eBUS SDK 6.4.0

Installation Instructions and Release Notes

Introduction

eBUS™ SDK release 6.4.0 is custom version of the Pleora Technologies software development kit (SDK), which includes additional functionality for JAI.

This document provides installation and usage instructions. It also includes release notes that pertain to the software, as of February 2024.

Installation Instructions

Installing the eBUS SDK for JAI with eBUS Python API package on Windows

Since the eBUS SDK Python API for JAI is not part of the eBUS SDK for JAI on Windows, we must first install the base eBUS SDK for JAI using the Windows installation package, then install the appropriate eBUS Python for JAI package for the specific version of Python you are using. You can download the eBUS SDK for JAI and the associated eBUS Python for JAI packages from the Pleora Support Center at supportcenter.pleora.com.

Note: If you use the Linux operating system, you must install the eBUS SDK for JAI as superuser. For full details about installing the eBUS SDK for JAI on Linux, see the eBUS SDK for Linux Quick Start Guide, available at the Pleora Support Center (supportcenter.pleora.com).

To install eBUS SDK for JAI 6.4.0 on the Windows operating system

1. Uninstall the existing eBUS SDK for JAI installation package from your computer. A reboot may be required.
2. Copy the eBUS SDK for JAI installation package to your computer.
3. Run the eBUS SDK for JAI installation package (**eBUS SDK 64-bit for JAI.**6.4.0<***SDK build #***>**)** and follow the installation wizard prompts.
4. After the eBUS SDK for JAI is installed, open the JAI version of eBUS Player from the Windows Start menu. It appears under **eBUS** in the Windows **Start** menu.

To install eBUS Python for JAI API 6.4.0 on the Windows operating system

NOTE: eBUS Python for JAI API is only available for 64-bit Windows for JAI, see the "Supported Python Versions" section, on page 22.

Pleora Technologies

The following dependency packages are required for eBUS Python for JAI on Windows:

- Python (3.6, 3.7, 3.8, 3.9, 3.10 and /or 3.11)
- **pip**
- **numpy**
- **opencv-python** (optional for some samples)
- eBUS Python for JAI

Step 1: How to install Python on Windows

Installing and using Python on Windows is very simple. The installation procedure involves just three steps:

1. Download the binaries (Python Releases for Windows | Python.org)
2. Run the executable installer.
3. Add Python to PATH environmental variables.

Step 2: How to upgrade pip on Windows

From a terminal, run the following command:

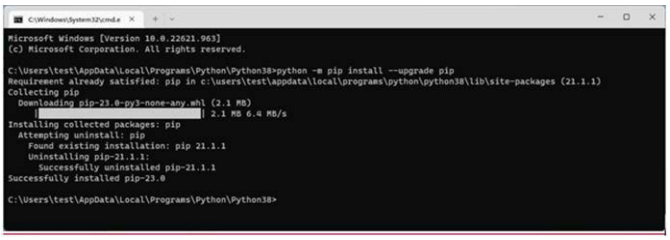
```
python -m pip install --upgrade pip
```

If you have only one Python package installed, and you have added Python in your PATH. You can call python everywhere.

If you have several Python packages installed, you should launch python from the installed location.

From a terminal run the following command with the default Python path, if you use python 3.8:

```
C:\Users\<username>\AppData\Local\Programs\Python\Python38>python -m pip install--
upgrade pip
```



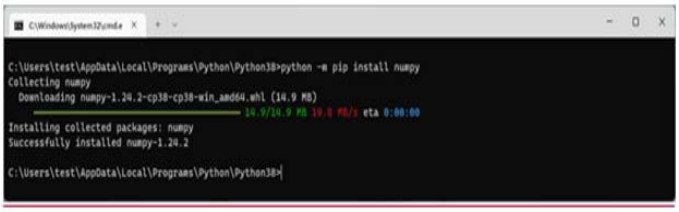
Step 3: How to install numpy on Windows

From a terminal, run the following command:

```
python -m pip install numpy
```

If you have several Python packages installed, you need to specify the python executable. From a terminal run the following command with the default Python path, if you use python 3.8:

```
C:\Users\<username>\AppData\Local\Programs\Python\Python38>python -m pip install numpy
```



Step 5: (optional) How to install opencv-python on Windows

From a terminal, run the following command:

```
python -m pip install opencv-python--prefer-binary
```

If you have several Python packages installed, you need to specify the python executable. From a terminal run the following command with the default Python path, if you use python 3.8:

```
C:\Users\<username>\AppData\Local\Programs\Python\Python38>python -m pip install
opencv-python--prefer-binary
```

Step 6: How to install eBUS Python for JAI API on Windows

From a terminal, run the following command:

```
python -m pip install <path of the package>\ebus_python-6.3.1-<build number>_jai-
py<python version>-none-win_amd64.whl
```

If you have several Python packages installed, you need to specify the python executable. From a terminal run the following commands with the default Python path, if you use python 3.8:

```
C:\Users\<username>\AppData\Local\Programs\Python\Python38\python -m pip install

<path of the package>\ebus_python-6.3.1-<build number>_jai-py38-none-win_amd64.whl
```

Location of installed eBUS Python API on Windows

```
C:\Users\<username>\AppData\Local\Programs\Python\Python<Python
version>\Lib\site-packages\ebus-python
```

For example, with Python 3.8:

```
C:\Users\<username>\AppData\Local\Programs\Python\Python38\Lib\site-
packages\ebus-python
```

Installing the eBUS SDK for JAI with eBUS Python for JAI API package on Linux

Use the installation packages to install the eBUS SDK for JAI and eBUS Python for JAI API for Linux.

To install the eBUS SDK for JAI (/Runtime) and eBUS Python for JAI API

1. The following prerequisites are critical to be able to save MP4 video along with other functionality within eBUS SDK for JAI. They are tightly bound to eBUS SDK for JAI so they need to be installed regardless of whether or not MP4 video is being saved. Note that there are additional steps required to enable MP4 video saving in eBUS Player (these are described later on in this section).
    - 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 Ubuntu 22.04 (64-bit) execute the command:
        - `sudo apt-get install libavcodec58`
    - For CentOS 8 Stream execute the following commands:
        - `sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release- latest-8.noarch.rpm`
        - `sudo yum install https://download1.rpmfusion.org/free/el/rpmfusion-free-release-8.noarch.rpm https://download1.rpmfusion.org/nonfree/el/rpmfusion-nonfree-release-8.noarch.rpm`
        - `sudo yum config-manager --set enabled powertools`
        - `sudo yum install ffmpeg`
    - For RHEL 8:
        - `sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release- latest-8.noarch.rpm`
        - `sudo yum install https://download1.rpmfusion.org/free/el/rpmfusion-free-release-8.noarch.rpm https://download1.rpmfusion.org/nonfree/el/rpmfusion-nonfree-release-8.noarch.rpm`
        - `sudo subscription-manager repos --enable codeready-builder-for-rhel- 8x-86_64-rpms`
        - `sudo yum install ffmpeg`
2. If not already installed, install Qt5 on your system. 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**.
    - For Ubuntu 18.04 and 20.04 (64-bit) execute the command:
        - `sudo apt-get install qt5-default`
    - For Ubuntu 22.04 (64-bit) execute the command:
        - `sudo apt-get install qtbase5-dev qt5-qmake`
    - For On RHEL8/CentOS Stream 8 execute the command (should be run as a super user):
        - `yum install qt5-qtbase-devel`
3. Install other required dependencies:
    - For Ubuntu 18.04, 20.04, or 22.04, execute the command**:**
        - `sudo apt-get install build-essential`
        - `sudo apt-get python3-numpy (required only when installing eBUS Python)`
        - `sudo apt-get python3-opencv (required only when installing eBUS Python)`

- For Ubuntu 22.04, execute the following command to update the compiler to build the eBUS driver when the linux kernel version is 6.5 or later.
  - sudo apt-get install gcc-12
- For CentOS 8 Stream and RedHat 8, execute the command:
  - `sudo yum install elftutil-libelf-devel`
  - `sudo yum install python3-numpy (required only when installing eBUS Python)`
  - `python3 -m pip install --user --upgrade pip (optional when installing eBUS Python)`
  - `python3 -m pip install --user opencv-python --prefer-binary (optional when installing eBUS Python)`
- For ARM64 (aarch64) on eBUS SDK for JAI platform (or eBUS Runtime), execute the command:
  - On JetPack 4.6.3:
    - `sudo apt-get install libyaml-cpp0.5v5`
  - On JetPack 5.1:
    - `sudo apt-get install libyaml-cpp0.6`

4. Copy the eBUS SDK for JAI and eBUS Python for JAI installation packages to your workstation or embedded computer. The installation packages are available for download at supportcenter.pleora.com.
5. From the terminal, execute the following commands according to your system. The command varies depending on the distribution you are using.
   - Ubuntu x86_64:
     - `sudo dpkg -i eBUS_SDK_JAI_<distribution_targeted>-<6.3.1>-<SDK build #>.deb`
     - `sudo dpkg -i eBUS_Python_JAI_<distribution_targeted>-<6.3.1>-<SDK build #>.deb`
     - Where <*distribution_targeted*> can be:
       - Ubuntu-22.04 x86_64
       - Ubuntu-20.04 x86_64
       - Ubuntu-18.04 x86_64
   - For ARM platforms:
     - `sudo dpkg -i eBUS_SDK_JAI_<distribution_targeted>-<6.3.1>-<SDK build #>.deb`
     - `sudo dpkg -i eBUS_Python_JAI_<distribution_targeted>-<6.3.1>-<SDK build #>.deb`
     - Where <*distribution_targeted*> can be
       - For eBUS SDK 6.3 (and later):
         - Jetson_4.6_linux-aarch64-arm
         - Jetson_5.1_linux-aarch64-arm
   - On RHEL/CentOS
     - `sudo rpm -i eBUS_SDK_JAI_<distribution targeted>-x86_64-<6.3.1>-<SDK build #>.rpm`
     - `sudo rpm -i eBUS_Python_JAI_<distribution targeted>-x86_64-<6.3.1>-<SDK build #>.rpm`
     - Where <*distribution_targeted*> can be:
       - RHEL-CentOS (for RedHat 8.7)
       - CentOS-RHEL (for CentOS 8 Stream)

If any components failed to install, see the notes at the end of this procedure.

6. For RHEL8 or CentOS 8 Stream, the firewall needs to be disabled in order to connect to a GigE Vision device over the network.
   - As super user, execute this command:
     - systemctl disable firewalld

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

The eBUS SDK for JAI and eBUS Python for JAI API are installed in the following directory:

- `/opt/jai/ebus_sdk/<distribution_targeted>/`
  - For example: `/opt/jai/ebus_sdk/Ubuntu-20.04-x86_64`

Enabling MP4 Video Saving in eBUS Player for Linux (Optional):

The following step is only needed if the eBUS SDK application, such as eBUS Player, requires MP4 video saving functionality.

1. Install required dependencies
   - For Ubuntu 18.04/20.04/22.04 LTS (x86_64) and JetPack 4.6.3/5.1
     - sudo apt-get install libswscale-dev libavcodec-dev libavformat-dev
   - For RHEL 8 and CentOS 8 Stream
     - sudo yum install ffmpeg-devel
2. Navigate to `/opt/jai/ebus_sdk/<distribution_targeted>/samples`.
3. Make a copy of the following directory, as a backup for the original source files:
   - `/opt/jai/ebus_sdk/<distribution_targeted>/share/samples`
4. Navigate to the directory that contains the copy of the sample code that you have "write" access.
5. Edit sample.Makefile by adding the following highlighted code in the locations shown below:
   - For Ubuntu 18.04/20.04/22.04 LTS (x86_64) and JetPack 4.6.3/5.1

```
endif
CFLAGS    += -D_UNIX_ -D_LINUX_ -fPIC -std=c++11
CPPFLAGS  += -D_UNIX_ -D_LINUX_ -DQT_GUI_LIB -fPIC -std=c++11

LDFLAGS             += -L$(PUREGEV_ROOT)/lib          \
                       -lPvAppUtils                   \
                       -lPtConvertersLib              \
                       -lPvBase                       \
                       -lPvBuffer                     \
                       -lPvGenICam                    \
                       -lPvSystem                     \
                       -lPvStream                     \
                       -lPvDevice                     \
                       -lPvTransmitter                \
                       -lPvVirtualDevice              \
                       -lPvPersistence                \
                       -lPvSerial                     \
                       -lPvCameraBridge

LDFLAGS += -lswscale \
           -lavcodec\
           -lavformat\
           -lavutil

CPPFLAGS  += -DPV_ENABLE_MP4
```

   - For RHEL 8 and CentOS 8 Stream

```
endif
CFLAGS    += -D_UNIX_ -D_LINUX_ -fPIC -std=c++11
CPPFLAGS  += -D_UNIX_ -D_LINUX_ -DQT_GUI_LIB -fPIC -std=c++11

LDFLAGS             += -L$(PUREGEV_ROOT)/lib          \
                       -lPvAppUtils                   \
                       -lPtConvertersLib              \
                       -lPvBase                       \
                       -lPvBuffer                     \
                       -lPvGenICam                    \
                       -lPvSystem                     \
                       -lPvStream                     \
                       -lPvDevice                     \
                       -lPvTransmitter                \
                       -lPvVirtualDevice              \
                       -lPvPersistence                \
                       -lPvSerial                     \
                       -lPvCameraBridge

LDFLAGS += -lswscale \
           -lavcodec\
           -lavformat\
           -lavutil

CPPFLAGS  += -D __STDC_CONSTANT_MACROS -D PV_ENABLE_MP4 -I /usr/include/ffmpeg
```

6. Navigate to /opt/jai/ebus_sdk/<*distribution_targeted*>/samples/eBUSPlayer/
7. Type "make" to recompile.

Release Notes

These release notes pertain to release 6.4.0 of the eBUS SDK.

Note: On the Windows operating system, if you uninstalled eBUS SDK 5.1.10 (or earlier) and you had previously placed eBUS SDK receive license files in the **Licenses** folder, you must do the following:

1. In Windows Explorer, move the eBUS SDK receive license files from one of the following locations:

   32-bit operating systems: \Program Files\Pleora Technologies Inc\eBUS SDK\licenses

   64-bit operating systems: \Program Files (x86)\Pleora Technologies Inc\eBUS SDK\licenses
2. To the following location (the location is now the same for 32-bit and 64-bit Windows operating systems):

   \Program Files\JAI\eBUS SDK\licenses
3. Delete the original folder (\Program Files\Pleora Technologies Inc\eBUS SDK\licenses or \Program Files (x86)\Pleora Technologies Inc\eBUS SDK\licenses). It is no longer required.

What's new in Release 6.4.0?

eBUS SDK 6.4.0 is the latest eBUS GA release. Release 6.4.0 provides the following new functionality:

- Introduced support for de-compression of Pleora's **RapidPIX** lossless compression technology, currently for the **NTx-Mini-LC** embedded video interfaces.
  Note: When moving from the previous version to eBUS SDK 6.4.0, you must uninstall and then install.
  - Added new classes: **PvCompressionFilter**, **PvDecompressionFilter**, **PvPleoraCompressed** to support Pleora lossless de-compression. Issue ID EBUS-5444.
  - Added new **GetPleoraCompressed** method in **PvBuffer**. This method returns the **PvPleoraCompressed** interface of the buffer when the payload type is **PvPayloadTypePleoraCompressed**. Issue ID EBUS-5444.
  - Added lossless decompression support in **eBUS Player**. Issue ID EBUS-5442.
  - Added the functionality to save lossless compressed images with Pleora Compressed format (PTC1) from **eBUS Player**. Issue ID EBUS-5472.
  - Added the functionality to save decompressed images with Pleora Decompression (BIN) from eBUS Player. Issue ID EBUS-5520
  - Added **PTC Evaluation Tool** with eBUS SDK installer and **eBUS Player** toolkit installer. The tool is ONLY available on Windows 64-bit. Issue ID EBUS-5443.
  - Extended C++, .NET and Python **PvStreamSample** and **PvPipelineSample** to illustrate how to decompress an image. Issue ID EBUS-5441.
- Added new C++ and Python **EventSample** to show how to handle incoming events and event data. Issue ID EBUS-5340.
- Added PNG file saving support to eBUS SDK/**eBUS Player**. Issue ID EBUS-5399.
  - Made the PNG format the default image saving format in eBUS Player. Issue ID EBUS-5399.
- Extended Python API support on Windows to include Python 3.11. Issue ID EBUS-5305.
- Extended GenICam File Transfers for **SoftDeviceGEV** application to read out the eBUS SDK log files. Issue ID EBUS-5331.

Fixed in Release 6.4.0

- Extended the minimal chunk size supported when transmitting a multi-part container with **eBUS Edge**. Issue ID EBUS-5410.
- Fixed a property of eBUS Edge's **TriggerSoftware** feature. **TriggerSoftware** is now configured to not be streamable. Issue ID EBUS-5448.
- Changed the default verbosity of all **Keylok** logging to info in **PtUtilsLib** subsystem. Issue ID EBUS-5496.
- Removed **PvDotNet.dll** and **PvGUIDotNet.dll** in Binaries folder when installing **eBUS SDK** 32-bit on Windows. Issue ID EBUS-5548.
- Upgraded Linux kernel support to version 6.5 on Ubuntu 22.04.3 LTS for building the **eBUS Universal Pro** driver. Issue ID EBUS-5550.
- Removed the macro definition associated to **PvPixelMono8Signed** type in the eBUS SDK C++ API documentation. Issue ID EBUS-5501.
  Note: For more information regarding the required installation steps to update the driver on an Ubuntu 22.04 system running Kernel 6.5.0 refer to eBUS SDK Linux Quick Start Guide.

eBUS SDK 6.3.3

What's new in Release 6.3.3?

- Provided bug fixes.

Fixed in Release 6.3.3

- Implemented inter-process synchronization to USB key dongle for eBUS licensing, with added robustness to use-cases involving multiple eBUS applications to run concurrently. Issue ID EBUS-5272.

eBUS SDK 6.3.2

What's new in Release 6.3.2?

This is an engineering load release of eBUS SDK 6.3.2 provided bug fixes.

Fixed in Release 6.3.2

- Unblocked the image acquisition of SoftDeviceGEV when using hardware trigger. Issue ID EBUS-5464
- Fixed an issue of eBUS SDK 6.3.1.6545 to install and load the eBUS Universal Pro driver on Windows 7. Issue ID EBUS-5455.

eBUS SDK 6.3.1

What's new in Release 6.3.1?

eBUS SDK 6.3.1 is a Controlled Introduction (CI) release. Release 6.3.1 provides the following new functionality:

- Updated support of eBUS SDK on linux-arm for Jetpack 5.1 to Jetpack 5.1.1
- Added support of eBUS Python on linux-arm for Jetpack 4.6.3 with Python 3.8. Issue ID EBUS-5223
- Added support for the large leader/large trailer extension for the multi-part payload from GigE Vision 2.1 for both eBUS Edge and eBUS Receive. Issue ID EBUS-5277, EBUS-4937
    - Added new C++ function in PvBuffer class (IsLargeLeaderTrailerEnabled()) to check if the Large Data Leader and Trailer is enabled on the buffer which contains Multi-part.
    - Added new virtual C++ function (SetLargeLeaderTrailerEnabled) in classes IPvStreamingChannelSource and PvStreamingChannelSourceDefault to set whether the streaming source which transmit multi-part data uses large data leader and trailer packets.
    - Improved the SoftDeviceGEVMultipart samples (C++ and Python) to support the use of large data leader and trailer packets.
- Moved PvH264Decoder class from PvAppUtils library to the new PvCodec library on Linux and Windows. Issue ID EBUS-5275.
- Added chunk data payload support to eBUS Edge. Issue ID EBUS-5231
- Added new C++ and Python samples (**SoftDeviceGEVChunkData**) to transmit continuously chunk data payload. Issue ID EBUS-5231

- Added support of PvStream::SetUserModeRxPacketBufferSize on Windows to allow increasing eBUS receive packet buffer size when in User Mode. Issue ID EBUS-5371.
- Added support on Windows of PvStreamGEV::SetUserModeSocketRxBufferSize() and PvStreamGEV::GetUserModeSocketRxBufferSize() to allow setting the socket receive buffer size in User Mode. Issue ID EBUS-5377.
- Added GenICam file transfer code to SoftDeviceGEV sample. Issue ID EBUS-5230
- Added a new C++ method in PvImage class (HasWatermark()) to check if a watermark was applied on the image. Issue ID EBUS-4943

Fixed in Release 6.3.1

- Resolved the issue to connect to eBUS Edge when multi-part and chunk are enabled. Issue ID EBUS-5318
- Resolved the issues to pass the latest GigE Validation Framework with eBUS Edge. Issue ID EBUS-5311.
- Corrected the MISSING_PACKETS error code reported from Data Receiver to RESEND_FAILURE when packets expected are unavailable or IMAGE_ERROR when packets are received with an overrun error. Issue ID EBUS-4301.
- Resolved an occasional crash with the eBUS Universal Pro driver for USB3 Vision when the USB3 cable is disconnected from the host computer. Issue ID EBUS-5379.
- Resolved a BSOD issue when disabling the NIC, changing the NIC speed or setting the jumbo packets while eBUS SDK is connected to a GigE Vision device. Issue ID EBUS-5271, EBUS-3867 and EBUS-3891
- Resolved the Windows security issue with the publisher of the eBUS driver when updating eBUS Universal Pro for Ethernet driver from 6.2.4 to 6.3.1. Issue ID EBUS-5269.
- Fixed IPvRegister.ReadBytes in Python. Issue ID EBUS-5273
- Resolved missing SourceIDValue checks in DualSource. Issue ID EBUS-5257
- Added robustness to the Python sample to not allow opening of a stream when the selected device doesn't have SourceIDValue or SourceStreamChannel features. Issue ID EBUS-5232
- Fixed the wrong sample title observed from terminal when launching ImageProcessing.py. Issue ID EBUS-5227
- Resolved the issue in MultiSource.py sample to retrieve SourceID when using Pleora Analog-Pro device. Issue ID EBUS-5229
- Resolved the error compilation of SoftDeviceGEV C++ sample with Visual Studio 2010. Issue ID EBUS-5228
- Resolved and removed all warning messages when compiling eBUSPlayer sample with MP4 saving option. Issue ID EBUS-5080, EBUS-4914, EBUS-4631
- Increased the maximum AnswerTimeout value to 4294967295 in GigE Vision communication parameters. Issue ID EBUS-5065
- Resolved an issue to run ImageProcessing.py when opencv-python is not installed. Issue ID EBUS-5235

eBUS SDK 6.3.0

What's new in Release 6.3.0?

eBUS SDK 6.3.0 is the latest eBUS GA release. Release 6.3.0 provides the following new functionality:

- Introduces Python API support for 64-bit Operating Systems for eBUS SDK, including both eBUS Receive and eBUS Edge.
    - For a full outline of supported Python versions, see Supported Python Versions on page 23
- Updates the supported Operating Systems:
    - Introduces support for the following Operating Systems:

        - Microsoft® Windows 11 (64-bit)
        - Ubuntu 22.04 LTS (64-bit)
        - CentOS 8 Stream (64-bit)
        - RedHat 8 (64-bit)
        - NVIDIA JetPack 5.1 (Ubuntu 20.04 LTS based (64-bit)) for NVIDIA Jetson Edge ARM Devices
    - Continues to support the following Operating Systems:
        - Microsoft® Windows 10, 8.1, and 7 (32-bit or 64-bit)
        - Ubuntu 20.04 LTS (64-bit)
        - Ubuntu 18.04 LTS (64-bit)
        - NVIDIA JetPack 4.6 (Ubuntu 18.04 LTS based (64-bit)) for NVIDIA Jetson Edge ARM Devices
    - Removes support for the following Operating Systems:
        - Ubuntu 14.04 LTS (32-bit and 64-bit)
        - Ubuntu 16.04 LTS (32-bit and 64-bit)
        - CentOS / RedHat 7 (64-bit)
    - Full system requirements are provided in "System Requirements" on page 14.
- Introduces eBUS Runtime packages for Linux x86_64/ARM.
- Introduces the following new API's for eBUS Edge:
    - **IPvSoftDeviceGEV::SetUserSetNotify()** and **PvSoftDeviceGEV:: SetUserSetNotify()** were introduced as callbacks for **UserSet** state changes (i.e. **UserSetLoad** and **UserSetSave** operations). Issue ID EBUS-4936
    - New APIs **IPvRegisterInfo::IsArrayOfBytes()** and **IPvRegisterFactory::AddByteArray()** were introduced to properly handle byte-order on accessing the registers through **READREG_CMD**, **READMEM_CMD**, **WRITERREG_CMD**, and **WRITERMEM_CMD** commands, for **String** and **IRegister** feature types. Issue ID EBUS-5127, EBUS-5113
- Introduces the following new API's for eBUS Receive:
    - **PvStreamGEV::SetUserModeSocketRxBufferSize()** and **PvStreamGEV::GetUserModeSocketRxBufferSize()** were introduced to allow the user to set the socket Receive Buffer Size through socket option SO_RCVBUF when streaming to eBUS SDK's User Mode Data Receiver on Linux. Issue ID EBUS-4940.
    - **PvStream::SetUserModeRxPacketBufferSize()** was introduced to allow the user to increase the maximum eBUS SDK's receiver packet buffer size when streaming to User Mode Data Receiver when using a virtual NIC on Linux. Issue ID EBUS-5108.
    - **PvInterface::GetPleoraDriverVersion()** was added to allow the user to retrieve the eBUS driver version. Issue ID EBUS-5103.
- Added **IRegister** support for enhanced file transfer speeds using the GenICam File Transfer mechanism.
    - The **IRegister** support is now added to interface of **IPvGenApiFactory** class.
- Provided ability to override Pleora defined Trigger Features (**TriggerSelector**, **TriggerSource**, etc.) so that the users can define their own trigger features at the application layer.
    - The API can be found in **PvStreamingChannelSourceTrigger.h** class header file.
- Introduction of a new **SoftDeviceGEVTrigger** sample to illustrate how to over-ride the Pleora defined trigger features for your own custom trigger features.

Fixed in Release 6.3.0

- Increased the maximum value allowed for **DeviceLinkHeartbeatTimeout** for **SoftDeviceGEV** to 60 minutes. Issue ID EBUS-5084

- Modified the API signature from **PvStreamingChannelSourceTrigger::AddSource**(**uint32_t aindex, const char* aName**) to **PvStreamingChannelSourceTrigger::AddSource(uint32_t aIndex const PvString &aName**). Issue ID EBUS-5192

- Added a user readable min/max value in the GenICam description for nodes **FirstPacketTimeout**, **RequestTimeout**, **InterPacketTimeout**, **PreemptiveResendTimeout** and **ResetOnIdle**. Issue ID EBUS-4794
- Corrected the error code returned from **PvBufferConverter::SetConversionThreadsPriority** to **PvResult::INVALID_PARAMETER** as indicated in the documentation. Issue ID EBUS-5134
- Fixed an issue that prevents eBUS Edge from being connected to the eBUS SDK 4.1 receiver if the eBUS Edge's **ManufacturerName** and **ModelName** contain non-alphanumeric characters. Issue ID EBUS-5203
- Fixed an issue where **IPvRegister::ReadDouble()** and **IPvRegister::WriteDouble()** used the wrong byte-order. Issue ID EBUS-5086
- Fixed an issue where unreferenced parameters warning messages were displayed when compiling eBUS samples on a Linux platform. Issue ID EBUS-4917
- The maximum number of **UserSets** allowed for eBUS Edge was increased from 16 to 256. Issus ID EBUS-5067
- The **Coord3D_ABC32f** pixel format on **PvSoftDeviceGEV** now shows the correct bit depth of 96bpp. Issue ID EBUS-4957
- The **data_purpose_id** field is now mapped to the **ComponentIDValue** feature in **SoftDeviceGEVMultiPart**. Issue ID EBUS-5066

eBUS SDK 6.2.15

What's new in Release 6.2.15?

- Provides bug fixes.

Fixed in Release 6.2.15

- Fixed an issue that prevented the MP4 file from being closed properly when streaming in **MultiFrame AcquisitionMode**. Issue ID EBUS-5070
- Fixed an issue where MP4 video quality for **eBUS Player** on Linux was pixelated. An adjustable parameter to configure the MP4 saving bitrate for **eBUS Player** was added. Issue ID EBUS-5077
- Fixed an issue where **eBUS Player** was not compiling using Visual Studio 2010. Issue ID EBUS-5119
- Fixed an issue that caused the **eBUS Player** application to crash while saving images as MP4 and BMP files. Issue ID EBUS-4901
- Fixed an issue where saving an MP4 file from images with color pixel formats resulted in improper colors. Issue ID EBUS-4900

eBUS SDK 6.2.12

What's new in Release 6.2.12?

- Provides bug fixes

Fixed in Release 6.2.12

- Fixed an issue where there was no incremental activity shown on the progress bar when reading large files using eBUS Player's File Transfer (Tools menu). The progress bar now updates during the read operation for large files. Issue ID EBUS-4952

eBUS SDK 6.2.10

What's New in Release 6.2.10?

- Provides bug fixes

Fixed in Release 6.2.10

- Resolved an issue where eBUS Edge can consume too much dynamic memory. Issue ID EBUS-4879
- Resolved an issue where the interrupt link deadlocked when trying to programmatically disconnect a device that's been disconnected already (powered off or removed from the network). Issue ID EBUS-4895

eBUS SDK 6.2.8

What's New in Release 6.2.8?

- eBUS Tx is rebranded as eBUS Edge starting in Release 6.2.8.
- The following pixel formats are added to eBUS SDK:
    - Bayer[xx]14
    - Mono14p
    - Bayer[xx]14p
    - Mono32
- eBUS SDK introduces support for JetPack 4.6.
- The GenApi libraries are updated to version 3.3.0.

Fixed in Release 6.2.8

- Resolved an issue where **PvSystem::FindDevice** with a Persistent IP Address failed on the first time it was called. Issue ID EBUS-4810.
- Resolved an issue when running on Linux systems, where the Display Name attribute of an Enumeration GenICam feature was not displayed properly. Issue ID EBUS-4852.
- Corrected an issue where, on some PC's, saving the current image in .bmp format caused eBUS Player to crash. Issue ID EBUS-4819.
- Resolved an issue where the extra padding in the saved .bmp image was not removed when using eBUS Player. Issue ID EBUS-4820.
- Removed the extra dot at the end of installation path for a PUREGEV_ROOT environment variable when installing eBUS Runtime package. Issue ID EBUS-4862.
- Resolved an issue where the Visual_C++_2015-2019_Redistribution package had to be installed prior to installing eBUS SDK 6.2 on Windows 7, 8.1, and 10 (32-bit and 64-bit). Issue ID EBUS-4771.
- Resolved an issue where, occasionally when data packets for a block were not received, the eBUS Data Receiver asked for the corresponding lost packets as expected however, in some cases the requested packet ID

was for a packet ID which did not exist as part of the block. For example, if the packet trailer ID of block was 36, the data receiver requested packet ID 37 which was unexpected. Issue ID EBUS-4315.

- Corrected an issue where when a **PvSoftDeviceGEV**-based application failed to resend packet data, nothing was sent (whereas the GigE Vision standard requires that the **PvSoftDeviceGEV** send a streaming data packet with a header that includes the status). Issue ID EBUS-4311.
- Modified the log error message to be more specific, indicating "Setting the IP configuration" when running a SoftDeviceGEV application as root. Issue ID EBUS-4798.
- Corrected an issue where on the Windows 7 operating system, a message would appear when installing the eBUS SDK that indicated that the publisher of the eBUS Universal Pro driver cannot be verified. Issue ID EBUS-4188.
- Fixed an issue on Windows where the saved BMP image size is larger than expected when the width of the image is not divisible by 32. A black strip padding is added at the right side of the image. Issue ID EBUS-4734.
- Corrected an issue where when accessing the C++ Code Samples and .NET Code Samples pages (index.html) using the Microsoft Edge browser, clicking the links to the sample applications would not open the corresponding folder. Issue ID EBUS-4135.

eBUS SDK 6.2.4

What's New in Release 6.2.4?

- eBUS SDK is improved in Release 6.2.4 to reduce CPU consumption, thereby enabling additional on-board processing capabilities to be leveraged for eBUS Tx applications deployed on embedded devices for 3D Linescan applications. These enhancements ensure that sufficient resources are available on the embedded system for accurate triggering in Linescan applications, as well as ensuring that all 3D data analysis (for example, point cloud calculation) can be performed on the embedded device.
- eBUS SDK introduces support for Ubuntu 20.04 LTS (64-bit) on x86 platforms in Release 6.2.4.
- eBUS SDK Release 6.2.4 adds Visual Studio 2019 to the list of supported integrated development environments for Windows application development.
- In previous eBUS SDK versions (version 4.0 to 6.1) stopping and restarting the eBUS Daemon was required when activating a license on a Linux OS. eBUS SDK is enhanced in Release 6.2 to remove the eBUS Daemon for the Linux operating system. As of eBUS SDK Release 6.2, you must simply close and re-launch your application after applying a license on the Linux OS for the license to be activated.

Fixed in Release 6.2.4

- Corrected an issue where the data in the Bulk serial ports receive buffer is not flushed after reading. This can cause stale date to be read back until new data is received. Issue ID EBUS-4732.
- Corrected an issue that did not allow for packet resends to be enabled for Video Server API based applications. Issue ID EBUS-4685.
- Corrected an issue where a high CPU load could be seen with an eBUS Tx device when a triggered acquisition was performed and it was not streaming. Issue ID EBUS-4669.
- Corrected an issue that prevented the use of the display window with the MultiSource sample. Issue ID EBUS-4589.
- Corrected an issue when saving the Destination Port and Tiling modes in the configuration file of .NET TransmitTiledImages sample. Issue ID EBUS-4526.
- Updated the MulticastSlave .NET sample, to ensure the display rate is properly updated when receiving images. Issue ID EBUS-4522.
- libMedia files are no longer installed with eBUS SDK on all supported Ubuntu distributions. Issue ID EBUS-4447.
- PvSystem::GetU3VSupportedVersion() now returns the proper version of the USB3 Vision specification. The eBUS SDK supports version 1.0 of the USB3 Vision specification. Issue ID EBUS-4443.

- Corrected an overflow issue with eBUS Player that prevented bitmap (BMP) images from saving properly when the payload size was larger than 64 Mbytes. Issue ID EBUS-4410.
- Corrected an issue where you could save bitmap files (not TIPP or raw images) using the Save Current Image option on the Tools menu. Issue ID EBUS-4307.
- Corrected an issue with the size available for RxBufferSize of PvDeviceSerialPort. Previously, the buffer size had to be increased to ensure all data was received. Issue ID EBUS-4249.
  Corrected an issue where the wrong units were used internally for the ResendRequestTimeout of the data receiver. Issue ID EBUS-4426.
- Corrected an issue on the Windows 7 operating system, where a message appeared when installing the eBUS SDK indicating that the publisher of the eBUS Universal Pro driver could not be verified. Issue ID EBUS-4188.
- Fixed an issue on Windows where the saved BMP image size is larger than expected when the width of the image is not divisible by 32. Issue ID EBUS-4734.

eBUS SDK 6.1.8

What's New in Release 6.1.8?

- Provides an eBUS SDK 6.1.8 installation package for the 64-bit Jetson Linux ARM platforms, as described in "Installation Instructions" on page 1.

eBUS SDK 6.1.7

What's New in Release 6.1.7?

- Adds new methods for **IPvSoftDeviceGEVInfo** to access the **DeviceFirmwareVersion** register, and to replace the default Pleora eBUS SDK version number with a user-defined version number. Issue ID EBUS-4476.

Fixed in Release 6.1.7

- Changed the name of the **UserSetsControl** category to **UserSetControl**, for compliance with the *GenICam Standard Features Naming Convention* (SFNC). Issue ID EBUS-4456.

eBUS SDK 6.1.5

What's New in Release 6.1.5?

- Provides bug fixes.
- Increases the maximum number of streams from 32 to 64.
- Provides eBUS SDK 6.1.5 installation packages for the Microsoft Windows 10, 64-bit operating system.

Fixed in Release 6.1.5

- Corrected an issue that prevented you from re-enabling the eBUS Universal Pro for Ethernet driver in the eBUS Driver Installation Tool's **Network Adapter Configuration** dialog box (accessible by clicking **Advanced**). Issue ID eBUS-4468.

- Corrected an issue that caused the **PvDotNet.dll** to fail with an unhandled exception when trying to select a Video Server API transmitter from a **PvGUIDot.PvDeviceFinderForm** object. **Note:** The Video Server API is not recommended for new designs. Issue ID EBUS-4465.

- Corrected an issue that prevented you from enabling the eBUS Universal Pro driver for a network adapter with the **EbSetupLib** sample. Issue ID EBUS-2932.

eBUS SDK 6.1.4

What's New in Release 6.1.4?

- Provides bug fixes.

Fixed in Release 6.1.4

- Upgraded Linux kernel support to version 5 (verified with version 5.3.6) on Ubuntu 18.04 LTS for building the eBUS Universal Pro driver. Issue ID EBUS-4347.
- The **PvSoftDeviceGEV** now checks for IP address conflicts before using a persistent IP address. Issue ID EBUS-4303.
- An unhandled segmentation fault no longer occurs when restoring default preferences with eBUS Player on Linux. Issue ID EBUS-4280.
- Corrected a pixel conversion issue when saving from **RGB12V1Packed** format to TIFF. Issue ID EBUS-4417.
- Fixed an issue with the eBUS Tx API that prevented you from creating a SwissKnife float feature with the source selector as the only variable. Issue ID EBUS-4372.
- Corrected an issue that prevented the GigE Vision event channel communication with the firewall enabled. Issue ID EBUS-4397.
- Added the following baud rates to **PvSerialBridge**: 230400, 460800, 921600. Issue ID EBUS-4362.
- The NetCommand C++ sample and binary are no longer included with the eBUS SDK. Issue ID EBUS-4242.

eBUS SDK 6.1.2

What's New in Release 6.1.2?

Introduces the following new functionality and provides bug fixes.

- For integer and float GenICam XML features in the eBUS Tx API, this release adds the ability to set the minimum, maximum, increment, and value attributes to point to other GenApi nodes, similar to a Swiss Knife. For example, when implementing a binning feature, you may want the minimum, maximum, and increment attributes to be dependent on the **Width**, **Height**, **OffsetX**, and **OffsetY** features.
- The following new **IPvGenApiFactory** functions have been added:
    - To override the existing **PvSoftDeviceGEV**-managed features: **SetPMinFor**, **SetPMaxFor**, **SetPIncFor**, and **SetPValueFor**.
    - To set the value for the minimum, maximum, increment, and value for custom features: **SetPMin**, **SetPMax**, **SetPInc**, and **SetPValue**. Issue ID EBUS-4305.

Fixed in Release 6.1.2

- Custom device level features implemented with **PvSoftDeviceGEV** are now saved to the userset. Issue ID EBUS-4302.
- **PvSoftDeviceGEV**-based applications will now successfully pass version 2.1.2 of the GigE Vision Validation Framework on the Windows 7 operating system. Issue ID EBUS-4276.
- In the eBUS Driver Installation Tool, when you click **Help** > **About**, the available driver version is now reported for the eBUS Universal Pro Driver for USB3 Vision. Issue ID EBUS-4264.
- Corrected an issue that prevented packets from being resent when using the **PvSoftDeviceGEV** in legacy mode (**GevGVSPExtendedIDMode** is set to **Off**). By default, extended ID mode is enabled. Issue ID EBUS-4281.

eBUS SDK 6.1.1

What's New in Release 6.1.1?

Introduces new functionality, provides bug fixes, and upgrades the supported Linux for ARM platforms and operating systems, as listed below.

- In the eBUS Tx API:
    - Adds support for the multi-part payload type. The multi-part payload type is specified by the GigE Vision standard (version 2.1 and later). It lets you bring together multiple types of data and transmit them together in a single block. This is useful for applications where you want to keep related data together, so it can be processed by a single receiver.
    - Adds userset capability, which allows you to persist the configuration of your GigE Vision device across power cycles and restarts.
    - Adds the ability in the eBUS Tx API to add invalidators to **PvSoftDeviceGEV**-owned features.
    - Adds the ability to create custom registers and GenApi features with the eBUS Tx API on a per-streaming source basis (in addition to the existing ability to create global custom registers and GenApi features).
    - Adds a new **AddEnumEntry** overload to **IPvGenApiFactory** that accepts a custom display name and GenICam namespace for the enumeration entry.
    - Updates the mechanism that is used to create the GenICam XML file for the **PvSoftDeviceGEV** sample application. The GenICam XML file is now compressed and provided to the controller application as a zip file. The zip file is automatically created on **PvSoftDeviceGEV Start**, immediately after the GenICam XML file is generated.
    - Introduces a backup implementation of the IP configuration module. If the **PvSoftDeviceGEV** cannot initialize the COM stack as required, it will now use a simpler IP configuration module. When in this mode, the behavior of the **PvSoftDeviceGEV** is similar to running the application WITHOUT Administrator mode: **FORCEIP_CMD**, static IP configuration, control on DHCP, and LLA will not be available. However, the **PvSoftDeviceGEV** will start.
    - Extends the **IPvGenApiFactory** interface:
        - Adds GenICam XML SwissKnife support with the **AddVariable**, **CreateIntSwissKnife**, and **CreateFloatSwissKnife** methods.
        - Adds GenICam XML Converter support with the **AddVariable**, **CreateIntConverter**, and **CreateFloatConverter** methods.

Note: A SwissKnife integer/float is a GenApi read-only feature that resolves a formula whenever it is read. The formula is defined as a string and evaluates to an integer/float. A converter integer/float is a read-write GenApi construct that allows reading from and writing to a referenced feature using SwissKnife-like formulas.

- Adds unit attribute assignment for integer and float GenICam XML features using the **SetUnit** method.
- Adds **pValue** feature support with the **SetPValue** method. A **pValue** replaces the register feature implementation with a link to another feature.
- Adds **pIsAvailable** feature attribute support with the **SetPIsAvailable** method. The **Is Available** attribute of a feature is used to represent a temporary state of unavailability for a feature. For example, a SwissKnife feature could be used to control whether a feature is available or not.

  These additions are considered to be advanced GenICam XML concepts. For detailed information, see the *eBUS SDK C++ API CHM* file. These additions are also demonstrated in **MyEventSink.cpp** in the **SoftDeviceGEV** sample.
- Adds two new methods to the **IPvStreamingChannelSource** interface: **IsPayloadTypeSupported** and **SetTestPayloadFormatMode**. These methods must be implemented for all multi-part stream sources.
  - **IsPayloadTypeSupported** must return **true** when **aPayloadType** is **PvPayloadTypeMultiPart**. It must return **false** otherwise.
  - **SetTestPayloadFormat** must set the stream source to test mode when **aPayloadType** is **PvPayloadTypeMultiPart**. It must set the device back to normal operation mode when it is **PvPayloadTypeNone**.
  - The **SoftDeviceGEVMultiPart** sample application shows how these two new methods should be implemented.

- Adds support for zipped folders to the Pleora Firmware Updater utility (in addition to existing .DFW file support), in preparation for a new firmware update mechanism that will be introduced for Pleora video interfaces in the upcoming months. The Pleora Firmware Updater utility is installed in the following location as part of the eBUS SDK installation: C:\Program Files\Common Files\Pleora\eBUS SDK. On 32-bit systems, the utility is located in C:\Program Files (x86)\Common Files\Pleora\eBUS SDK.

Fixed in Release 6.1.1

- Addressed the following issues that could occur when using the **DualSource** sample application to connect to a software-based GigE Vision device created with the **SoftDeviceGEVSimple** sample application:
  - Corrected an unhandled exception that could occur when selecting an acquisition mode in the **Acquisition Mode** list. Issue ID EBUS-4149.
  - Corrected the mapping between the **PvSoftDeviceGEV** streaming sources and the **AcquisitionMode** list in the **DualSource** sample application code. Issue ID EBUS-4149.
  - Corrected an unhandled exception that could occur when clicking **Stop** after losing connection to the **PvSoftDeviceGEV** while streaming. Issue ID EBUS-4150.
- Corrected an issue that caused the **Acquisition Mode** list in the **SimpleGUIApplication** C++ sample application to be disabled when connected to a software-based GigE Vision device created with the **SoftDeviceGEVSimple** sample application. Issue ID EBUS-4151.
- The **PUREGEV_ROOT** environment variable is now created when installing the eBUS SDK Runtime package. Issue ID EBUS-4168.

- On the Linux operating system, corrected an issue that prevented eBUS Player from saving some default communication preferences when clicking **Tools** > **Default GigE Vision Communication Parameters** or **Default USB3 Vision Communication Parameters**. Issue ID EBUS-4259.
- Corrected an issue that prevented the stream statistics from being updated in the .NET version of the **PvStreamSample**, **PvPipelineSample**, and **ImageProcessing** sample applications. Issue ID EBUS-4261.
- **PvSoftDeviceGEV** events now work properly with the **OnEventGenICam** callback. Issue ID EBUS-4185.
- Instead of using the specified **AnswerTimeout** and **CommandRetry** values, the default values are used (**AnswerTimeout** resets to 1000 and **CommandRetry** resets to 3). Issue ID EBUS-4162.

eBUS SDK 6.0.2

What's New in Release 6.0.2?

This is the first eBUS SDK 6.x GA release. It provides bug fixes, introduces new functionality, and upgrades the supported operating systems, as listed below:

- Adds eBUS Tx functionality, which allows users to create a software-based GigE Vision device with full GVCP and GVSP support. This functionality is implemented by a new set of classes and functions that coexist alongside the existing Video Server API transmitter classes and functions. eBUS Tx is supported on the Windows and Linux operating systems (although the Windows 7 operating system is not recommended for new eBUS Tx designs).

  Please note that the Video Server API and associated sample applications are not recommended for new designs. We recommend that you use the new eBUS Tx API (**PvSoftDeviceGEV**) when developing new transmitter applications.
- Adds a separate installation package for the eBUS Player Toolkit, so you can install the eBUS Player application independent of the eBUS SDK.
- Upgrades GenAPI support to 3.1.0.
- Updates the supported operating systems:
  - Continues to support Microsoft® Windows 10, 8.1, and 7 (32-bit or 64-bit). Removes support for Windows Server 2008 and 2012.
  - Upgrades x86 Linux support to Ubuntu 18.04 LTS, 64-bit (in addition to Ubuntu 14.04 and 16.04 LTS, 32 and 64-bit), using a single installation package. Continues to support x86 Linux for RHEL 7 and CentOS 7, 64-bit.
  - Full system requirements are provided in "Installation Instructions" on page 1.
- Updates the supported Integrated Development Environments (IDEs):
  - Adds support for Visual Studio 2017 (in addition to Visual Studio 2015, 2013, 2012, and 2010). To compile the .NET samples, version 4.6 of the .Net Framework is required.
  - Full system requirements are provided in "Installation Instructions" on page 1.
- Introduces two new eBUS SDK licensing packages: **GEV-Tx License File** and **Developer Seat license for eBUS SDK**. For more information, see the eBUS SDK Datasheet, available at https://www.pleora.com/products/ebus-sdk/.

  Note: There has been no change to the existing licensing packages. Existing licenses continue to be supported in eBUS SDK 6.x.

Fixed in Release 6.0.2

- Corrected an issue that caused the chunk ID and length for the image data to be added to the payload twice when using **PvImage::Attach** and **PvBuffer::AddChunk**, which resulted in a data payload that was twice the expected size. Issue ID EBUS-4009.
- Resolved a timeout issue that introduced a delay in returning from **PvTransmitter::Close()**. Issue ID EBUS-4031.
- Made the **TimestampLatch** and **TimestampReset** features available for GigE Vision devices created with the **SoftDeviceGEV** sample application. Issue ID EBUS-4183.
- Includes bug fixes from earlier releases that were not made generally available:
    - Added two new functions to **PvDeviceInfoGEV** (**GetIPConfigCurrentString** and **GetIPConfigOptionsString**), as specified by the GigE Vision standard. The device information **(ip config options** and **ip config current**) now appears in the device finder. Issue EBUS-3955.
    - Fixed a deadlock when simultaneously handling GenICam events and performing serial writes. Issue ID EBUS-3943.
    - Corrected an issue that prevented GenICam events from returning parameter data. Issue ID EBUS-3944.

Important Note about eBUS SDK Licenses for Users of eBUS SDK 5.0.2 (or Earlier)

In eBUS SDK 5.1.2, the eBUS SDK installation path was changed on the Windows operating system, including the location of the **Licenses** folder. If you were using eBUS SDK 5.0.2 (or earlier) and you had previously placed eBUS SDK receive license files in the **Licenses** folder, you must do the following:

1. In Windows Explorer, move the eBUS SDK receive license files from one of the following locations:
    - **32-bit:** \Program Files\Pleora Technologies Inc\eBUS SDK\Licenses
    - **64-bit:** \Program Files (x86)\Pleora Technologies Inc\eBUS SDK\Licenses

    To one of the following locations, depending on which release 6.x package you installed:

    - **eBUS Player Toolkit:** \Program Files\Pleora Technologies Inc\eBUS Player\Licenses
    - **eBUS SDK:** \Program Files\Pleora Technologies Inc\eBUS SDK\Licenses

2. On 64-bit operating systems, after you copy the license files, the **\Program Files (x86)\Pleora Technologies Inc\eBUS SDK\Licenses** is no longer required and can be deleted.

Known Issues

- No images are saved with Pleora Decompressed (BIN) format when the incoming data is not compressed. Issue ID EBUS-5512.
- When executing PvCompressionFilter::Execute, the output buffer is 1.5 times larger than the input buffer. In order to get only the compressed data, create a new buffer and fill this new buffer with only the Pleora Compressed data from the output buffer. Issue ID EBUS-5513.
- PTC Evaluation Tool 32-bit is not available for Windows 32-bit. Issue ID EBUS-5548.
- eBUS Python samples do not work with the Python IDLE shell because of an issue with msvcrt.getch(). You can use a CMD terminal to run eBUS Python Samples. Using an IDE such as Visual Studio Code (tested 1.76.2) to debug a Python sample is suggested. Issue ID EBUS-5242.

- The unexpected warning message "Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome Use QT_QPA_PLATFORM=wayland to run Wayland anyway" appears when starting eBUS GUI applications on Ubuntu 22.04 and CentOS 8 Stream. Issue ID EBUS-5171.
- The MultiSource.py sample fails to work with Analog-Pro due to an error retrieving SourceID. Issue ID EBUS-5229.
- The SoftDeviceGEV C++ sample does not compile in Visual Studio 2010. Issue ID EBUS-5228.
- The eBUS HTML documentation cannot be opened with a built-in Firefox application on Ubuntu 22.04. Follow the steps in https://www.omgubuntu.co.uk/2022/04/how-to-install-firefox-deb-apt-ubuntu-22-04 OR copy the documentation to a folder in your user space before opening it. Issue ID EBUS-5195.
- Prior to installing eBUS SDK 6.4 on Linux x86 platforms, you must first uninstall previous versions of the eBUS SDK which are installed on your system. Issue ID EBUS-4745.
- On Ubuntu, the **eBUS Player** application could freeze if you are using the connection recovery mechanism while connected to a USB3 Vision that is streaming at high rates. Specifically, this can occur if the device is streaming at 65 FPS (or higher) and the USB connection is plug cycled when **LinkRecovery** is enabled. Issue ID EBUS-4772.
- When saving mp4 video in eBUS Player on a Windows 7 (64 bit) operating system, the yellow component may be discolored. Issue ID EBUS-4765.
- Image filtering of **eBUS Player** is not applied on all saved BMP images when the frame rate of the input device is high. Issue ID EBUS-4789.
- In the **GenICamParameters** .NET sample application, an unhandled exception occurs when selecting a feature that is "not available". Issue ID EBUS-4298.
- The eBUS Player histogram feature is not working properly with color images. Issue ID EBUS-4299.
- For some devices, the list of options available in the Mode list on the main page of eBUS Player may vary from those available in the **Acquisition Mode** > **AcquisitionMode** list in the eBUS Player **Device Control** dialog box. To access all acquisition modes, use the eBUS Player **Device Control** dialog box. Issue ID EBUS-3720.
- The **TransmitChunkData** sample application does not display the correct number of transmitted images. The value that is displayed is lower than the actual value. Issue ID EBUS-4330.
- When using the **SoftDeviceGEV** sample application, the **SampleString** and **SampleBoolean** features are not automatically updated when you select a different **EnumEntry** in the **SampleEnum** list. Issue ID EBUS-4341.
- **PvDevice::GetAccessType** returns a non-OK status when using **PvDevice::Connect** with **PvAccessExclusive** mode. Issue ID EBUS-4398.
- Attempting to connect to a USB3 Vision device with a non-empty **UserDefinedName** using **PvDeviceU3V** will fail. Instead, you should use **PvDevice**. Issue ID EBUS-4442.
- **PvSoftDeviceGEV::Start(const PvString &aMACAddress)** returns **true** if the **aMACAddress** is an IP address (instead of returning **false**). Issue ID EBUS-4394.
- The GenICam parameter **GevSCPD** is not fully implemented for eBUS Tx. Issue ID EBUS-4366.
- When using **u3v** option with **PleoraFirmwareUpdater**, the USB3 Vision device is not reset at the end of the process to apply the new firmware load. Upon completion of an update, a power cycle of the USB3 Vision device or executing the **DeviceReset** GenICam command is required to apply the new firmware load. Issue ID EBUS-4882.
- Loading .zip Pleora firmware files on a GigE Vision or USB3 Vision Pleora devices with **PleoraFirmwareUpdater** on Linux x86/ARM platforms is not currently supported. Issue ID EBUS-4884, EBUS-4883.

System Requirements

Supported Operating Systems

For the Windows Operating System:

- Microsoft Windows 11, 64-bit
- Microsoft Windows 10, 32-bit or 64-bit
- Microsoft Windows 8.1, 32-bit or 64-bit
- Microsoft Windows 7 with Service Pack 1 (or later), 32-bit or 64-bit

  Important note: Windows 7/8.1 (32-bit and 64-bit) and Windows 10 (32-bit) Operating Systems are not recommended for new eBUS SDK designs.

For the Linux platform:

- Red Hat Enterprise Linux 8, 64-bit with kernel version 4.18.0
- CentOS 8 Stream, 64-bit with kernel version 4.18.0
- Ubuntu 22.04 LTS, 64-bit with kernel versions 5.15.0, 5.19.0, 6.2.0 and 6.5.0
- Ubuntu 20.04 LTS, 64-bit with kernel versions 5.4.0, 5.8.0, 5.11.0, 5.13, 5.15.0, 5.19.0
- Ubuntu 18.04 LTS, 64-bit with kernel versions 4.4.0-143 and 5.4.0
  Note: The eBUS SDK is supported on the following Linux ARM platforms:
  - NVIDIA Jetson Nano, Jetson AGX Xavier, Jetson Xavier NX, Jetson TX2i, Jetson TX2 NX, and Jetson TX2 platforms (Ubuntu 18.04 with Jetpack 4.6)
  - NVIDIA Jetson AGX Xavier, Jetson Xavier NX, Jetson AGX Orin, and Jetson Orin NX and Jetson Orin Nano platforms (Ubuntu 20.04 and JetPack 5.1)

Required Hardware

The following hardware is required:

- Gigabit Ethernet network card or USB 3.0 host controller.

Pleora has validated and supports the following USB 3.0 chipsets:

- NEC Electronics/Renesas Electronics chipset
- Intel® Ivy Bridge chipset

Supported Development Environments

The following development environments are supported:

For the Windows operating system:

- For C++ API and .NET API*:
  - Visual Studio 2019, 2017, 2015, 2013, 2012, and 2010
    - * - The .NET assemblies use Version 4 of the .NET framework and require the .NET framework.
- For Python API:
  - Visual Studio Code: version 1.76.2

For the Linux operating system, Qt and qmake are required to compile GUI-based samples:

- For Ubuntu 22.04 Desktop (64-bit): Qt 5.15.3
- For Ubuntu 20.04 Desktop (64-bit): Qt 5.12.8
- For Ubuntu 20.04 for ARM (64-bit): Qt 5.12.8

- For Ubuntu 18.04 Desktop (64-bit): Qt 5.9.5
- For Ubuntu 18.04 for ARM (64-bit): Qt 5.9.5
- For RHEL 8 (64-bit) Qt 5.15.3
- For CentOS 8 Stream (64-bit): Qt 5.15.37: Qt 5.9.2

Supported Python Versions:

The following Python versions are supported:

Operating System	Python Version*					
	3.6	3.7	3.8	3.9	3.10	3.11
Windows 11/10/8.1 (64-bit)	Supported.	Supported.	Supported.	Supported.	Supported.	Supported.
Windows 7 (64-bit)	Supported.	Supported.	Supported.	Not Supported.	Not Supported.	Not Supported.
Windows 10/8.1/7 (32-bit)	Not Supported.	Not Supported.	Not Supported.	Not Supported.	Not Supported.	Not Supported.
Ubuntu 22.04 LTS (x86_64)	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Supported. Python 3.10 is installed by default for this Operating System.	Planned for in future release of eBUS SDK for JAI.
Ubuntu 20.04 LTS (x86_64)	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Supported. Python 3.8 is installed by default for this Operating System.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.
Ubuntu 18.04 LTS (x86_64)	Supported. Python 3.6 is installed by default for this Operating System.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.
RedHat 8 (x86_64)	Supported. Python 3.6 is installed by default for this Operating System.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.
CentOS 8 Stream (x86_64)	Supported. Python 3.6 is installed by default for this Operating System.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.
JetPack 4.6 (Ubuntu 18.04 based)	Supported. Python 3.6 is installed by default for this Operating System.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.
JetPack 5.1 (Ubuntu 20.04 based)	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Supported. Python 3.8 is installed by default for this Operating System.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.	Planned for in future release of eBUS SDK for JAI.

* - "eBUS Python for JAI" packages are to be installed after eBUS SDK for JAI is installed on your system.

GenApi, GenCP, GigE Vision, and USB3 Vision Support

The following table lists the supported GenApi, GenCP, GigE Vision, and USB3 Vision versions.

Table 1: GenApi, GenCP, GigE Vision, and USB3 Vision Support

Component	Supported version
GenApi (GenICam)	Version 3.3.0
GenCP	Version 1.0
GigE Vision	Version 2.1 (and earlier)
USB3 Vision	Version 1.0

For More Information

When you install the eBUS SDK, you can access documentation that describes the interfaces, classes, and functions that are available. This documentation is only available when you install the eBUS SDK (not the eBUS Player Toolkit).

Table 2: eBUS SDK Documentation (C++ and .NET APIs)

Operating system	eBUS SDK documentation location
Windows	Windows Start menu > eBUS Or: C:\Program Files\JAI\eBUS SDK\Documentation
Linux (.NET not supported)	<*installation_directory*>/share/doc/sdk/index.html

Table 3: eBUS SDK Documentation (Python API)

Operating system	eBUS SDK documentation location
Windows	PYTHON_INSTALLATION_PATH\Lib\site-packages\ebus-python\docs\index.html
Linux	/<*installation_directory*>/share/doc/python/index.html

Additional documentation is available on the Pleora Technologies Support Center (http://www.pleora.com/support-center), such as:

- *eBUS Player User Guide*, available for Windows, and Linux
- *eBUS Player Quick Start Guide*
- *eBUS SDK Quick Start Guides*, available for C++, .NET, Python, and Linux
- *Getting Started with eBUS Edge*
- *eBUS SDK 3.x to 4.x Migration Guide*
- *Vision SDK to eBUS SDK Migration Guide*
- *eBUS SDK Licensing Overview Knowledge Base Article*

Copyright Information

Document Version

RN-0046 Version 7.0, February 2024