http://www.candelatech.com
sales@candelatech.com
+1 360 380 1618 [PST, GMT -8]

# Creating Connections with the FIREmod Script

**Goal**: Create, destroy, start and stop connections and endpoints without needing to use the LANforge GUI.

Traffic emulation can be run unattended and using automated tools without use of the LANforgeGUI using Perl scripts provided with the LANforge Server. These scripts can be run from within the LANforge server or outside the LANforge Server (on a Windows desktop). The output of the scripts needs to be redirected into a text file for you to process the results.


## Script Capabilities

The `lf_firemod.pl` script has a lot of options because endpoints have a lot of features. Basic actions:

- Creating and Deleting Endpoints and Cross Connects: *create_endp, delte_endp, create_cx, delete_cx*
- Modifying an Endpoints TX Speed: *set_endp*
- Listing and Monitoring Ports, Endpoints and Cross Connects: *list_ports, show_endp, list_cx, show_cx*
- Reporting on Ports, Endpoints and Cross Connects: *show_port, show_endp, show_cx*
- Controlling Traffic: *do_cli, start_endp, stop_endp*. To start bi-directional traffic, start both endpoints.
- Pass direct CLI commands: *do_cmd*. Use this to help configure aspects of your testing scenario that are options presented in this script. Like secondary IPs on a port.

Creating a basic cross connect requires two endpoints, and each endpoint requires a port (network interface). Script options often begin by stating the manager, resource and action:

```
C:\> perl .\lf_firemod.pl --mgr 192.168.100.1 --resource 2 --action create_endp ...more options
```

## Script Actions, arguments to `--action`

### Creating Endoints: `create_endp`

We use these parameters when creating and endpoint:

**--endp_name**
: name this endpoint

**--port_name**
: name of the port this endpoint uses

**--speed**
speed of the endpoint transmission in bps

**--tos**
type of service

**--max_speed**
Maximum port speed if different than minimum speed of port, in bps

**--endp_type**
Endpoint Types: *tcp, udp, tcp6, udp6*. To create multicast endpoint types, use *mc_udp* and *mc_udp6*.

**--min_pkt_sz/--max_pkt_sz**
Minimum and maximum packet sizes

**--use_csums**
Enable checksums

**--ttl**
packet Time To Live

**--report_timer**
the update interval for the endpoint

### Example of creating a tcp connection endpoint with debugging:

```
lf_firemod.pl   --action create_endp \
  --mgr 192.168.45.34        --mgr_port 4002 \
  --endp_name web_1          --speed 154000 \
  --endp_type tcp            --port_name eth1 \
  --quiet no
```

### Creating a multicast udp connection:

```
lf_firemod.pl   --action create_endp \
  --mgr 192.168.45.34        --mgr_port 4002 \
  --endp_name mcast_xmit_1   --speed 154000 \
  --endp_type mc_udp         --mcast_addr 224.9.9.8 --mcast_port 9998 \
  --rcv_mcast NO             --port_name eth1 \
  --min_pkt_sz 1072          --max_pkt_sz 1472 \
  --use_csums NO             --ttl 32 \
  --quiet no                 --report_timer 1000
```

### Create a connection with specific test-manager

```
lf_firemod.pl  --action create_endp \
  --mgr 192.168.45.34     --mgr_port 4002 \
  --endp_name web_1       --speed 154000 \
  --endp_type tcp         --port_name eth1 \
  --quiet no              --test_manager web_tm
```

### Show Endpoint Stats: `show_endp`

By default, using the show_endp action shows all endpoints. It might be useful to place output like this right into a file or to immediate use grep to find the rows you want.

```
 $ ./lf_firemod.pl --action show_endp --mgr cholla-f19

 RSLT: 0  Cmd: 'nc_show_endp'


FileEndp [e2#0-nfs-100] (NOT_RUNNING, WRITING, WRITE_RATE_BURSTY, CHECK_MOUNT, AUTO_MOUNT, UN_MOUNT, O_TRUNC)
     Shelf: 1, Card: 1  Port: 10  Endpoint: 1  Type: FILE_NFS  Pattern: INCREASING
     MinWriteRate: 1544000bps  MaxWriteRate: 0bps  MinRead/WriteSz: 4096B  MaxRead/WriteSz: 32768B
     MinReadRate:  1544000bps  MaxReadRate:  1544000bps  QuiesceAfterFiles: -1
     NumFiles: 2  MinFileSize: 26214400B  MaxFileSize: 26214400B
     Directory: AUTO  Prefix: AUTO  Volume:
     Server-Mount: 10.41.0.1:/tank/tmp  Mount-Dir: AUTO  Mount-Options:
     RptTimer: 1000ms  RunningFor: 0s  StopIn: 0s  Quiesce: 3
     LastRpt: 0.000 secs ago     RealWriteRate: 0bps   RealReadRate: 0bps
     RetryTimer: 1000ms  InFailedIO: 0ms
       Buffers Read:     Total: 0          Time: 0s    Cur: 0        0/s
       Bytes Read:       Total: 0          Time: 0s    Cur: 0        0/s
       Files Read:       Total: 0          Time: 0s    Cur: 0        0/s
       Bytes Written:    Total: 0          Time: 0s    Cur: 0        0/s
       Buffers Written:  Total: 0          Time: 0s    Cur: 0        0/s
       Files Written:    Total: 0          Time: 0s    Cur: 0        0/s
       Read CRC Failed:  Total: 0          Time: 0s    Cur: 0        0/s

FileEndp [e2#0-nfs-101] (NOT_RUNNING, WRITING, WRITE_RATE_BURSTY, CHECK_MOUNT, AUTO_MOUNT, UN_MOUNT, O_TRUNC)
     Shelf: 1, Card: 1  Port: 12  Endpoint: 2  Type: FILE_NFS  Pattern: INCREASING
     MinWriteRate: 1544000bps  MaxWriteRate: 0bps  MinRead/WriteSz: 4096B  MaxRead/WriteSz: 32768B
     MinReadRate:  1544000bps  MaxReadRate:  1544000bps  QuiesceAfterFiles: -1
     NumFiles: 2  MinFileSize: 26214400B  MaxFileSize: 26214400B
     Directory: AUTO  Prefix: AUTO  Volume:
     Server-Mount: 10.41.0.1:/tank/tmp  Mount-Dir: AUTO  Mount-Options:
     RptTimer: 1000ms  RunningFor: 0s  StopIn: 0s  Quiesce: 3
     LastRpt: 0.000 secs ago     RealWriteRate: 0bps   RealReadRate: 0bps
```

You can redirect all output into a file:

```
 $ ./lf_firemod.pl --action show_endp --mgr cholla-f19 > /var/tmp/endp-stats.txt
```

It is possible to print out one-word attributes, such as MaxWriteRate tx_bps or rx_bps:

```
./lf_firemod.pl --mgr 127.0.0.1 --quiet 1 --action show_endp --endp_name cx_0-B --endp_vals tx_
Rx Bytes: 99938104
Tx Bytes: 99993112
```

### Configure Endpoint: `set_endp`

This is for changing the attributes of an endpoint, such as endpoint TX rate.

```
 $ ./lf_firemod.pl --mgr cholla-f19 --action set_endp --endp_name cx_0-A --speed 2000000
```

### Show Port Stats: `show_port`

This is pretty useful for getting transmit rate on ports during a connection while not having to use the lf_portmod script. If you do not specify `--port_name`, all ports will be listed.

```
 $ ./lf_firemod.pl --action show_port --mgr cholla-f19 --port_name eth2#0

Shelf: 1, Card: 1, Port: 10  Type: MacVLAN  Alias:
  Win32-Name:   Win32-Desc:   Parent/Peer: eth2  Rpt-Timer: 8000  CPU-Mask: 0
   Current:     UP LINK-UP TSO UFO GSO GRO PROBE_ERROR
   Supported:   UP SEND_TO_SELF
   Partner:     UP
   Advertising: 10bt-HD 10bt-FD 100bt-HD 100bt-FD 1000-FD TSO-ENABLED UFO-ENABLED GSO-ENABLED GI
   IP: 10.41.0.10  MASK: 255.255.255.0  GW: 0.0.0.0  VID: 0  ResetState: COMPLETE
   DNS Servers:
   IPv6-Global: DELETED
   IPv6-Link: fe80::a00:27ff:fe09:183d/64
   IPv6-Gateway: DELETED
   MAC: 08:00:27:09:18:3d  DEV: eth2#0  MTU: 1500  TX Queue Len: 0
   LastDHCP: 0ms  Driver: macvlan Tx-Rate: 1000000Kbps
   Bus-Speed: 0/0  Bus-Width: 0/0
   Bridge-Port-Cost: Ignore  Prio: Ignore  Aging: 0
   DHCP-Client-ID: NONE  DHCP-Vendor-ID: NONE
     pps_tx: 0  pps_rx: 0  bps_tx: 0  bps_rx: 0
     Rxp: 5652  Txp: 21  Rxb: 1932984  Txb: 1826  RxERR: 0  TxERR: 0
     RxDrop: 0  TxDrop: 0  Multi: 5652  Coll: 0  RxLenERR: 0  RxOverFlow: 0
     RxCRC: 0  RxFrame: 0  RxFifo: 0  RxMissed: 0  TxAbort: 0  TxCarrier: 0
     TxFifo: 0  TxHeartBeat: 0  TxWindow: 0  RxBytesLL: 2068632  TxBytesLL: 2330
```

### List Ports, action: `list_ports`

This is the same as `--show_port` without the port_name option.

### Direct LANforge Command: `do_cmd`

In case you wanted to pass a CLI command directly in. Below is an example of setting the TOS flag for an endpoint:

```
C:\> perl .\lf_firemod.pl --mgr 192.168.100.1 --action do_cmd \
   --cmd "set_endp_tos cx_01-A LOWDELAY 10"
```

See the LANforge CLI User Guide for more info.

### Remove endpoint: `delete_endp`

Remember to remove the cross connect before removing the endpoint.

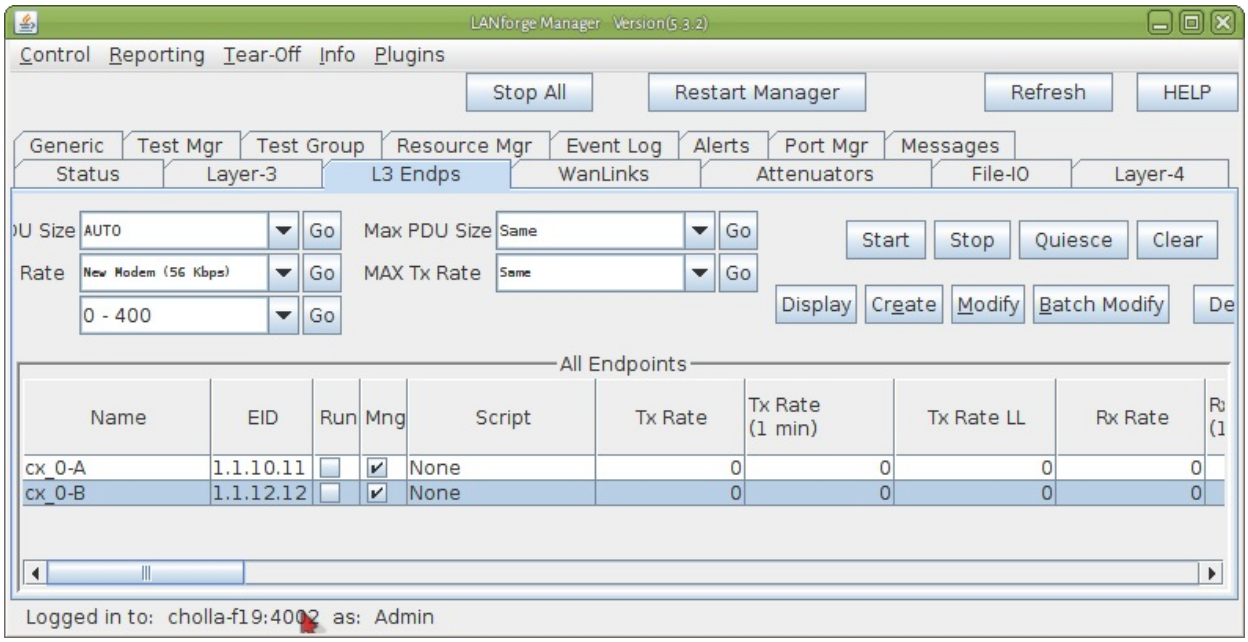
```
 $ ./lf_firemod.pl --action delete_endp  --mgr cholla-f19 --endp_name cx-0-A
```

### Create Cross-connect: `create_cx`

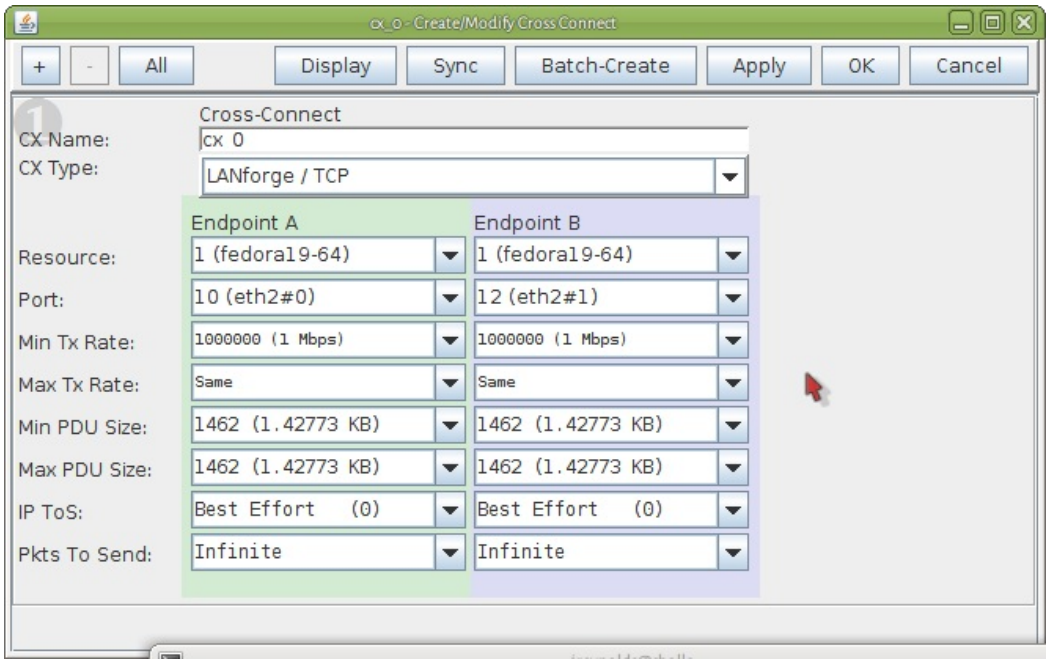
First you want to create two endpoints. You will add those endpoints to your cross connect. This example below shows all three steps:

```
$ ./lf_firemod.pl --action create_endp --mgr cholla-f19 --port_name eth2#0 \
     --endp_name cx_0-A --speed 1000000 --endp_type tcp --min_pkt_sz 1462 --report_timer 1000

$ ./lf_firemod.pl --action create_endp --mgr cholla-f19 --port_name eth2#1 \
     --endp_name cx_0-B --speed 1000000 --endp_type tcp --min_pkt_sz 1462 --report_timer 1000

$ ./lf_firemod.pl --action create_cx --mgr cholla-f19 --cx_name  cx_0 \
     --cx_endps cx_0-A,cx_0-B --report_timer 1000
```

Below we see the endpoints created:




and the CX details screen:




### Show Cross Connects: `list_cx`

This shows the cross connects and their enpoints:

```
 $ ./lf_firemod.pl --action list_cx --mgr cholla-f19
CX cx_0, endpoint cx_0-A, endpoint cx_0-B
```

### Show Cross Connect Stats: `show_cx`

The output of this command begins with the basic stats for the CX and includes the statistics of each endpoint.

```
$ ./lf_firemod.pl --action delete_endp  --mgr cholla-f19 --endp_name cx_0-A
```

```
 LANFORGE_TCP CX:  cx_0  id: 12  type: LANFORGE_TCP  DesiredState: UN_INITIALIZED RealState: STOPPED
   tx_endpoint: cx_0-A  rx_endpoint: cx_0-B report_timer: 1.000000s  TestMgr: default_tm

Endpoint [cx_0-A] (NOT_RUNNING, FIXED_PLD_SIZE, RATE_BURSTY, IP_PORT_AUTO)
     Shelf: 1, Card: 1  Port: 10  Endpoint: 11 Type: LANFORGE_TCP  Pattern: INCREASING
     MinTxRate: 1000000bps  MaxTxRate: 1000000bps  MinPktSize: 1462B  MaxPktSize: 1462B
     DestMAC: 08:00:27:69:1a:3d  DestIpAddr: 10.41.0.11  DestIpPort: 0  Quiesce: 3
     SrcMAC:  08:00:27:09:18:3d  SrcIp:  0.0.0.0  IpPort:  0-0  IpTOS: DONT-SET  Priority: 0
     Role: CONNECT  RptTimer: 1000ms  RunningFor: 0s  StopIn: 0s  Avg-Jitter: 0ms
     Latency: 0 -:0:- 0  [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ] (1)
     Pkt-Gaps: 0 -:0:- 0  [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ] (1)
     Last-Rpt: 0.000 secs ago  RealTxRate: 0bps  RealRxRate: 0bps  TTL: 0  Conn-Timeout: 10000
     FileName:   SendBadCrc: 0  RcvBuf: 0  SndBuf: 0  CWND: 0 SND-MSS: 0
     RxDrop%-SEQ:  0.0000  RxDrop%-CX: 0.0000  Conn-Timer: -1-0ms  Conn-Pause: 0-0ms
     PktsToSend: 0
  Multi-Conn: 0  Active-Connections: 0  Files-Played: 0
     RunningInGroup: NONE  Script-steps-completed: 0  Steps-Failed: 0
     First-Rx: -1ms  Mcast-Source: 0.0.0.0:0
       Rx Pkts:            Total: 0           Time: 60s   Cur: 0         0/s
       Rx Pkts (On Wire):  Total: 0           Time: 60s   Cur: 0         0/s
       Rx Bytes:           Total: 0           Time: 60s   Cur: 0         0/s
       Rx Bytes (On Wire): Total: 0           Time: 60s   Cur: 0         0/s
       Rx OOO Pkts:        Total: 0           Time: 60s   Cur: 0         0/s
       RX Wrong Dev:       Total: 0           Time: 60s   Cur: 0         0/s
       RX CRC Failed:      Total: 0           Time: 60s   Cur: 0         0/s
       RX Bit Errors:      Total: 0           Time: 3s    Cur: 0         0/s
       Rx Dropped Pkts:    Total: 0           Time: 3s    Cur: 0         0/s
          Cx Detected:     0
       Rx Duplicate Pkts:  Total: 0           Time: 60s   Cur: 0         0/s
       Tx Pkts:            Total: 0           Time: 60s   Cur: 0         0/s
       Tx Pkts (On Wire):  Total: 0           Time: 60s   Cur: 0         0/s
       Tx Bytes:           Total: 0           Time: 60s   Cur: 0         0/s
```

### Remove Cross Connect: `delete_cx`

Remember to delete a cross connect before you delete its endpoints.

```
 $ ./lf_firemod.pl --action delete_cx  --mgr cholla-f19 --cx_name cx_0
```

## Controlling Traffic

### You need to use `do_cmd` to control Unicast traffic.

By default, cross connects are created in the *default_tm* test manager. To control them, you want to specify *default_tm* in your *set_cx_state* CLI command.

```
./lf_firemod.pl --mgr 127.0.0.1 --quiet 0 --action do_cmd --cmd "set_cx_state default_tm cx_0 |
```

The format of the command is specified in the CLI User Guide: set_cx_state. Possible CX states include:

- RUNNING
- SWITCH
- QUIECE
- STOPPED
- DELETED

### For Multicast traffic, use `start_endp/stop_endp`

```
$ ./lf_firemod.pl --mgr cholla-f19 --action stop_endp --endp_name cx_0-A
```

## Multicast Endpoints

There are different options for creating multicast endpoints.

```
$ ./lf_firemod.pl --action create_endp --endp_name mcast_xmit_1 \
  --endp_type mc_udp      --speed 154000 \
  --mcast_addr 224.9.9.8  --mcast_port 9998 \
  --rcv_mcast NO          --port_name eth1 \
  --min_pkt_sz 1072       --max_pkt_sz 1472 \
  --use_csums NO          --ttl 32
```

## Add secondary IPs to a Port

This is not a default script option, so we use the `do_cmd` action:

```
C:\> perl .\lf_firemod.pl --mgr 192.168.100.1
  --action do_cmd "set_sec_ip 1 1 eth1 10.26.0.20-250/24"
```

See the LANforge CLI User Guide for more info.

## Present Options

This is the output of `lf_firemod.pl --help`:

```
./lf_firemod.pl  --action { create_endp | show_endp | set_endp | show_port | list_ports
                          | do_cmd | start_endp | stop_endp | delete_endp
                          | create_cx | list_cx | show_cx | delete_cx } ]
 [--endp_vals {key,key,key,key}]
     # show_endp output can be narrowed with key-value arguments
     # Examples:
     #  --action show_endp --endp_vals MinTxRate,DestMAC,Avg-Jitter
     # Not available: Latency,Pkt-Gaps, or rows below steps-failed.
     # Special Keys:
     #  --endp_vals tx_bps         (Tx Bytes)
     #  --endp_vals rx_bps         (Rx Bytes)
 [--mgr       {host-name | IP}]
 [--mgr_port  {ip port}]
 [--cmd       {lf-cli-command text}]
 [--endp_name {name}]
 [--port_name {name}]
 [--resource  {number}]
 [--speed     {speed in bps}]
 [--tos       { DONT-SET | LOWDELAY | THROUGHPUT | RELIABILITY | LOWCOST },{priority}]
 [--max_speed {speed in bps}]
 [--quiet     { yes | no }]
 [--endp_type { lf_udp | lf_udp6 | lf_tcp | lf_tcp6 | mc_udp | mc_udp6 }]
 [--mcast_addr {multicast address, for example: 224.4.5.6}]
 [--mcast_port {multicast port number}]
 [--min_pkt_sz {minimum payload size in bytes}]
 [--max_pkt_sz {maximum payload size in bytes}]
 [--rcv_mcast { yes (receiver) | no (transmitter) }]
 [--use_csums { yes | no  should we checksum the payload }]
```