



Rocky Enterprise Linux 9.2 Manual Pages on command 'systemd.net-naming-scheme.7'

C:\>man systemd.net-naming-scheme.7

SYSTEMD.NET-NAMING-SCHEME(7) systemd.net-naming-scheme SYSTEMD.NET-NAMING-SCHEME(7)

NAME

systemd.net-naming-scheme - Network device naming schemes

DESCRIPTION

Network interfaces names and MAC addresses may be generated based on certain stable interface attributes. This is possible when there is enough information about the device to generate those attributes and the use of this information is configured.

This page describes interface naming, i.e. what possible names may be generated.

Those names are generated by the `systemd-udev.service(8)` builtin `net_id` and exported as udev properties (`ID_NET_NAME_ONBOARD=`, `ID_NET_LABEL_ONBOARD=`, `ID_NET_NAME_PATH=`, `ID_NET_NAME_SLOT=`).

Names and MAC addresses are derived from various stable device metadata attributes.

Newer versions of udev take more of these attributes into account, improving (and thus possibly changing) the names and addresses used for the same devices.

Different versions of those generation rules are called "naming schemes". The default naming scheme is chosen at compilation time. Usually this will be the latest implemented version, but it is also possible to set one of the older

versions to preserve compatibility. This may be useful for example for distributions, which may introduce new versions of `systemd` in stable releases without changing the naming scheme. The naming scheme may also be overridden using the `net.naming-scheme=` kernel command line switch, see `systemd-udev.service(8)`.

Available naming schemes are described below.

After the udev properties have been generated, appropriate udev rules may be used to actually rename devices based on those properties. See the description of NamePolicy= and MACAddressPolicy= in systemd.link(5).

Note that while the concept of network interface naming schemes is primarily relevant in the context of systemd-udevd.service, the systemd-nspawn(1) container manager also takes it into account when naming network interfaces, see below.

NAMING

All names start with a two-character prefix that signifies the interface type.

Table 1. Two character prefixes based on the type of interface

??

?Prefix ? Description ?

??

?en ? Ethernet ?

??

?ib ? InfiniBand ?

??

?sl ? Serial line IP (slip) ?

??

?wl ? Wireless local area network ?

? ? (WLAN) ?

??

?ww ? Wireless wide area network ?

? ? (WWAN) ?

??

The udev net_id builtin exports the following udev device properties:

ID_NET_NAME_ONBOARD=prefixnumber

This name is set based on the numeric ordering information given by the firmware for on-board devices. The name consists of the prefix, letter o, and a number specified by the firmware. This is only available for PCI devices.

ID_NET_LABEL_ONBOARD=prefix label

This property is set based on textual label given by the firmware for on-board devices. The name consists of the prefix concatenated with the label. This is only available for PCI devices.

be renamed according to the configuration, even if they have been renamed already, if keep is not specified as the naming policy in the .link file. See systemd.link(5) for a description of NamePolicy=.

v241

MACAddressPolicy=persistent was extended to set MAC addresses based on the device name. Previously addresses were only based on the ID_NET_NAME_* attributes, which meant that interface names would never be generated for virtual devices. Now a persistent address will be generated for most devices, including in particular bridges.

Note: when userspace does not set a MAC address for a bridge device, the kernel will initially assign a random address, and then change it when the first device is enslaved to the bridge. With this naming policy change, bridges get a persistent MAC address based on the bridge name instead of the first enslaved device.

v243

Support for renaming netdevsim (simulated networking) devices was added.

Previously those devices were not renamed.

Previously two-letter interface type prefix was prepended to

ID_NET_LABEL_ONBOARD=. This is not done anymore.

v245

When systemd-nspawn(1) derives the name for the host side of the network interface created with --network-veth from the container name it previously simply truncated the result at 15 characters if longer (since that's the maximum length for network interface names). From now on, for any interface name that would be longer than 15 characters the last 4 characters are set to a 24bit hash value of the full interface name. This way network interface name collisions between multiple similarly named containers (who only differ in container name suffix) should be less likely (but still possible, since the 24bit hash value is very small).

v247

If the PCI slot is associated with PCI bridge and that has multiple child network controllers then all of them might derive the same value of ID_NET_NAME_SLOT property. That could cause naming conflict if the property is

selected as a device name. Now, we detect the situation, slot - bridge relation, and we don't produce the ID_NET_NAME_SLOT property to avoid possible naming conflict.

Note that latest may be used to denote the latest scheme known (to this particular version of systemd).

EXAMPLES

Example 1. Using udevadm test-builtin to display device properties

```
$ udevadm test-builtin net_id /sys/class/net/enp0s31f6
...
Using default interface naming scheme 'v243'.
ID_NET_NAMING_SCHEME=v243
ID_NET_NAME_MAC=enx54ee75cb1dc0
ID_OUI_FROM_DATABASE=Wistron InfoComm(Kunshan)Co.,Ltd.
ID_NET_NAME_PATH=enp0s31f6
...
```

Example 2. PCI Ethernet card with firmware index "1"

```
ID_NET_NAME_ONBOARD=eno1
ID_NET_NAME_ONBOARD_LABEL=Ethernet Port 1
```

Example 3. PCI Ethernet card in hotplug slot with firmware index number

```
# /sys/devices/pci0000:00/0000:00:1c.3/0000:05:00.0/net/ens1
ID_NET_NAME_MAC=enx000000000466
ID_NET_NAME_PATH=enp5s0
ID_NET_NAME_SLOT=ens1
```

Example 4. PCI Ethernet multi-function card with 2 ports

```
# /sys/devices/pci0000:00/0000:00:1c.0/0000:02:00.0/net/enp2s0f0
ID_NET_NAME_MAC=enx78e7d1ea46da
ID_NET_NAME_PATH=enp2s0f0
# /sys/devices/pci0000:00/0000:00:1c.0/0000:02:00.1/net/enp2s0f1
ID_NET_NAME_MAC=enx78e7d1ea46dc
ID_NET_NAME_PATH=enp2s0f1
```

Example 5. PCI WLAN card

```
# /sys/devices/pci0000:00/0000:00:1c.1/0000:03:00.0/net/wlp3s0
ID_NET_NAME_MAC=wlx0024d7e31130
```

```
ID_NET_NAME_PATH=wlp3s0
```

Example 6. PCI IB host adapter with 2 ports

```
# /sys/devices/pci0000:00/0000:00:03.0/0000:15:00.0/net/ibp21s0f0
```

```
ID_NET_NAME_PATH=ibp21s0f0
```

```
# /sys/devices/pci0000:00/0000:00:03.0/0000:15:00.1/net/ibp21s0f1
```

```
ID_NET_NAME_PATH=ibp21s0f1
```

Example 7. USB built-in 3G modem

```
# /sys/devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1.4/2-1.4:1.6/net/wwp0s29u1u4i6
```

```
ID_NET_NAME_MAC=wwx028037ec0200
```

```
ID_NET_NAME_PATH=wwp0s29u1u4i6
```

Example 8. USB Android phone

```
# /sys/devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1.2/2-1.2:1.0/net/enp0s29u1u2
```

```
ID_NET_NAME_MAC=enxd626b3450fb5
```

```
ID_NET_NAME_PATH=enp0s29u1u2
```

Example 9. s390 grouped CCW interface

```
# /sys/devices/css0/0.0.0007/0.0.f5f0/group_device/net/encf5f0
```

```
ID_NET_NAME_MAC=enx026d3c00000a
```

```
ID_NET_NAME_PATH=encf5f0
```

SEE ALSO

udev(7), udevadm(8), Predictable Network Interface Names[1], systemd-nspawn(1)

NOTES

1. Predictable Network Interface Names

https://systemd.io/PREDICTABLE_INTERFACE_NAMES

systemd 245

SYSTEMD.NET-NAMING-SCHEME(7)