Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'systemd.net-naming-scheme.7' command

$ man systemd.net-naming-scheme.7

SYSTEMD.NET-NAMING-SCHEME(7systemd.net-naming-scheSYSTEMD.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-udevd.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-

udevd.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=prefixonumber, ID_NET_NAME_ONBOARD=prefixdnumber

This name is set based on the numeric ordering information given by the firmware for on-board devices. Different schemes are used depending on the firmware type, as described in the table below.

Table 2. Onboard naming schemes

??????????????????????????????????????

?Format ? Description ?

??????????????????????????????????????

?prefixonumber ? PCI onboard index ?

??????????????????????????????????????

?prefixdnumber ? Devicetree alias index ?

??????????????????????????????????????

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.

ID_NET_NAME_MAC=prefixxAABBCCDDEEFF

This name consists of the prefix, letter x, and 12 hexadecimal digits of the MAC address. It is available if the device has a fixed MAC address. Because this name is based on an attribute of the card itself, it remains "stable" when the device is moved (even between machines), but will change when the hardware is replaced.

ID_NET_NAME_SLOT=prefix[Pdomain]sslot[ffunction][nport_name|ddev_port],
ID_NET_NAME_SLOT=prefixvslot, ID_NET_NAME_SLOT=prefixxslot,
ID_NET_NAME_SLOT=prefix[Pdomain]sslot[ffunction][nport_name|ddev_port]bnumber,
ID_NET_NAME_SLOT=prefix[Pdomain]sslot[ffunction][nport_name|ddev_port]uport...[cconfig][iinterface],
ID_NET_NAME_SLOT=prefix[Pdomain]sslot[ffunction][nport_name|ddev_port]vslot

This property describes the slot position. Different schemes are used depending on the bus type, as described in the table below. In case of USB, BCMA, and SR-VIO devices, the full name consists of the prefix, PCI slot identifier, and USB or BCMA or SR-VIO slot identifier. The first two parts are denoted as "..." in the table below.

Table 3. Slot naming schemes

??

?Format ? Description ?

??

?prefix [Pdomain] sslot [ffunction] [nport_name | ddev_port] ? PCI slot number ?

??

?prefix vslot ? VIO slot number (IBM ?

? ? PowerVM) ?

??

?prefix Xnumber ? VIF interface number (Xen) ?

??

?... bnumber ? Broadcom bus (BCMA) core ?

? ? number ?

??

?... uport... [cconfig] [iinterface] ? USB port number chain ?

??

?... vslot ? SR-VIO slot number ?

??

The PCI domain is only prepended when it is not 0. All

multi-function PCI devices will carry the ffunction number in the

device name, including the function 0 device. For

non-multi-function devices, the number is suppressed if 0. The port

name port_name is used, or the port number ddev_port if the name is

not known.

For BCMA devices, the core number is suppressed when 0.

For USB devices the full chain of port numbers of hubs is composed.

If the name gets longer than the maximum number of 15 characters,

the name is not exported. The usual USB configuration number 1 and

interface number 0 values are suppressed.

SR-IOV virtual devices are named based on the name of the parent

interface, with a suffix of v and the virtual device number, with

any leading zeros removed. The bus number is ignored.

In some configurations a parent PCI bridge of a given network

controller may be associated with a slot. In such case we don't

generate this device property to avoid possible naming conflicts.

ID_NET_NAME_PATH=prefixcbus_id,

ID_NET_NAME_PATH=prefixavendormodeliinstance,

ID_NET_NAME_PATH=prefixiaddressnport_name,

ID_NET_NAME_PATH=prefix[Pdomain]pbussslot[ffunction][nphys_port_name|ddev_port],

ID_NET_NAME_PATH=prefix[Pdomain]pbussslot[ffunction][nphys_port_name|ddev_port]bnumber,

ID_NET_NAME_PATH=prefix[Pdomain]pbussslot[ffunction][nphys_port_name|ddev_port]uport...[cconfig][iinterface]

This property describes the device installation location. Different schemes are used depending on the bus type, as described in the table below. For BCMA and USB devices, PCI path information must known, and the full name consists of the prefix, PCI slot identifier, and USB or BCMA location. The first two parts are denoted as "..." in the table below.

Table 4. Path naming schemes

Format	Description
prefix cbus_id	CCW or grouped CCW device identifier
prefix avendor model iinstance	ACPI path names for ARM64 platform devices
prefix iaddress nport_name	Netdevsim (simulated networking device) device number and port name

??

?prefix [Pdomain] pbus sslot [ffunction] [nphys_port_name | ddev_port] ? PCI geographical location ?

??

??

?... bnumber ? Broadcom bus (BCMA) core ?

? ? number ?

??

??

?... uport... [cconfig] [iinterface] ? USB port number chain ?

??

??

CCW and grouped CCW devices are found in IBM System Z mainframes.

Any leading zeros and dots are suppressed.

For PCI, BCMA, and USB devices, the same rules as described above

for slot naming are used.

HISTORY

The following "naming schemes" have been defined (which may be chosen

at system boot-up time via the net.naming-scheme= kernel command line

switch, see above):

v238

This is the naming scheme that was implemented in systemd 238.

v239

Naming was changed for virtual network interfaces created with

SR-IOV and NPAR and for devices where the PCI network controller

device does not have a slot number associated.

SR-IOV virtual devices are named based on the name of the parent

interface, with a suffix of "vport", where port is the virtual

device number. Previously those virtual devices were named as if

completely independent.

The ninth and later NPAR virtual devices are named following the

scheme used for the first eight NPAR partitions. Previously those devices were not renamed and the kernel default ("ethN") was used.

Names are also generated for PCI devices where the PCI network controller device does not have an associated slot number itself, but one of its parents does. Previously those devices were not renamed and the kernel default was used.

v240

The "ib" prefix and stable names for infiniband devices are introduced. Previously those devices were not renamed.

The ACPI index field (used in ID_NET_NAME_ONBOARD=) is now also used when 0.

A new naming policy NamePolicy=keep was introduced. With this policy, if the network device name was already set by userspace, the device will not be renamed again. Previously, this naming policy applied implicitly, and now it must be explicitly requested. Effectively, this means that network devices will 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

When a PCI slot is associated with a PCI bridge that has multiple child network controllers, the same value of the ID_NET_NAME_SLOT property might be derived for those controllers. This would cause a naming conflict if the property is selected as the device name. Now, we detect this situation and don't produce the ID_NET_NAME_SLOT property.

v249

PCI hotplug slot names for the s390 PCI driver are a hexadecimal representation of the function_id device attribute. This attribute is now used to build the ID_NET_NAME_SLOT. Before that, all slot names were parsed as decimal numbers, which could either result in an incorrect value of the ID_NET_NAME_SLOT property or none at all.

Some firmware and hypervisor implementations report unreasonably high numbers for the onboard index. To prevent the generation of bogus onbard interface names, index numbers greater than 16381 (2??-1) were ignored. For s390 PCI devices index values up to 65535 (2??-1) are valid. To account for that, the limit was increased to 65535.

The udev rule NAME= replaces ":", "/", and "%" with an underscore ("_"), and refuses strings which contain only numerics.

v250

Added naming scheme for Xen netfront "vif" interfaces based on the guest side VIF number set from the Xen config (or the interface index in AWS EC2).

v251

Since version v247 we no longer set ID_NET_NAME_SLOT if we detect that a PCI device associated with a slot is a PCI bridge as that would create naming conflict when there are more child devices on that bridge. Now, this is relaxed and we will use slot information to generate the name based on it but only if the PCI device has multiple functions. This is safe because distinct function number is a part of the device name for multifunction devices.

v252

Added naming scheme for platform devices with devicetree aliases.

rhel-9.0

Since version v247 we no longer set ID_NET_NAME_SLOT if we detect that a PCI device associated with a slot is a PCI bridge as that would create naming conflict when there are more child devices on that bridge. Now, this is relaxed and we will use slot information to generate the name based on it but only if the PCI device has multiple functions. This is safe because distinct function number is a part of the device name for multifunction devices.

rhel-9.1

Same as naming scheme rhel-9.0.

rhel-9.2

Same as naming scheme rhel-9.0.

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