TRP LABORATORIES

Using udev

Version 1

July 9, 2024

Developed by Transportation Laboratories

Version History

Version	Date	Revision Description
1	7/9/2024	Initial publication

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.

  Example: Select the `cmdapp-`*`relVersion`*`-`*`buildVersion`*`.zip` file….

- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.

  Example: **Type**: Click **Select Type** to display drop-down menu options.

- Cross-references are designated in Arial italics.

  Example: Refer to *Figure 1*…

- Click intra-document cross-references and page references to display the stated destination.

  Example: Refer to *Section 1 udev Overview and Rules* on page 1.

  The clickable cross-references in the preceding example are *1*, *Overview*, and on page 1.

CyFlex Documentation

CyFlex manuals are available at https://cyflex.com/. View **Help & Docs** topics or use the **Search** facility to find topics of interest.

Table of Contents

1 UDEV OVERVIEW AND RULES 1
1.1 udev Resources and Reference Information 1
2 FINDING DEVICE INFORMATION WITH UDEVADM 2
3 FINDING INFORMATION WITH LSUSB 6
4 WRITING RULES 10
4.1 Rules Guidelines 10
4.2 Reloading udev Rules without Rebooting 12

1 udev Overview and Rules

`udev` supplies the system software with device events, manages permissions of device nodes and may create additional `symlink`s in the `/dev` directory, or renames network interfaces. The kernel usually just assigns unpredictable device names based on the order of discovery. Meaningful `symlink`s or network device names provide a way to reliably identify devices based on their properties or current configuration.

The `udev` daemon, `systemd-udevd.service(8)` receives device `uevents` directly from the kernel whenever a device is added or removed from the system or when it changes its state. When `udev` receives a device event, it matches its configured set of rules against various device attributes to identify the device. Rules that match may provide additional device information to be stored in the udev database or to be used to create meaningful `symlink` names.

All device information udev processes is stored in the udev database and sent out to possible event subscribers. Access to all stored data and the event sources is provided by the library `libudev`.

1.1 udev Resources and Reference Information

```
man udev
man udevadm
man udev.conf
man systemd-udevd
```

Most recent version of *Writing udev Rules* by Daniel Drake: writing_udev_rules

Example of `udev` Rules: docs.kernel.org

Managing System Devices in Oracle Linux 8

`udev` reference: Arch wiki

2 Finding Device Information with udevadm

As documented in udevadm(8) — Linux manual page:

udevadm expects a command and command specific options. It controls the runtime behavior of **systemd-udevd**, requests kernel events, manages the event queue, and provides simple debugging mechanisms.

Enter:

```
$ udevadm info -a -n /dev/path/to/device
```

where:

- `info` indicates a query of the udev database for device information.
- `-a` (`attribute-walk`) prints all `sysfs` properties of the specified device that can be used in `udev` rules to match the specified device. It prints all devices along the chain, up to the root of `sysfs` that can be used in udev rules.
- `-n` (`name=`*`file`*) specify the name of the device node or a `symlink` to query, e.g. `/dev/path/to/device`

Generic use of `udevadm` to find information on a device that is in the virtual file system:

```
$ udevadm info -a -n /dev/pcan-usb/0/can0
```

The following example finds information about a PEAK can device `can0`.

```
udevadm info starts with the device specified by the devpath and then walks
up the chain of parent devices. It prints for every device found, all
possible attributes in the udev rules key format. A rule to match can be
composed by the attributes of the device and the attributes from one single
parent device.

  looking at device '/devices/pci0000:00/0000:00:14.0/usb1/1-4/1-4.3/1-
4.3:1.0/pcan/pcanusb32':
    KERNEL=="pcanusb32"
    SUBSYSTEM=="pcan"
    DRIVER==""
    ATTR{adapter_name}=="PCAN-USB"
    ATTR{adapter_number}=="0"
    ATTR{adapter_partnum}=="IPEH-002021/002022"
    ATTR{adapter_version}=="2.8.1"
    ATTR{base}=="0x0"
    ATTR{btr0btr1}=="0x001c"
    ATTR{bus_state}=="0"
    ATTR{clk_drift}=="0"
    ATTR{clock}=="8000000"
    ATTR{ctrlr_number}=="0"
    ATTR{devid}=="32"
    ATTR{errors}=="0"
    ATTR{hwtype}=="11"
    ATTR{init_flags}=="0x00000000"
    ATTR{irq}=="0"
    ATTR{irqs}=="0"
    ATTR{led}=="0"
    ATTR{minor}=="32"
    ATTR{ndev}=="can32"
    ATTR{nom_bitrate}=="500000"
    ATTR{nom_brp}=="1"
    ATTR{nom_sample_point}=="8750"
    ATTR{nom_sjw}=="1"
    ATTR{nom_tq}=="125"
    ATTR{nom_tseg1}=="13"
    ATTR{nom_tseg2}=="2"
    ATTR{read}=="0"
    ATTR{rx_error_counter}=="0"
    ATTR{rx_frames_counter}=="0"
    ATTR{rx_irqs}=="0"
    ATTR{serialno}=="4294967295"
    ATTR{status}=="0x0000"
    ATTR{ts_mode}=="2"
    ATTR{tx_error_counter}=="0"
    ATTR{tx_fifo_ratio}=="0.00"
    ATTR{tx_frames_counter}=="0"
    ATTR{tx_irqs}=="0"
    ATTR{type}=="usb"
    ATTR{write}=="0"

  looking at parent device '/devices/pci0000:00/0000:00:14.0/usb1/1-4/1-
4.3/1-4.3:1.0':
    KERNELS=="1-4.3:1.0"
    SUBSYSTEMS=="usb"
    DRIVERS=="pcan"
    ATTRS{authorized}=="1"
    ATTRS{bAlternateSetting}==" 0"
    ATTRS{bInterfaceClass}=="00"
    ATTRS{bInterfaceNumber}=="00"
    ATTRS{bInterfaceProtocol}=="00"
    ATTRS{bInterfaceSubClass}=="00"
    ATTRS{bNumEndpoints}=="04"
    ATTRS{supports_autosuspend}=="0"

  looking at parent device '/devices/pci0000:00/0000:00:14.0/usb1/1-4/1-4.3':
    KERNELS=="1-4.3"
    SUBSYSTEMS=="usb"
    DRIVERS=="usb"
    ATTRS{authorized}=="1"
    ATTRS{avoid_reset_quirk}=="0"
    ATTRS{bConfigurationValue}=="1"
    ATTRS{bDeviceClass}=="00"
    ATTRS{bDeviceProtocol}=="00"
    ATTRS{bDeviceSubClass}=="00"
    ATTRS{bMaxPacketSize0}=="16"
    ATTRS{bMaxPower}=="200mA"
    ATTRS{bNumConfigurations}=="3"
    ATTRS{bNumInterfaces}==" 1"
    ATTRS{bcdDevice}=="1c20"
    ATTRS{bmAttributes}=="80"
    ATTRS{busnum}=="1"
    ATTRS{configuration}==""
    ATTRS{devnum}=="7"
    ATTRS{devpath}=="4.3"
    ATTRS{idProduct}=="000c"
    ATTRS{idVendor}=="0c72"
    ATTRS{ltm_capable}=="no"
    ATTRS{maxchild}=="0"
    ATTRS{quirks}=="0x0"
    ATTRS{removable}=="removable"
    ATTRS{rx_lanes}=="1"
    ATTRS{speed}=="12"
    ATTRS{tx_lanes}=="1"
    ATTRS{urbnum}=="34"
    ATTRS{version}==" 1.00"

  looking at parent device '/devices/pci0000:00/0000:00:14.0/usb1/1-4':
    KERNELS=="1-4"
    SUBSYSTEMS=="usb"
    DRIVERS=="usb"
    ATTRS{authorized}=="1"
    ATTRS{avoid_reset_quirk}=="0"
    ATTRS{bConfigurationValue}=="1"
    ATTRS{bDeviceClass}=="09"
    ATTRS{bDeviceProtocol}=="01"
    ATTRS{bDeviceSubClass}=="00"
    ATTRS{bMaxPacketSize0}=="64"
    ATTRS{bMaxPower}=="100mA"
    ATTRS{bNumConfigurations}=="1"
    ATTRS{bNumInterfaces}==" 1"
    ATTRS{bcdDevice}=="6070"
    ATTRS{bmAttributes}=="e0"
    ATTRS{busnum}=="1"
    ATTRS{configuration}==""
    ATTRS{devnum}=="3"
    ATTRS{devpath}=="4"
    ATTRS{idProduct}=="0608"
    ATTRS{idVendor}=="05e3"
    ATTRS{ltm_capable}=="no"
    ATTRS{maxchild}=="4"
    ATTRS{product}=="USB2.0 Hub"
    ATTRS{quirks}=="0x0"
    ATTRS{removable}=="fixed"
    ATTRS{rx_lanes}=="1"
    ATTRS{speed}=="480"
    ATTRS{tx_lanes}=="1"
    ATTRS{urbnum}=="56"
    ATTRS{version}==" 2.00"

  looking at parent device '/devices/pci0000:00/0000:00:14.0/usb1':
    KERNELS=="usb1"
    SUBSYSTEMS=="usb"
    DRIVERS=="usb"
    ATTRS{authorized}=="1"
    ATTRS{authorized_default}=="1"
    ATTRS{avoid_reset_quirk}=="0"
    ATTRS{bConfigurationValue}=="1"
    ATTRS{bDeviceClass}=="09"
    ATTRS{bDeviceProtocol}=="01"
    ATTRS{bDeviceSubClass}=="00"
    ATTRS{bMaxPacketSize0}=="64"
    ATTRS{bMaxPower}=="0mA"
    ATTRS{bNumConfigurations}=="1"
    ATTRS{bNumInterfaces}==" 1"
    ATTRS{bcdDevice}=="0504"
    ATTRS{bmAttributes}=="e0"
    ATTRS{busnum}=="1"
    ATTRS{configuration}==""
    ATTRS{devnum}=="1"
    ATTRS{devpath}=="0"
    ATTRS{idProduct}=="0002"
    ATTRS{idVendor}=="1d6b"
    ATTRS{interface_authorized_default}=="1"
    ATTRS{ltm_capable}=="no"
    ATTRS{manufacturer}=="Linux 5.4.17-2136.318.7.2.el8uek.x86_64 xhci-hcd"
    ATTRS{maxchild}=="16"
    ATTRS{product}=="xHCI Host Controller"
    ATTRS{quirks}=="0x0"
    ATTRS{removable}=="unknown"
    ATTRS{rx_lanes}=="1"
    ATTRS{serial}=="0000:00:14.0"
    ATTRS{speed}=="480"
    ATTRS{tx_lanes}=="1"
    ATTRS{urbnum}=="108"
    ATTRS{version}==" 2.00"

  looking at parent device '/devices/pci0000:00/0000:00:14.0':
    KERNELS=="0000:00:14.0"
    SUBSYSTEMS=="pci"
    DRIVERS=="xhci_hcd"
    ATTRS{ari_enabled}=="0"
    ATTRS{broken_parity_status}=="0"
    ATTRS{class}=="0x0c0330"
    ATTRS{consistent_dma_mask_bits}=="64"
    ATTRS{d3cold_allowed}=="1"
    ATTRS{dbc}=="disabled"
    ATTRS{device}=="0x43ed"
    ATTRS{dma_mask_bits}=="64"
    ATTRS{driver_override}=="(null)"
    ATTRS{enable}=="1"
    ATTRS{index}=="3"
    ATTRS{irq}=="125"
    ATTRS{label}=="Onboard - Other"
    ATTRS{local_cpulist}=="0-15"
    ATTRS{local_cpus}=="ffff"
    ATTRS{msi_bus}=="1"
    ATTRS{numa_node}=="-1"
    ATTRS{revision}=="0x11"
    ATTRS{subsystem_device}=="0x7d09"
    ATTRS{subsystem_vendor}=="0x1462"
    ATTRS{vendor}=="0x8086"

  looking at parent device '/devices/pci0000:00':
    KERNELS=="pci0000:00"
    SUBSYSTEMS==""
    DRIVERS==""
```

3 Finding Information with lsusb

For USB type adapter devices, enter the following to list verbose information on USB devices.

```
$ lsusb -v
```

The following is an example of output for one device.

```
Bus 001 Device 007: ID 0c72:000c PEAK System PCAN-USB
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               1.00
  bDeviceClass            0
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0        16
  idVendor           0x0c72 PEAK System
  idProduct          0x000c PCAN-USB
  bcdDevice           1c.20
  iManufacturer           0
  iProduct                3
  iSerial                 0
  bNumConfigurations      3
  Configuration Descriptor:
    bLength                 9
    bDescriptorType         2
    wTotalLength       0x002e
    bNumInterfaces          1
    bConfigurationValue     1
    iConfiguration          0
    bmAttributes         0x80
      (Bus Powered)
    MaxPower              200mA
    Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        0
      bAlternateSetting       0
      bNumEndpoints           4
      bInterfaceClass         0
      bInterfaceSubClass      0
      bInterfaceProtocol      0
      iInterface              0
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x81  EP 1 IN
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0010  1x 16 bytes
        bInterval              20
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x01  EP 1 OUT
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0010  1x 16 bytes
        bInterval              20
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x82  EP 2 IN
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0040  1x 64 bytes
        bInterval               1
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x02  EP 2 OUT
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0040  1x 64 bytes
        bInterval               1
  Configuration Descriptor:
    bLength                 9
    bDescriptorType         2
    wTotalLength       0x002e
    bNumInterfaces          1
    bConfigurationValue     2
    iConfiguration          0
    bmAttributes         0x80
      (Bus Powered)
    MaxPower              394mA
    Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        0
      bAlternateSetting       0
      bNumEndpoints           4
      bInterfaceClass         0
      bInterfaceSubClass      0
      bInterfaceProtocol      0
      iInterface              0
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x81  EP 1 IN
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0010  1x 16 bytes
        bInterval              20
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x01  EP 1 OUT
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0010  1x 16 bytes
        bInterval              20
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x82  EP 2 IN
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0040  1x 64 bytes
        bInterval               1
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x02  EP 2 OUT
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0040  1x 64 bytes
        bInterval               1
  Configuration Descriptor:
    bLength                 9
    bDescriptorType         2
    wTotalLength       0x002e
    bNumInterfaces          1
    bConfigurationValue     3
    iConfiguration          0
    bmAttributes         0x80
      (Bus Powered)
    MaxPower              200mA
    Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        0
      bAlternateSetting       0
      bNumEndpoints           4
      bInterfaceClass         0
      bInterfaceSubClass      0
      bInterfaceProtocol      0
      iInterface              0
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x81  EP 1 IN
        bmAttributes            3
          Transfer Type            Interrupt
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0010  1x 16 bytes
        bInterval               1
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x01  EP 1 OUT
        bmAttributes            3
          Transfer Type            Interrupt
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0010  1x 16 bytes
        bInterval               1
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x82  EP 2 IN
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0040  1x 64 bytes
        bInterval               1
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x02  EP 2 OUT
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0040  1x 64 bytes
        bInterval               1
```

4 Writing Rules

`udev` rules are written to a file in the `/etc/udev/rules.d` directory. There are other directories that can be used, but this is the most common. The rules file should have a number at the beginning of the file name. This number is used to determine the order in which the rules are processed. The lower the number, the earlier the rule is processed. All rules are processed even if a match is found early in the list. Rule files must have the `.rules` extension.

4.1 Rules Guidelines

- The rules file must have a `.rules` extension.
- The rules file must have a number at the beginning of the file name.
- The rules file must be in the `/etc/udev/rules.d` directory.
- The rules file must have the correct syntax.
- Matching attributes for the device:
    - `KERNEL`
    - `SUBSYSTEM`
    - `ATTR`
    - `DRIVER`
- Matching attributes of parent devices:
    - `KERNELS`
    - `SUBSYSTEMS`
    - `ATTRS`
    - `DRIVERS`
- Only a single parent device can be used for matching.
- `udev` rules do not support line continuation. Keep the rules on a single line.

For more information, refer to the most recent version of *Writing udev Rules* by Daniel Drake: writing_udev_rules.

The following is an example of a rules file:

```
/etc/udev/rules.d/11-usb-tty.rules
# this works
KERNEL=="pcanusb32", SUBSYSTEM=="pcan", ATTR{serialno}=="4294967295",
SYMLINK+="ucandoit"

# this also works
SUBSYSTEM=="pcan", ATTR{serialno}=="4294967295", SYMLINK+="ucandoit2"

# fails on the ACTION
ACTION="add", SUBSYSTEM=="PCAN", ATTRS{adapter_name}=="PCAN-USB",
ATTRS{serialno}=="4294967295", SYMLINK+="uCANdoit"

# this also works
SUBSYSTEM=="pcan", ATTR{serialno}=="4294967295", ATTRS{idProduct}=="000c",
ATTRS{idVendor}=="0c72", SYMLINK+="uCANdoit"
```

```
# this also works
SUBSYSTEM=="pcan", ATTR{adapter_name}=="PCAN-USB",
ATTR{serialno}=="4294967295", SYMLINK+="uCANdoit2"

# this also works
ATTR{adapter_name}=="PCAN-USB", ATTR{serialno}=="4294967295",
SYMLINK+="uCANdoit3"

# this fails on using more than one parent device for matching
SUBSYSTEM=="pcan", ATTR{serialno}=="4294967295", ATTRS{idProduct}=="000c",
ATTRS{serial}=="0000:00:14.0", SYMLINK+="uCANdoit4"
```

`SYMLINK+="link_name"`

This is the line that will create a `symlink` to the device file normally in the `/dev` directory. This is most likely the whole point that a `udev` rule is for when in context with devices.

Every line in the rules file contains at least one key-value pair. Except for empty lines or lines beginning with "#", which are ignored. There are two kinds of keys: match and assignment. If all match keys match against their values, the rule gets applied and the assignment keys get the specified values assigned.

A matching rule may rename a network interface, add `symlink`s pointing to the device node, or run a specified program as part of the event handling.

A rule consists of a comma-separated list of one or more key-value pairs. Each key has a distinct operation, depending on the used operator. Valid operators are:

Operator	Description
`==`	Compare for equality.
`!=`	Compare for inequality.
`=`	Assign a value to a key. Keys that represent a list are reset and only this single value is assigned.
`+=`	Add the value to a key that holds a list of entries.
`-=`	Remove the value from a key that holds a list of entries.
`:=`	Assign a value to a key finally; disallow any later changes.

The following are some key names can be used to match against device properties. See man 7 udev for more details. Some of the keys also match against properties of the parent devices in sysfs, not only the device that has generated the event. If multiple keys that match a parent device are specified in a single rule, all these keys must match at one and the same parent device.

Keyword	Description
`ACTION`	Match the name of the event action.
`DEVPATH`	Match the `devpath` of the event device.
`KERNEL`	Match the name of the event device.
`NAME`	Match the name of a network interface. It can be used once the `NAME` key has been set in one of the preceding rules.
`SYMLINK`	Match the name of a `symlink` targeting the node. It can be used once a `SYMLINK` key has been set in one of the preceding rules. There may be multiple `symlink`s; only one needs to match.
`SUBSYSTEM`	Match the subsystem of the event device.
`DRIVER`	Match the driver name of the event device. Only set this key for devices which are bound to a driver at the time the event is generated.
`ATTR{filename}`, `SYSCTL{kernel parameter}`	Match `sysfs` attribute values of the event device. Trailing whitespace in the attribute values is ignored unless the specified match value itself contains trailing whitespace. Match a kernel parameter value.

4.2 Reloading udev Rules without Rebooting

Normally `udev` rules are parsed at boot, and frequently this might be the best option. The following are a few steps that can be tried without rebooting to process any new udev rules.

```
$ sudo udevadm test $(udevadm info -q path -n /dev/pcan-usb/0/can0)
$ sudo udevadm control --reload && udevadm trigger
```

The `test` parameter will simulate running all the `udev` rules on the device. This does not actually do anything, but in OL8 it seems to be necessary before running `udevadm control`. The `test` parameter will show results including whether any errors occurred. This is a good check to determine whether the rules written against the device are going to work. If no errors are found and the results are expected, then running `control` and `trigger` will apply the new `udev` rules.