

PLEORA TECHNOLOGIES INC.



eBUS SDK for macOS® Quick Start Guide

eBUS SDK Version 6.0



Copyright © 2019 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

PureGEV, eBUS, iPORT, vDisplay, AutoGEV, AutoGen, 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 Number

EX001-017-0015, Version 5.0, 3/13/19

Table of Contents

About this Guide	1
What this Guide Provides	2
Related Documents	2
Introducing the eBUS SDK for macOS	3
About the eBUS SDK for macOS	4
eBUS SDK Licenses	4
Activating an eBUS SDK License	4
Installing the eBUS SDK for macOS	5
System Requirements	6
Installing the eBUS SDK for macOS	6
Uninstalling the eBUS SDK	7
Managing the eBUS Universal Pro for Ethernet Driver and eBUS Daemon	7
Enabling Jumbo Frames	9
Using eBUS Player to Configure Devices and Stream Images	11
Compiling and Running Sample Applications	13
About the Samples	14
Using H.264 Encoding with the eBUSPlayer Sample Application	16
Accessing the Sample Code	17
Developing your Application with Xcode	17
Reviewing the eBUS SDK Log Files	19
Technical Support	21

Chapter 1



About this Guide

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

The following topics are covered in this chapter:

- [“What this Guide Provides”](#) on page 2
- [“Related Documents”](#) on page 2

What this Guide Provides

This guide provides you with the steps to use the eBUS SDK on macOS High Sierra (10.13). The guide assumes a general understanding of macOS.

You can use the eBUS SDK sample applications to see how the API classes and methods work together for device configuration and control, unicast and multicast communication, image and data acquisition, image display, and diagnostics. You can also use the sample code to verify that your system is working properly (that is, determine whether there is a problem with your code or your equipment).

Related Documents

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

- *eBUS Player Quick Start Guide for macOS*
- *eBUS Player User Guide for macOS*
- *eBUS SDK Licensing Overview Knowledge Base Article*
- eBUS SDK API documentation. Navigate to `/Library/Frameworks/eBUS.framework/Versions/Current/Documentation/index.html`

Chapter 2



Introducing the eBUS SDK for macOS

This chapter describes the eBUS SDK for macOS, which is an application development toolset that allows you to develop custom vision systems to acquire and transmit images and data using the macOS operating system.

The following topics are covered in this chapter:

- [“About the eBUS SDK for macOS”](#) on page 4
- [“eBUS SDK Licenses”](#) on page 4

About the eBUS SDK for macOS

Designed for use in high-performance digital video systems and cameras, Pleora's eBUS™ Software Development Kit (SDK) is a feature-rich toolkit that allows Windows, Linux and Mac developers to produce video applications in short timeframes, while reducing risk and lowering design and support costs.

The eBUS SDK is based on a clean, modular architecture that uses a single set of functions to receive video over GigE, 10 GigE, USB, and IEEE 802.11 wireless. This helps “future-proof” application software because it can be modified quickly and easily for different media. It also helps developers work efficiently, and reduces support for portfolios with multiple interface offerings.

Pleora's eBUS SDK features a huge library of sample code, which serves as a quick-start platform for each development project. It also includes the eBUS Universal Pro driver, which helps developers achieve high-performance results by maximizing end-to-end throughput and ensuring video is delivered with low, consistent latency.

The eBUS SDK complies fully with the GigE Vision®, USB3 Vision™ and GenICam™ standards, and is compatible with every version of each. It interoperates seamlessly with Pleora's extensive portfolio of video interface products, as well as with standards-compliant products from other manufacturers.

eBUS SDK Licenses

While the eBUS SDK for macOS is a licensed product, you can trial the API without purchasing a license. However, the following limitations apply:

- Received images (received from third-party devices, such as non-Pleora enabled cameras) have an embossed watermark.
- Raw data is only received from Pleora-enabled cameras.

Activating an eBUS SDK License

For detailed information about licensing, including details on activating a license, see the *eBUS SDK Licensing Overview Knowledge Base Article*, available on the Pleora Technologies Support Center at supportcenter.pleora.com.

Chapter 3



Installing the eBUS SDK for macOS

This chapter provides system requirements and installation steps that you will use to install the eBUS SDK for macOS. It also provides information to help you manage the eBUS Universal Pro for Ethernet driver and the eBUS daemon. The eBUS daemon is used for connection to USB3 Vision devices and is also used for licensing of the eBUS SDK's transmit and receive capabilities.

The following topics are covered in this chapter:

- “System Requirements” on page 6
- “Installing the eBUS SDK for macOS” on page 6
- “Uninstalling the eBUS SDK” on page 7
- “Managing the eBUS Universal Pro for Ethernet Driver and eBUS Daemon” on page 7

System Requirements

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

The following hardware is required:

- At least one Gigabit Ethernet NIC (if you are using GigE Vision devices) or at least one USB 3.0 port (if you are using USB3 Vision devices).
- macOS Mojave (version 10.14)
- Xcode version 10.1

Installing the eBUS SDK for macOS

Follow the installation prompts that appear when you run the eBUS SDK installation package. As part of the installation, sample applications, the eBUS Universal Pro for Ethernet driver, and the eBUS daemon are installed on your computer.

Uninstalling the eBUS SDK

Use the **eBUSUninstallAll.sh** script to uninstall the eBUS SDK. This script uninstalls the eBUS SDK, including the sample applications, driver, and the eBUS daemon that are installed on your computer.



You must use the **eBUSUninstallAll.sh** script to uninstall the eBUS SDK, including eBUS Player. Dragging the eBUS Player app to the Trash will not fully remove the eBUS SDK.

Dragging the eBUS Player app to the Trash only removes the eBUS Player app, not the entire eBUS SDK.

To uninstall the eBUS SDK

- Using Terminal, run `sudo ./eBUSUninstallAll.sh` from the following location:
`/Library/Frameworks/eBUS.framework`



It is not necessary to uninstall the eBUS SDK before installing a different eBUS SDK version. The currently-installed version is automatically uninstalled as part of this process.

Managing the eBUS Universal Pro for Ethernet Driver and eBUS Daemon

During installation of the eBUS SDK, the eBUS Universal Pro for Ethernet driver, which supports GigE Vision devices, is installed on your computer and is loaded automatically. The eBUS daemon is also installed, which is used for connection to USB3 Vision devices and is also used for licensing of the eBUS SDK's transmit and receive capabilities.

If you need to temporarily disable the driver or eBUS daemon, or if you want to completely remove either one from your computer, you can use the **eBUSUniversalProForEthernetHelper.sh** and **eBUSDaemonHelper.sh** scripts, located in `/Library/Application Support/Pleora/eBUS SDK`

To load, unload, or uninstall the eBUS Universal Pro for Ethernet driver

- You can run the following commands using Terminal:
 - **To manually load the driver (one time only).** Use this command to load the driver. The next time you reboot the computer, the driver will not be loaded.

```
sudo ./eBUSUniversalProForEthernetHelper.sh --manualload
```
 - **To unload the driver.** Use this command to unload the driver. To load the driver again, use `manualload` or `autoload`.

```
sudo ./eBUSUniversalProForEthernetHelper.sh --unload
```
 - **To configure the driver to load automatically at startup** (running this command stops and restarts the driver).

```
sudo ./eBUSUniversalProForEthernetHelper.sh --autoload
```
 - **To uninstall the driver (for advanced users only).**

```
sudo ./eBUSUniversalProForEthernetHelper.sh --uninstall
```
 - **To view the eBUS driver status**

```
sudo ./eBUSUniversalProForEthernetHelper.sh --diagnostic
```

To load, unload, or uninstall the eBUS daemon

- You can run the following commands using Terminal:
 - **To manually start the eBUS daemon (one time only).** Use this command to start the eBUS daemon. The next time you reboot the computer, the daemon will not be run.

```
sudo ./eBUSDaemonHelper.sh --manualload
```
 - **To stop the eBUS daemon.** Use this command to stop the eBUS daemon. To start the eBUS daemon again, use `manualload` or `autoload`.

```
sudo ./eBUSDaemonHelper.sh --unload
```
 - **To configure the eBUS daemon to run automatically at startup** (running this command stops and restarts the eBUS daemon).

```
sudo ./eBUSDaemonHelper.sh --autoload
```
 - **To uninstall the eBUS daemon (for advanced users only).**

```
sudo ./eBUSDaemonHelper.sh --uninstall
```
 - **To view the eBUS daemon status**

```
sudo ./eBUSDaemonHelper.sh --diagnostic
```



After you install an eBUS SDK license on your computer, run `sudo ./eBUSDaemonHelper.sh --autoload`, to stop and restart the eBUS daemon. For more information about licensing, see the *eBUS SDK Licensing Overview Knowledge Base Article*, available on the Pleora Technologies Support Center.

Chapter 4

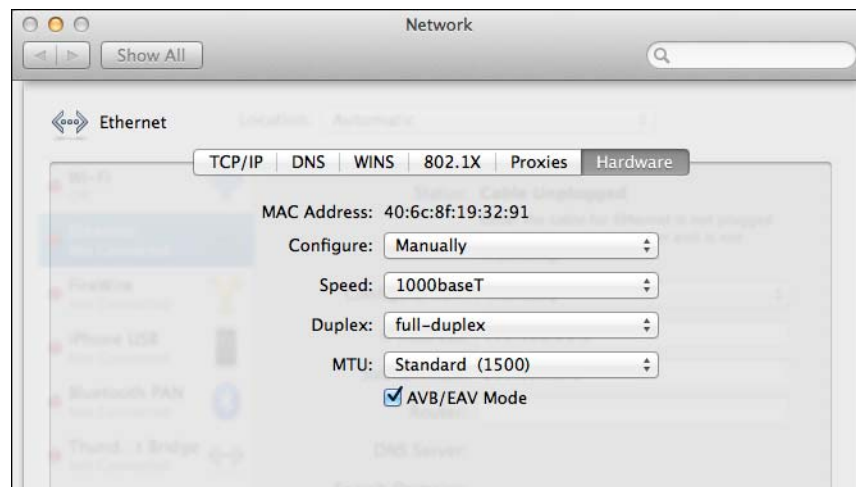


Enabling Jumbo Frames

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 frames

1. In macOS **System Preferences**, click **Network**.
2. Click **Ethernet** and then click **Advanced**.
3. On the **Hardware** tab, click **Manually** in the **Configure** list.
4. In the list **Speed** list, ensure **1000baseT** is selected.
5. In the **Duplex** list, ensure **full-duplex** is selected.
6. In the **MTU** list, click **Jumbo (9000)**.



7. Click **OK**.
8. Close the open dialogs.

Chapter 5



Using eBUS Player to Configure Devices and Stream Images

You can use the eBUS Player application to connect to, configure, and stream images from GigE Vision and USB3 Vision devices. You can open eBUS Player by clicking the eBUS Player icon in Launchpad.



Chapter 6



Compiling and Running Sample Applications

To illustrate how you can use the eBUS SDK for macOS to acquire and transmit images, the SDK includes sample code that you can use. This chapter provides a description of the sample code and provides general information about accessing the code to create sample applications.

The following topics are covered in this chapter:

- “About the Samples” on page 14
- “Accessing the Sample Code” on page 17
- “Developing your Application with Xcode” on page 17

About the Samples

The eBUS SDK for macOS contains two types of samples:

- **GUI-based samples**, which are written in Objective-C. The GUI elements of the sample, such as PvDisplay and PvGenBrowser, are provided by PvGUI, which is an Objective-C library. The non-GUI elements of the sample, such as PvDevice, PvStream, and PvGenParameterArray, are provided by C++ libraries.
- **Command line samples**, which are written in C++.



Files with the .mm extension may contain both Objective-C and C++ code.

The eBUS SDK libraries are compiled with Apple LLVM 9.0 for the macOS 10.13 Base SDK which is the standard toolchain on macOS High Sierra (the eBUS Framework does not support gcc). 64-bit targets are provided.

The following table provides a description of the sample code that is available for the eBUS SDK for macOS.



For developers who have worked with earlier releases of the eBUS SDK, please take note that the following code samples have been deprecated in eBUS SDK 6.0: **TransmitTestPattern** and **TransmitTiledImages**.

Table 1: Sample Code

Sample code	Function	Type of application that is created
eBUS Demo Application		
eBUS Player	eBUS Player is a full-featured application that showcases the functionality of the eBUS Rx SDK, which is used to detect, connect, and configure GigE Vision and USB3 Vision devices, and display and stream images. This sample provides advanced examples of the eBUS SDK classes and functions that are available and provides full source code.	<ul style="list-style-type: none">• UI-based, Cocoa
Getting Started		
PvStreamSample	This “Hello World” sample shows you how to connect to a GigE Vision or USB3 Vision device, receive an image stream, stop streaming, and disconnect from the device.	<ul style="list-style-type: none">• Command line

Table 1: Sample Code (Continued)

Sample code	Function	Type of application that is created
SimpleGUIApplication	<p>This UI sample provides a basic user interface using C++ and MFC to detect, connect, and configure GigE Vision and USB3 Vision devices, and display and stream images.</p> <p>This sample is a good starting point to show how to create a GUI project to receive images from a GigE Vision or USB3 Vision camera.</p>	<ul style="list-style-type: none"> • UI-based, Cocoa
Image Streaming		
PvPipelineSample	<p>This sample extends the "Hello World" PvStreamSample by showing how buffers are managed internally by the PvPipeline class. This removes some of the complexity of buffer management from the application when compared to the PvStream sample.</p>	<ul style="list-style-type: none"> • Command line
MultiSource for GigE Vision devices	<p>This command line sample for GigE Vision devices shows you how to receive images from a GigE Vision device that has multiple streaming sources.</p>	<ul style="list-style-type: none"> • Command line
DualSource	<p>This GUI-based sample for GigE Vision devices extends the MultiSource sample by allowing you to view image streams from a GigE Vision device that has two streaming sources.</p>	<ul style="list-style-type: none"> • UI-based, Cocoa
MulticastMaster for GigE Vision devices	<p>This sample shows how to connect to a GigE Vision device and initiate a multicast stream to allow multiple slave devices to receive and process the image stream simultaneously.</p> <p>This sample is used in conjunction with the MulticastSlave sample, which listens to the multicast stream.</p>	<ul style="list-style-type: none"> • Command line
MulticastSlave for GigE Vision devices	<p>This sample shows how to configure the eBUS SDK to receive an image stream from a GigE Vision device that is configured for multicast mode.</p> <p>This sample is used in conjunction with the MulticastMaster sample, which initiates the multicast stream.</p>	<ul style="list-style-type: none"> • Command line
Discovery and Connection		
DeviceFinder	<p>This sample shows how to detect and enumerate GigE Vision and USB3 Vision devices on the network.</p>	<ul style="list-style-type: none"> • UI based
ConnectionRecovery	<p>This sample shows how to automatically recover from connectivity issues, such as accidental disconnects and power interruptions, to build more robustness into your eBUS SDK application.</p>	<ul style="list-style-type: none"> • Command line

Table 1: Sample Code (Continued)

Sample code	Function	Type of application that is created
Configuration and Event Monitoring		
DeviceSerialPort	This sample shows how to send commands to a camera or other device that accepts serial input commands through a compatible Pleora iPORT video interface using the Pleora device's General Purpose Input/Output (GPIO) signals, including UART or BULK.	<ul style="list-style-type: none"> • Command line
CameraBridge	This sample shows how to control a Camera Link camera through a compatible Pleora iPORT CL Series External Frame Grabber using the following Camera Link protocols: CLProtocol and GenCP.	<ul style="list-style-type: none"> • Command line
GenICamParameters	This sample shows how to enumerate and display the GenICam features and settings of a GenICam-compatible device by discovering and accessing the features of the device's node map. The node map is built programmatically from the device's GenICam XML file.	<ul style="list-style-type: none"> • Command line
ConfigurationReader	This sample shows how to save the configuration state of your GigE Vision or USB3 Vision device, your eBUS SDK application preferences, and any custom strings and property lists to a file. It also illustrates how to open and restore this information.	<ul style="list-style-type: none"> • Command line

Using H.264 Encoding with the eBUSPlayer Sample Application

You can use eBUS Player to encode images in H.264 video format and save them on your computer in an MPEG-4 file.

If you are interested in adding H.264 encoding to your application, you can refer to the `Mp4WriterMac` class in the eBUS Player sample application for an example.

This feature is not available by default in eBUS Player. You must add the `PV_ENABLE_MP4` preprocessor macro in the build settings of the eBUS Player sample in Xcode.

Accessing the Sample Code

The sample code is available in the following location, along with an `index.html` file that lists each sample and provides brief descriptions:

```
/Library/Frameworks/eBUS.framework/Versions/Current/Samples
```



You must copy the sample code to another location on your computer (such as Documents) before you open the sample code in Xcode. Access to the `Samples` directory is restricted.

Developing your Application with Xcode

The eBUS SDK and its headers are bundled in a framework called the eBUS Framework. When developing your applications with the eBUS SDK, you must add the eBUS Framework to your project in Xcode.

To add the eBUS Framework to an Xcode project

1. Open your Xcode project.
2. In the project navigator, select the location in which you want to add the framework.
3. On the **File** menu, click **Add Files to <App_Name>**.
4. Select the eBUS Framework and then click **Add**. The framework is in the following location:
`/Library/Frameworks/eBUS.framework`

Chapter 7

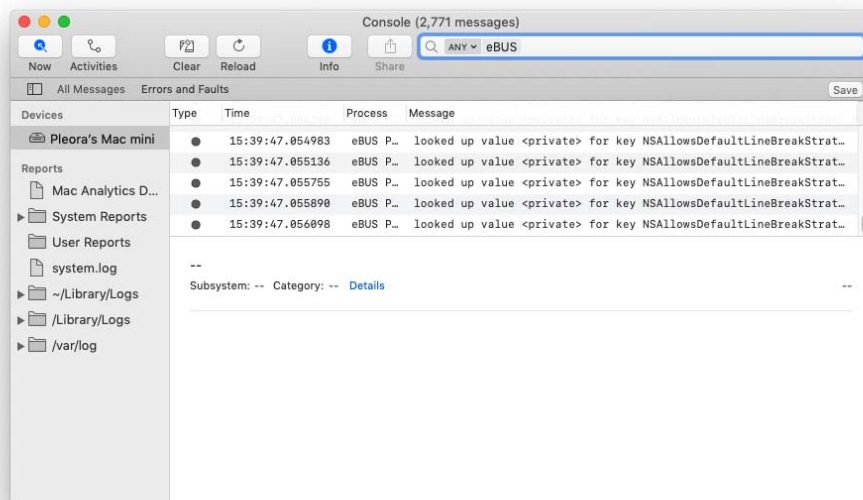


Reviewing the eBUS SDK Log Files

The eBUS SDK log files can help you debug issues that you encounter. To review the log files, open the macOS Console app and filter the entries to show eBUS SDK information.

To review the eBUS SDK log files

1. Launch the macOS Console application.
2. In the **Console** menu under **Action**, select **Include Info Messages** and **Include Debug Messages**.
3. Type **eBUS** in the Console application's **Search** bar and then press **ENTER**.
4. Start the eBUS SDK application from which you want to capture logs.



Chapter 8



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 and click **Support Center**.
If you have not registered yet, you are prompted to register.
Accounts are usually validated within one business day.

