

Learn CLI commands used to operate WiFi stations.

Goal: Compare and learn script and CLI commands used when creating and operating stations.

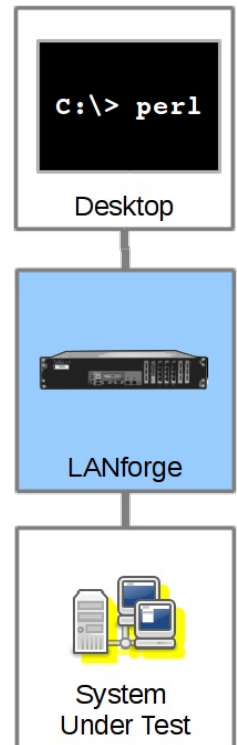
The LANforge perl scripts have always been able to print out the CLI commands used to communicate with the LANforger manager. These examples show recent modifications that allow you to collect the CLI commands more easily using the `--log_cli` switch. (Not all scripts have this feature yet). The `--log_cli` switch prints CLI commands to your console. You can collect those commands in a file using the switch with an argument of the file name:

```
$ ./lf_vue_mod.sh --log_cli /tmp/clilog.txt
```

It is possible to repeat these commands using the `lf_firemod.pl` script:

```
$ ./lf_firemod.pl --mgr localhost --action do_cmd \  
--cmd "set_port 1 2 sta200 NA NA NA NA 0 NA NA NA NA 8388610"
```

Requires a LANforge CT520 (or better) system and an access point.



Examples of CLI commands

1. Creating Stations
2. Using Open Authentication
3. Using WPA2 Authentication
4. Static IP Addresses
5. Station DHCP IP Address
6. Creating a Station with a MAC Address Pattern
7. Admin Down
8. Admin Up
9. Deleting a Station
10. Creating Connections and Running Traffic

11. Starting and Stopping Traffic
12. Create a Layer 3 TCP Connection
13. Create a Layer 3 UDP Connection
14. Create a Layer 4 Web Connection

Setting for Examples

This was done in a two-machine LANforge cluster, the manager named *jedtest* and the second resource named *kedtest*. The CLI output of these CLI commands has been discarded as well as any `show_port` commands.

The `show_port` commands are useful for inspecting the results of previous commands. Often there is useful wait before issuing the `show_port` command to allow processing time on the manager. Please inspect the scripts in the `/home/lanforge/scripts` directory for how and when they tend to sleep.

These commands are also found in the `/home/lanforge/DB/DFLT` directory files. You cannot run those DB files directly, because they are executed in certain order. However, you can grep for connection- and station-names in those files to find results of GUI commands.

Creating Stations

Using Open Authentication

This station is created with DHCP enabled. That is controlled via *flags* that are described in the `add_sta` command.

Shell script:

```
./lf_vue_mod.sh --mgr jedtest --resource 2 --create_sta --name sta100 --radio wiphy0 --security open --ssid jedtest
```

Perl script:

```
./lf_associate_ap.pl --mgr jedtest --resource 2 --quiet yes --action add --radio wiphy0 --security open --ssid jedtest --passphrase --first_sta sta100 --first_ip DHCP --num_stations 1
```

CLI command:

```
set_wifi_radio 1 2 wiphy0 NA -1 NA NA NA NA NA NA NA NA 0x1 NA
add_sta 1 2 wiphy0 sta100 0 jedtest NA [BLANK] AUTO NA 00:0e:8e:8d:8d:e9 8 NA NA NA NA NA
set_port 1 2 sta100 0.0.0.0 255.255.0.0 0.0.0.0 NA 2147483648 00:0e:8e:8d:8d:e9 NA NA NA 8
```

Using WPA2 Authentication

This station is created with DHCP enabled. That is controlled via *flags* that are described in the `add_sta` command.

Shell script:

```
$ ./lf_vue_mod.sh --mgr jedtest --resource 2 --create_sta --name sta200 \
--radio wiphy1 --security wpa2 --ssid jedtest --passphrase jedtest1 \
--log_cli /tmp/clilog.txt
```

Perl script:

```
./lf_associate_ap.pl --mgr jedtest --resource 2 \  
--action add --radio wiphy1 --security wpa2 --ssid jedtest \  
--passphrase jedtest1 --first_sta sta200 --first_ip DHCP --num_stations 1
```

CLI command:

```
set_wifi_radio 1 2 wiphy1 NA -1 NA NA NA NA NA NA NA NA 0x1 NA  
add_sta 1 2 wiphy1 sta200 1024 jedtest NA jedtest1 AUTO NA 00:0e:8e:6f:01:62 8 NA NA NA NA  
set_port 1 2 sta200 0.0.0.0 255.255.0.0 0.0.0.0 NA 2147483648 00:0e:8e:6f:01:62 NA NA NA 8
```

Static IP Addresses

Here is an example of creating a virtual station with a static address: 10.26.2.14/255.255.254.0

Shell Script:

```
--
```

Perl Script:

```
./lf_associate_ap.pl --mgr jedtest --resource 2 --action add --radio wiphy1 --first_sta sta203 --  
first_ip 10.26.2.4 --netmask 255.255.254.0 --ssid jedtest --security wpa2 --passphrase jedtest1 --  
num_stations 1 --wifi_mode abgnAC --log_cli /tmp/clilog.txt
```

CLI Command:

```
set_wifi_radio 1 2 wiphy1 NA -1 NA NA NA NA NA NA NA NA 0x1 NA  
show_port 1 2 wiphy1  
add_sta 1 2 wiphy1 sta203 1024 jedtest NA jedtest1 AUTO NA 00:0e:8e:63:50:62 8 NA NA NA NA  
set_port 1 2 sta100 10.26.2.4 255.255.254.0 0.0.0.0 NA 0 00:0e:8e:63:50:62 NA NA NA 838865
```

Station DHCP IP Address

For the station to gain a DHCP IP address, you have to *admin-up* the station.

Creating a Station with a MAC Address Pattern

The `lf_associate_ap` script contains logic that parses a MAC address pattern and produces new MAC addresses. This is not a feature of the LANforge Manager. Your CLI calls to the LANforge manager will not parse the mask.

The pattern nomenclature of the LANforge GUI can also be used when specifying a MAC address for stations:

| | |
|---------|--------------------------------|
| xx | keep parent radio octet |
| * | randomize this octet |
| 00 - ff | assign this value to the octet |

Shell script:

```
--
```

Perl script:

```
./lf_associate_ap.pl --mgr jedtest --resource 2 --action add --radio wiphy1 --first_sta sta205 --  
first_ip 10.26.2.4 --netmask 255.255.254.0 --ssid jedtest --security wpa2 --passphrase jedtest1 --  
num_stations 1 --mac-pattern '4e:xx:xx:xx:*:01' --log_cli /tmp/clilog.txt
```

CLI command:

```
set_wifi_radio 1 2 wiphy1 NA -1 NA NA NA NA NA NA NA NA 0x1 NA
```

```
show_port 1 2 wiphy1
add_sta 1 2 wiphy1 sta205 1024 jedtest NA jedtest1 AUTO NA 4e:0e:8e:43:f1:01 8 NA NA NA NA
set_port 1 2 sta205 10.26.2.4 255.255.254.0 0.0.0.0 NA 0 4e:0e:8e:43:f1:01 NA NA NA 838865
```

Admin Down

Shell script:

```
./lf_vue_mod.sh --mgr jedtest --resource 2 --down --name sta200 --log_cli /tmp/clilog.txt --quiet 1
```

Perl script:

```
./lf_portmod.pl --manager jedtest --card 2 --port_name sta200 --set_ifstate down
```

CLI command:

```
set_port 1 2 sta200 NA NA NA NA 1 NA NA NA NA 8388610
```

Admin Up

Shell script:

```
./lf_vue_mod.sh --mgr jedtest --resource 2 --up --name sta200 --log_cli /tmp/clilog.txt --quiet 1
```

Perl script:

```
./lf_portmod.pl --manager jedtest --card 2 --port_name sta200 --set_ifstate up
```

CLI command:

```
set_port 1 2 sta200 NA NA NA NA 0 NA NA NA NA 8388610
```

Delete Station

Shell script:

```
./lf_vue_mod.sh --mgr jedtest --resource 2 --delete_sta --name sta200 --log_cli /tmp/clilog.txt -
-quiet 1
```

Perl script:

```
./lf_associate_ap.pl --mgr jedtest --resource 2 --action del --port_del sta200
```

CLI command:

```
rm_vlan 1 2 sta100
```

Creating Connections and Running Traffic

LANforge can create Layer-3 and Layer-4 connections using the `lf_vue_mod.sh` script. When connections are created, they exist in a *stopped* state. Connections can then have their state changed to *RUNNING* to start traffic.

Starting and Stopping Traffic

Layer-3 and Layer-4 connections both subject to the states STOPPED, RUNNING, and QUIESCSE.

Shell script:

```
./lf_vue_mod.sh --mgr jedtest --start_cx --name tcp200 --log_cli /tmp/clilog.txt --quiet 1
```

```
./lf_vue_mod.sh --mgr jedtest --stop_cx --name tcp200 --log_cli /tmp/clilog.txt --quiet 1
```

Perl script:

```
./lf_firemod.pl --mgr jedtest --action do_cmd --cmd "set_cx_state default_tm tcp200 RUNNING"
```

```
./lf_firemod.pl --mgr jedtest --action do_cmd --cmd "set_cx_state default_tm tcp200 STOPPED"
```

CLI commands:

```
set_cx_state default_tm tcp200 RUNNING
```

```
set_cx_state default_tm tcp200 STOPPED
```

Create a Layer 3 TCP Connection

Shell script:

```
./lf_vue_mod.sh --mgr jedtest --resource 2 --create_cx --name tcp200 --tcp --sta sta200 --port eth1 --bps 1000000 --log_cli /tmp/clilog.txt --quiet 1
```

Perl script:

```
./lf_firemod.pl --mgr jedtest --resource 2 --action create_endp --endp_name tcp200-A --spe  
./lf_firemod.pl --mgr jedtest --resource 2 --action create_endp --endp_name tcp200-B --spe  
./lf_firemod.pl --mgr jedtest --resource 2 --action create_cx --cx_name tcp200 --cx_endps
```

CLI commands:

```
add_endp tcp200-A 1 2 sta200 lf_tcp -1 NA 1000000 1000000 NA -1 -1 increasing NO NA 0 0  
set_endp_report_timer tcp200-A 5000  
add_endp tcp200-B 1 2 eth1 lf_tcp -1 NA 1000000 1000000 NA -1 -1 increasing NO NA 0 0  
set_endp_report_timer tcp200-B 5000  
add_cx tcp200 default_tm tcp200-A tcp200-B  
set_cx_report_timer default_tm tcp200 5000 NA
```

Create a Layer 3 UDP Connection

Shell script:

```
./lf_vue_mod.sh --mgr jedtest --resource 2 --create_cx --name upd200 --udp --sta sta200 --port eth1 --bps 2000000 --log_cli /tmp/clilog.txt --quiet 1
```

Perl script:

```
./lf_firemod.pl --mgr jedtest --resource 2 --log_cli /tmp/clilog.txt --quiet 1 --action cr  
./lf_firemod.pl --mgr jedtest --resource 2 --log_cli /tmp/clilog.txt --quiet 1 --action cr  
./lf_firemod.pl --mgr jedtest --resource 2 --log_cli /tmp/clilog.txt --quiet 1 --action cr
```

CLI commands:

```
add_endp udp200-A 1 2 sta200 lf_udp -1 NA 2000000 2000000 NA -1 -1 increasing NO NA 0 0  
set_endp_report_timer udp200-A 5000  
add_endp udp200-B 1 2 eth1 lf_udp -1 NA 2000000 2000000 NA -1 -1 increasing NO NA 0 0  
set_endp_report_timer udp200-B 5000
```

```
add_cx udp200 default_tm udp200-A udp200-B
set_cx_report_timer default_tm udp200 5000 NA
```

Create a Layer 4 Web Connection

Layer-4 connections are created with a one-sided technique, the curl command always operates on the **A-side** and the **B-side** is unmanaged. The endpoint and connection naming does not follow the Layer-3 convention.

Shell script:

```
./lf_vue_mod.sh --mgr jedtest --resource 2 --create_l4 --name yh200 --sta sta200 --url
http://www.yahoo.com/ --utm 2400 --log_cli /tmp/cliilog.txt --quiet 1
```

Perl script:

Commands are set using `lf_firemod.pl --action do_cmd --cmd ...`

CLI commands:

```
add_l4_endp yh200 1 2 sta200 l4_generic 0 10000 2400 'dl http://www.yahoo.com/ /dev/null'
set_endp_tos yh200 DONT-SET 0
set_endp_flag yh200 L4Enable404 0
set_endp_report_timer yh200 5000
set_endp_flag yh200 ClearPortOnStart 0
set_endp_quiesce yh200 3
add_cx CX_yh200 default_tm yh200
```