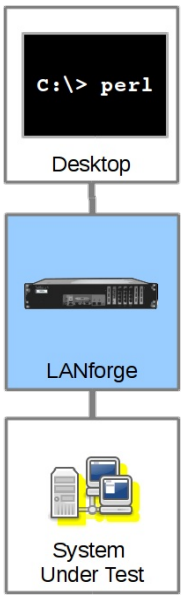
sales@candelatech.com
support@candelatech.com
+1 (360) 380-1618 [PST, GMT -8]

Associating stations with the lf_associate_ap script.

Goal: Create, destroy, start and stop virtual stations without needing to use the LANforge GUI.

Automated wireless traffic is possible using the `lf_associate_ap.pl` script. This script can be run within the LANforge server or outside the LANforge Server (on a windows desktop). The output of the script should be redirected to a text file if you want to review the resuts. Use this file in conjunction with the `lf_firemod.pl` script to create traffic. Requires a LANforge CT520 (or better) system and an access point.



Script Capabilities

The `lf_assocatiate_ap.pl` script has many options, but here are the basic actions:

- Create stations and cross connects with them, running traffic for a specified amount of time (action: step1)
- Generate stress on the AP by repeatedly bringing up stations and taking them down (action: step2).

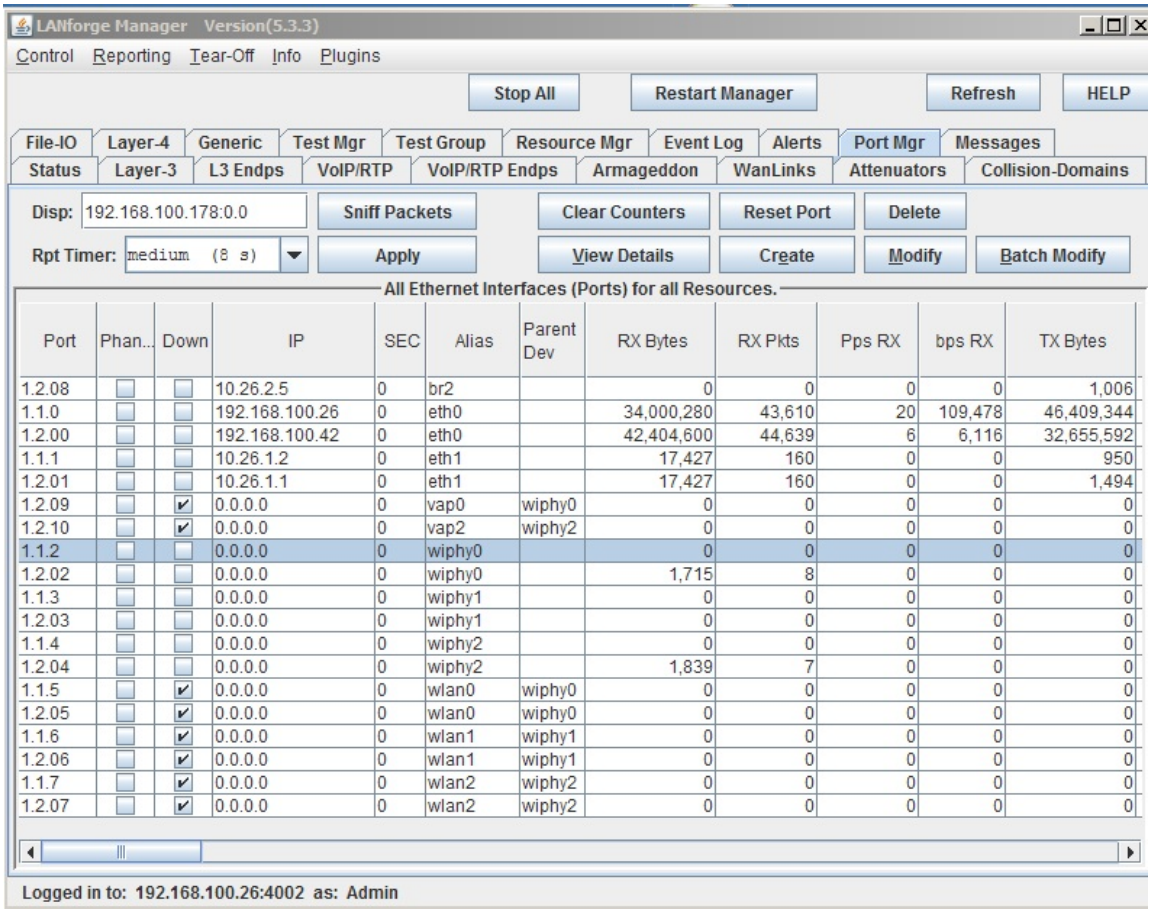
Before you begin

1. We assume you have a separate WiFi access point in routed mode. These examples can be used on a CT523 (or better) system with more than one radio if you want to practice the techniques. You would dedicate a radio to be a virtual AP (see cookbook).
2. For these examples, our AP will be open with no username or password, and the SSID will be **jedtest**
3. If you want to run scripts from your Windows desktop, you have ActivePerl installed.

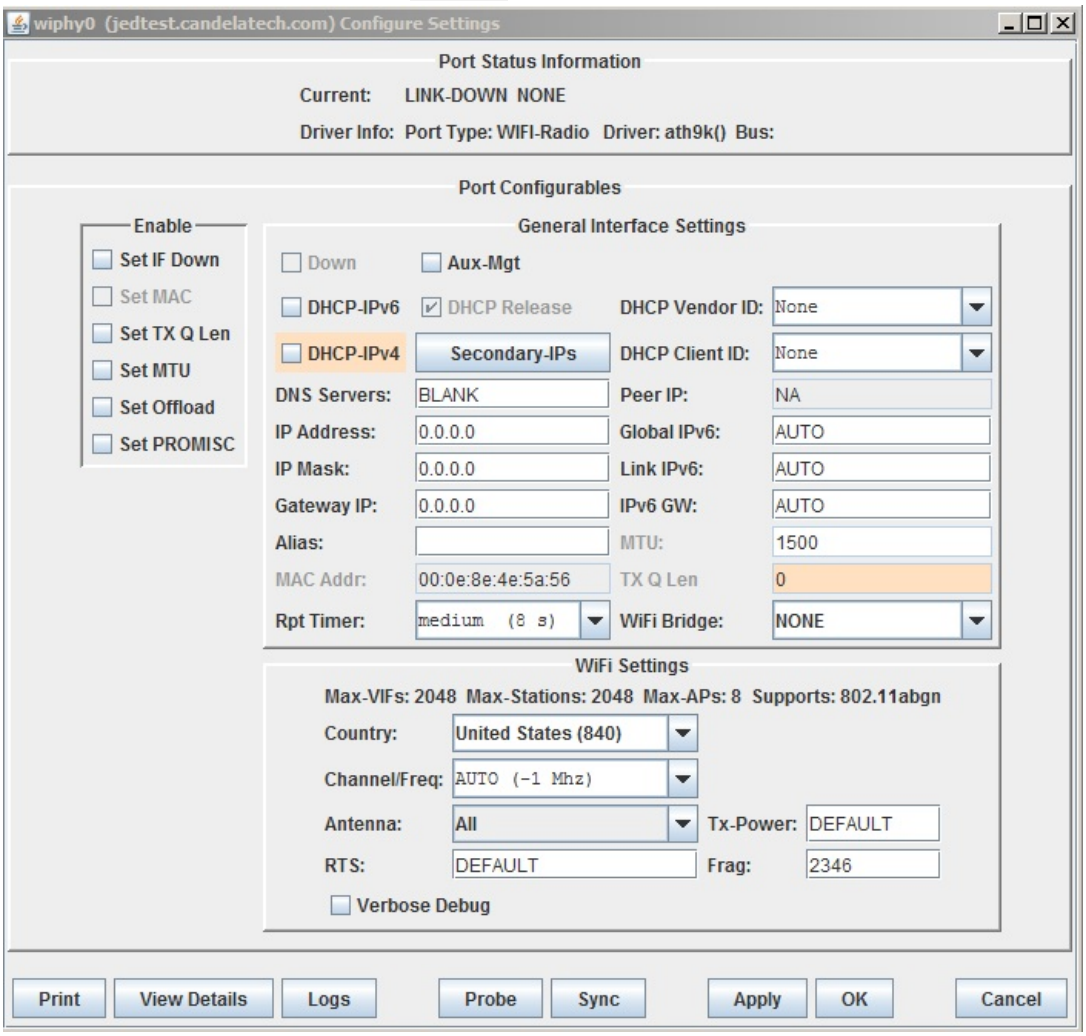
Creating a virtual station with traffic

Using `lf_associate_ap` on Windows

1. In the LANforge GUI, we will inspect our `wiphy0` radio.



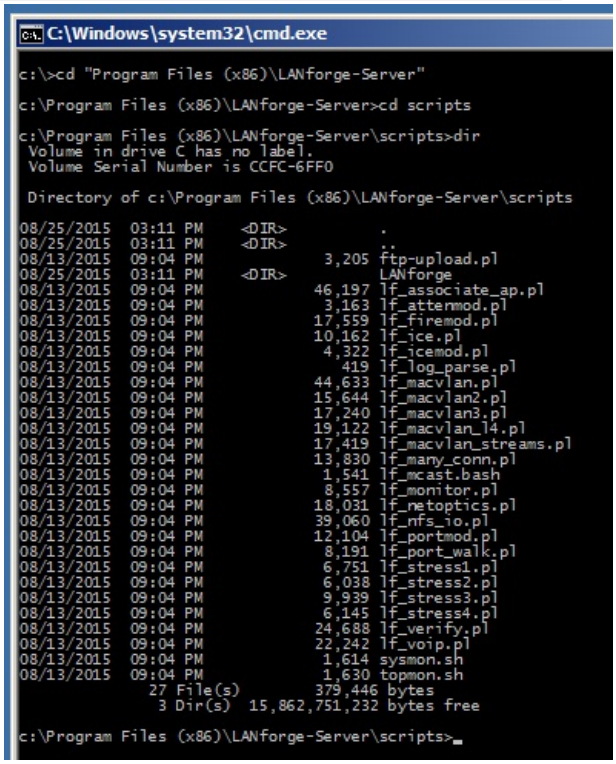
And the radio should be set to channel `-1 AUTO`



i CMD window shortcut: ⊞ R `cmd` ↵

i LANforge Scripts are at `C:\Program Files\LANforge-Server\scripts`

2. `cd C:\Program Files\LANforge-Server\scripts`



3. `perl .\lf_associate_ap.pl --help` Will show you the script options.

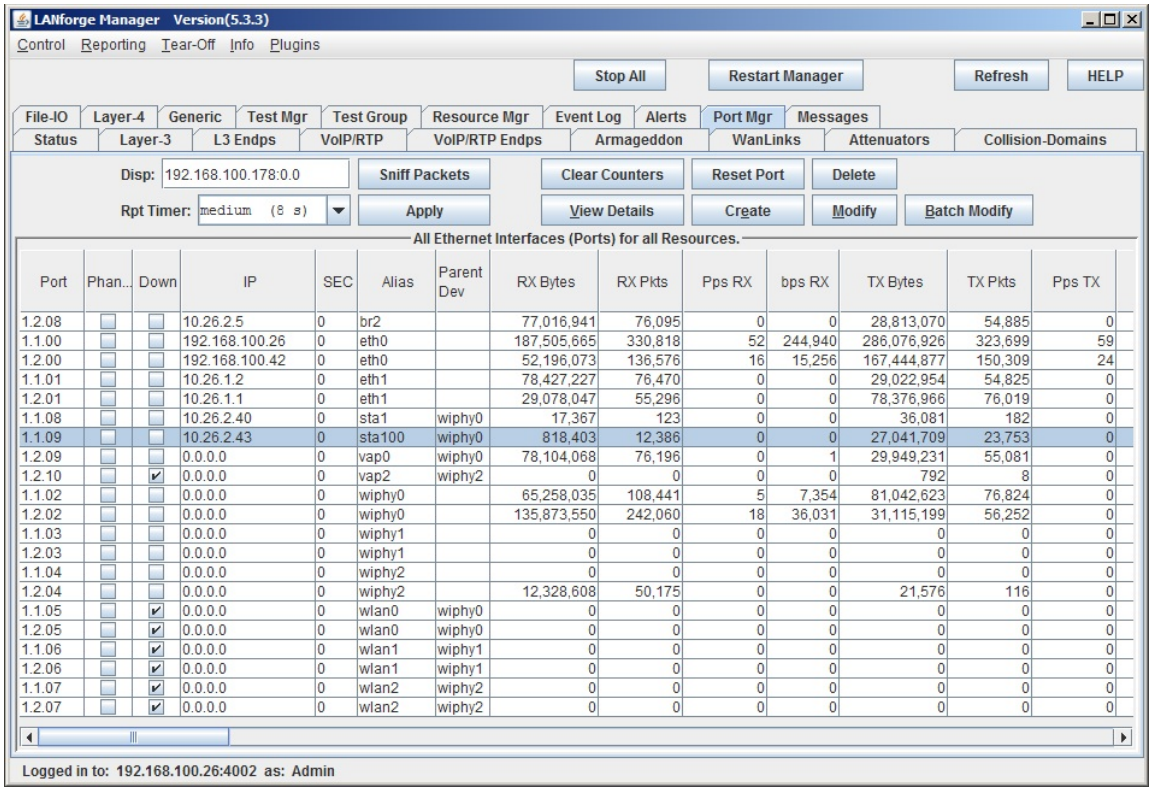
4. We can create a virtual station with this command:

```
perl .\lf_associate_ap.pl --resource 1  --resource 1 --mgr jedtest ^
   --action step1       --radio wiphy0    --ssid jedtest ^
   --first_sta sta100   --num_stations 1  --duration 20 ^
   --first_ip DHCP      --upstream eth1 --security wpa2 --passphrase jedtest1
```

i Long DOS commands and be continued on the next line with the ^ character.

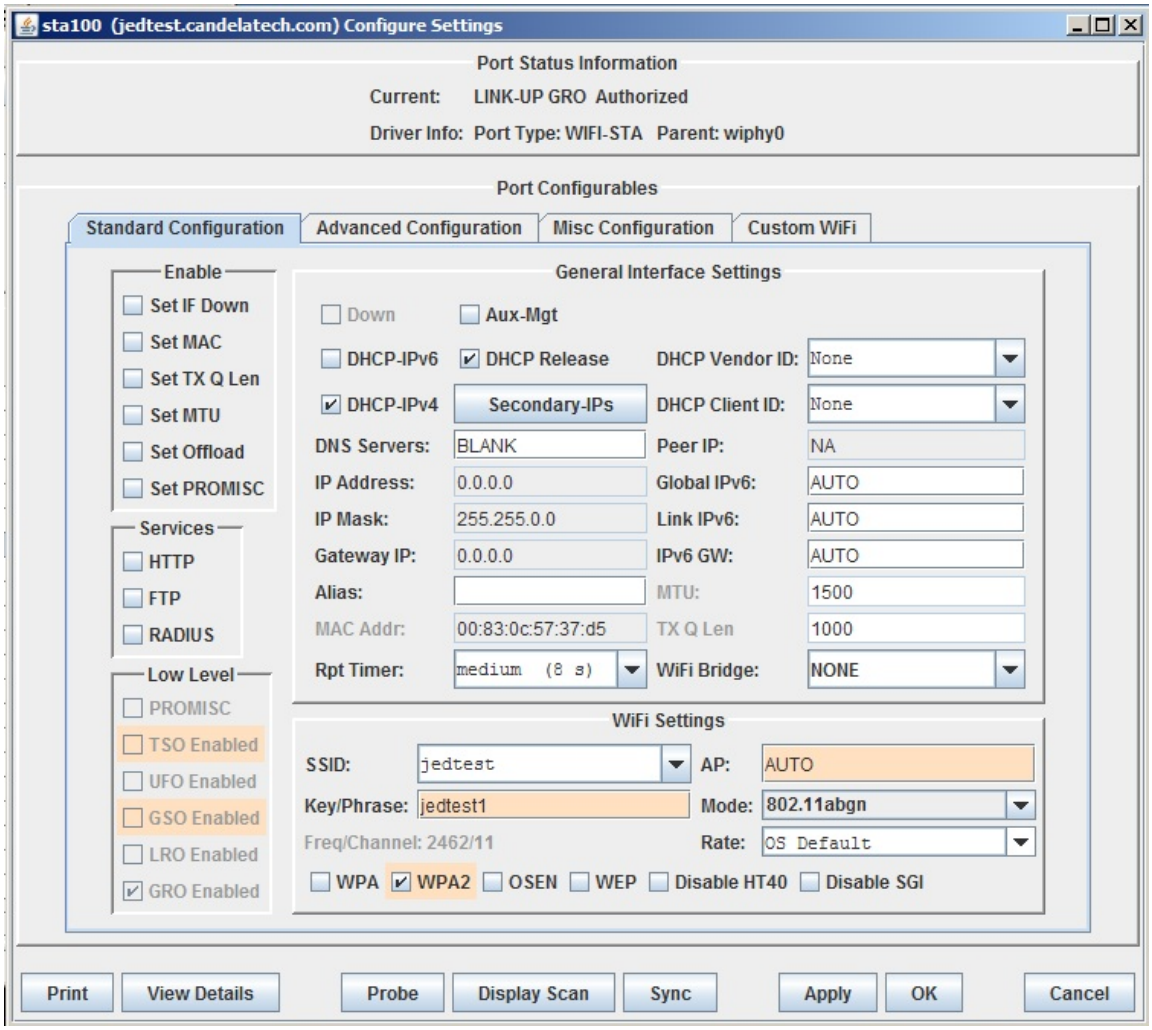
```
c:\Program Files (x86)\LANforge-Server\scripts>perl .\lf_associate_ap.pl --resource 1  --resource 1 --mgr jedtest ^
More?    --action step1       --radio wiphy0    --ssid jedtest ^
More?    --first_sta sta100   --num_stations 1  --duration 20 ^
More?    --first_ip DHCP      --upstream eth1 --security wpa2 --passphrase jedtest1
Removing old cross-connects, and endpoints ...
 cx-100 (ep-A100 - ep-B100)... done.
Deleting ports:...sta100 port sta100 not present,  not found,  done.
Waiting for 1 stations to be removed... sta100, Old stations removed
 Creating new stations:  sta100  Created 1 stations
Waiting for stations to associate... 1/1 seen to associate

Creating connections:  cx-100 (sta100 - eth1), done.
Adjusting cx min/max tx for upload test:  cx-100...done.
 started uploads.
  ep-A100  Rx-bps/Tx-B  ep-B100  Rx-bps/Tx-B |
  0bps / 6MB      10Mbps / 0B    |
  0bps / 12MB     10Mbps / 0B    |
  0bps / 18MB     10Mbps / 0B    |
  0bps / 24MB     10Mbps / 0B    |
ep-A100:        Tx Bytes:           Total: 24998120    Time: 60s   Cur: 25000203  416670/s
                Rx Bytes:           Total: 0           Time: 60s   Cur: 0         0/s
ep-B100:        Tx Bytes:           Total: 0           Time: 60s   Cur: 0         0/s
                Rx Bytes:           Total: 24998120    Time: 60s   Cur: 24999369  416656/s
Adjusting tx_rate for download... cx-100...done

Started download...
  ep-A100  Rx-bps/Tx-B  ep-B100  Rx-bps/Tx-B |
  0bps / 0B       0bps / 0B      |
  0bps / 0B       0bps / 0B      |
  0bps / 0B       0bps / 0B      |
  0bps / 0B       0bps / 0B      |
ep-A100:        Tx Bytes:           Total: 0           Time: 60s   Cur: 0         0/s
                Rx Bytes:           Total: 0           Time: 60s   Cur: 0         0/s
ep-B100:        Tx Bytes:           Total: 0           Time: 60s   Cur: 0         0/s
                Rx Bytes:           Total: 0           Time: 60s   Cur: 0         0/s

c:\Program Files (x86)\LANforge-Server\scripts>
```

5. We can see the port appear in the LANforge GUI:



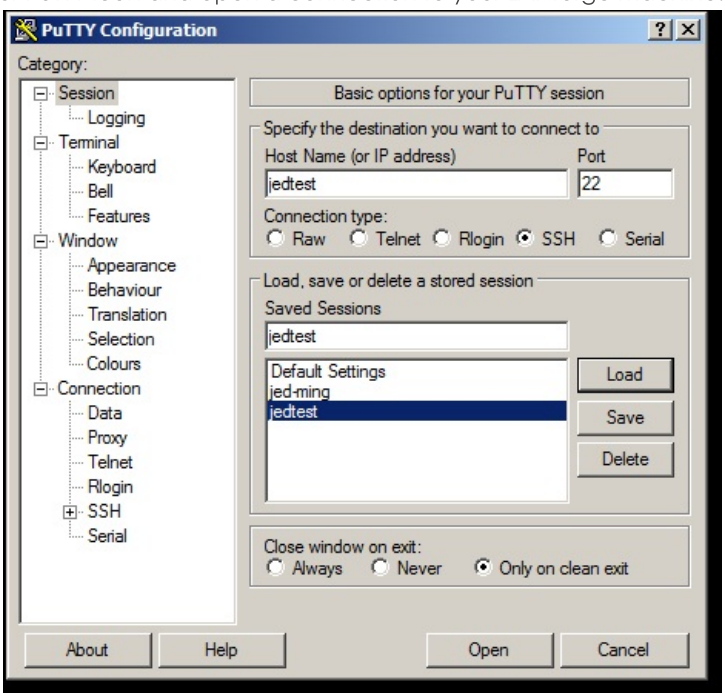
Port	Phan...	Down	IP	SEC	Alias	Parent Dev	RX Bytes	RX Pkts	Pps RX	bps RX	TX Bytes	TX Pkts	Pps TX
1.2.08	☐	☐	10.26.2.5	0	br2		77,016,941	76,095	0	0	28,813,070	54,885	0
1.1.00	☐	☐	192.168.100.26	0	eth0		187,505,665	330,818	52	244,940	286,076,926	323,699	59
1.2.00	☐	☐	192.168.100.42	0	eth0		52,196,073	136,576	16	15,256	167,444,877	150,309	24
1.1.01	☐	☐	10.26.1.2	0	eth1		78,427,227	76,470	0	0	29,022,954	54,825	0
1.2.01	☐	☐	10.26.1.1	0	eth1		29,078,047	55,296	0	0	78,376,966	76,019	0
1.1.08	☐	☐	10.26.2.40	0	sta1	wiphy0	17,367	123	0	0	36,081	182	0
1.1.09	☐	☐	10.26.2.43	0	sta100	wiphy0	818,403	12,386	0	0	27,041,709	23,753	0
1.2.09	☐	☐	0.0.0.0	0	vap0	wiphy0	78,104,068	76,196	0	1	29,949,231	55,081	0
1.2.10	☐	☑	0.0.0.0	0	vap2	wiphy2	0	0	0	0	792	8	0
1.1.02	☐	☐	0.0.0.0	0	wiphy0		65,258,035	108,441	5	7,354	81,042,623	76,824	0
1.2.02	☐	☐	0.0.0.0	0	wiphy0		135,873,550	242,060	18	36,031	31,115,199	56,252	0
1.1.03	☐	☐	0.0.0.0	0	wiphy1		0	0	0	0	0	0	0
1.2.03	☐	☐	0.0.0.0	0	wiphy1		0	0	0	0	0	0	0
1.1.04	☐	☐	0.0.0.0	0	wiphy2		0	0	0	0	0	0	0
1.2.04	☐	☐	0.0.0.0	0	wiphy2		12,328,608	50,175	0	0	21,576	116	0
1.1.05	☐	☑	0.0.0.0	0	wlan0	wiphy0	0	0	0	0	0	0	0
1.2.05	☐	☑	0.0.0.0	0	wlan0	wiphy0	0	0	0	0	0	0	0
1.1.06	☐	☑	0.0.0.0	0	wlan1	wiphy1	0	0	0	0	0	0	0
1.2.06	☐	☑	0.0.0.0	0	wlan1	wiphy1	0	0	0	0	0	0	0
1.1.07	☐	☑	0.0.0.0	0	wlan2	wiphy2	0	0	0	0	0	0	0
1.2.07	☐	☑	0.0.0.0	0	wlan2	wiphy2	0	0	0	0	0	0	0

and we can inspect it.

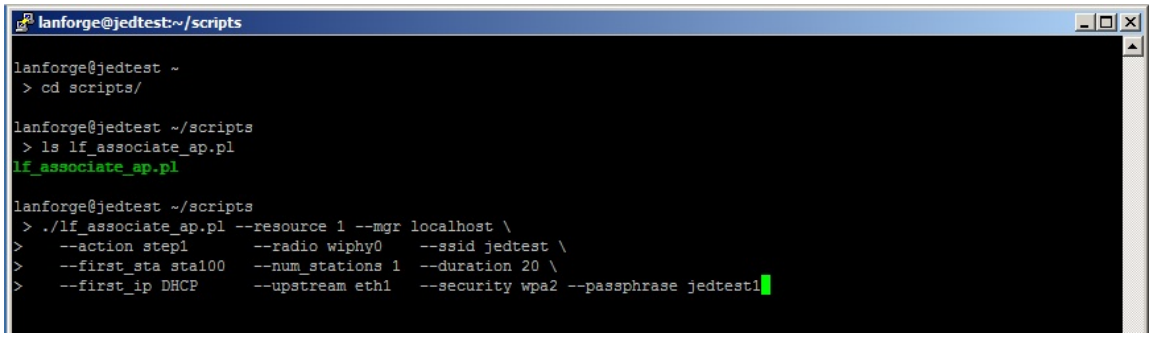


Using `lf_associate_ap` on Linux

1. Double click on your PuTTY icon and open a connection to your LANforge machine.



2. The `lf_associate_ap.pl` script is in the scripts sub directory.

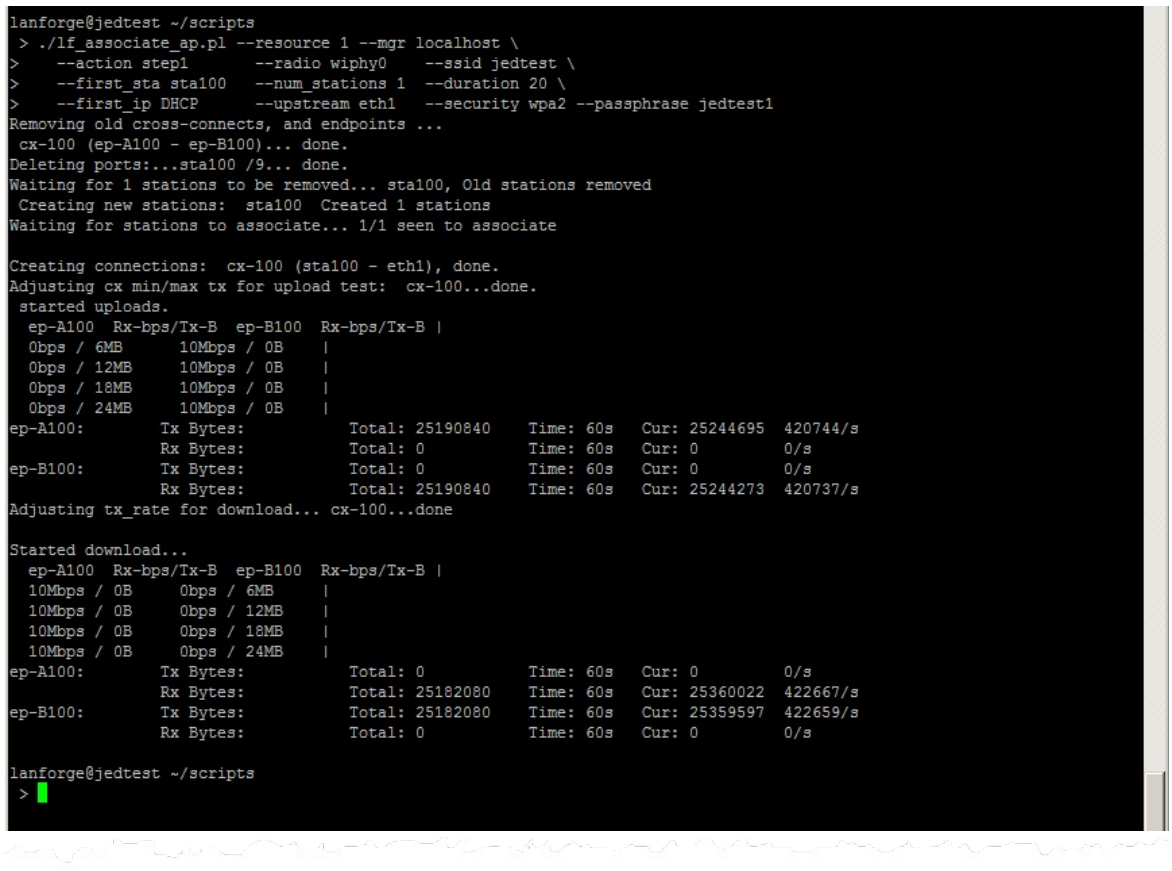


3. Our command is basically the same.

> i Long shell commands and be continued on the next line with the \ character.

```
./lf_associate_ap.pl --resource 1 --mgr localhost \
   --action step1       --radio wiphy0    --ssid jedtest \
   --first_sta sta100   --num_stations 1  --duration 20 \
   --first_ip DHCP      --upstream eth1   --security wpa2 --passphrase jedtest1
```

4. We will see similar output:



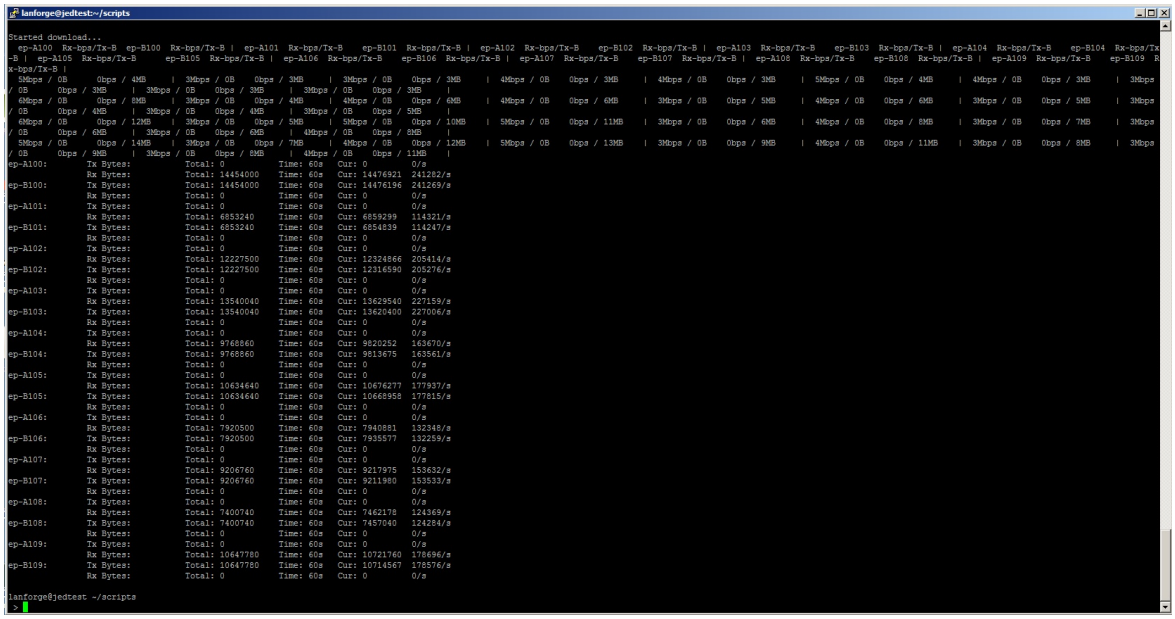
More Traffic Examples

1. Creating Multiple stations that transm

```
./lf_associate_ap.pl --resource 1 --mgr localhost \
   --action step1       --radio wiphy0    --ssid jedtest \
   --first_sta sta100 --num_stations 10 --duration 20 \
   --first_ip DHCP      --upstream eth1   --security wpa2 --passphrase jedtest1
```

2. Creating TCP/IP bursty traffic from 30Mbps to 450 Mbps

```
./lf_associate_ap.pl --resource 1 --mgr localhost \
   --action step1       --radio wiphy0    --ssid jedtest \
   --first_sta sta100   --num_stations 10 --duration 120 \
   --first_ip DHCP      --upstream eth1   --security wpa2 --passphrase jedtest1 \
   --cxtype tcp --bps-min 30Mpbs \
   --bps-max 450Mbps
```



3. Capturing that report with redirection

```
./lf_associate_ap.pl --resource 1 --mgr localhost \
   --action step1       --radio wiphy0    --ssid jedtest \
   --first_sta sta100   --num_stations 10 --duration 120 \
   --first_ip DHCP      --upstream eth1   --security wpa2 --passphrase jedtest1 \
   --cxtype tcp --bps-min 30Mpbs --bps-max 450Mbps &> report.txt
```

> i Both DOS and Linux command output can be saved to a file with the **`&>`** operator.

> i Both DOS and Linux files can be viewed with the **`more`** command.

4. Creating steady UDP traffic to at 450Mbps

```
$ ./lf_associate_ap.pl --resource 1 --mgr localhost \
   --action step1       --radio wiphy0    --ssid jedtest \
   --first_sta sta100   --num_stations 10 --duration 120 \
   --first_ip DHCP      --upstream eth1   --security wpa2 --passphrase jedtest1 \
   --cxtype udp --bps-min 450Mpbs \
   --bps-max SAME &> report.txt
$  more report.txt
```

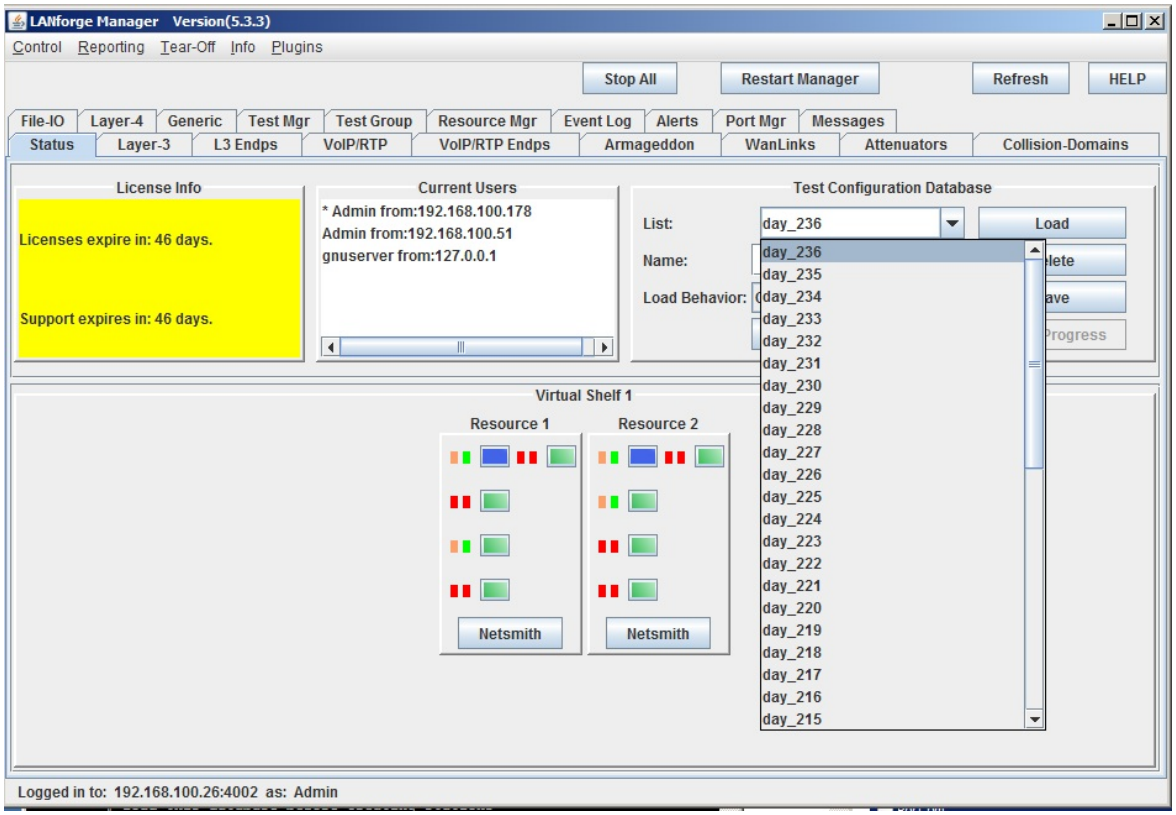
5. Associating to an open AP

```
./lf_associate_ap.pl --resource 1 --mgr localhost \
   --action step1       --radio wiphy0    --ssid jedtest \
   --first_sta sta100   --num_stations 10 --duration 120 \
   --first_ip DHCP      --upstream eth1   --security open
```

6. Connecting a station at 802.11/abg speeds

```
./lf_associate_ap.pl --resource 1 --mgr localhost \
   --action step1       --radio wiphy0    --ssid jedtest \
   --first_sta sta100   --num_stations 10 --duration 120 \
   --first_ip DHCP      --upstream eth1 --security open \
   --wifi_mode abg
```

7. Initializing your test secenario by pre-loading a database. The database is the same name as the dropdown in the GUI Status tab.

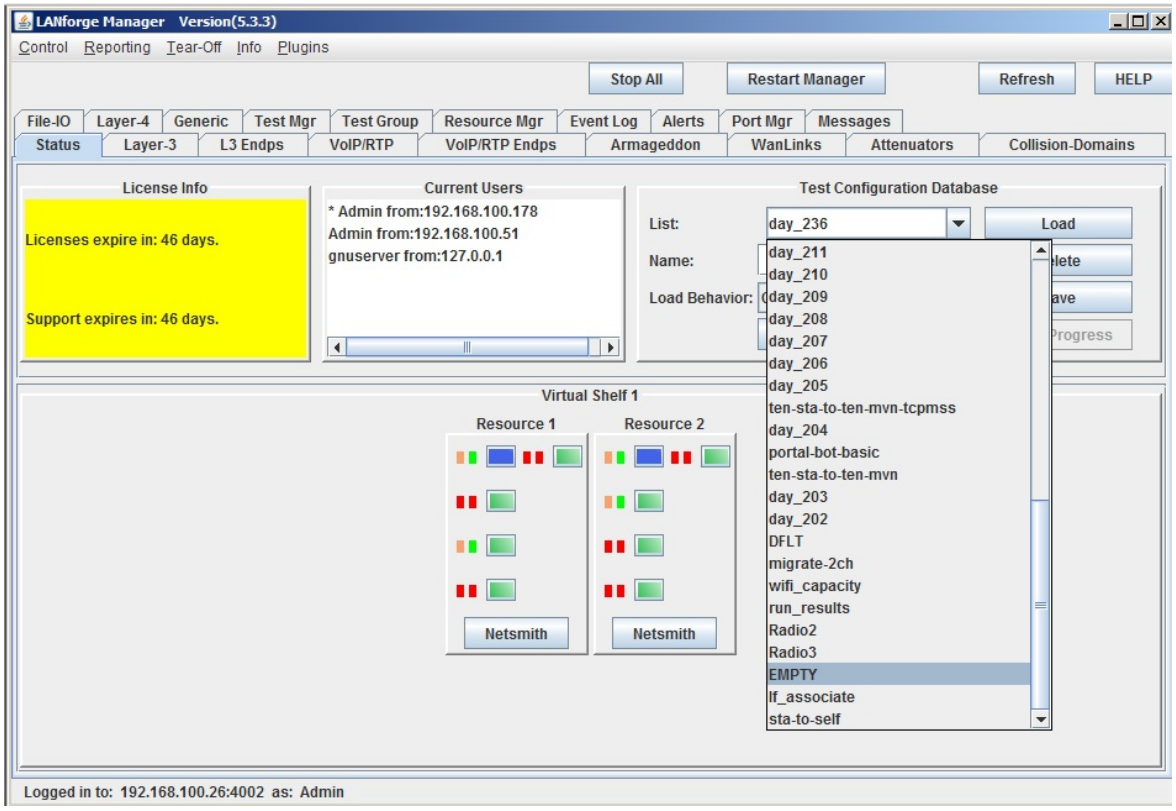



```
./lf_associate_ap.pl --resource 1 --mgr localhost \
   --action step1       --radio wiphy0    --ssid jedtest \
   --first_sta sta100   --num_stations 10 --duration 120 \
   --first_ip DHCP      --upstream eth1 --security open \
   --db_preload day_236
```

8. Saving your test state after completing a traffic run

```
./lf_associate_ap.pl --resource 1 --mgr localhost \
   --action step1       --radio wiphy0    --ssid jedtest \
   --first_sta sta100   --num_stations 10 --duration 120 \
   --first_ip DHCP      --upstream eth1 --security open \
   --db_preload day_236 --db_save station_results
```

9. Cleaning out your scenario settings after completing a traffic run. We can do this by loading the EMPTY database with the `db_postload` switch.



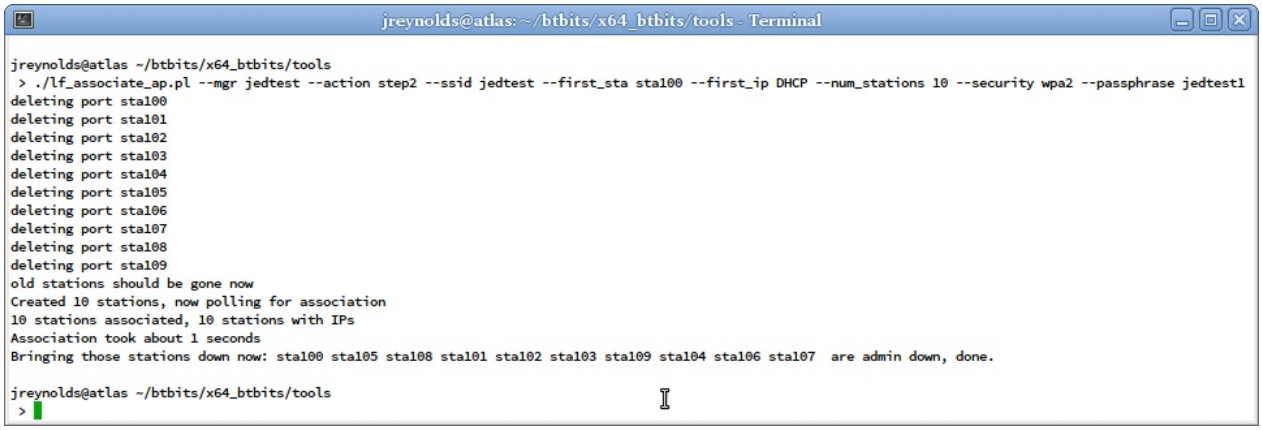

```
./lf_associate_ap.pl --resource 1 --mgr localhost \
   --action step1       --radio wiphy0    --ssid jedtest \
   --first_sta sta100   --num_stations 10 --duration 120 \
   --first_ip DHCP      --upstream eth1 --security open \
   --db_preload day_236 --db_save station_results --db_postload EMPTY
```

Using lf_associate_ap to stress test an AP

We can have a series of stations associate and unassociate over and over. This can be quite a bit of exercise for an AP. Below is a command that tests five clients connecting.

```
./lf_associate_ap.pl --mgr jedtest --action step2 \
 --ssid jedtest --first_sta sta100 --first_ip DHCP \
 --num_stations 10 --security wpa2 --passphrase jedtest1
```

This will create set of ten stations bring them up and then take them down.



Script Options

These might have been update since publication, please check --help output for your version of the script.

```
./lf_associate_ap.pl   [--mgr {host-name | IP}]
      [--mgr_port {ip port}]     # use if on non-default management port
      [--resource {resource}]    # use if multiple lanforge systems; defaults to 1
      [--quiet { yes | no }]     # debug output; -q

      ##       AP selection
      [--radio {name}]           # e.g. wiphy2
      [--ssid {ssid}]            # e.g. jedtest
      [--security {open|wep|wpa|wpa2}] # station authentication type
      [--passphrase {...}]       # implies wpa2 if --security not set
      [--wifi_mode {a|abg|abgn|abgnAC|an|anAC|b|bg|bgn|g}]

      ##       station configuration
      [--num_stations {10}]
      [--first_sta {sta100}]
      [--first_ip {DHCP |ip address}]
      [--netmask {255.255.0.0}]

      ##       connection configuration
      [--cxtype {tcp/tcp6/udp/udp6}]   # use a tcp/udp connection, default tcp
      [--upstream {name|eth1}]
         # could be AP or could be port on LANforge
         # connected to WAN side of AP
      [--bps-min {10000000}]         # minimum tx bps
      [--bps-max {SAME|bps-value}]  # maximum tx bps, use SAME or omit for SAME
      [--duration {30}]      # connection duration, seconds, default 60
      [--poll-time {5}]    # nap time between connection displays
      [--action {step1,step2}]
         # step1: creates [num_stations] stations and L3 connections
         # step2: does bringup test

      [--traffic_type {separate|concurrent}]
         # for step1: separate does download then upload
         # concurrent does upload and download at same time

      [--db_preload {scenario name}]
         # load this database before creating stations
         # option intended as a cleanup step

      [--db_save {name}]
         # save the state of this test scenario after running the
         # connections, before --db_postload

      [--db_postload {scenario name}]
         # load this database after running connections,
         # option intended as a cleanup step
```