

# Bobcat GEV Linux Installer Manual

## Note to our customers

Thank you for buying our product. We hope that you will find it as exciting and versatile as we do. Our staff is available to answer any questions, suggestions or concerns you may have. We can be reached at:

**Email:** [techsupport@imperx.com](mailto:techsupport@imperx.com)

**Toll Free** 1 (866) 849-1662 or (+1) 561-989-0006

**Fax:** (+1) 561-989-0045

**Visit our Web Site:** [www.imperx.com](http://www.imperx.com)

## TABLE OF CONTENTS

1	Bobcat GEV Linux General Information .....	3
2	Requirements for Bobcat GEV Linux Installation Instructions .....	3
3	Bobcat GEV Installation Instructions.....	19
4	Uninstall Bobcat GEV Instructions.....	20
4	Configuration of the Network Interfaces.....	20
5	Bobcat GEV SDK samples .....	22

## 1 BOBCAT GEV LINUX GENERAL INFORMATION

---

Bobcat GEV Linux Installer includes the kernel modules for the driver and a set of sample C++ Qt projects which demonstrates the usage of Bobcat GEV eBUS SDK API functions.

## 2 REQUIREMENTS FOR BOBCAT GEV LINUX INSTALLATION INSTRUCTIONS

---

The Bobcat GEV Linux driver has dependencies on some packages prior to running the installer. Please make sure your system has the following packages: GCC, make, and Qt. Also, make sure you are login as ROOT for all instructions.

**NOTE!** Prior to installing the driver, you must disable the network firewall. This is necessary for the sample applications to be able to detect the Bobcat GEV camera.

### 2.1 NETWORK CONNECTIONS

The network connection of your Bobcat GigE Camera needs to be set. This can be confirmed by running the 'ifconfig' command from the terminal window as shown below:

```
[users@testbuilder ~]$ ifconfig
```

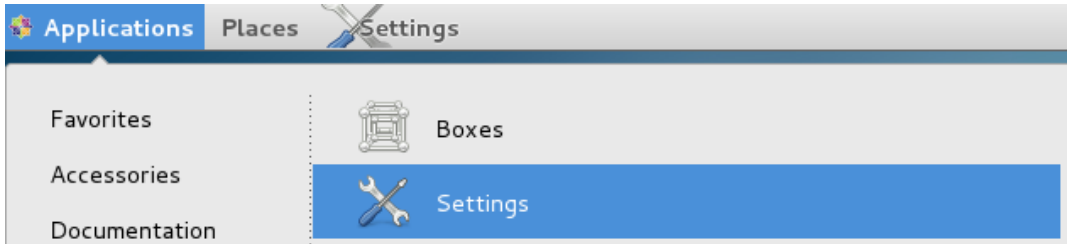
You will see your Ethernet settings. Please make sure the Ethernet setting corresponding to your Bobcat GEV camera has an Ethernet address 'inet addr'.

For example, **inet addr: 10.10.0.1**

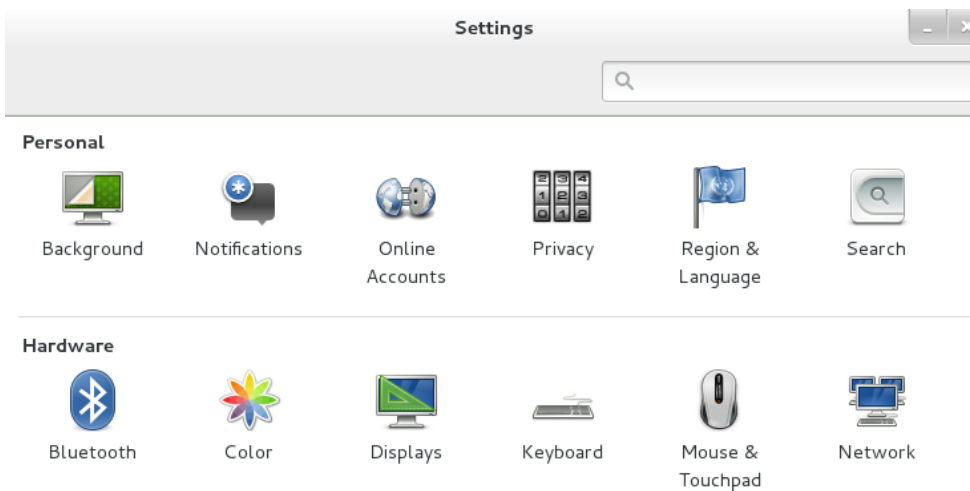
If you do not see an Ethernet address, then you will need to assign your Ethernet address manual as follows.

**For Centos 7/RHEL 64-bit operation system:**

Click on the Applications->Settings menu tab.

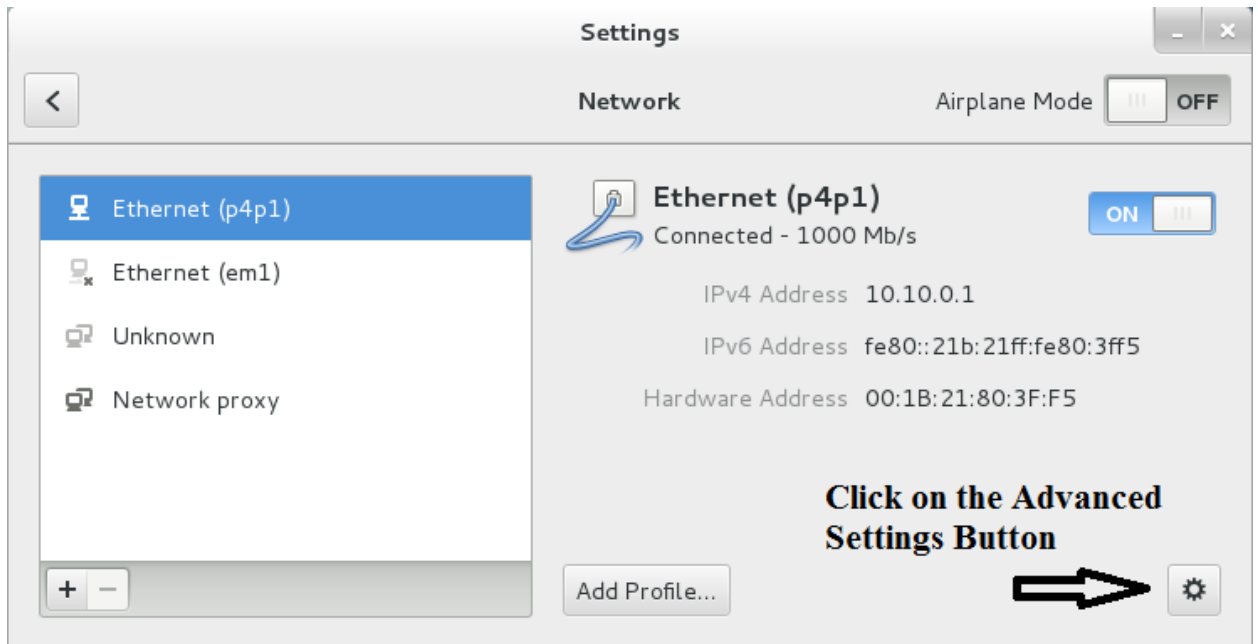


A Settings Dialog box will appear as shown below:



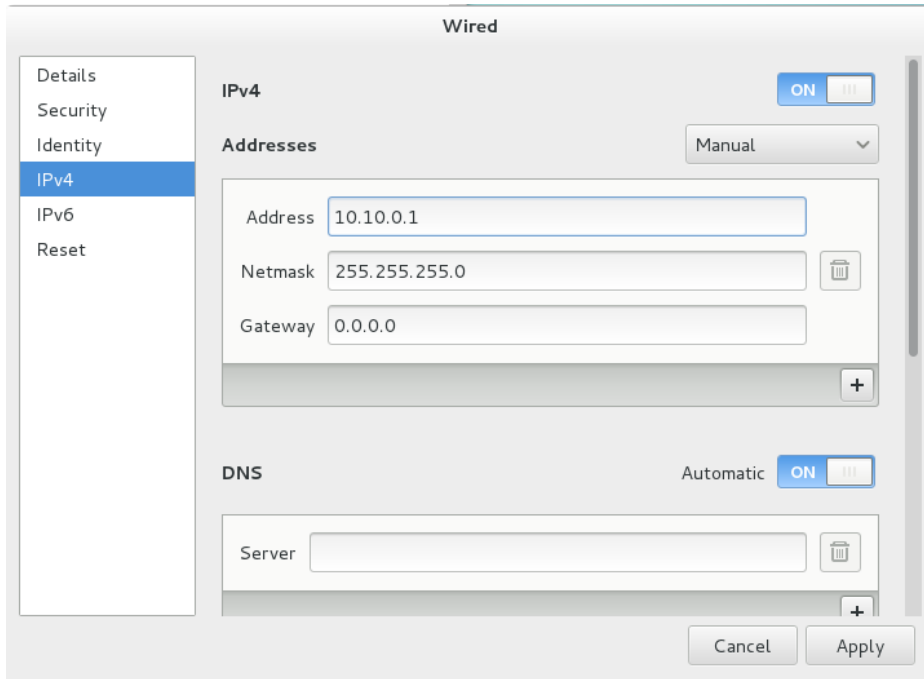
Click on the **Network** Icon.

The **Settings->Network** will appear as shown.



Click on the **'Wired'** <Ethernet name> of your Bobcat GEV Camera device. Then, click on **the Advanced Settings Button** as shown below for the Ethernet wire connection.

The Wired Dialog box will appear.



In the '**Addresses**' field, set the combo-box list to '**Manual**'. Under the '**Addresses**' field please enter the '**ethernet address**' you are assigning this Bobcat GEV camera (device). Set the '**Netmask**' to 255.255.255.0 and '**Gateway**' to 0.0.0.0. Then, press the '**Apply**' button.

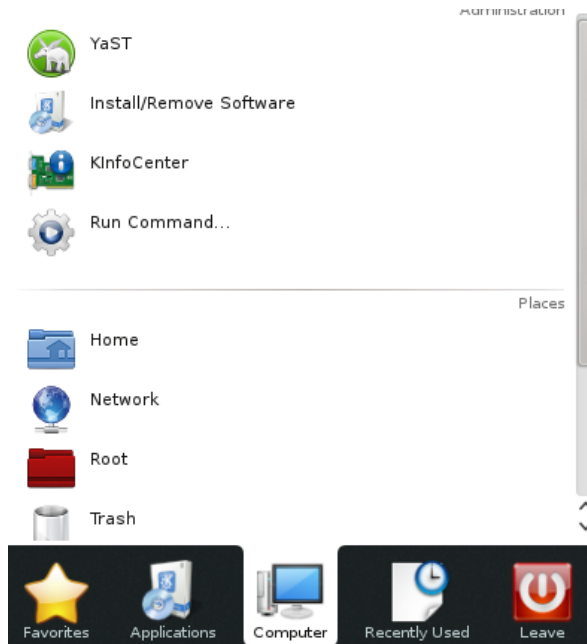
Rerun the 'ifconfig' command from the terminal. You should see your 'inet address' you assigned for your Ethernet camera. Please make sure you install or have installed the following packages prior to running the Bobcat GEV Linux installer.

**For openSUSE 64-bit operation system:**

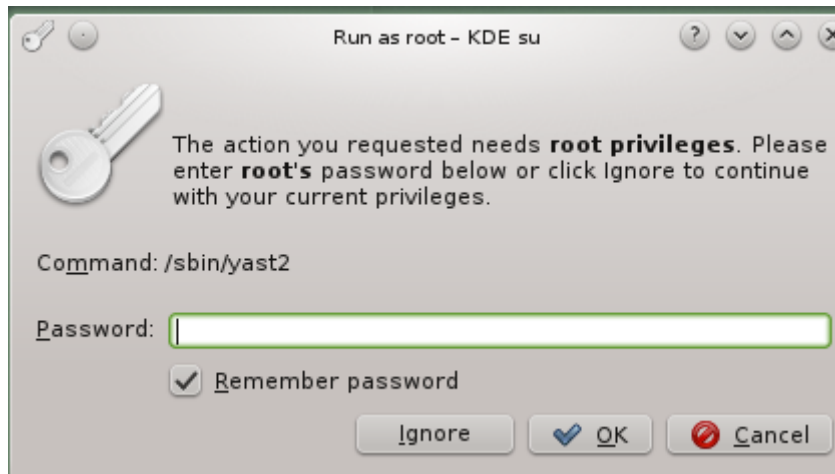
The network connection of your internet needs to be setup. Likewise, we need Bobcat GigE Camera to be set. You must run ifconfig to see if the internet is connected below:

```
[users@testbuilder ~]$ ifconfig
```

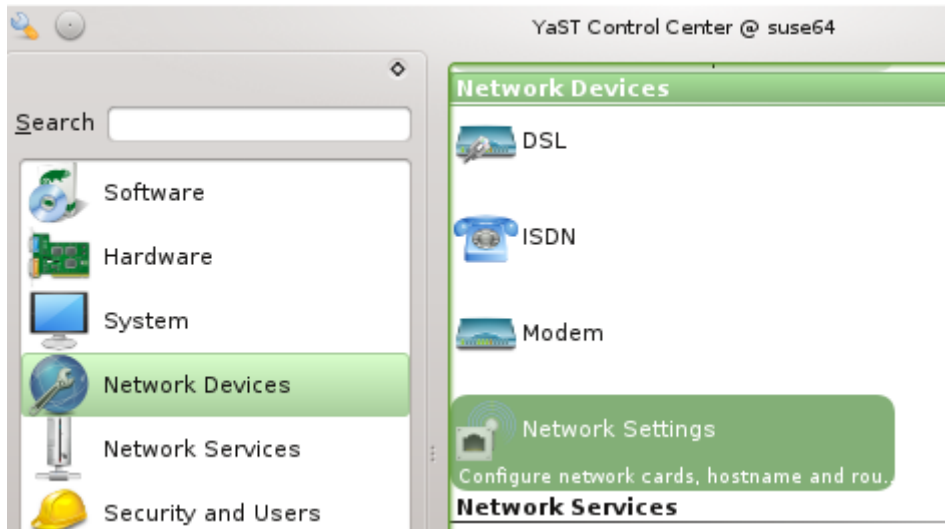
If the internet is not setup, please setup. This can be done as follows. Click on Yast



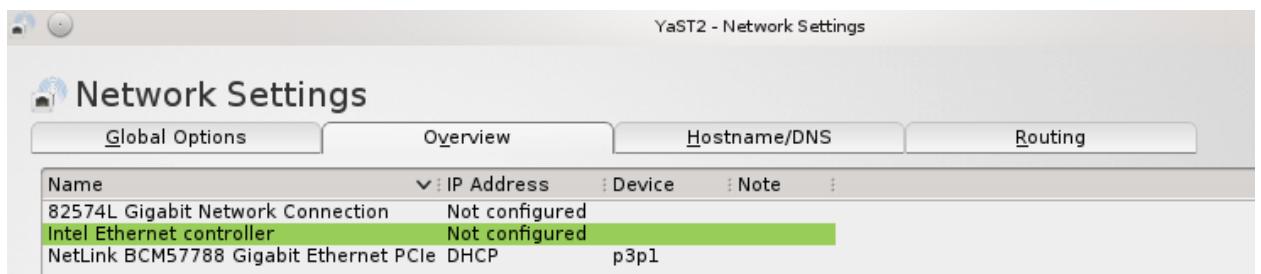
Then, enter your Root Password.



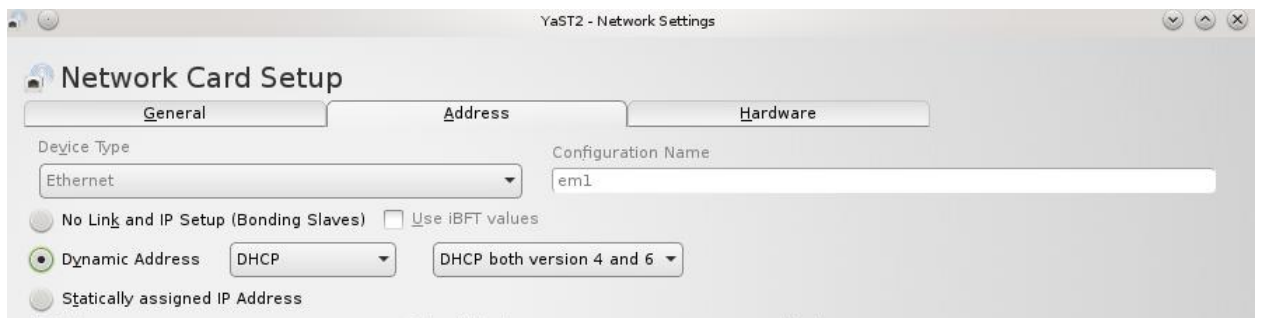
Then, Click on Network Devices -> Network Settings:



Look for device for your internet.



In my case, 'Intel Ethernet controller'. Press the 'Edit' button and set it for 'Dynamic Address' (DHCP).



Then, press OK. **REBOOT Computer for the network setting to be activated.**

Then, review routing table from the Terminal window. See if the gateway interface has been set. If not, you must go back to the Network Devices -> Network Settings and re-edit the interface with the MAC address for the internet and reset it to 'dynamic address' (DHCP) and then press 'OK'. Then, from the terminal window, recheck router table. You should see your gateway Flag 'G' setup.

```
[users@testbuilder ~]$ /sbin/route -n
```

Next, setup your Ethernet for your Bobcat Camera. In my case, I will use 'enp5s0' interface. Look at Yast->Network Devices->Network Settings to figure out the device name Network Connection of your Bobcat Camera. In my case, it is 'enp5s0'. From the terminal window, please setup your ipaddress.

```
[users@testbuilder ~]$ ifconfig enp5s0 inet 10.10.0.1 netmask 255.255.255.0
```

Verify interfaces are setup.

```
[users@testbuilder ~]$ ifconfig
```

**Note: It is important that the internet is working to download all the necessary packages to compile samples and kernel. Also, please make sure firewall is disabled. Talk to network administrator for approach.**

## 2.2 FIREWALL

The network firewall must be disabled when using the Bobcat GEV camera (device).

**NOTE:** Disabling of the network firewall is critical otherwise the application will not detect the device (camera).

### HOW TO DISABLE FIREWALL

#### For Centos 7/RHEL (64-bit):

**Step 1:** Login as ROOT user

**Step 2:** Enter the following commands to stop firewall

```
[root@testbuilder ~]$ systemctl stop firewalld
```

**For openSUSE:****Step 1:** Login as ROOT user**Step 2:** Enter the following commands to disable firewall

```
[root@testbuilder ~]$ /sbin/rcSuSEfirewall2 stop
```

**HOW TO ENABLE FIREWALL****For Centos 7 /RHEL (64-bit):****Step 1:** Login as ROOT user**Step 2:** Enter the following commands to enable firewall

```
[root@testbuilder ~]$ systemctl start firewalld
```

**For openSUSE:****Step 1:** Login as ROOT user**Step 2:** Enter the following commands to enable firewall

```
[root@testbuilder ~]$ /sbin/rcSuSEfirewall2 start
```

**RE-BOOTING COMPUTER (CENTOS 7) and Firewall ISSUES**

**Note:** When restarting computer, you might need to stop/start/stop the firewall to have access to the device (camera)

For example,

**Step 1:** `systemctl stop firewalld`

**Step 2:** `systemctl start firewalld`

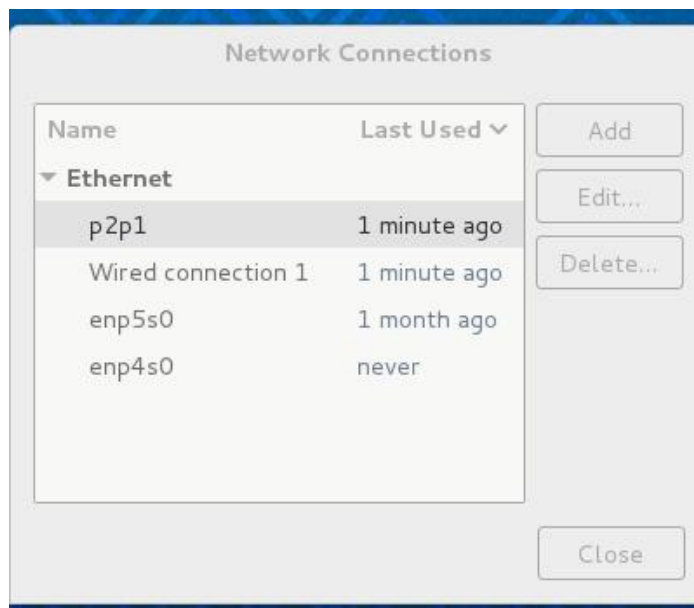
**Step 3:** `systemctl stop firewalld`

**(FEDORA ONLY) TRUSTED ZONE FOR FIREWALL**

You must change the network connection zone for the interface. Please click on the “**Options->Change Zones of Connections ...**”

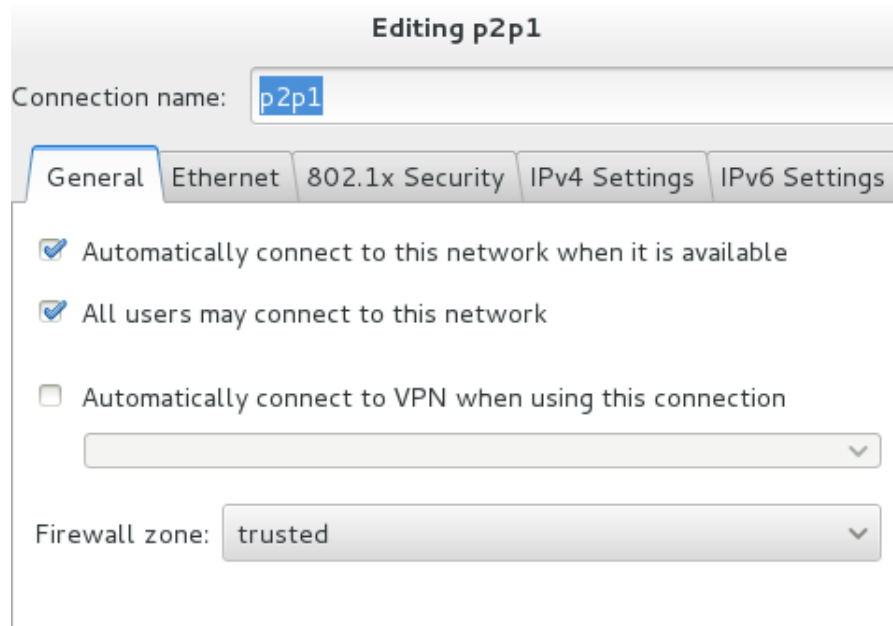


Click on your ethernet device. In my case, p2p1. If you don't see your interface name. It may be called 'Wired connection 2'. Click on the 'edit' button.



Change the Firewall Zone to 'Trusted' and then hit 'Save'.

**Note:** You can edit the Connection name to match your Ethernet device.



## 2.3 INSTALLATION PACKAGES

The following packages need to be installed in order for the kernel module and sample C++ Qt applications to be compiled and run.

### ***EXPAT (Xml) and Qt Tools***

The Xml and Qt tools are essential tools needed to run the sample applications. The following installation instructions are in **BOLD**. The installation of packages are installed from the command line from the terminal window.

#### **For Ubuntu 32-bit/64-bit:**

Installing dependency packages	Command Line
Install 'libexpat.so'	<b>\$ apt-get install libexpat1-dev</b>
Install Qt	<b>\$ apt-get install qt-sdk</b>

**For Centos 7/RHEL/Fedora 64-bit:**

Installing dependency packages	Command Line
Install 'libexpat.so.0'	<p>Please download and save on thumb drive the compat-expat1-1.95.8-8.el6.x86_64. You can download from this website:</p> <p><a href="http://rpm.pbone.net">http://rpm.pbone.net</a></p> <p><b>Search for package.</b></p> <p>Then, run:</p> <pre>rpm -Uvh compat-expat1-1.95.8-8.el6.x86_64</pre>
Install Qt	<pre>\$ yum install qt-devel.x86_64</pre> <p><b>\$yum install qt-devel (For Centos 7 only)</b></p>
Install qtcreator	<p>Download from <a href="http://download.qt-project.org/archive/qtcreator/2.5/">http://download.qt-project.org/archive/qtcreator/2.5/</a></p> <p>Then, change the permission of binary file and then install package.</p> <pre>chmod +x &lt;binary_file&gt;.bin</pre>

**NOTE:** In order to search which package contains the library/executable file you need, you can use the following command

**In Centos 7/RHEL:**

```
[user@testbuilder ~]$ yum whatprovides <filename>
```

For example, if you want to find what package contains 'libexpat.so.0'. You are able to use the following line:

```
[user@testbuilder ~]$ yum whatprovides libexpat.so.0
```

In Ubuntu:

You must have '**apt-file**' installed to use the tool to help you search for files which are in specific packages

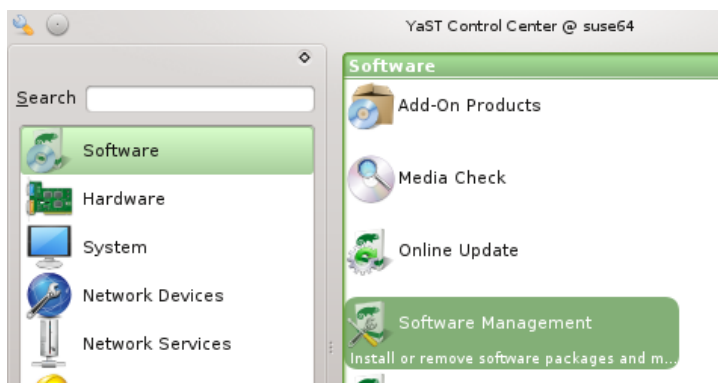
```
[user@testbuilder ~]$ apt-file --package-only search <filename>
```

For example, if you want to find what package contains 'libexpat.so'. You are able to use the following line:

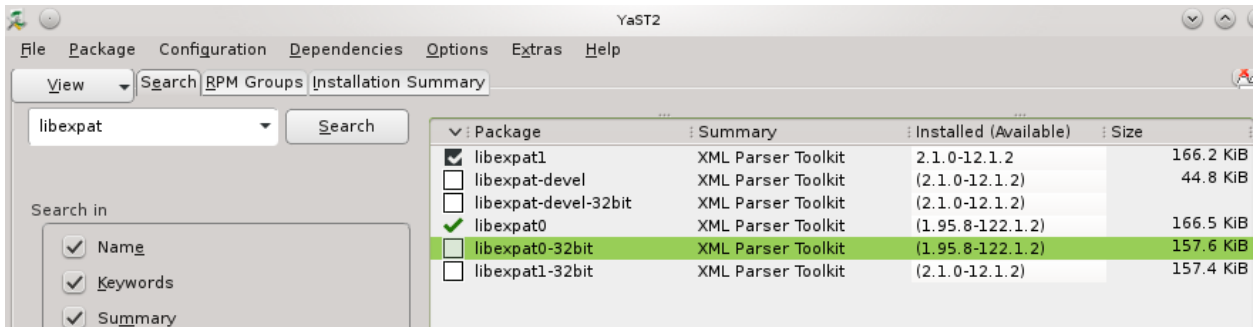
```
[user@testbuilder ~]$ apt-file --package-only search libexpat.so
```

#### **For openSUSE 64-bit:**

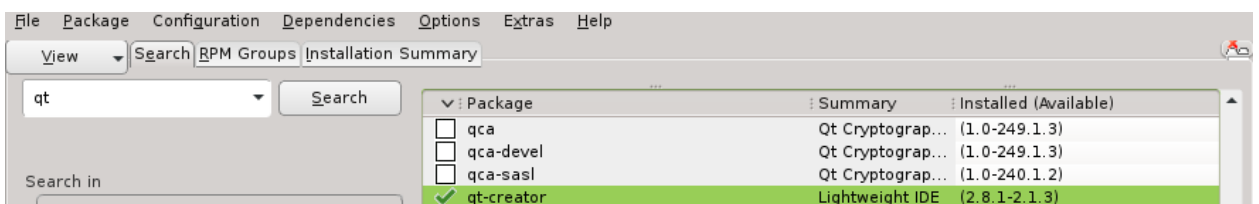
Click on Yast->Software->Software Management



Click on the 'Search' tab and search for 'libexpat.so'. Check the radio box that has 'libexpat0'. Then, click on the next several buttons to accept license agreement and run configuration/installation process.



Then, search for qt. Click on the 'qt-creator'. Check the radio button for 'qt-creator'. He will install the 'Qt' in the process. Please Accept and Continue to install packages for Qt creator.



### **LIBUDEV.SO.0 (openSUSE and Ubuntu 14.04 64-bit only)**

The libudev.so.0 make need to be installed to run some applications. The following installation instructions are in **BOLD**. The installation of packages are installed from the command line from the terminal window.

Installing dependency packages	Command Line
Install 'libudev.so.0'	Please download and save on thumb drive the libudev0-182-8.10.x86_64. You can download from this website: <a href="http://rpm.pbone.net">http://rpm.pbone.net</a> <b>Search for package.</b> Then, run: <b>rpm -Uvh libudev0-182-8.10.x86_64</b>

### ***Development Tools/Build Essentials (All Operating Systems)***

Only install '**Development Tools (Centos 7/RHEL)**' or '**build-essential(Ubuntu)**' if current operating system does not have 'gcc', 'make'.

You can check if these tools exist by typing the following lines on the command prompt in your terminal window:

```
$gcc -v
$make -v
```

This will return information about gcc and make file version numbers.

#### **For Ubuntu 32-bit/64-bit:**

Installing dependency packages	Command Line
Install 'gcc, make', etc (development tools)	\$ <b>apt-get install build-essential</b>

#### **For Centos7/RHEL 64-bit:**

Installing dependency packages	Command Line
Install 'gcc, make', etc (development tools)	\$ <b>yum groupinstall "Development Tools"</b>

Install **kernel-headers** and **kernel-devel** source code if they do not exist in your current system.

#### **For Ubuntu 32-bit/64-bit:**

Installing dependency packages	Command Line
Install 'kernel-headers' for your current kernel	\$ <b>apt-get install kernel-headers-`uname -r`</b>
Install 'kernel-devel' for your current kernel	\$ <b>apt-get install kernel-devel-`uname -r`</b>

#### **For Centos 7/RHEL 64-bit:**

Installing dependency packages	Command Line
Install 'kernel-headers' for your current kernel	\$ <b>yum install kernel-headers-`uname -r`</b>
Install 'kernel-devel' for your current kernel	\$ <b>yum install kernel-devel-`uname -r`</b>

**Note:** If package does not exist in 'yum'. You will need to download from website. Then, install using 'rpm -uvh'. You can download from this website:

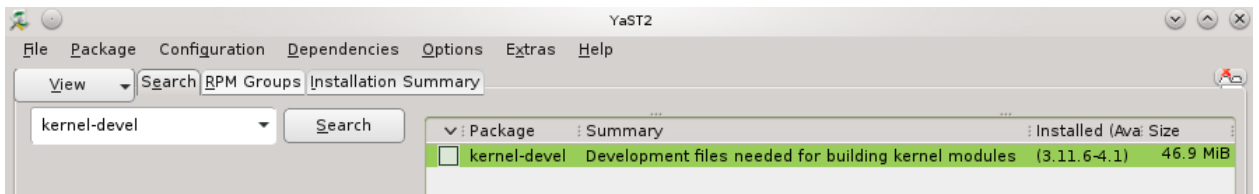
<http://rpm.pbone.net>

**Search for package.**

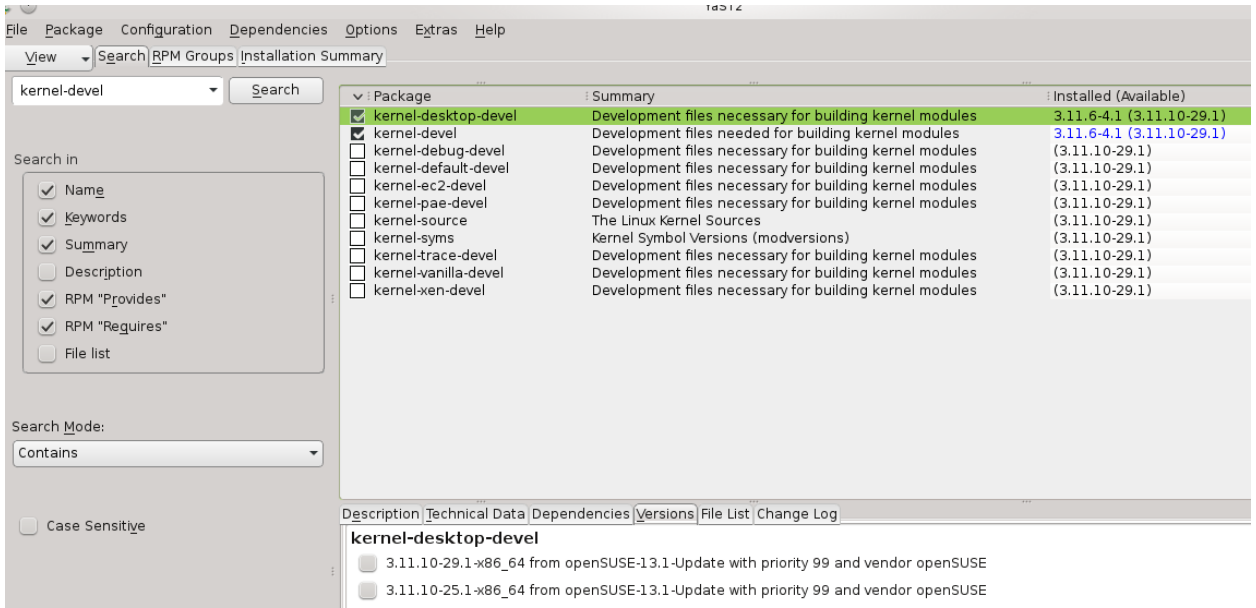
<kernel rpm>

### **For openSUSE 64-bit:**

Install kernel-devel from the 'Yast' -> Software -> Software Management. Search for kernel-devel. Then, check the radio button box and accept and download kernel-devel.



If your version of OpenSuse requires the **Desktop** version of the kernel. Please check the check box **RPM "Provides"** and **RPM "Requires"**. Then, press search. All the available kernel-devel packages will be shown. If your system requires another version of the kernel, please press the **Versions** tab and select the appropriate revision as shown below:



### 3 BOBCAT GEV INSTALLATION INSTRUCTIONS

The Bobcat GEV Linux installer is named 'bobcat\_gev.run'. Please copy this file to your home directory.

**Step 1:** At the terminal window, please make sure you are login as root and enter password:

```
[user@testbuilder ~]$ su
```

```
password: *****
```

**NOTE:** The su command – runs a shell with substitute user or group ID. In our case, running su without a USER name not given, assumes root.

**Step 2:** Please make sure you are able to run the installer by making the installer executable

```
[user@testbuilder ~]$ chmod u+x bobcat_gev_vXXX.run
```

The vXXX corresponds to the Bobcat eBUS version number with build number.

**Step 3:** Run Installer and follow prompted questions.

```
[user@testbuilder ~]$ ./bobcat_gev.run
```

The Bobcat GEV for Linux is installed on /opt/imperx/bobcat\_gev

Step 4: Load the Bobcat GEV device driver

```
[user@testbuilder~]$ /opt/imperx/bobcat_gev/module/./load.sh
```

## 4 UNINSTALL BOBCAT GEV INSTRUCTIONS

---

To uninstall the Bobcat GEV please follow the following steps.

**Step 1:** From the terminal window, make sure you are login as ROOT.

**Step 2:** Run the uninstaller

```
[user@testbuilder ~]$ /opt/imperx/bobcat_gev/bin/uinstall.sh
```

**Step 3:** The prompt will ask you if you are sure if you want to uninstall Bobcat GEV. Type 'Yes' or 'No'

The software will be removed if you answered 'Yes'.

## 4 CONFIGURATION OF THE NETWORK INTERFACES

---

The configuration of the network interfaces are essential to maximize the performance of the Bobcat GEV Camera using the eBUS-PureGEV SDK. Each operating system (Centos, Ubuntu, Fedora, etc.) has a different technique for configuring the network interface.

### Ubuntu Operating System

We will use the 'ifconfig' command to configure a network interface. From the linux command window, you will be able to type 'man ifconfig' to display the linux programmer's manual documentation of this feature. If you type, 'ifconfig' on the command line. It will display all the network devices for your operating system. For instances, in my current environment I have the following three interfaces: 'eth2', 'eth3', and 'lo'.

'eth2' is a Ethernet adapter installed on this machine.

'eth3' is another Ethernet adapter installed on this machine.

'lo' is the loopback interface

### Change the IP address

After reviewing the ifconfig network interfaces of our machine. We can change/or add an IP address for the network interface. In our case, we can change the IP address of the 'eth0' network interface as follows.

**Step 1:** Bring down the driver for this 'eth0' network interface

```
[user@testbuilder~]$ ifconfig eth0 down
```

**Step 2:** Set the ipaddress. This could be from the recognized supported address families of your operating system. In this case, we will use inet (TCP/IP).

```
[user@testbuilder~]$ ifconfig eth0 inet <ip-address>
```

**Step 3:** Bring up the driver for the 'eth0' network interface

```
[user@testbuilder~]$ ifconfig eth0 up
```

**Step 4:** Rerun ifconfig. It should show your new <ip-address> for your interface

### Change the Maximum Transmission Unit (MTU)

In the communication protocol layer, the Maximum Transmission Unit (MTU) / packet size of the network interface can be changed. Modifying the MTU will help in utilizing maximum performance of your device. The procedure is as follows:

**Step 1:** Bring the 'eth0' network interface you currently want to change the MTU/packet size

```
[user@testbuilder~]$ ifconfig down
```

**Step 2:** Identify the packet size. In our case, we have 8164 bytes

```
[user@testbuilder~]$ ifconfig eth0 mtu 8164
```

**Step 3:** Bring up the 'eth0' network interface

```
[user@testbuilder~]$ ifconfig eth0 up
```

**Note:** Cascading Multiple Command Options are also supported. Please reference the linux operation system supporting documentation *Linux Programmer's Manual*.

```
[user@testbuilder~]$ man ifconfig
```

## 5 BOBCAT GEV SDK SAMPLES

---

Bobcat GEV SDK samples are installed as part of SDK during the Bobcat GEV software installation. C++ samples.

To build C++ samples, please follow the following steps.

**Step 1:** Please make sure you are login as ROOT

**Step 2:** Source the PureGev 'set\_puregev\_env' Environment Variables

```
[user@testbuilder ~]$ source /opt/imperx/bobcat_gev/bin/set_puregev_env
```

**Note!** Step 2 will need to be repeated every time you open a new terminal window.

**Step 3:** Change the directory to location of the SDK samples

```
[user@testbuilder ~]$ cd /opt/imperx/bobcat_gev/share/samples
```

**Step 4:** Run the compile scripts. This will compile all the samples from the command line.

```
[user@testbuilder samples]$ ./build.sh
```

The following samples will have compiled and are located in the corresponding directory under /opt/imperx/bobcat\_gev/share/samples:

Samples
ConfigurationReader
ConnectionRecovery
DeviceFinder
DeviceSerialPort
DualSource
eBUSPlayer
GenICamParameters
IGVExposureUISample
IGVGainUISample

IGVTriggerUISample
IGVTrueSenseUISample
ImageProcessing
MulticastMaster
MulticastSlave
MultiSource
PvPipelineSample
PvStreamSample
TransmitProcessedImage
TransmitTestPattern

### eBUSPlayer Sample Compilation

The eBUSPlayer Sample will not compile as expected until you edit the ImageSaving.cpp file to set the desired directory path for the saved images. These steps are shown below:

For the **eBUSPlayer** sample, you will have to edit the ImageSaving.cpp file located at 'opt/imperx/bobcat\_gev/share/samples/eBUSPlayer' in your computer. Replace line 92 shown below  
mPath = ...

with your desired directory path for the saved images. An example is shown below:

```
#ifdef LINUX
    mPath = "/opt/imperx/bobcat_gev/share/samples/eBUSPlayer"; //...
#endif // LINUX
```