

RHEL/CentOS 7: Revert To Old Network Interface Naming Convention

Finnbarr P. Murphy

(fpm@fpmurphy.com)

RHEL7 and downstream distributions such as CentOS 7 use *systemd predictable network interface names* by default instead of the traditional interface naming scheme which existed in Linux distributions since the earliest days of Linux, i.e. *eth0*, *wlan0*, etc.

This is somewhat problematic when studying for the RHCSA or RHCE exams as most current books use the old network interface naming conventions such as *eth0*, *eth1*, etc. One such book is Asghar Ghori's RHSCA & RHCE Training and Exam Preparation Guide.

Here is the default network configuration created during the installation of CentOS 7 build 1511 in a VM running on VMware Workstation 12:

```
# ip addr
1: lo: mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eno16777736: mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:50:56:35:8e:c9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.101/24 brd 192.168.0.255 scope global dynamic eno16777736
        valid_lft 1716sec preferred_lft 1716sec
    inet6 fe80::250:56ff:fe35:8ec9/64 scope link
        valid_lft forever preferred_lft forever
#
```

The interface name *eno16777736* indicates that it is an onboard interface at a firmware provided index number. More precisely, *em* indicates an Ethernet interface, *o* indicates a (onboard) firmware device index number, and *16777736* is the device's *acpi_index* (ACPI_DSM instance number).

```
# cat /sys/class/net/eno16777736/device/acpi_index
16777736
#
```

Here is the contents of the *eno16777736* interface configuration file created by the installation process:

```
# cat /etc/sysconfig/network-scripts/ifcfg-eno16777736
HWADDR="00:50:56:35:8E:C9"
TYPE="Ethernet"
BOOTPROTO="dhcp"
DEFROUTE="yes"
```

```

PEERDNS="yes"
PEERROUTES="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_PEERDNS="yes"
IPV6_PEERROUTES="yes"
IPV6_FAILURE_FATAL="no"
NAME="eno16777736"
UUID="d033918c-8bae-433c-a87a-171d35ba8769"
ONBOOT="yes"
DEVICE="eno16777736"

```

Note that there is some known issues as to the value of the *acpi_index* returned by VMware Workstation. See [here](#), [here](#) and [here](#) for more information. However, that issue is of no real interest to us in this post.

Suppose you want to permanently assign an IPv4 address to this interface and manually manage the interface yourself (including assigning an interface name) so you have a stable network interface for your RHCSA or RHEL lab VM. Suppose also that you wish to do all of this without changing options on the kernel command line and/or rebooting your system. This post will show you how.

There is a lot of invalid and obsolete information on how to do this on the Internet. In RHEL7, and downstream distributions, *systemd*, *udev* rules, *NetworkManager* and more, all get involved in the configuration of network interfaces. As of the date of this post, *systemd-networkd* components are not shipped nor officially supported in RHEL7 and hence in CentOS. Obviously, this may change in future releases. Note that *systemd-networkd* is available in the RHEL7 Optional channel, but that channel is for unsupported packages offered as a convenience to customers.

The first thing you need to do is stop and disable *NetworkManager* which by default is currently managing the network interface.

```

# systemctl stop NetworkManager
# systemctl disable NetworkManager
# systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; disabled; vendor preset:
   enabled)
   Active: inactive (dead)
#

```

In my humble opinion *NetworkManager* is evil and has absolutely no place on a device that is configured as a server.

Next use *ip link* to rename the interface:

```

# ip link set eno16777736 down
# ip link set eno16777736 name eth0
# ip link set eth0 up
# ip addr
1: lo: mtu 65536 qdisc noqueue state UNKNOWN
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: mtu 1500 qdisc pfifo_fast state UP qlen 1000

```

```
link/ether 00:50:56:35:8e:c9 brd ff:ff:ff:ff:ff:ff
inet 192.168.0.101/24 brd 192.168.0.255 scope global dynamic eth0
valid_lft 1111sec preferred_lft 1111sec
```

These few commands are sufficient to change the network interface name - provided you never need to reboot the system and are only using IPv4.

To persist the network interface name change across a reboot, you need to create an appropriate network interface configuration file for *eth0*.

```
# cd /etc/sysconfig/network-scripts/
# mv ifcfg-eno16777736 ifcfg-eth0
```

Use your favorite editor to modify *ifcfg-eth0* as follows:

```
WADDR="00:50:56:35:8E:C9"
TYPE="Ethernet"
BOOTPROTO="none"
NM_CONTROLLED="no"
IPADDR="192.168.0.100"
NETMASK="255.255.255.0"
DEFROUTE="yes"
PEERDNS="yes"
PEERROUTES="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_PEERDNS="yes"
IPV6_PEERROUTES="yes"
IPV6_FAILURE_FATAL="no"
NAME="eth0"
UUID="d033918c-8bae-433c-a87a-171d35ba8769"
ONBOOT="yes"
DEVICE="eth0"
```

The *HWADDR=* is a critical directive. It is used to determine the Ethernet network interface you wish to configure. is the hardware address of the Ethernet device in the form *AA:BB:CC:DD:EE:FF*. This directive is useful for machines with multiple NICs to ensure that the interfaces are assigned the correct device names regardless of the configured load order for each NIC's module. This directive should not be used in conjunction with *MACADDR*.

```
# systemctl restart network
# ip addr
1: lo: mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:50:56:35:8e:c9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.100/24 brd 192.168.0.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe35:8ec9/64 scope link
        valid_lft forever preferred_lft forever
```

Your *eth0* network interface configuration is now persistent and will survive a reboot.

Note that if you change the virtual MAC address assigned by VMware Workstation to the virtual network adapter, you will need to modify the MAC address in the *HWADDR* directive in the corresponding network interface file in */etc/sysconfig/network-scripts/*.

Suppose you needed to modify *ifcfg-eth0* to change the MAC address of the *HWADDR* directive. Obviously, you could use your favorite text editor, or you could write a shell script similar to the following:

```
#!/bin/bash

cd /etc/sysconfig/network-scripts/
MAC=$(cat /sys/class/net/eth0/address)
# optional, I happen to prefer an uppercased MAC address
MAC=$(echo $MAC | tr [:lower:] [:upper:])
sed -ire '/HWADDR/d' ifcfg-eth0
echo 'HWADDR="$MAC"' >> ifcfg-eth0
```

You then need execute *systemd restart network* for the new MAC address to be recognized.

Enjoy!