

Helpdesk XIMEA

Portal > Knowledgebase > xiAPI & Software Package > Software Package > Linux SP Knowledge Base

Linux SP Knowledge Base

Support SK - 2022-04-11 - in Software Package

Linux SP Knowledge Base [1](#)

Usage of API [1](#)

Enumeration and Opening of cameras from multiple processes [1](#)

Symptom:

Calling of xiGetNumberDevices and/or xiOpenDevice returns unexpected results when running from multiple processes.

For example - multiple applications are started in one time and each calls the xiGetNumberDevices.

Resolution:

- Possibility 1:
 - Use only one application in one time. It can run multiple threads if needed.
- Possibility 2
 - Use one process for enumeration (using xiGetNumberDevices and xiOpenDevice) and other processes for other tasks.
 - Ensure that only one process in time can call xiGetNumberDevices and xiOpenDevice.

Application Development [1](#)

xiQ/MQ camera is not accessible after Application Crash [1](#)

Symptoms:

API returns zero on call xiGetNumberDevices after camera was opened by application that crashed before while acquisition was active.

Resolution:

- Possibility 1:
 - Disconnect and connect the camera
 - Start the application again
- Possibility 2:

- reload xhci_hcd driver, either using rmmod/modprobe or /sys/bus/pci/drivers/xhci_hcd/(un)bind interface (needs root privileges);
- Possibility 3:
 - reset the USB device by execute shell command


```
echo 1 > /sys/bus/usb/devices/4-2/remove
```

 # 4-2 is correct bus-device address for camera which can be looked up via lsusb for example
- Possibility 4:
 - use code at [usb_cam_reset_linux.cpp](#) to reset camera before starting

Trace Output Settings¶

In Linux, trace output is visible in error output (stderr).

For redirecting error output to a file use **2>** like the following:

```
./xiSample 2> stderr.txt
```

For canceling error output (removing it) simply redirect it to the null like the following:

```
./xiSample 2> /dev/null
```

Cancelling error output can also be done by setting the xiAPI parameter

XI_PRM_DEBUG_LEVEL to the value **XI_DL_DISABLED**:

```
xiSetParamInt(0, XI_PRM_DEBUG_LEVEL, XI_DL_DISABLED);
```

Changing Real-Time Clock¶

Background¶

API is using synchronization primitives, such as WaitForSingleObject. They are based on Linux/macOS on real time clock. API implements multiple mechanisms to overcome disadvantages of these implementations. However there are still some imperfections in behavior:

Behavior¶

- API versions V4.17.42 or newer:
 - If cameras delivers data periodically (e.g. one image every second) then you should notice no issue even if real-time clock is changed.
 - If cameras delivers data sporadically, your application depends on xiGetImage timeouts and real-time clock is changed to past, the timeout will increase

accordingly.

- Example: When camera does not deliver any image and time is changed by 1 seconds to past, the call of `xiGetImage(h,image,4)` take 5 seconds.
- Workaround: change real-time in smaller increments than used in `xiGetImage` calls.
- Changing of real-time clock to future does not affect the acquisition.
- API versions V4.17.41 or older:
 - Changing of real-time clock (e.g. by command `date -s`) affects acquisition of cameras. Workaround: If your system needs to change real-time, perform this before opening of first camera (`xiOpenDevice`) or after close of last camera (`xiCloseDevice`).

Tags

Linux