

## Candela Technologies CLI Scripting Examples

LANforge has a large collection of shell and perl scripts that can create testing scenarios using the LANforge CLI API. These scripts do most of the common tasks, including:

- Creating VLAN ports and virtual stations
- Toggling ports and stations up and down
- Creating Layer 3 connections

We suggest you read the first three [scripting cookbooks](#) to become acquainted with scripting:

- [CLI Introduction](#)
- [Monitor and Reset Ports with the portmod script](#)
- [Cross Connects and Endpoints Tutorial](#)

The LANforge CLI scripts are maintained in the [Lanforge-Scripts github project](#). Please get updated scripts there. If you have trouble, find bugs, or need help, please email [support@candelatech.com](mailto:support@candelatech.com).

### Perl Script Requirements

NOTE: LANforge perl script require many of the perl-core package components. LANforge systems come with these already installed. The `lf_kinstall.pl` script will also install them.

If you want to install these scripts on a system without LANforge, start with these:

```
$ sudo dnf install -y perl perl-core perl-CPAN perl-JSON perl-JSON-PP perl-JSON-XS perl-Number-Format "perl(Net::Telnet)" "perl(Net::SSLeay)"
```

We recommend you install these dependencies using your package manager, **not** CPAN. It is possible to mess up your perl installation using CPAN.

### (Much) Older Perl Script Examples

Please see the README at the bottom of [github.com/greearb/lanforge-scripts](https://github.com/greearb/lanforge-scripts) for up to date scripts.

| Description   | Script                          |
|---|---------------------------------|
| This is a collection of Perl modules that automate some of the tasks involving setting and parsing objects through the LANforge CLI. They are a work in progress and may change significantly as I learn more Perl! Unzip this in the directory you run your Perl scripts in, or study up on Perl and install it in a more system-wide area.  | <a href="#">LANforge.pm.tgz</a> |
| This script sets up connections of types: lf_udp, lf_tcp, custom_ether, custom_udp, and custom_tcp across 3 ports on 2 machines. It then continuously starts and stops the connections.   | <a href="#">lf_stress1.pl</a>   |
| This creates a few fast connections between 3 ports on two machines. It then starts/stops them with a fairly lengthy run between them..   | <a href="#">lf_stress2.pl</a>   |
| This script is used to test 4 high-end machines. Two of them have GigE NICs in them, and will be configured to run back-to-back. Two other machines have a 4-port NIC and 2 single-port NICs. These ports will be configured to talk to each other..  | <a href="#">lf_stress3.pl</a>   |
| This script uses the new LANforge perl modules, and is used in verification regression testing. It has the ability to set values and then test to see if the values were actually set correctly. You may find it interesting as an example of how to use the LANforge perl modules. Requires the LANforge.pm collection above.  | <a href="#">lf_verify.pl</a>    |
| The purpose of this script is to create 10 (or more) TCP and/or UDP connections on specified ports. The connections will run for a short period of time, and then 10 more will be created on a new set of ports (the next 10). It writes it's cmds to a log file so you can get an idea of what it's doing.   | <a href="#">lf_port_walk.pl</a> |
| This script should be useful for people who are testing firewalls and other types of systems that care about what ports the data is transmitted on...   |                                 |
| The purpose of this script is to create as many TCP connections as possible. It creates 40 cross-connects (cmd-line arguments may over-ride the defaults). It then falls into an endless loop that starts all CXs, waits until both side has received traffic, and then stops the connections.  | <a href="#">lf_many_conn.pl</a> |
| The purpose of this script is to test for the maximum number of simultaneous connections possible on either a firewall or router. The script sets up one UDP connection, then one TCP connection at a time up to 200 or more connections until the UDP starts dropping packets. Once the UDP connection starts dropping packets, the script records the number of TCP connections and runs the entire test again for 3 loops then calculates the average. | <a href="#">lf_macvlan3.pl</a>  |