

Interface Renaming Scripts

June 2018

Please see the end of this page for the two scripts

For anyone that has used CentOS/Red Hat prior to version 7, or indeed other versions of Linux, the norm for network interface naming is to use **eth0**, **eth1**, **eth2** etc.

CentOS 7 changed this naming convention, for two main reasons.

1. Security (knowing particular interface names is considered a security risk).
2. Deterministic Naming. While the interfaces names used have always been the same, the mapping has been a bit hit and miss, and so on 10 identical servers, while the names are the same, eth0 may not always be the same physical interface (like Windows it can move).

Here we are looking at a couple of scripts written by an engineer here at work (not by me). While these scripts can work very well, there are a couple of things to be aware of.

Firstly the two scripts, these are:

1. `store_macaddr_mapping.sh` - stores server MAC addresses for use by the `rename_network_interfaces.sh` script.
2. `rename_network_interfaces.sh` - renames the interfaces using the information generated by `store_macaddr_mapping.sh`

`store_macaddr_mapping.sh`

The `store_macaddr_mapping.sh` is the trickier of the two scripts. I don't know what hardware this script was written for, but for me it has been a little troublesome. But it provides a good starting point for the renaming process. Copy this file to the server that needs the interfaces renaming, then give the file execute rights.

```
chmod 777 store_macaddr_mapping.sh
```

Now execute the file using the following syntax:

```
./store_macaddr_mapping.sh --output network_mapping.csv
```

A file `network_mapping.csv` will have been created. you can view this file using:

```
cat network_mapping.csv
```

Here is an example of what I have:

```
enp4s0f0;00:1e:67:d1:24:0e
;00:1e:67:d1:24:0f
;00:1e:67:d1:24:10
;00:1e:67:d1:24:11
;00:1e:67:d8:b3:68
;00:1e:67:d8:b3:69
```

The file should contain the following:

```
interface;mac
interface;mac
interface;mac
interface;mac
etc.
```

There are two issues here:

1. The interface names need to be eth0, eth1 etc.
2. The MAC Addresses are in the incorrect order, and this will break the mapping.

Both of these are easy to fix. Firstly you can manually edit the file, so that the interface names are correct, but more importantly, you need to get the MAC address order correct.

Use **ip link show** to show interfaces and MACS

```
ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens513f0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN mode DEFAULT qlen 1000
    link/ether 00:1e:67:d1:24:0e brd ff:ff:ff:ff:ff:ff
3: ens513f1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN mode DEFAULT qlen 1000
    link/ether 00:1e:67:d1:24:0f brd ff:ff:ff:ff:ff:ff
4: ens513f2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN mode DEFAULT qlen 1000
    link/ether 00:1e:67:d1:24:10 brd ff:ff:ff:ff:ff:ff
5: ens513f3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN mode DEFAULT qlen 1000
    link/ether 00:1e:67:d1:24:11 brd ff:ff:ff:ff:ff:ff
6: enp4s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT qlen 1000
    link/ether 00:1e:67:d8:b3:68 brd ff:ff:ff:ff:ff:ff
7: enp4s0f3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN mode DEFAULT qlen 1000
    link/ether 00:1e:67:d8:b3:69 brd ff:ff:ff:ff:ff:ff
```

From here we can see which MAC is associated with which interface (so I know that enp4s0f0 is the first interface on my server) this allows me to build the following file:

```
eth0;00:1e:67:d8:b3:68
eth1;00:1e:67:d8:b3:69
eth2;00:1e:67:d1:24:0e
eth3;00:1e:67:d1:24:0f
eth4;00:1e:67:d1:24:10
eth5;00:1e:67:d1:24:11
```

So now the **network_mapping.csv** file is ready to use.

rename_network_interfaces.sh

The next part is much simpler. Copy the rename_network_interfaces.sh to your server, and give it execute privileges using the following syntax:

```
chmod 777 rename_network_interfaces.sh
```

Now execute the script using the following syntax:

```
./rename_network_interfaces.sh --csv network_mapping.csv
```

If successful the following output will be observed (depending on what interface scripts already existed).

```
*** Getting macaddr to register ***
```

```

*** Using network_mapping.csv for network interface mapping ***
    Backup of /etc/sysconfig/network-scripts/ifcfg-ens513f3 in /etc/sysconfig/network-
scripts/backup/ifcfg-ens513f3
    Backup of /etc/sysconfig/network-scripts/ifcfg-ens513f1 in /etc/sysconfig/network-
scripts/backup/ifcfg-ens513f1
    Backup of /etc/sysconfig/network-scripts/ifcfg-enp4s0f0 in /etc/sysconfig/network-
scripts/backup/ifcfg-enp4s0f0
    Backup of /etc/sysconfig/network-scripts/ifcfg-enp4s0f3 in /etc/sysconfig/network-
scripts/backup/ifcfg-enp4s0f3
    Backup of /etc/sysconfig/network-scripts/ifcfg-ens513f2 in /etc/sysconfig/network-
scripts/backup/ifcfg-ens513f2
    Backup of /etc/sysconfig/network-scripts/ifcfg-ens513f0 in /etc/sysconfig/network-
scripts/backup/ifcfg-ens513f0
    Generating file /etc/sysconfig/network-scripts/ifcfg-eth0
    Generating file /etc/sysconfig/network-scripts/ifcfg-eth1
    Generating file /etc/sysconfig/network-scripts/ifcfg-eth2
    Generating file /etc/sysconfig/network-scripts/ifcfg-eth3
    Generating file /etc/sysconfig/network-scripts/ifcfg-eth4
    Generating file /etc/sysconfig/network-scripts/ifcfg-eth5

```

Check the /etc/sysconfig/network-scripts location to see the new files.

Note: No configuration information is copied from the old interface scripts, so before you reboot, you need to re-add all of your IP/Netmask/Gateway/DNS information.

Reboot the server to check the interfaces have been renamed, and the mapping is correct.

The Scripts

store_macaddr_mapping

```

#!/bin/bash

# this script is based on infos got from blog page:
# http://ask.xmodulo.com/change-network-interface-name-centos7.html

# exit at first error
set -o errexit

export IP_INTERFACES="ip addr"
export INTERFACE_ANCHOR="state UP"
export MACADDR_ANCHOR="link/ether"

export OUTPUT_FILE

function show_help() {
cat <<- EOF
Generate csv file with interface name; mac addr. Output file is a mandatory parameter.
    --output                Output file path where mac addre mapping will be stored
    --help|-h              [Optional] Display this help and exit
EOF
}

while [ $# -gt 0 ] ; do
    case $1 in
        --output) OUTPUT_FILE="$2";shift 2;;
        --help|-h) show_help;exit 0;;
        *) if ! [ -z "$1" ] ; then echo "Unknown parameter $1" ; show_help ; exit 1 ; else shift 1; fi
    ;;
    esac
done

[ -z "$OUTPUT_FILE" ] && show_help && exit 1

rm -f "$OUTPUT_FILE" > /dev/null 2>&1

```

```

echo "Generating $OUTPUT_FILE"
interface_name_list=$(($IP_INTERFACES | grep "$INTERFACE_ANCHOR" | awk '{print $2}' | sed 's|:|||'))
interface_name_list=( $interface_name_list )
macaddr_list=$(($IP_INTERFACES | grep "$MACADDR_ANCHOR" | awk '{print $2}' | tr '[:upper:]' '[:lower:]'))
macaddr_list=( $macaddr_list )
cnt=0
arraySize="${#macaddr_list[@]}"

while [ "$cnt" != "$arraySize" ] ; do
    echo "${interface_name_list[$cnt]};${macaddr_list[$cnt]}" >> "$OUTPUT_FILE"
    cnt=$(( cnt + 1 ))
done

```

rename_network_interfaces.sh

```

#!/bin/bash

# this script is based on infos got from blog page:
# http://ask.xmodulo.com/change-network-interface-name-centos7.html

# exit at first error
set -o errexit

IP_INTERFACES="ip addr"
MACADDR_ANCHOR="link/ether"
NET_NAME_TEMPLATE="eth"

CSV_FILE=

NETWORK_SCRIPT_DIR="/etc/sysconfig/network-scripts"
NETWORK_BACKUP_DIR="/etc/sysconfig/network-scripts/backup"
NETWORK_SCRIPT_TEMPLATE='NAME="<network_name>"
DEVICE="<network_name>"
ONBOOT=yes
NETBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
# HWADDR key is mandatory to use custom interface name: do not delete it
HWADDR=<mac_addr>'

function show_help() {
cat <<- EOF
Update network interfaces names. A csv file with the expected mapping can be given as arg.
    --csv                               Csv file containing <interface name>;<mac addr> as they
should be defined
    --help|-h                           [Optional] Display this help and exit

EOF
}

while [ $# -gt 0 ] ; do
    case $1 in
        --csv) CSV_FILE="$2";shift 2;;
        --help|-h) show_help;exit 0;;
        *) if ! [ -z "$1" ] ; then echo "Unknown parameter $1" ; show_help ; exit 1 ; else shift 1; fi
    ;;
    esac
done

echo "**** Getting macaddr to register ****"
macaddr_list=$(($IP_INTERFACES | grep "$MACADDR_ANCHOR" | awk '{print $2}'))

if ! [ -z "$CSV_FILE" ] && [ -f "$CSV_FILE" ] ; then
    echo "**** Using $CSV_FILE for network interface mapping ****"
    ! [ -d "$NETWORK_BACKUP_DIR" ] && mkdir -p "$NETWORK_BACKUP_DIR"
    for old_script_file in $(find $NETWORK_SCRIPT_DIR -maxdepth 1 -name "ifcfg-*" -a -not -name "ifcfg-lo" -a -not -name "ifcfg-*\.*"); do
        echo -e "\tBackup of $old_script_file in $NETWORK_BACKUP_DIR/${basename "$old_script_file"}"
        mv -f "$old_script_file" $NETWORK_BACKUP_DIR/.
    done

```

```
done
while read interface_data
do
    [ -z "$interface_data" ] && continue
    interface_name=$(echo "$interface_data" | cut -d \; -f1)
    mac_addr=$(echo "$interface_data" | cut -d \; -f2 | tr '[:upper:]' '[:lower:]')
    [ -z "$interface_name" ] && continue
    [ -z "$mac_addr" ] && continue
    # check that given mac addr exist
    if ! echo "$macaddr_list" | grep "$mac_addr" > /dev/null 2>&1 ; then
        echo "Unable to find $mac_addr in available network interfaces, please review file content
of $CSV_FILE"
        exit 1
    fi
    echo -e "\tGenerating file $NETWORK_SCRIPT_DIR/ifcfg-$interface_name"
    echo "$NETWORK_SCRIPT_TEMPLATE" > $NETWORK_SCRIPT_DIR/ifcfg-"$interface_name"
    sed -i "s|<network_name>|$interface_name|g" $NETWORK_SCRIPT_DIR/ifcfg-"$interface_name"
    sed -i "s|<mac_addr>|$mac_addr|g" $NETWORK_SCRIPT_DIR/ifcfg-"$interface_name"
done < "$CSV_FILE"

else
    echo "*** Using default naming (${NET_NAME_TEMPLATE}x) for network interface mapping ***"
    ! [ -d "$NETWORK_BACKUP_DIR" ] && mkdir -p "$NETWORK_BACKUP_DIR"
    cnt=0
    for old_script_file in $(find $NETWORK_SCRIPT_DIR -maxdepth 1 -name "ifcfg-*" -a -not -name "ifcfg-
lo" -a -not -name "ifcfg-*\.*"); do
        echo -e "\tBackup of $old_script_file in $NETWORK_BACKUP_DIR/$(basename "$old_script_file")"
        mv -f "$old_script_file" $NETWORK_BACKUP_DIR/.
    done
    for mac_addr in ${macaddr_list[*]}; do
        mac_addr=$(echo "$mac_addr" | tr '[:upper:]' '[:lower:]')
        echo -e "\tGenerating file $NETWORK_SCRIPT_DIR/ifcfg-$NET_NAME_TEMPLATE$cnt"
        echo "$NETWORK_SCRIPT_TEMPLATE" > $NETWORK_SCRIPT_DIR/ifcfg-$NET_NAME_TEMPLATE$cnt
        sed -i "s|<network_name>|$NET_NAME_TEMPLATE$cnt|g" $NETWORK_SCRIPT_DIR/ifcfg-
$NET_NAME_TEMPLATE$cnt
        sed -i "s|<mac_addr>|$mac_addr|g" $NETWORK_SCRIPT_DIR/ifcfg-$NET_NAME_TEMPLATE$cnt
        cnt=$(( cnt + 1 ))
    done
fi

echo "*** Please check consistency of generated files in directory $NETWORK_SCRIPT_DIR ***"
```

From:
<https://wiki.alanwalker.uk/> - **WalkerWiki** - wiki.alanwalker.uk

Permanent link:
https://wiki.alanwalker.uk/doku.php?id=wiki:interface_renaming_scripts

Last update: **2018/07/31 07:26**

