files, which are required to install the eBUS Universal Pro for Ethernet Filter driver. The Linux kernel headers are typically installed on your system when you install RHEL/CentOS or Ubuntu. If they are not present, as indicated by an error message when installing the driver, see "Error message: Cannot find the files to build kernel module in this PC" on page 26.

- For the Linux ARM platform:
  - NVIDIA Jetson Nano, Jetson AGX Xavier, Jetson Xavier NX, and Jetson TX2 platforms (Ubuntu 18.04 with Jetpack 4.6)

Note: Support for the i.MX8 platform (Yocto 3.0 (Kernel version 5.4)) is also available. Contact your Pleora Support Representative for more information.

- For the x86 Linux and Linux ARM platforms, Qt is required to compile GUI-based samples:
  - **For Ubuntu 20.04 Desktop:** Qt 5.12.8
  - **For Ubuntu 18.04 Desktop:** Qt 5.9.5
  - **For Ubuntu 16.04 Desktop:** Qt 5.5.1
  - **For Ubuntu 14.04 Desktop:** Qt 5.2.1
  - **For Ubuntu 18.04 for ARM:** Qt 5.9.5
  - **For RHEL/CentOS7:** Qt 5.9.2

Tip: To install Qt on your system, execute one of the following commands:

- On Ubuntu: `sudo apt-get install qt5-default`
- On RHEL/CentOS: `yum install qt-devel`

For RHEL/CentOS, install Qt as a super user, or you can use the `sudo` command if the sudo group exists.

If Qt is currently installed on your system, ensure that you have the correct version and append the *QT_Install_Path*/bin on the $PATH.

Ensure that you uninstall any previous versions of eBUS SDK from your machine before installing BUS SDK 6.2.

Installing the eBUS SDK for Linux

Use the installation package to install the eBUS SDK for Linux.

To install the eBUS SDK

1. Install one of the following prerequisites, depending on your operating system (if required):
   - For Ubuntu 14.04 (32-bit and 64-bit) execute the command:
     ```
     sudo apt-get install libavcodec54
     ```
   - For Ubuntu 16.04 (32-bit and 64-bit) execute the command:
     ```
     sudo apt-get install ffmpeg
     ```
   - For Ubuntu 18.04 (64-bit) execute the command:
     ```
     sudo apt-get install libavcodec57
     ```
   - For Ubuntu 20.04 (64-bit) execute the command:
     ```
     sudo apt-get install libavcodec58
     ```

- For RHEL and CentOS 7 (64-bit), install FFmpeg on CentOS 7. See <u>How to Install and Use FFMpeg</u> for details.

> After you have installed FFMpeg, disable the firewall. As super user, execute the command: `systemctl disable firewalld`

- For CentOS 7, as super user, execute the command:
  ```
  yum install elftutil-libelf-devel
  ```
- For ARM64 (aarch64) on NVIDIA Jetson platforms, execute the command:
  ```
  sudo apt-get install -y libyaml-cpp-dev
  ```

2. If you are installing the eBUS SDK on Ubuntu 14.04, 18.04, or 20.04, you must install build-essential. Execute the command:
   ```
   sudo apt-get install build-essential
   ```

> You must install build-essential before installing eBUS SDK in order to allow the compilation of the eBUS Universal Pro driver during the eBUS SDK installation. Build-essential is also required to compile samples. The build-essential package is installed automatically on Ubuntu 16.04.

3. Copy the eBUS SDK installation package to your workstation or embedded computer.

   The installation package is available for download at <u>supportcenter.pleora.com</u>.

4. From the terminal, execute one of the following commands. The command varies depending on the distribution you are using.
   - On Ubuntu:
     ```
     sudo dpkg -i eBUS_SDK_<distribution_targeted>-<6.x.x>-<SDK build #>.deb
     ```
     Where *<distribution_targeted>* can be:
     - Ubuntu-20.04-x86_64
     - Ubuntu-18.04-x86_64
     - Ubuntu-16.04-x86_64
     - Ubuntu-16.04-i686
     - Ubuntu-14.04-x86_64
     - Ubuntu-14.04-i686
   - On RHEL/CentOS, as super user:
     ```
     rpm -i eBUS_SDK_<distribution_targeted>-<6.x.x>-<SDK build #>.rpm
     ```
     If any components failed to install, see the notes at the end of this procedure.

5. We recommend that you reboot your workstation or embedded computer to ensure that the correct environment variables are set at startup.

   The eBUS SDK is installed in the following directory:
   ```
   opt/pleora/ebus_sdk/<distribution_targeted>/
   ```
   For example: `/opt/pleora/ebus_sdk/Ubuntu-16.04-i686`

> **Tip:** 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" on page 18.

Note: 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" on page 26.

Choosing Optimal Settings for the Jetson Modules

We strongly recommend that you modify the Jetson Nano, AGX Xavier, Xavier NX, and TX2 configuration for power and performance management. If you do not perform these steps, you may encounter freezes or the eBUS SDK may stop responding.

To choose the optimal nvpmodel mode

- Execute the following command:

  sudo nvpmodel -m 0

To increase the CPU clock to the maximum value

1. Back up the default value to a file by executing the following command:

   sudo jetson_clocks.sh --store <filename_to_store>

2. Increase the CPU clock speed by executing the following command:

   sudo jetson_clocks

To enable jumbo frames

- The use of jumbo frames reduces the amount of Ethernet, IP, UDP, and GVSP packet overhead when transmitting images, which reduces the CPU load and memory requirements. Using the script that is detailed in "To enable jumbo packets" on page 19, set jumbo frames to **9000**.

Uninstalling the eBUS SDK for Linux

You can use the dpkg command to uninstall the eBUS SDK for Linux (Ubuntu) and the rpm command for RHEL/CentOS.

To uninstall the eBUS SDK

1. From the terminal, execute one of the following commands. The command varies depending on the distribution you are using.

   - On Ubuntu:
     sudo dpkg -r ebus_sdk_<package_name>

   Where <package_name> can be:
     - Ubuntu-20.04-x86_64
     - Ubuntu-18.04-x86_64
     - Ubuntu-16.04-x86_64
     - Ubuntu-16.04-i686
     - Ubuntu-14.04-x86_64

- Ubuntu-14.04-i686
- For ARM platforms:
  - linux-aarch-arm

  For example: `sudo dpkg -r ebus_sdk_ubuntu-16.04-i686`
  or `dpkg -r ebus_sdk_linux-aarch-arm`
- On RHEL/CentOS, as super user:
  `rpm -e ebus_sdk_<package_name>`

  For example: `rpm -e ebus_sdk_rhel-centos-x86_64`

> **Tip:** To see installed packages, you can execute one of the following commands:
> - **On Ubuntu:**
>
>   `dpkg -l | grep ebus`
> - **On RHEL/CentOS:**
>
>   `rpm -qa | grep ebus`

Chapter 3

Activating eBUS SDK Licenses

A license is required to take full advantage of the eBUS SDK's transmit and receive capabilities. 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.

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) or macOS. If you need to use Linux (ARM platforms) or macOS, install the license file, which is available for purchase online.

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 representative.
- DO NOT disable or remove the NIC (or WiFi adapter) that is associated with the license.

To activate an eBUS SDK license

1. On your workstation (eBUS Receive or eBUS SDK Seat license) or embedded computer (eBUS Edge license), copy the license file to:

   `/opt/pleora/ebus_sdk/<distribution_targeted>/licenses`

2. Select your next step:
   - If you are using an earlier version of eBUS SDK (version 4.0 to 6.1), go to step 3.
   - If you are using eBUS SDK version 6.2 or later, go to step 4.

3. Stop and restart the eBUS daemon by executing the following commands:

   `sudo service ebusd stop`

   `sudo service ebusd restart`

4. 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.

If the licenses folder is not created automatically, create the folder by executing the following command:

`sudo mkdir /opt/pleora/ebus_sdk/<distribution_targeted>/licenses`

Additional Information: Stopping the eBUS daemon (applies to eBUS SDK versions 4.0 to 6.1)

The eBUS daemon is used by the eBUS SDK for connection to USB3 Vision devices and is also used to license the eBUS SDK's transmit and receive capabilities.

With GigE Vision devices

The GigE Vision device will continue streaming images, even when the eBUS SDK daemon is not running.

Stopping the eBUS daemon has an impact on the licensing of your system. If a valid license is present when you stop the eBUS daemon, you can continue streaming images until you disconnect the GigE Vision device. At this point, an invalid license will be reported. To return to normal operation, start the eBUS daemon and restart any applications created with the eBUS SDK, such as eBUS Player or the sample applications.

With USB3 Vision devices

When you stop the eBUS daemon, any connected USB3 Vision devices will be disconnected. To connect to USB3 Vision devices with eBUS Player, the sample applications, or applications created with the eBUS SDK, the eBUS SDK must be running.

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.

Chapter 4

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:

   sudo /opt/pleora/ebus_sdk/*<distribution_targeted>*/bin/set_udev_rules.sh script.

2. When prompted, enter the vendor ID assigned to the device's manufacturer. This number often appears on the device's label.

3. For eBUS versions 4.0 to 6.1, stop and restart the eBUS daemon by executing the following commands:

   sudo service eBUSd restart

In eBUS SDK versions 4.0 to 6.1, the eBUS daemon is used by the eBUS SDK for connection to USB3 Vision devices and is also used to license the eBUS SDK's transmit and receive capabilities. The eBUS daemon is removed in eBUS SDK version 6.2 and later.

Chapter 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.

The following topics are covered in this chapter:

- "Using the eBUS Universal Pro for Ethernet Filter Driver" on page 18
- "Enabling Jumbo Ethernet Frames" on page 19
- "Additional Optimization Steps" on page 19

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/<distribution_targeted>/module`

2. Execute the following command:
   `sudo ./build.sh --kernel=/usr/src/<linux_headers>`

   Where *linux_headers* is the corresponding headers for the Linux4Tegra version that is installed. For example, **linux-headers-4.4.38-tegra**.

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/<distribution_targeted>/module/install_driver.sh`

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/<distribution_targeted>/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`

Enabling Jumbo Ethernet Frames

If supported by your network card, GigE Vision device, and switch (if applicable), we recommend that you enable jumbo Ethernet frames when using GigE Vision devices. The use of jumbo frames reduces the amount of Ethernet, IP, UDP, and GVSP packet overhead when transmitting images, which reduces the CPU load and memory requirements.

To enable jumbo packets

- Execute the following command:

    ```
    sudo ifconfig <network_interface_ID> mtu [SIZE]
    ```

    where, [*network_interface_ID*] is the name of the network interface card (NIC)

    [*SIZE*] = desired frame size

    For example:

    ```
    sudo ifconfig eth0 mtu 9000
    ```

Tips:

To see the name of the NIC in your system, execute the following command:

```
sudo lshw -c network -businfo
```

To see the current maximum transmission unit (MTU) value, execute the following command:

```
ifconfig | grep MTU
```

Additional Optimization Steps

If you are using an Intel Pro/1000-based network interface, there are additional configuration settings that can yield performance improvements. If you have an Intel Pro/1000 NIC, refer to the *Configuring Network Adapters for Best Performance Knowledge Base Article* (available on the Pleora Support Center) for more information. For other NICs, see the documentation accompanying the NIC.
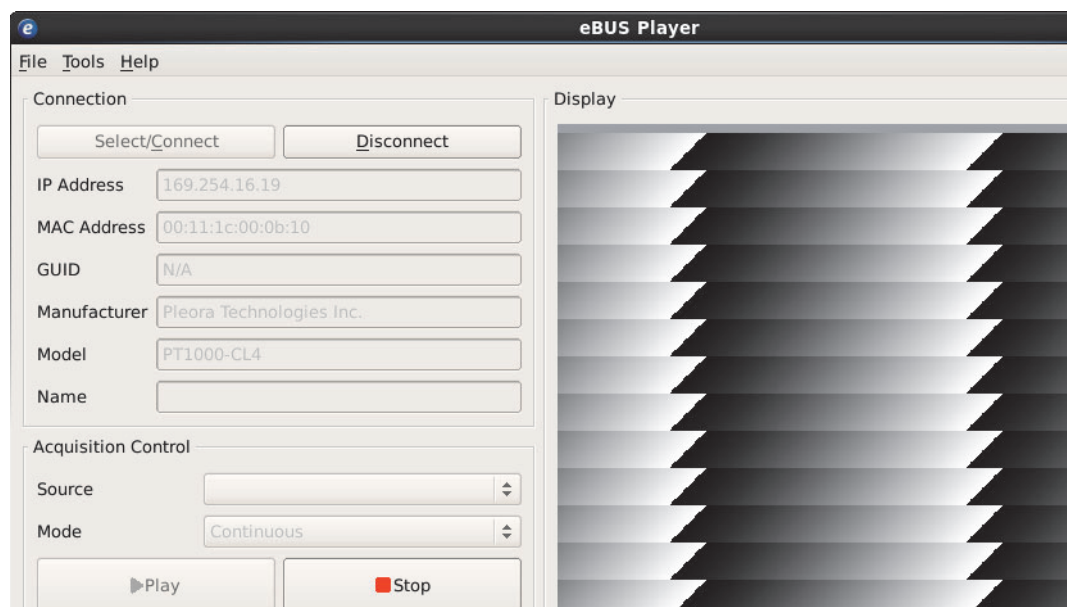
Chapter 6

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 and 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" on page 17.

Note: In the `opt/pleora/<distribution_targeted>/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.

Chapter 7

Compiling and Running Sample Applications

We strongly recommend that you copy the sample applications from the `share/samples` directory to your personal development workspace before modifying or compiling them.

A list of samples, with a description of each, can be found in the index.html file in the `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" on page 5.

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

To compile the eBUS SDK sample applications

1. If you have not restarted your computer since installing the eBUS SDK, we recommend that you do so now. This ensures that the correct environment variables are set at startup.

2. Install the following prerequisite:

   `sudo apt install libyaml-cpp-dev`

3. Make a copy of the following directory, as a backup for the original source files:

   `/opt/pleora/ebus_sdk/<distribution_targeted>/share/samples`

4. Navigate to the directory that contains the copy of the sample code from step 1.

5. Do one of the following:

- **To compile all sample applications at one time**: Execute the `build.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.

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." on page 26.

Chapter 8

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/<distribution_targeted>/bin/set_rp_filter.sh
```

Cannot detect or connect to USB3 Vision devices.

For eBUS SDK versions 4.0 to 6.1, ensure that the eBUS daemon is started by executing the command:
```
sudo service eBUSd status
```
- If the daemon is not started, execute the command: `sudo service eBUSd start`

- If the daemon is not installed, run the `sudo /opt/pleora/ebus_sdk/<distribution_targeted>/bin/install_daemon.sh` script and follow the prompts.

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" on page 15.

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/<distribution_targeted>/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` script to set the environment variables (`source set_puregev_env`).

Tip: You can also launch the sample applications using the `RunFromEnv.sh` script. For example:

```
source ./RunFromEnv.sh && ~/samples/PvStreamSample/PvStreamSample
```

Note: 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_ROOT_V3_3 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" on page 11 or the *eBUS SDK Licensing Overview Knowledge Base Article*, available on the Pleora Support Center (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:

- On Ubuntu:
  ```
  sudo apt-get install linux-headers-$(uname -r)
  ```
- On RHEL/CentOS, as super user:
  ```
  yum install kernel-devel
  ```

The eBUS SDK applications freeze or stop responding on the Jetson module

Ensure that you have chosen the nvpmodel mode, increased the CPU clock the maximum value, and configured jumbo packets for 9000 bytes. For more information, see "Choosing Optimal Settings for the Jetson Modules" on page 8.

The eBUSd daemon is not loaded and cannot restart (in eBUS SDK versions 4.0 to 6.1)

If you are using eBUS SDK versions 4.0 to 6.1 and want to activate a license on the Linux operating system, but the eBUSd is failing to restart and is reported as "not loaded", for example:

```
sudo service eBUSd stop
Failed to stop eBUSd.service: Unit eBUSd.service not loaded
```

Ensure that the eBUS daemon is installed and configured to automatically load at startup:

```
cd     /opt/pleora/ebus_sdk/<Distribution_Targeted_by_eBUS_installer>/bin
sudo  ./install_daemon.sh
```

Chapter 9

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.