

## LANforge CLI User Guide

To connect to the LANforge-CLI, open a tcp socket connection to the IP address of the management interface and IP port 4001. For instance:

```
telnet localhost 4001
```

Notes about entering commands:

- Numbers may be entered as decimal or HEX. If entering in HEX, you must prepend 0x and ensure that the next number after that is not a zero (or it will be parsed as OCTAL instead of HEX. For example, if you want to enter decimal 11, you could enter: 11 or 0xB
- Strings consisting of a single word may be entered by themselves, but if you wish to have a multi-word string considered a single token by the parser, surround it with single quotes. Adding single quotes around a single word token is OK too, and may make scripting easier in some cases.
- As of release 5.3.8, an time there are two single-quotes in a row, the parser will treat it as a single single quote. For previous releases, single-quoted tokens had this behaviour, but an un-quoted token would not combine two single-quotes into one. So, for 5.3.8 and higher the tokens are parsed like this:

- o 'token space' = token space
- o 'to"ken' = to'ken
- o to"ken = to'ken

For 5.3.7 and lower the tokens are parsed like this:

- o 'token space' = token space
- o 'to"ken' = to'ken
- o to"ken = **to"ken**

- Arguments are sensitive to position. You cannot just skip arguments, but you can use `NA` for most of them and have LANforge ignore them. You may leave off any trailing arguments and they will be treated the same as if they were `NA`.

- Parameter subscripts:

|              |   |
|--------------|---|
| [R]          | Required for JSON calls.  |
| [R,a-z]      | JSON calls require value in range [a-z].                                |
| [D:value]    | JSON interpreter provides a default value if missing.                   |
| [R][D:value] | JSON interpreter provides a default value if missing, null, None or NA. |

- `adb` Execute adb command on LANforge resource.
- `adb_timeout` Execute adb command on LANforge resource w/ timeout (ms) for non-forked cmds
- `adb_gui` Launch remote desktop over adb for Android devices.
- `add_adb` Add adb device and configure its info.
- `adb_wifi_event` This is used internally by LANforge to listen for WiFi events from adb.
- `rm_adb` Remove an adb device
- `add_arm_endp` Add an Armageddon (Kernel accelerated UDP) endpoint.
- `add_cx` Add a cross-connect to a test-manager.
- `add_cd` Add a Collision Domain (grouping of WanLinks).
- `add_cd_endp` Add an Endpoint to a Collision Domain.
- `add_cd_vr` Add a Virtual Router to a Collision Domain.
- `add_chamber` Add/Modify a Chamber entry.
- `add_chamber_cx` Add/Modify a Chamber connection.
- `add_chamber_path` Set/Add a path to a Chamber.
- `add_dut` Add/Modify a Device-Under-Test entry.
- `add_dut_ssid` SSID configuration for a Device-Under-Test entry.
- `add_dut_notes` Set/Add DUT notes text.
- `add_file_endp` Add a File endpoint to the LANforge Manager.
- `add_gen_endp` Add a Generic endpoint to the LANforge Manager.
- `add_l4_endp` Add a Layer 4-7 endpoint to the LANforge Manager.
- `add_channel_group` Add a grouping of DS0 channels to be used by PPP connections.
- `add_ppp_link` Add a PPP interface connection.
- `add_profile` Add LANforge device profile.
- `add_profile_notes` Set/Add Device Profile notes text.

25. `add_traffic_profile` Add LANforge traffic profile.
26. `add_traffic_profile_notes` Set/Add Traffic Profile notes text.
27. `add_text_blob` Set/Add free-form text storage.
28. `add_t1_span` Add a T1/E1 SPAN to the LANforge Manager.
29. `add_voip_endp` Add a VOIP endpoint to the LANforge Manager.
30. `add_vr` Add or modify a Virtual Router object.
31. `add_vr_bgp` Add BGP configuration to a virtual router.
32. `add_bgp_peer` Add/Modify BGP peer configuration to a virtual router.
33. `add_vrcx` Add or modify a Virtual Router Connection Endpoint object.
34. `add_vrcx2` Modify a Virtual Router Connection Endpoint object.
35. `set_vrcx_cost` Modify a Virtual Router Connection interface cost.
36. `add_endp` Add an endpoint to the LANforge Manager.
37. `add_event` Add a new event or modify an existing one.
38. `add_bond` Add a Linux Bond Device.
39. `add_br` Add a Linux Bridge Device.
40. `add_mvlan` Add a MAC based VLAN (Requires kernel support).
41. `add_rdd` Add a Redirect-Device (Requires kernel support).
42. `add_gre` Add a GRE Tunnel device.
43. `add_sec_ip` Add or update secondary IP Address(es).
44. `add_vlan` Add an 802.1Q VLAN (Requires kernel support).
45. `add_venue` Add/modify a Venue.
46. `add_sta` Add/modify a WIFI Virtual Station (Virtual STA) interface.
47. `add_vap` Add/modify a WIFI Virtual Access Point (VAP) interface.
48. `add_monitor` Add/modify a WIFI Monitor interface.
49. `add_tm` Create and add a new test manager to the system.
50. `add_group` Create a new test group.
51. `add_tgcx` Adds CX to test group.
52. `add_wl_endp` Add a WanLink (ICE) endpoint to the LANforge Manager.
53. `add_wanpath` Add a WanPath (ICE) personality to a WanLink.
54. `admin` Various admin commands.
55. `apply_vr_cfg` Apply all of the virtual routing settings for this Resource.
56. `cancel_vr_cfg` Cancel a virtual-router configuration process for this Resource.
57. `clear_cx_counters` Clear counters for one or all cross-connects.
 

```
PORTS_TOO | 0x01    # Clear port counters this CX uses as well.
SEND_EVENT | 0x02    # Send event when clearing counters.
```
58. `clear_endp_counters` Clear counters for one or all endpoints.
59. `clear_cd_counters` Clear counters for one or all Collision Domains.
60. `clear_group` Clears all cross-connects in a test group.
61. `clear_port_counters` Clear one or all port counters or other items.
62. `clear_resource_counters` Clear counters on one or all resources.
63. `clear_wifi_profiles` Clear wifi profile from device in question.
64. `clear_wp_counters` Clear WanPath counters for one endpoint.
65. `discover` Force discovery of nodes on the management network.
66. `diag` Get diagnostic information from the LANforge server.
67. `notify_dhcp` Handle input from the DHCP client process.
68. `do_pesq` Start a PESQ calculation.
69. `file` Transfer files through LANforge API.
70. `gossip` Send a message to everyone else logged in to the server.
71. `getinrxrate` Get tx packets per second rate over the last 3 seconds.
72. `getinrxrate` Get rx packets per second rate over the last 3 seconds.
73. `getinrxbps` Get rx bits-per-second per second rate over the last 3 seconds.
74. `gettxpkts` Get the total tx packets sent.
75. `getrxpkts` Get the total rx packets sent.
76. `getpktdrops` Get the total packets dropped (based on sequence number gaps).
77. `getavglatency` Get the average latency for an endpoint.
78. `getrxporterrpkts` Get the total error packets detected on the receiving port (interface)
79. `getrxendperrpkts` Get the total error packets detected on the endpoint.
80. `getipadd` Get the IP for an endpoint.

81. `getmask` Get the IP Mask for an endpoint.
82. `getmac` Get the MAC address for an endpoint.
83. `?` Show help for command(s).
84. `init_wiser` Initialize the Wiser NCW/HNW module.
85. `ios` For IPC - SwiftUI sending information to BTSERVER
86. `licenses` Print out license information. See also: `set_license`
87. `load` Load a previously saved test database.
88. `login` Login as the client name you enter.
89. `create_client` Create a new client.
90. `log_capture` Save logs to a specified location.
91. `log_level` Query or modify the logging level.
92. `log_msg` Send an message to the LANforge log file.
93. `motd` Get the message of the day (alerts, etc)
94. `nc_show_endpoints` Non-Cached Show one or all endpoints.
95. `nc_show_pesq` Non-Cached Show PESQ results for one or all VOIP endpoints.
96. `nc_show_ports` Show one/all ports for one/all resources in one/all shelves. No caching.
97. `c_show_ports` Show one/all ports for one/all resources in one/all shelves. Always uses cache.
98. `nc_show_channel_groups` Show one/all ChannelGroups for one/all resources in one/all shelves. An empty specifier will be treated as 'all'. Will always request the absolute latest information from the remote system(s)
99. `nc_show_spans` Show one/all Spans for one/all resources in one/all shelves. An empty specifier will be treated as 'all'. Will always request the absolute latest information from the remote system(s)
100. `nc_show_vr` Show one/all Virtual Routers for one/all resources in one/all shelves. An empty specifier will be treated as 'all'. This command will always request the absolute latest information from the remote system(s)
101. `nc_show_vrcx` Show one/all Virtual Router Connections for one/all resources in one/all shelves. Only Connections on the 'free-list', those not associated with any Virtual Router will be shown with this command unless you exactly specify the VRCX Name. If the VRCX is in a virtual router, only cached results will be shown. Connections associated with routers will be shown with the 'show\_vr' command with the rest of the router information. This command will always request the absolute latest information from the remote system(s)
102. `nc_show_cd` Show one/all Collision Domains.
103. `nc_show_ppp_links` Show one/all PPP Links for one/all resources in one/all shelves. An empty specifier will be treated as 'all'.
104. `probe_port` Probe and report low-level details for a port.
105. `probe_ports` Check for the existence of new (virtual) interfaces.
106. `port_reset_completed` Internal command used by port-reset script to notify LANforge the reset has completed. This is only valid for Resource processes.
107. `exit` Log out of the LANforge control server.
108. `report` Configure server-side reporting.
109. `reset_port` Reset an Ethernet port or ports.
110. `reset_serial_span` Reset a serial span.
111. `reboot_os` Restart the OS on a remote resource.
112. `rm_attenuator` Remove attenuator configuration.
113. `rm_chamber` Remove Chamber configuration.
114. `rm_chamber_path` Remove a chamber path.
115. `rm_dut` Remove DUT configuration.
116. `rm_rfggen` Requests removal of rf-generator configuration.
117. `rm_cd` Remove a Collision Domain.
118. `rm_cd_endp` Remove an Endpoint from a Collision Domain.
119. `rm_cd_vr` Remove a Virtual Router from a Collision Domain.
120. `rm_endp` Remove one or all endpoints.
121. `rm_channel_group` Remove a channel group, or set of groups.
122. `rm_event` Remove one or more events from the event log.
123. `rm_group` Deletes a new test group.
124. `rm_profile` Remove Device Profile configuration.
125. `rm_text_blob` Remove Text Blob.
126. `rm_traffic_profile` Remove Traffic Profile configuration.
127. `rm_threshold` Remove existing threshold-alert for a particular entity.
128. `rm_tgcx` Removes CX from test group.
129. `rm_venue` Remove a venue.
130. `rm_vr` Remove one or all Virtual Routers.

131. `rm_vrcx` Remove one or all Virtual Router Connections on the free-list. Underlying objects will be deleted if they were auto-created to begin with unless you specify the last argument as 'vrcx\_only'.
132. `rm_span` Remove a Serial Span (T1, etc), or a set of spans.
133. `rm_ppp_link` Remove a PppLink.
134. `rm_client` Delete a stored client profile.
135. `rm_cx` Delete a cross-connect from the system.
136. `rm_wanpath` Remove one or all wanpaths from an endpoint.
137. `rm_db` Delete a database.
138. `rm_resource` Remove a phantom Resource and all of its config.
139. `rm_sec_ip` Remove secondary IP Address(es).
140. `rm_vlan` Remove a virtual interface.
141. `rm_test_mgr` Remove a single test manager.
142. `save` Save the current configuration to a file, to be loaded later.
143. `scan_wifi` Scan for WiFi access points.
144. `set_arm_info` Set Armageddon Endpoint configuration.
145. `set_attenuator` Set attenuation value on specified attenuator module.
146. `set_rfggen` Set RF Noise-generator (RADAR) config.
147. `blink_attenuator` Visually identify attenuator by blinking LEDs or changing LCD colors or similar.
148. `flash_attenuator` Upload new software image to specified attenuator.
149. `set_chamber` Set configuration for chambers with turntables.
150. `set_cx_report_timer` Set time between reports from Test-Manager(s) to client(s).
151. `set_endp_proxy` Set the proxy information for L3 endpoints.
152. `set_endp_report_timer` Set the report timer for and endpoint.
153. `set_cx_state` Set the state of the Cross-Connect(s).
154. `set_l4_endp` Set some extra layer-4 endpoint configuration
155. `set_license` Install license keys on the manager machine.
156. `set_password` Set the password for the current or another client.
157. `set_ppp_link_state` Set the state of the PPP Link(s).
158. `set_resource` Set the Resource configuration.
159. `set_script` Add or modify a script for a particular entity.
160. `set_test_id` Set the test ID on specified resource(s).
161. `rpt_script` Internal command, see `set_script`, syntax is same.
162. `add_threshold` Add or modify a threshold-alert for a particular entity.
163. `set_wifi_radio` Modify a WIFI Radio interface.
164. `set_wifi_extra` Configure advanced wifi settings.
165. `set_wifi_extra2` Configure more advanced wifi settings.
166. `set_wifi_txo` Configure wifi TX rate control overrides.
167. `set_wifi_corruptions` Configure corruptions for wifi devices.
168. `set_wifi_custom` Set/Add custom hostapd or wpa\_supplicant config file contents.
169. `set_ifup_script` Set the post-ifup-script for a port.
170. `set_endp_addr` Set the MAC, IP, and Port addresses for an UN\_MANAGED endpoint.
171. `set_endp_payload` Payload type and payload for an endpoint.
172. `set_endp_details` Modify low-level settings such as TCP window sizes.
173. `set_event_interest` Set event interest.
174. `set_event_priority` Set event priority.
175. `set_mc_endp` Set multicast-specific info for multicast endpoints.
176. `show_adb` Show ADB devices
177. `show_chamber` Show Chamber object
178. `show_dut` Show Devices Under Test (DUT)
179. `show_events` Show recent events.
180. `show_alerts` Show active Alerts.
181. `show_event_interest` Display Event settings.
182. `show_err` Send an error message to everyone else logged in to the server.
183. `start_endp` Start an endpoint.
184. `show_profile` Show Device Profiles
185. `show_text_blob` Show Text Blob
186. `show_traffic_profile` Show Traffic Profiles
187. `start_group` Starts all cross-connects in a test group.
188. `start_ppp_link` Start a PppLink.

189. `stop_endp` Stop an endpoint.
190. `quiesce_endp` Quiesce an endpoint.
191. `stop_group` Stops all cross-connects in a test group.
192. `quiesce_group` Quiesces all cross-connects in a test group.
193. `stop_ppp_link` Stop a PppLink.
194. `set_endp_tos` Type of Service metrics for transmitted packets from this endpoint.
195. `set_endp_quiesce` Set the quiesce timer, in seconds.
196. `set_endp_pld_bounds` Set the min/max payload size bounds for an endpoint.
197. `set_endp_tx_bounds` Set the min/max transmit rate bounds for an endpoint.
198. `set_fe_info` Set read/write size and file information for File Endpoints.
199. `set_gen_cmd` Set command to be executed for this generic endpoint.
200. `set_endp_flag` Set a flag to modify some Endpoint option.
201. `set_flag` Set a flag to modify some client option.
202. `set_gps_info` Set information that could be obtained from a GPS device.
203. `set_poll_mode` Set mode to polling or push algorithm.
204. `set_port` Configure the attributes on an Ethernet port.
205. `set_port2` Set additional port configuration for existing port.
206. `set_port_alias` Set the alias for a virtual interface specified by MAC or 802.1Q VLAN-ID.
207. `set_sec_ip` Set new list of secondary IP Address(es).
208. `set_voip_info` Set various VOIP endpoint related values.
209. `set_wanpath_filter` Set the Filter type for the WanPath
210. `set_wanpath_running` Set the Running state of the WanPath
211. `set_wanpath_corruption` Set corruption values on a WanLink.
212. `set_wanlink_info` Set various WAN-Link Endpoint data members.
213. `set_wanlink_pcap` Set the WanLink packet capture information.
214. `set_wl_corruption` Set corruption values on a WanLink.
215. `set_wl_qdisc` Set the Queuing Discipline for a WanLink.
216. `set_endp_file` Set the file name for a particular endpoint. Used for packet playback.
217. `show_attenuators` Show Attenuator information.
218. `show_rfgen` Show RF-Generators configured and/or discovered.
219. `show_resources` Show one or all resources for one or all shelves.
220. `show_clients` Show all unique clients that have registered in the past.
221. `show_cx` Show one or all cross-connects for one or all test managers.
222. `show_cxe` Show one or all cross-connects and their endpoints.
223. `show_cd` Show one/all Collision Domains.
224. `show_rt` Show Virtual Router's routing table.
225. `show_vr` Show Virtual Routers for one/all resources
226. `show_vrcx` Show Virtual Router connections for all resources
227. `show_dbs` Show all available databases that may be loaded.
228. `show_endpoints` Show one or all endpoints.
229. `show_script_results` Show results of last script run for one or all endpoints.
230. `show_pesq` Show PESQ results for one or all VOIP endpoints.
231. `show_endp_payload` Show the payloads for one or all endpoints.
232. `show_files` Show files in a particular directory.
233. `show_ports` Show one/all ports for one/all resources in one/all shelves.
234. `show_channel_groups` Show one/all ChannelGroups for one/all resources in one/all shelves. An empty specifier will be treated as 'all'.
235. `show_spans` Show one/all Spans for one/all resources in one/all shelves. An empty specifier will be treated as 'all'.
236. `show_ppp_links` Show one/all PPP Links for one/all resources in one/all shelves. An empty specifier will be treated as 'all'.
237. `show_tm` Show one or all test managers.
238. `show_group` Show one or all Test Groups.
239. `show_venue` Show one or more venues.
240. `show_wps` Show one or all WanPaths for one or all WanLink Endpoints.
241. `shutdown` Restart the LANforge manager process.
242. `shutdown_resource` Restart all LANforge processes on a remote resource.
243. `shutdown_os` Shutdown the OS on a remote resource.
244. `sniff_port` Launch Wireshark on a traffic generator port.

- 245. **tail** Stream the content of a file.
- 246. **tm\_register** Register interest in one or all test managers.
- 247. **tm\_unregister** Un-register interest in one or all test managers.
- 248. **version** Print out the version of the LANforge server.
- 249. **wiser\_reset** Reset WISER library on the specified machine.
- 250. **who** Show who is currently logged into the system.
- 251. **wifi\_event** This is used internally by LANforge to listen for WiFi events.
- 252. **wifi\_cli\_cmd** Pass command to wpa\_cli or hostapd\_cli process for the specified station or AP.
- 253. **xorpsh** Connect to a Virtual Router's xorpsh shell or send cmds to the xorpsh.

1. **adb**

adb is used to control Android devices connected to LANforge systems via USB and/or IP network.

**Related Commands**

| Argument           | Description  |
|--------------------|--|
| 1. <i>shelf</i>    | Shelf name/id. Required. [R][D:1]  |
| 2. <i>resource</i> | Resource number. [W]   |
| 3. <i>adb_id</i>   | Android device identifier, use NA if it should not be used/specified. [W]  |
| 4. <i>key</i>      | Key to be used in response messages, NA for generic keyed message. Key should not have - or spaces or other non-alphanumeric characters in it. |
| 5. <i>adb_cmd</i>  | All remaining text after adb_id will be sent to the adb command. Unescaped Value   |

Syntax: `adb shelf resource adb_id key adb_cmd`

2. **adb\_timeout**

adb is used to control Android devices connected to LANforge systems via USB and/or IP network.

**Related Commands**

| Argument           | Description  |
|--------------------|--|
| 1. <i>shelf</i>    | Shelf name/id. Required. [R][D:1]  |
| 2. <i>resource</i> | Resource number. [W]   |
| 3. <i>adb_id</i>   | Android device identifier, use NA if it should not be used/specified. [W]  |
| 4. <i>key</i>      | Key to be used in response messages, NA for generic keyed message. Key should not have - or spaces or other non-alphanumeric characters in it. |
| 5. <i>max_dur</i>  | Num of milliseconds to let this command run before killing it. 0 indicates no timeout.   |
| 6. <i>adb_cmd</i>  | All remaining text after adb_id will be sent to the adb command. Unescaped Value   |

Syntax: `adb_timeout shelf resource adb_id key max_dur adb_cmd`

3. **adb\_gui**

Utilize the MonkeyRemote or srcrcpy project to provide an interactive UI for Android devices via the adb protocol. If you do not specify the `DISPLAY`, LANforge will attempt to guess it based on your connecting IP address.

For PCs, you can use the [exceed program from Hummingbird software](#).

To enable X access on Unix/Linux, run this command:

```
xhost +
```

This can open your machine to security threats, so read up on `xhost` before you run this command on a mission critical machine not protected by a good firewall!

If using `scrcpy`, `screen-size` can be > 1.0. 1.0 means '800' screen width. If using `MonkeyRemote`, then value is percentage of default size.

**Flags** are defined as follows. You can enter the value in HEX if you prefix it with `0x`.

```
USE_SCRCPY | 0x1 # Use scrcpy instead of MonkeyRemote
```

| Argument                    | Description   |
|-----------------------------|---|
| 1. <i>shelf</i>             | Shelf name/id. Required. [R][D:1]   |
| 2. <i>resource</i>          | Resource number. [W]  |
| 3. <i>adb_id</i>            | Android device identifier.  |
| 4. <i>display</i>           | The DISPLAY option, for example: 192.168.1.5:0.0. Will guess if left blank.         |
| 5. <i>screen_size_pctnt</i> | 0.1 to 1.0, screen size percentage for the Android display.                         |
| 6. <i>flags</i>             | See flags defined above.  |
| 7. <i>max_size</i>          | Limit both the width and height of the video to value. (scrcpy only). 0 is default. |

Syntax: `adb_gui shelf resource adb_id display screen_size_pctnt flags max_size`

#### 4. **add\_adb**

Add adb device and configure its settings. ADB Device will be phantom until it is discovered by the LANforge resource.

| Argument                  | Description   |
|---------------------------|---|
| 1. <i>shelf</i>           | Shelf name/id. Required. [R][D:1]                         |
| 2. <i>resource</i>        | Resource number. [W]                                      |
| 3. <i>adb_id</i>          | Android device identifier (serial number).                |
| 4. <i>adb_product</i>     | Android device product ID                                 |
| 5. <i>adb_model</i>       | Android device model ID                                   |
| 6. <i>adb_device</i>      | Android device device ID                                  |
| 7. <i>lf_username</i>     | LANforge Interop app user-name                            |
| 8. <i>sdk_version</i>     | Android sdk version (example: 19)                         |
| 9. <i>sdk_release</i>     | Android sdk release (example: 4.4.2)                      |
| 10. <i>app_identifier</i> | Identifier that App and adb can both query (mac of wlan0) |
| 11. <i>device_type</i>    | Interop device type                                       |

Syntax: `add_adb shelf resource adb_id adb_product adb_model adb_device lf_username sdk_version sdk_release app_identifier device_type`

#### 5. **adb\_wifi\_event**

This is used internally by LANforge to listen for WiFi events from adb.

| Argument          | Description                          |
|-------------------|--------------------------------------|
| 1. <i>device</i>  | ADB device name. [R]                 |
| 2. <i>event</i>   | What happened. [R]                   |
| 3. <i>status</i>  | Status on what happened.             |
| 4. <i>msg</i>     | Entire event in human readable form. |
| 5. <i>status2</i> | Status on what happened.             |

Syntax: `adb_wifi_event device event status msg status2`

## 6. **rm\_adb**

Remove an adb device. ADB Device must be phantom to be removed.

| Argument           | Description                                |
|--------------------|--|
| 1. <i>shelf</i>    | Shelf name/id. Required. [R][D:1]          |
| 2. <i>resource</i> | Resource number. [W]                       |
| 3. <i>adb_id</i>   | Android device identifier (serial number). |

Syntax: `rm_adb shelf resource adb_id`

## 7. **add\_arm\_endp**

Add an Armageddon endpoint. Armageddon endpoints are kernel accelerated, and often run many times faster than regular LANforge endpoints, especially for smaller packets. The feature set is optimized for quickly generating lots of packets from different source and destination addresses (mac, IP, ip-port, etc).

### Related Commands

| Argument            | Description  |
|---------------------|--|
| 1. <i>alias</i>     | Name of endpoint. [R]  |
| 2. <i>shelf</i>     | Shelf name/id. Required.[D:1]                                  |
| 3. <i>resource</i>  | Resource number.   |
| 4. <i>port</i>      | Port number or name.   |
| 5. <i>type</i>      | Endpoint Type : arm_udp.                                       |
| 6. <i>pps</i>       | Packets per second to generate.                                |
| 7. <i>pkt_sz</i>    | Minimum packet size, including all Ethernet headers.           |
| 8. <i>mx_pkt_sz</i> | Maximum packet size, including all Ethernet headers.           |
| 9. <i>cpu_id</i>    | Preferred CPU ID on which this endpoint should run.            |
| 10. <i>tos</i>      | The Type of Service, can be HEX. See set_endp_tos for details. |

Syntax: `add_arm_endp alias shelf resource port type pps pkt_sz mx_pkt_sz cpu_id tos`

## 8. **add\_cx**

Add a cross-connect to a test-manager. The endpoints must have already been created.

### Related Commands

| Argument           | Description   |
|--------------------|---|
| 1. <i>alias</i>    | Name of the Cross Connect to create. [R]                    |
| 2. <i>test_mgr</i> | Name of test-manager to create the CX on. [W][D:default_tm] |
| 3. <i>tx_endp</i>  | Name of Transmitting endpoint. [R]                          |
| 4. <i>rx_endp</i>  | Name of Receiving endpoint. [W]                             |

Syntax: `add_cx alias test_mgr tx_endp rx_endp`

## 9. **add\_cd**

Add a Collision Domain (CD). A CD is a group of WanLinks and/or Virtual-Routers that are considered to be in the same collision domain. For instance, when emulating clients talking to an AP, all of the WanLinks associated with this emulated AP should be in the same Collision Domain.

All WanLinks or Virtual Routers in a CD must be on the same Resource (machine). Currently only the 'WIFI' type is supported unless you have the third-party WISER module loaded (contact your sales rep for info.)

The WIFI emulation counts bandwidth when it is transmitted or received (ie, it emulates stations ↔ AP behaviour.) The WISER emulation emulates special military waveforms. An Ethernet Hub emulation is planned for future releases.



**Flags** are defined as follows. The `state` field over-rides the `running` flag if `state` is not `NA`. You can enter the value in HEX if you prefix it with `0x`.

|                      |  |                  |                         |
|----------------------|--|------------------|-------------------------|
| <code>RUNNING</code> |  | <code>0x1</code> | # Set to running state. |
| <code>ERR</code>     |  | <code>0x2</code> | # Set to kernel mode.   |

| Argument                     | Description   |
|------------------------------|---|
| 1. <code>shelf</code>        | Shelf name/id. [R][D:1]   |
| 2. <code>resource</code>     | Resource number. [W]  |
| 3. <code>alias</code>        | Name of Collision Domain. [W]   |
| 4. <code>type</code>         | CD Type: WIFI, WISER_SURFACE, WISER_SURFACE_AIR, WISER_AIR_AIR, WISER_NCW |
| 5. <code>bps</code>          | Maximum speed at which this collision domain can run.                     |
| 6. <code>report_timer</code> | How often to report stats.  |
| 7. <code>state</code>        | RUNNING or STOPPED (default is RUNNING). Use this to start/stop.          |
| 8. <code>flags</code>        | See above. Leave blank or use 'NA' for no default values.                 |

Syntax: `add_cd shelf resource alias type bps report_timer state flags`

#### 10. **add\_cd\_endp**

Add an Endpoint to a Collision Domain. The endpoint must be a WanLink Endpoint. If the endpoint is currently in another Collision Domain, it will be migrated to the new one safely.

| Argument             | Description                   |
|----------------------|-------------------------------|
| 1. <code>cd</code>   | Name of Collision Domain. [R] |
| 2. <code>endp</code> | Endpoint name/id. [R]         |

Syntax: `add_cd_endp cd endp`

#### 11. **add\_cd\_vr**

Add a Virtual Router to a Collision Domain. If the VR is currently in another Collision Domain, it will be migrated to the new one safely.

| Argument           | Description                   |
|--------------------|-------------------------------|
| 1. <code>cd</code> | Name of Collision Domain. [R] |
| 2. <code>vr</code> | Virtual-Router name/ID. [R]   |

Syntax: `add_cd_vr cd vr`

#### 12. **add\_chamber**

Add/Modify a Chamber entry. A chamber may have up to 4 LANforge resources and up to 4 DUTs. It may also have up to 16 RF Connections defined, including Connections with attenuation configured.

##### chamber\_flags:

|                      |  |                  |  |
|----------------------|--|------------------|--|
| <code>PHANTOM</code> |  | <code>0x1</code> | # (1) Chamber is not actually here right now.          |
| <code>VIRTUAL</code> |  | <code>0x2</code> | # (2) No real chamber, open-air grouping of equipment. |
| <code>OPEN</code>    |  | <code>0x4</code> | # (3) Door is open, no real isolation right now.       |

##### Chamber Type:

|                       |  |                |
|-----------------------|--|----------------|
| <code>UNKNOWN</code>  |  | <code>0</code> |
| <code>MEDIUM</code>   |  | <code>1</code> |
| <code>LARGE</code>    |  | <code>2</code> |
| <code>2D-LARGE</code> |  | <code>3</code> |

##### Turntable Type:

|                     |  |                |  |
|---------------------|--|----------------|--|
| <code>CT850A</code> |  | <code>0</code> | # TCP-IP Connected turntable in CT850A 2D chamber. |
| <code>COMXIM</code> |  | <code>1</code> | # ComXim stand-alone USB connected turn-table.     |
| <code>CT840A</code> |  | <code>2</code> | # Modbus API turntable in CT840A 2D chamber.       |

The resource-id only needs to be set in case there is a serial connection to the chamber turntable or other management control device. For TCP-IP connections, the manager process can

connect directly. See **add\_chamber\_cx** to configure connection objects.

| Argument                  | Description  |
|---------------------------|--|
| 1. <i>name</i>            | Name of Chamber, unique identifier. [R]                              |
| 2. <i>flags</i>           | Flag field for Chamber, see above.                                   |
| 3. <i>isolation</i>       | Estimated isolation in db for this chamber.                          |
| 4. <i>chamber_type</i>    | Chamber type, see above. Use 1 for Medium if uncertain.              |
| 5. <i>dut_name1</i>       | Name of first DUT in this chamber or NA                              |
| 6. <i>dut_name2</i>       | Name of second DUT in this chamber or NA                             |
| 7. <i>dut_name3</i>       | Name of third DUT in this chamber or NA                              |
| 8. <i>dut_name4</i>       | Name of fourth DUT in this chamber or NA                             |
| 9. <i>lanforge1</i>       | EID of first LANforge Resource in this chamber or NA                 |
| 10. <i>lanforge2</i>      | EID of second LANforge Resource in this chamber or NA                |
| 11. <i>lanforge3</i>      | EID of third LANforge Resource in this chamber or NA                 |
| 12. <i>lanforge4</i>      | EID of fourth LANforge Resource in this chamber or NA                |
| 13. <i>flags_mask</i>     | Mask of what flags to pay attention to, or NA for all.               |
| 14. <i>X</i>              | X coordinate to be used when drawn in the LANforge-GUI.              |
| 15. <i>Y</i>              | Y coordinate to be used when drawn in the LANforge-GUI.              |
| 16. <i>width</i>          | Width to be used when drawn in the LANforge-GUI.                     |
| 17. <i>height</i>         | Height to be used when drawn in the LANforge-GUI.                    |
| 18. <i>resource</i>       | LANforge Resource ID for controlling turn-table via serial protocol. |
| 19. <i>turntable_type</i> | Turn-Table type: see above.  |
| 20. <i>sma_count</i>      | Number of SMA connectors on this chamber, default is 16.             |

Syntax: **add\_chamber** name flags isolation chamber\_type dut\_name1 dut\_name2 dut\_name3 dut\_name4 lanforge1 lanforge2 lanforge3 lanforge4 flags\_mask X Y width height resource turntable\_type sma\_count

### 13. **add\_chamber\_cx**

Add/Modify a Chamber connection. A chamber may have up to 32 connections defined.

**chamber\_cx\_flags:**

```
CONNECTED | 0x1      # (1) Connected to something. If flag is not set, connection is open to the air (maybe 1
TERMINATED | 0x2      # (2) Connection is terminated, signal shall not pass!
```

| Argument                 | Description  |
|--------------------------|--|
| 1. <i>name</i>           | Name of Chamber, unique identifier. [R]                                |
| 2. <i>connection_idx</i> | Connection index, currently up to 32 connections supported (0-31) [R]  |
| 3. <i>internal</i>       | Internal (1) or not (0): Internal connections are no longer supported. |
| 4. <i>flags</i>          | Flag field for Chamber Connection, see above.                          |
| 5. <i>a_id</i>           | EidAntenna in string format for A side connection.                     |
| 6. <i>b_id</i>           | EidAntenna in string format for B side connection.                     |

|                      |   |
|----------------------|---|
| 7. <i>atten_id</i>   | EID for the Attenuator module if one is inline on this connection.  |
| 8. <i>flags_mask</i> | Mask of what flags to pay attention to, or NA for all.  |
| 9. <i>min_atten</i>  | Specify minimum attenuation in 10ths of a db. Distance logic will not set atten below this.                   |
| 10. <i>zrssi2</i>    | Specify 2.4Ghz zero-attenuation RSSI in 10ths of a db. Distance logic will consider this in its calculations. |
| 11. <i>zrssi5</i>    | Specify 5Ghz zero-attenuation RSSI in 10ths of a db. Distance logic will consider this in its calculations.   |

Syntax: `add_chamber_cx name connection_idx internal flags a_id b_id atten_id flags_mask min_atten zrssi2 zrssi5`

#### 14. **add\_chamber\_path**

This text (x,y,ticks triples) will be added to the end of the specified path. The text must be entered one line at a time, primarily due to CLI parsing limitations. X and Y are coordinates, with 0,0 being top-left. Ticks are units of time that the chamber object should stay in the specified location before moving to the next waypoint. When re-playing a path, the ticks will be converted into units of time based on the specified replay speed. Setting the path content to [BLANK] will delete it. You can also use the 'rm\_chamber\_path' command to delete one or all paths.

| Argument          | Description   |
|-------------------|---|
| 1. <i>chamber</i> | Chamber Name. [R]   |
| 2. <i>path</i>    | Path Name [R]   |
| 3. <i>content</i> | [BLANK] will erase all content, any other text will be appended to existing text. Unescaped Value |

Syntax: `add_chamber_path chamber path content`

#### 15. **add\_dut**

Add/Modify a Device-Under-Test (DUT) entry. The DUT is primarily informational and used to help customize reports and automate high-level test cases.

##### dut\_flags:

```

STA_MODE      | 0x1    # (1) DUT acts as Station.
AP_MODE       | 0x2    # (2) DUT acts as AP.
INACTIVE      | 0x4    # (3) Ignore this in ChamberView, etc
WEP           | 0x8    # Use WEP encryption on all ssids, deprecated, see add_dut_ssids.
WPA           | 0x10   # Use WPA encryption on all ssids, deprecated, see add_dut_ssids.
WPA2          | 0x20   # Use WPA2 encryption on all ssids, deprecated, see add_dut_ssids.
DHCPD-LAN    | 0x40   # Provides DHCP server on LAN port
DHCPD-WAN    | 0x80   # Provides DHCP server on WAN port
WPA3         | 0x100  # Use WPA3 encryption on all ssids, deprecated, see add_dut_extras.
11r          | 0x200  # Use .11r connection logic on all ssids, deprecated, see add_dut_ssids.
EAP-TTLS     | 0x400  # Use EAP-TTLS connection logic on all ssids, deprecated, see add_dut_ssids.
EAP-PEAP     | 0x800  # Use EAP-PEAP connection logic on all ssids, deprecated, see add_dut_ssids.
NOT-DHCPD    | 0x1000 # Station/edge device that is NOT using DHCP.
              # Otherwise, automation logic assumes it is using dhcp client.

```

| Argument              | Description  |
|-----------------------|--|
| 1. <i>name</i>        | Name of DUT, cannot contain ':' [R]  |
| 2. <i>flags</i>       | Flag field for DUT, see above.   |
| 3. <i>img_file</i>    | File-Name for image to represent DUT.  |
| 4. <i>sw_version</i>  | DUT Software Version information   |
| 5. <i>hw_version</i>  | DUT Hardware Version information   |
| 6. <i>model_num</i>   | DUT Model information  |
| 7. <i>serial_num</i>  | DUT Identifier (serial-number, etc)  |
| 8. <i>serial_port</i> | Resource and Serial port name on LANforge that connects to DUT (1.2.ttyS0). Serial port does not need to be on resource holding wan_port or lan_port |
| 9. <i>wan_port</i>    | IP/Mask for WAN port   |

|                           |  |
|---------------------------|--|
| 10. <i>lan_port</i>       | IP/Mask for LAN port                                   |
| 11. <i>ssid1</i>          | WiFi SSID that can be used to connect to DUT           |
| 12. <i>passwd1</i>        | WiFi Password that can be used to connect to DUT       |
| 13. <i>ssid2</i>          | WiFi SSID that can be used to connect to DUT           |
| 14. <i>passwd2</i>        | WiFi Password that can be used to connect to DUT       |
| 15. <i>ssid3</i>          | WiFi SSID that can be used to connect to DUT           |
| 16. <i>passwd3</i>        | WiFi Password that can be used to connect to DUT       |
| 17. <i>mgt_ip</i>         | Management IP Address to access DUT                    |
| 18. <i>api_id</i>         | DUT API Identifier (none specified yet)                |
| 19. <i>flags_mask</i>     | Optional mask to specify what DUT flags are being set. |
| 20. <i>antenna_count1</i> | Antenna count for first radio.                         |
| 21. <i>antenna_count2</i> | Antenna count for second radio.                        |
| 22. <i>antenna_count3</i> | Antenna count for third radio.                         |
| 23. <i>bssid1</i>         | BSSID for first radio.                                 |
| 24. <i>bssid2</i>         | BSSID for second radio.                                |
| 25. <i>bssid3</i>         | BSSID for third radio.                                 |
| 26. <i>top_left_x</i>     | X Location for Chamber View.                           |
| 27. <i>top_left_y</i>     | X Location for Chamber View.                           |
| 28. <i>eap_id</i>         | EAP Identifier, for EAP-PEAP.                          |

Syntax: `add_dut name flags img_file sw_version hw_version model_num serial_num serial_port wan_port lan_port ssid1 passwd1 ssid2 passwd2 ssid3 passwd3 mgt_ip api_id flags_mask antenna_count1 antenna_count2 antenna_count3 bssid1 bssid2 bssid3 top_left_x top_left_y eap_id`

#### 16. **add\_dut\_ssid**

SSID configuration for a Device-Under-Test (DUT) entry.

##### ssid\_flags:

```
WEP          | 0x8   # Use WEP encryption
WPA          | 0x10  # Use WPA encryption
WPA2         | 0x20  # Use WPA2 encryption
WPA3         | 0x100 # Use WPA3 encryption
11r          | 0x200 # Use .11r connection logic
EAP-TTLS    | 0x400 # Use EAP-TTLS connection logic
EAP-PEAP    | 0x800 # Use EAP-PEAP connection logic
```

| Argument                  | Description   |
|---------------------------|---|
| 1. <i>name</i>            | Name of DUT, cannot contain ':' [R]                 |
| 2. <i>ssid_idx</i>        | Index of the SSID. Zero-based indexing: (0 - 7) [W] |
| 3. <i>ssid</i>            | WiFi SSID that can be used to connect to DUT        |
| 4. <i>passwd</i>          | WiFi Password that can be used to connect to DUT    |
| 5. <i>bssid</i>           | BSSID for cooresponding SSID.                       |
| 6. <i>ssid_flags</i>      | SSID flags, see above.                              |
| 7. <i>ssid_flags_mask</i> | SSID flags mask                                     |

Syntax: `add_dut_ssid name ssid_idx ssid passwd bssid ssid_flags ssid_flags_mask`

#### 17. **add\_dut\_notes**

This text will be added to the end of the notes field for DUTs. The text must be entered one line at a time, primarily due to CLI parsing limitations.

| Argument       | Description  |
|----------------|--|
| 1. <i>dut</i>  | DUT Name. [R]  |
| 2. <i>text</i> | [BLANK] will erase all, any other text will be appended to existing text.<br>Unescaped Value |

Syntax: `add_dut_notes dut text`

## 18. `add_file_endp`

Add a File endpoint to the LANforge Manager. This endpoint can then be used to read and/or write data from/to the file system. This is most interesting when the file system in question is some sort of network file system like NFS or iSCSI. If the endpoint already exists, then this command may be used to update the values. This defaults to 4096 read/write sizes, but you can change that with the `set_fe_info` command.

### Payload Pattern:

|              |   |
|--------------|---|
| increasing   | # bytes start at 00 and increase, wrapping if needed.   |
| decreasing   | # bytes start at FF and decrease, wrapping if needed.   |
| random       | # generate a new random payload each time sent.   |
| random_fixed | # Means generate one random payload, and send it over<br># and over again.  |
| zeros        | # Payload is all zeros (00).  |
| ones         | # Payload is all ones (FF).   |
| PRBS_4_0_3   | # Use linear feedback shift register to generate pseudo random sequence.<br># First number is bit-length of register, second two are TAPS (zero-based indexes)<br># Seed value is always 1. |
| PRBS_7_0_6   | # PRBS (see above)  |
| PRBS_11_8_10 | # PRBS (see above)  |
| PRBS_15_0_14 | # PRBS (see above)  |
| custom       | # Enter your own payload with the <code>set_endp_payload</code> cmd.  |

### file\_flags:

|                   |        |  |
|-------------------|--------|--|
| CHECK_MOUNT       | 0x1    | # (1) Attempt to verify NFS and SMB mounts match the configured values.  |
| AUTO_MOUNT        | 0x2    | # (2) Attempt to mount with the provided information if not already mounted.   |
| AUTO_UNMOUNT      | 0x4    | # (4) Attempt to un-mount when stopping test.  |
| O_DIRECT          | 0x8    | # (8) Open file with O_DIRECT flag, disables caching. Must use block-size read/w.  |
| UNLINK_BW         | 0x10   | # (16) Unlink file before writing. This works around issues with CIFS for some fi.   |
| O_LARGEFILE       | 0x20   | # (32) Open files with O_LARGEFILE. This allows greater than 2GB files on 32-bit :   |
| UNMOUNT_FORCE     | 0x40   | # (64) Use -f flag when calling amount   |
| UNMOUNT_LAZY      | 0x80   | # (128) Use -l flag when calling amount  |
| USE_FSTATFS       | 0x100  | # (256) Use fstatfs system call to verify file-system type when opening files.<br># This can take a bit of time on some file systems, but it can be used<br># to detect un-expected file-system unmounts and such. |
| O_APPEND          | 0x200  | # (512) Open files for writing with O_APPEND instead<br># of O_TRUNC. This will cause files to grow ever larger.   |
| DO_CRC            | 0x400  | # calculate 32 bit crc for each read/write   |
| SYNC_AFTER_WRITE  | 0x800  | # call sync(2) after writing each block  |
| SYNC_BEFORE_CLOSE | 0x1000 | # call sync(2) before closing the file   |

### File Endpoint type:

|              |                                  |
|--------------|----------------------------------|
| fe_generic   | # Uses unspecified file protocol |
| fe_nfs       | # Does an NFSv3 mount            |
| fe_nfs4      | # Does an NFSv4 mount            |
| fe_cifs      | # Does a CIFS (Samba) mount      |
| fe_iscsi     | # Does a iSCSI mount             |
| fe_cifs/ip6  | # Does an IPv6 CIFS mount        |
| fe_nfs/ip6   | # Does a NFSv3 IPv6 mount        |
| fe_nfs4/ip6  | # Does a NFSv4 IPv6 mount        |
| fe_smb2      | # Does a SMB v2.0 mount          |
| fe_smb2/ip6  | # Does a SMB v2.0 IPv6 mount     |
| fe_smb21     | # Does a SMB v2.1 mount          |
| fe_smb21/ip6 | # Does a SMB v2.1 IPv6 mount     |
| fe_smb30     | # Does a SMB v3.0 mount          |
| fe_smb30/ip6 | # Does a SMB v3.0 IPv6 mount     |

### Related Commands

| Argument           | Description           |
|--------------------|-----------------------|
| 1. <i>alias</i>    | Name of endpoint. [R] |
| 2. <i>shelf</i>    | Shelf name/id. [D:1]  |
| 3. <i>resource</i> | Resource number.      |

|                            |   |
|----------------------------|---|
| 4. <i>port</i>             | Port number or name.  |
| 5. <i>type</i>             | Endpoint Type (like <i>fe_nfs</i> )                                 |
| 6. <i>min_read_rate</i>    | Minimum read rate, bits-per-second.                                 |
| 7. <i>max_read_rate</i>    | Maximum read rate, bits-per-second.                                 |
| 8. <i>min_write_rate</i>   | Minimum write rate, bits-per-second.                                |
| 9. <i>max_write_rate</i>   | Maximum write rate, bits-per-second.                                |
| 10. <i>payload_pattern</i> | Payload pattern, see above.   |
| 11. <i>directory</i>       | The directory to read/write in. Absolute path suggested.            |
| 12. <i>prefix</i>          | The prefix of the file(s) to read/write.                            |
| 13. <i>server_mount</i>    | The server to mount, ex:<br>192.168.100.5/exports/test1             |
| 14. <i>mount_options</i>   | Optional mount options, passed to the mount command. 'NONE' clears. |
| 15. <i>fiio_flags</i>      | File-IO flags, see above for details.                               |
| 16. <i>mount_dir</i>       | Directory to mount/unmount (if blank, will use 'directory').        |
| 17. <i>volume</i>          | iSCSI volume to mount   |
| 18. <i>retry_timer</i>     | Number of milliseconds to retry errored IO calls before giving up.  |

Syntax: `add_file_endp alias shelf resource port type min_read_rate max_read_rate min_write_rate max_write_rate payload_pattern directory prefix server_mount mount_options fio_flags mount_dir volume retry_timer`

#### 19. **add\_gen\_endp**

Add a Generic endpoint to the LANforge Manager. This endpoint will cause an external program to be run, and the results will be sent back to the LANforge system. Due to parsing constraints, you can only use certain programs, but if LANforge does not support a program you want to use, please request the feature from Candela Technologies. Set the actual command to be executed command with `set_gen_cmd`

##### Related Commands

| Argument           | Description                                 |
|--------------------|---|
| 1. <i>alias</i>    | Name of endpoint. [R]                       |
| 2. <i>shelf</i>    | Shelf name/id. [D:1]                        |
| 3. <i>resource</i> | Resource number.                            |
| 4. <i>port</i>     | Port number or name.                        |
| 5. <i>type</i>     | Endpoint Type : gen_generic [D:gen_generic] |

Syntax: `add_gen_endp alias shelf resource port type`

#### 20. **add\_l4\_endp**

Add a Layer 4-7 (HTTP, FTP, TELNET, ...) endpoint to the LANforge Manager. This endpoint can then be used to handle URL(s). If the endpoint already exists, then this command may be used to update the values. If you do not wish to change certain fields from the current value, use `NA` for the value of these fields.

##### URL Syntax

When entering a URL, use this syntax: You need to single quote the whole value. Example download:

```
'dl http://www.candelatech.com/index.html /tmp/index.html'
```

Example upload:

```
'ul ftp://www.candelatech.com/uploads /tmp/data.txt'
```

If you want to upload data, use the **ftp** protocol. If you want to emulate HTTP form posts, please use a Generic Endpoint.

**Downloaded files** are typically `/dev/null` on Linux and `NUL` on Windows to save space. Multiple Layer 4 endpoints writing to the same file will probably create a corrupt output. Relative file names will be placed in either `/home/lanforge` on Linux or `%LOCALAPPDATA%\LANforge-GUI_{ver}` on Windows. Note that Windows can make folders under `%LOCALAPPDATA%` **read only** so you might want always write your destination files under `%TEMP%`. LANforge will not expand environmental variables in file names, so instead of writing `$TEMP` on Linux or `%TEMP%\{name}` in Windows you would need to write `/tmp/{name}` or `C:\Users\jreynolds\AppData\Local\Temp\{name}`.

URL **Protocols** can be those that **curl** supports. Here are common examples:

#### http

```
http:// or https://user:password@host/file
```

#### ftp

```
ftp://user:password@host/file
```

#### telnet

```
telnet://host:port/
```

#### fftp

```
fftp://host/file
```

### Endpoint Type

There is only one choice for Layer 4 type. This includes all URL protocols.

### URL List

If the `url-is-file` flag is set, then the URL entered below should be a local file name, and it should contain one or more URLs formatted according to our special syntax:

```
dl http://www.example.com/ /dev/null
ul ftp://www.example.com/uploads /home/lanforge/bigfile.bin
dl http://www.example.com/big.png /dev/null
```

### Authentication

The HTTP and Proxy authenticate methods and other flags are configured together. The `USE_PROXY_CACHE` is a special flag that lets the endpoint use cache values (for instance, as cached by squid). If this is NOT selected, cached values will not be allowed. Select one or more by adding the values together.

#### HTTP auth flags:

|              |     |                               |
|--------------|-----|-------------------------------|
| BASIC        | 0x1 | # Basic authentication        |
| DIGEST       | 0x2 | # Digest (MD5) authentication |
| GSSNEGOTIATE | 0x4 | # GSS authentication          |
| NTLM         | 0x8 | # NTLM authentication         |

**Proxy auth flags:** The `proxy_auth_type` field is overloaded with additional features. Notable is the `BIND_DNS` option that configures dns lookups to be made from the port interface and not via the default route.

|                         |         |         |   |
|-------------------------|---------|---------|---|
| BASIC                   | 0x1     | # 1     | Basic authentication  |
| DIGEST                  | 0x2     | # 2     | Digest (MD5) authentication                                     |
| GSSNEGOTIATE            | 0x4     | # 4     | GSS authentication  |
| NTLM                    | 0x8     | # 8     | NTLM authentication   |
| USE_PROXY_CACHE         | 0x20    | # 32    | Use proxy cache   |
| USE_GZIP_COMPRESSION    | 0x40    | # 64    | Use gzip compression  |
| USE_DEFLATE_COMPRESSION | 0x80    | # 128   | Use deflate compression   |
| INCLUDE_HEADERS         | 0x100   | # 256   | especially for IMAP   |
| BIND_DNS                | 0x200   | # 512   | Make DNS requests go out endpoints Port.                        |
| USE_IPV6                | 0x400   | # 1024  | Resolve URL is IPv6. Will use IPv4 if not selected.             |
| DISABLE_PASV            | 0x800   | # 2048  | Disable FTP PASV option (will use PORT command)                 |
| DISABLE_EPSV            | 0x1000  | # 4096  | Disable FTP EPSV option   |
| LF_L4_REAL_BROWSER_TEST | 0x2000  | # 8192  | Enable Real Browser Test  |
| MEDIA_PLAYBACKS_RANDOM  | 0x4000  | #       | Select random playback between 0 and media_playbacks            |
| MEDIA_SEEKS_RANDOM      | 0x8000  | #       | Select random media seek count between 0 and media_random_seeks |
| LF_L4_VIDEO_STREAM_TEST | 0x10000 | # 65536 | Enable Video Stream Test  |

### Speed

For configuring speeds, the minimum of the URLs per second and the `max_speed` is used.

### CX Construction

A Layer 4 connection is a one-legged cross connect. It is not necessary to create a B-endpoint. After creating your Layer 4 endpoint, create a cross connect with the name as `CX_{endpoint name}` and `rx_endp` as NA:

```
add_l4_endp '{alias}' 1 1 eth1 l4_generic NA 1000 600 'dl http://localhost/ /dev/null'
```

```
add_cx 'CX_{alias}' default_tm '{alias}' NA
```

## Related Commands

| Argument                     | Description   |
|------------------------------|---|
| 1. <i>alias</i>              | Name of endpoint. [R]   |
| 2. <i>shelf</i>              | Shelf name/id. [D:1]  |
| 3. <i>resource</i>           | Resource number.  |
| 4. <i>port</i>               | Port number or name.  |
| 5. <i>type</i>               | Endpoint Type : l4_generic  |
| 6. <i>proxy_port</i>         | HTTP Proxy port if you are using a proxy.   |
| 7. <i>timeout</i>            | How long to wait for a connection, in milliseconds  |
| 8. <i>url_rate</i>           | How often should we process the URL(s), per 10 minutes. <ul style="list-style-type: none"><li>◦ 600: 1/s</li><li>◦ 1200: 2/s</li><li>◦ 1800: 3/s</li><li>◦ 2400: 4/s</li></ul> [R][D:600] |
| 9. <i>URL</i>                | The URL, see syntax above. Can also be a local file.  |
| 10. <i>proxy_server</i>      | The name of our proxy server if using one.  |
| 11. <i>proxy_userpwd</i>     | The user-name and password for proxy authentication, format: <code>user:passwd</code> .   |
| 12. <i>ssl_cert_fname</i>    | Name of SSL Certs file.   |
| 13. <i>user_agent</i>        | User-Agent string. Leave blank for default. Also SMTP-TO: <a@b.com><c@d.com>...<q@x.com>  |
| 14. <i>proxy_auth_type</i>   | Bit-field for allowable proxy-authenticate methods.   |
| 15. <i>http_auth_type</i>    | Bit-field for allowable http-authenticate methods.  |
| 16. <i>dns_cache_timeout</i> | In seconds, how long to cache DNS lookups. 0 means no caching at all.   |
| 17. <i>max_speed</i>         | In bits-per-second, can rate limit upload or download speed of the URL contents. 0 means infinite.  |
| 18. <i>block_size</i>        | TFTP Block size, in bytes.  |
| 19. <i>smtp_from</i>         | SMTP From address.  |
| 20. <i>ip_addr</i>           | Local IP address, for binding to specific secondary IP.   |
| 21. <i>quiesce_after</i>     | Quiesce test after this many URLs have been processed.  |

Syntax: `add_l4_endp alias shelf resource port type proxy_port timeout url_rate URL proxy_server proxy_userpwd ssl_cert_fname user_agent proxy_auth_type http_auth_type dns_cache_timeout max_speed block_size smtp_from ip_addr quiesce_after`

### 21. **add\_channel\_group**

Add a grouping of DS0 channels to be used by PPP connections.

Supported formats for the channels entry include:

```
'0-23', '0,1,2,3,4,5,7' or '1-5,7,20-23'
```

**Channel types** (for Digium) are described here:

```
e&m | # Channel(s) are signalled using E&M signalling (specific
      # implementation, such as Immediate, Wink, or Feature Group D
      # are handled by the userspace library).
```



```

fxsls      | # Channel(s) are signalled using FXS Loopstart protocol.
fxsgs      | # Channel(s) are signalled using FXS Groundstart protocol.
fxsks      | # Channel(s) are signalled using FXS Koolstart protocol.
fxols      | # Channel(s) are signalled using FXO Loopstart protocol.
fxogs      | # Channel(s) are signalled using FXO Groundstart protocol.
fxoks      | # Channel(s) are signalled using FXO Koolstart protocol.
unused     | # No signalling is performed, each channel in the list remains idle
clear      | # Channel(s) are bundled into a single span. No conversion or
           | # signalling is performed, and raw data is available on the master.

indclear   | # Like 'clear' except all channels are treated individually and
           | # are not bundled. 'bchan' is an alias for this.

rawhdlc    | # The zaptel driver performs HDLC encoding and decoding on the
           | # bundle, and the resulting data is communicated via the master device.

fcshdlc    | # The zapdel driver performs HDLC encoding and decoding on the
           | # bundle and also performs incoming and outgoing FCS insertion
           | # and verification. 'dchan' is an alias for this.

nethdlc    | # The zaptel driver bundles the channels together into an
           | # hdlc network device, which in turn can be configured with
           | # sethdlc (available separately).

```

These are **not** currently supported:

```

sf         |# Channel(s) are signalled using in-band single freq tone.
           |#   Syntax as follows:
           |#   channel# -> sf:[rxfreq],[rxbw],[rxflag],[txfreq],[txlevel],[txflag]
           |#   rxfreq is rx tone freq in hz, rxbw is rx notch (and decode)
           |#   bandwidth in hz (typically 10.0), rxflag is either 'normal' or
           |#   'inverted', txfreq is tx tone freq in hz, txlevel is tx tone
           |#   level in dbm, txflag is either 'normal' or 'inverted'. Set
           |#   rxfreq or txfreq to 0.0 if that tone is not desired.

dacs       |# The zaptel driver cross connects the channels starting at
           |# the channel number listed at the end, after a colon

dacsrbs    |# The zaptel driver cross connects the channels starting at
           |# the channel number listed at the end, after a colon and
           |# also performs the DACSing of RBS bits.

```

| Argument            | Description   |
|---------------------|---|
| 1. <i>alias</i>     | Name for this Channel Group. [R]  |
| 2. <i>shelf</i>     | Shelf name/id. [R][D:1]   |
| 3. <i>resource</i>  | Resource number. [W]  |
| 4. <i>span_num</i>  | The span number. First span is 1, second is 2... [W]  |
| 5. <i>channels</i>  | List of channels to add to this group.  |
| 6. <i>type</i>      | The channel-type. Use 'clear' for PPP links.  |
| 7. <i>MTU</i>       | MTU (and MRU) for this channel group. Must be a multiple of the number of channels if configuring a T1 WanLink. |
| 8. <i>idle_flag</i> | Idle flag (byte) for this channel group, for instance: 0x7e   |

Syntax: `add_channel_group alias shelf resource span_num channels type MTU idle_flag`

## 22. **add\_ppp\_link**

Add a PPP interface connection. Currently we only support PPP over channel-groups on T1 interfaces. Some of the arguments below are passed directly to the pppd process which negotiates and otherwise creates the ppp interface. You may want to read the man page for pppd for more in-depth discussion of the features.

**channel\_groups** selects the hardware resources that the PPP link will use. For Multi-Link PPP, you can select multiple Channel-Groups, otherwise select a single one. If you are entering multiple groups, surround all groups with single quotes, like: 'cg1 cg2 cg3'

**mlppp\_descriptor** should start with 'magic:' and have some ascii-hex trailing it. For instance: magic:00:11:22:33:44 You can use 'NA' if you are not using Multi-Link PPP.

If you need to pass extra arguments to the pppd software, you can add those arguments to the 'extra\_args' value. Be sure to surround the input with single quotes so it is parsed correctly by LANforge.

| Argument | Description |
|----------|-------------|
|----------|-------------|

|                                 |   |
|---------------------------------|---|
| 1. <i>shelf</i>                 | Shelf name/id. [R]  |
| 2. <i>resource</i>              | Resource (machine) number. [W]  |
| 3. <i>unit</i>                  | Unit number for the PPP link. ie, the 7 in ppp7. [W]  |
| 4. <i>src_ip</i>                | Source IP address for this PPP connection.  |
| 5. <i>dst_ip</i>                | Destination IP address for this PPP connection.   |
| 6. <i>channel_groups</i>        | List of channel groups, see above.  |
| 7. <i>debug</i>                 | YES for debug, otherwise debugging for the ppp connection is off.                             |
| 8. <i>auth</i>                  | YES if you want to authenticate. Default is NO.   |
| 9. <i>persist</i>               | YES if you want to persist the connection. This is suggested.                                 |
| 10. <i>lcp-echo-interval</i>    | Seconds between LCP echos, suggest 1.   |
| 11. <i>lcp-echo-failure</i>     | LCP echo failures before we determine links is dead, suggest 5.                               |
| 12. <i>holdoff</i>              | Seconds between attempt to bring link back up if it dies, suggest 1.                          |
| 13. <i>mlppp_descriptor</i>     | A unique key for use with multi-link PPP connections.   |
| 14. <i>extra_args</i>           | Extra arguments to be passed directly to the pppd server.                                     |
| 15. <i>transport_type</i>       | What sort of transport this ppp link uses.  |
| 16. <i>pppoe_transport_port</i> | Port number (or name) for underlying PPPoE transport.   |
| 17. <i>tty_transport_device</i> | TTY device for PPP links associated with TTYS.  |
| 18. <i>run_time_min_ms</i>      | Minimum uptime (ms) for PPP link during an experiment, or 0 for the link to be always up.     |
| 19. <i>run_time_max_ms</i>      | Maximum uptime (ms) for PPP link during an experiment, or 0 for the link to be always up.     |
| 20. <i>down_time_min_ms</i>     | Minimum length of downtime (ms) for PPP link between runs, or 0 for the link to be always up. |
| 21. <i>down_time_max_ms</i>     | Maximum length of downtime (ms) for PPP link between runs, or 0 for the link to be always up. |

Syntax: `add_ppp_link shelf resource unit src_ip dst_ip channel_groups debug auth persist lcp-echo-interval lcp-echo-failure holdoff mlppp_descriptor extra_args transport_type pppoe_transport_port tty_transport_device run_time_min_ms run_time_max_ms down_time_min_ms down_time_max_ms`

### 23. **add\_profile**

Add LANforge device profile. This can give a high level description of how the LANforge system should act. The profile can then be selected in higher-level test cases to auto-generate lower level configuration.

#### **Wifi\_Mode**

| Input   | : Enum | Val | : Shown by nc_show_ports             |
|---------|--------|-----|--------------------------------------|
| AUTO    | 0      | #   | 802.11g                              |
| 802.11a | 1      | #   | 802.11a                              |
| b       | 2      | #   | 802.11b                              |
| g       | 3      | #   | 802.11g                              |
| abg     | 4      | #   | 802.11abg                            |
| abgn    | 5      | #   | 802.11abgn                           |
| bgn     | 6      | #   | 802.11bgn                            |
| bg      | 7      | #   | 802.11bg                             |
| abgnAC  | 8      | #   | 802.11abgn-AC                        |
| anAC    | 9      | #   | 802.11an-AC                          |
| an      | 10     | #   | 802.11an                             |
| bgnAC   | 11     | #   | 802.11bgn-AC                         |
| abgnAX  | 12     | #   | 802.11abgn-AX                        |
|         |        | #   | a/b/g/n/AC/AX (dual-band AX) support |
| bgnAX   | 13     | #   | 802.11bgn-AX                         |

```

anAX      | 14      # 802.11an-AX
aAX       | 15      # 802.11a-AX (6E disables /n and /ac)
abgn7     | 16      # 802.11abgn-EHT
          # a/b/g/n/AC/AX/EHT (dual-band AX) support
bgn7      | 17      # 802.11bgn-EHT
an7       | 18      # 802.11an-EHT
a7        | 19      # 802.11a-EHT (6E disables /n and /ac)

```

### profile\_type

```

as_is     | 0      # Make no changes to current configuration
sta       | 1      # Station device, most likely non mobile. The EIDs may specify radio(s) to use.
bridged_ap | 2      # AP device in bridged mode. The EIDs may specify radio and bridged port.
routed_ap | 3      # AP in routed mode. The EIDs may specify radio and upstream port.
upstream  | 4      # Upstream server device. The EIDs may specify which ports to use.
monitor   | 5      # Monitor device/sniffer. The EIDs may specify which radios to use.
mobile_sta | 6      # Mobile station device. Expects to connect to DUT AP(s) and upstream LANforge.
rdd       | 7      # Pair of redirect devices, typically associated with VR to act as traffic endpoint
client    | 8      # Client-side non-WiFi device (Ethernet port, for instance).
bond      | 9      # Bonded pair of Ethernet ports.
peer      | 10     # Edge device, client or server (Ethernet port, for instance).
uplink    | 11     # Uplink towards rest of network (can go in virtual router and do NAT)
vlan      | 12     # 802.1q VLAN. Specify VID with the 'freq' option.

```

### Profile Flags:

```

DHCP-SERVER | 0x1   # This should provide DHCP server.
WEP          | 0x2   # Use WEP encryption
WPA          | 0x4   # Use WPA encryption
WPA2         | 0x8   # Use WPA2 encryption
SKIP-DHCP-ROAM | 0x10  # Ask station to not re-do DHCP on roam.
WPA3         | 0x20  # Use WPA3 encryption
11r          | 0x40  # Use 802.11r roaming setup.
EAP-TTLS    | 0x80  # Use 802.1x EAP-TTLS
NAT          | 0x100 # Enable NAT if this object is in a virtual router
EAP-PEAP     | 0x200 # Enable EAP-PEAP
BSS-TRANS   | 0x400 # Enable BSS Transition logic
ALLOW-11W   | 0x800 # Set 11w (MFP/PMF) to optional.
ENABLE-POWERSAVE | 0x1000 # Enable power-save when creating stations.
RRM-IGNORE-BEACON-REQ | 0x2000 # Request station ignore RRM beacon measurement request.

```

For mac-address pattern, release 5.4.1 and higher also supports sub-byte randomizations. For instance, this will randomize just the low 4 bits of the second octet: xx:xx:xx:\*4:xx See Also: `add_traffic_profile`

| Argument                 | Description  |
|--------------------------|--|
| 1. <i>name</i>           | Profile Name. [R]  |
| 2. <i>profile_type</i>   | Profile type: See above.                                       |
| 3. <i>wifi_mode</i>      | WiFi Mode for this profile.                                    |
| 4. <i>antenna</i>        | Antenna count for this profile.                                |
| 5. <i>instance_count</i> | Number of devices (stations, vdevs, etc)                       |
| 6. <i>freq</i>           | WiFi frequency to be used, 0 means default.                    |
| 7. <i>ssid</i>           | WiFi SSID to be used, [BLANK] means any.                       |
| 8. <i>passwd</i>         | WiFi Password to be used (AP Mode), [BLANK] means no password. |
| 9. <i>profile_flags</i>  | Flags for this profile, see above.                             |
| 10. <i>flags_mask</i>    | Specify what flags to set.                                     |
| 11. <i>mac_pattern</i>   | Optional MAC-Address pattern, for instance: xx:xx:xx:*:*:xx    |
| 12. <i>bandwidth</i>     | 0 (auto), 20, 40, 80 or 160                                    |
| 13. <i>eap_id</i>        | EAP Identifier   |
| 14. <i>alias_prefix</i>  | Port alias prefix, aka hostname prefix.                        |
| 15. <i>vid</i>           | Vlan-ID (only valid for vlan profiles).                        |

Syntax: `add_profile name profile_type wifi_mode antenna instance_count freq ssid passwd profile_flags flags_mask mac_pattern bandwidth eap_id alias_prefix vid`

## 24. add\_profile\_notes

This text will be added to the end of the notes field for Profiles. The text must be entered one line

at a time, primarily due to CLI parsing limitations.

| Argument       | Description  |
|----------------|--|
| 1. <i>dut</i>  | Profile Name. [R]  |
| 2. <i>text</i> | [BLANK] will erase all, any other text will be appended to existing text.<br>Unescaped Value |

Syntax: `add_profile_notes dut text`

## 25. **add\_traffic\_profile**

Add LANforge traffic profile. This can give a high level description of how the LANforge system should generate and/or receive traffic. The profile can then be selected in higher-level test cases to auto-generate lower level configuration.

### Type

```
as_is | 0 # Make no changes to current configuration
udp | 1 #
tcp | 2 #
http | 3 # Not yet implemented
https | 4 # Not yet implemented
Iperf3-Server | 5 # iperf3 server
Iperf3-Client | 6 # iperf3 client
ARM-UDP | 7
ARM-TCP | 8
VOIP | 9
MCAST-TX | 10
MCAST-RX | 11
PING | 12
FTP | 13
```

### Traffic Profile Flags:

```
UP | 0x1 # Upload direction (this not set means download)
BI-DIRECTIONAL | 0x2 # Should we do bi-directional traffic?
IPERF_UDP | 0x4 # If Iperf, should use UDP. If not set, then will use TCP.
```

See Also: `add_profile`

| Argument                              | Description   |
|---------------------------------------|---|
| 1. <i>name</i>                        | Profile Name. [R]                                   |
| 2. <i>type</i>                        | Profile type: See above.                            |
| 3. <i>min_speed</i>                   | Main-Direction Speed in bps.                        |
| 4. <i>max_speed</i>                   | Main-Direction Speed in bps.                        |
| 5. <i>min_pdu</i>                     | Minimum PDU size                                    |
| 6. <i>max_pdu</i>                     | Minimum PDU size                                    |
| 7. <i>tos</i>                         | IP Type-of-Service                                  |
| 8. <i>instance_count</i>              | Number of connections per device                    |
| 9. <i>traffic_profile_flags</i>       | Flags for this profile, none defined at this point. |
| 10. <i>traffic_profile_flags_mask</i> | Specify what flags to set.                          |
| 11. <i>min_speed</i>                  | Opposite-Direction Speed in bps.                    |
| 12. <i>max_speed</i>                  | Opposite-Direction Speed in bps.                    |

Syntax: `add_traffic_profile name type min_speed max_speed min_pdu max_pdu tos instance_count traffic_profile_flags traffic_profile_flags_mask min_speed max_speed`

## 26. **add\_traffic\_profile\_notes**

This text will be added to the end of the notes field for Profiles. The text must be entered one line at a time, primarily due to CLI parsing limitations.

| Argument      | Description       |
|---------------|-------------------|
| 1. <i>dut</i> | Profile Name. [R] |

|                |  |
|----------------|--|
| 2. <i>text</i> | [BLANK] will erase all, any other text will be appended to existing text.<br>Unescaped Value |
|----------------|--|

Syntax: `add_traffic_profile_notes dut text`

## 27. **add\_text\_blob**

These objects are typically used by the GUI or other automated scripts and are not directly parsed or used by the LANforge server.

| Argument       | Description   |
|----------------|---|
| 1. <i>type</i> | Text type identifier stream, for instance 'cv-connectivity' [R]                           |
| 2. <i>name</i> | Text name, for instance '2-AP-test-case' [R]  |
| 3. <i>text</i> | [BLANK] will erase all, any other text will be appended to existing text. Unescaped Value |

Syntax: `add_text_blob type name text`

## 28. **add\_t1\_span**

Add a T1/E1 SPAN to the LANforge Manager. You will have to actually have T1/E1 hardware in the system before this is a useful thing to do. You will then be able to create channel-groups and PPP links. For the `first_channel`, the setting will depend on the T1/E1 port you wish to use. The first T1/E1 resource will have the `first_channel` of 1, the second at 25, the third at 49, etc.

### Build-out:

```
133_ft | 0 # 1-133 feet
266_ft | 1 # 122-266 feet
399_ft | 2 # 266-399 feet
533_ft | 3 # 399-533 feet
655_ft | 4 # 533-655 feet
-7.5db | 5 # -7.5db (CSU)
-15db | 6 # -15db (CSU)
-22.5db | 7 # -22.5db (CSU)
0db | 8 # 0db (CSU)
```

### PPP Link Types:

```
Sangoma_T1 |#
Sangoma_E1 |#
Digium_T1 |#
```

Framing NOTE: d4 is also known as 'sf' or 'superframe'.

| Argument                | Description   |
|-------------------------|---|
| 1. <i>shelf</i>         | Shelf name/id. [R][D:1]                                     |
| 2. <i>resource</i>      | Resource number. [W]  |
| 3. <i>type</i>          | Currently supported types listed above. [W]                 |
| 4. <i>span_num</i>      | The span number. First span is 1, second is 2... [W]        |
| 5. <i>first_channel</i> | The first DS0 channel for this span.                        |
| 6. <i>timing</i>        | Timing: 0 == do not use, 1 == primary, 2 == secondary..     |
| 7. <i>buildout</i>      | Buildout, Integer, see above.                               |
| 8. <i>framing</i>       | Framing: T1: esf or d4. E1: cas or ccs.                     |
| 9. <i>coding</i>        | Coding: T1: ami or b8zs. E1: ami or hdb3                    |
| 10. <i>pci_bus</i>      | PCI Bus number, needed for Sangoma resources.               |
| 11. <i>pci_slot</i>     | PCI slot number, needed for Sangoma resources.              |
| 12. <i>CPU_ID</i>       | CPU identifier (A, B, etc) for multiport Sangoma resources. |
| 13. <i>MTU</i>          | MTU for this span (used by in-band management, if at all).  |

Syntax: `add_t1_span shelf resource type span_num first_channel timing buildout framing coding pci_bus pci_slot CPU_ID MTU`

## 29. **add\_voip\_endp**

Add a VOIP (Voice over IP) to the LANforge Manager. If the endpoint already exists, then this command may be used to update the values. If the sip\_gateway is 'AUTO', then the management IP for that particular machine will be used.

| Argument                  | Description   |
|---------------------------|---|
| 1. <i>alias</i>           | Name of endpoint. [R]   |
| 2. <i>shelf</i>           | Shelf name/id. [D:1]  |
| 3. <i>resource</i>        | Resource number.  |
| 4. <i>port</i>            | Port number or name.  |
| 5. <i>phone_num</i>       | Phone number for Endpoint   |
| 6. <i>rtp_port</i>        | RTP port to use for send and receive.   |
| 7. <i>sip_gateway</i>     | SIP Gateway/Proxy Name, this is who to register with, or AUTO                                 |
| 8. <i>tx_sound_file</i>   | File name containing the sound sample we will be playing.                                     |
| 9. <i>rx_sound_file</i>   | File name to save received PCM data to. Will be in WAV format, or AUTO                        |
| 10. <i>VAD_timer</i>      | How much silence (milliseconds) before VAD is enabled.  |
| 11. <i>VAD_max_timer</i>  | How often should we force a packet, even if VAD is on.  |
| 12. <i>gateway_port</i>   | IP Port for SIP gateway (defaults to 5060).   |
| 13. <i>display_name</i>   | User-Name to be displayed. Use AUTO to display phone number. BT Identifier for Mobile.        |
| 14. <i>proxy_passwd</i>   | Password to be used when registering with proxy/gateway.                                      |
| 15. <i>peer_phone_num</i> | Use AUTO to use phone number of peer endpoint, otherwise specify a number: user[@host[:port]] |
| 16. <i>auth_user_name</i> | Use this field for authentication user name. AUTO or blank mean use phone number.             |
| 17. <i>ip_addr</i>        | Use this IP for local IP address. Useful when there are multiple IPs on a port.               |

Syntax: `add_voip_endp alias shelf resource port phone_num rtp_port sip_gateway tx_sound_file rx_sound_file VAD_timer VAD_max_timer gateway_port display_name proxy_passwd peer_phone_num auth_user_name ip_addr`

### 30. **add\_vr**

Add or modify a Virtual Router. Virtual Routers are used in conjunction with LANforge-ICE to provide advanced network emulation. **Flags** are defined as:

```
USE_XORP_OSPF      | 0x1      # Enable Xorp router daemon with OSPF (IPv4) protocol
USE_XORP_MCAST    | 0x2      # Enable Xorp Multicast routing (requires OSPF to be enabled currently)
USE_XORP_SHA      | 0x4      # Enable Telcordia's Xorp SHA option (requires OSPF to be enabled)
USE_IPV6_RADVD    | 0x8      # Enable IPv6 RADV Daemon for interfaces in this virtual router.
USE_IPV6          | 0x10     # Enable IPv6 OSPF routing for this virtual router.
ENABLE_BGP        | 0x20     # Set this to zero if you don't want BGP on this VR.
4BYTE_AS_NUMBER  | 0x40     # Sets corresponding Xorp flag.
ROUTE_REFLECTOR  | 0x80     # Act as BGP Route Reflector.
BGP_CONFED       | 0x100    # Configure BGP in a confederation.
BGP_DAMPING      | 0x200    # Enable BGP damping section in Xorp configuration file.
USE_RIP          | 0x400    # Enable RIP routing protocol in Xorp.
RIP_ACCEPT_DR    | 0x800    # Tell RIP to accept default-routes.
USE_XORP_OLSR    | 0x1000   # Enable OLSR routing protocol in Xorp.
```

| Argument           | Description                 |
|--------------------|-----------------------------|
| 1. <i>alias</i>    | Name of virtual router. [R] |
| 2. <i>shelf</i>    | Shelf name/id. [R][D:1]     |
| 3. <i>resource</i> | Resource number. [W]        |

|                  |  |
|------------------|--|
| 4. <i>notes</i>  | Notes for this Virtual Router. Put in quotes if the notes include white-space.       |
| 5. <i>X</i>      | X coordinate to be used when drawn in the LANforge-GUI.                              |
| 6. <i>Y</i>      | Y coordinate to be used when drawn in the LANforge-GUI.                              |
| 7. <i>width</i>  | Width to be used when drawn in the LANforge-GUI.                                     |
| 8. <i>height</i> | Height to be used when drawn in the LANforge-GUI.                                    |
| 9. <i>flags</i>  | Virtual router flags, see above for definitions.                                     |
| 10. <i>vr_id</i> | Leave blank, use NA or 0xFFFF unless you are certain of the value you want to enter. |

Syntax: `add_vr alias shelf resource notes X Y width height flags vr_id`

### 31. **add\_vr\_bgp**

Add BGP configuration to a virtual router. **Flags:**

```
ENABLE_BGP      | 0x20 # Set this to zero if you don't want BGP on this VR.
4BYTE_AS_NUMBER | 0x40 # Sets corresponding Xorp flag.
ROUTE_REFLECTOR | 0x80 # Act as BGP Route Reflector.
BGP_CONFED     | 0x100 # Configure BGP in a confederation.
BGP_DAMPING    | 0x200 # Enable BGP damping section in Xorp configuration file.
```

| Argument                | Description  |
|-------------------------|--|
| 1. <i>vr_id</i>         | Name of virtual router. [R]                                  |
| 2. <i>shelf</i>         | Shelf name/id. [R][D:1]                                      |
| 3. <i>resource</i>      | Resource number. [W]   |
| 4. <i>bgp_id</i>        | BGP Identifier: IPv4 Address                                 |
| 5. <i>local_as</i>      | BGP Autonomous System number, 1-65535                        |
| 6. <i>flags</i>         | Virtual router BGP flags, see above for definitions.         |
| 7. <i>cluster_id</i>    | Cluster ID, IPv4 Address. Use NA if not clustering.          |
| 8. <i>confed_id</i>     | Confederation ID 1-65535. Use NA if not in a confederation.  |
| 9. <i>half_life</i>     | Halflife in minutes for damping configuration.               |
| 10. <i>max_suppress</i> | Maximum hold down time in minutes for damping configuration. |
| 11. <i>reuse</i>        | Route flag damping reuse threshold, in minutes.              |
| 12. <i>suppress</i>     | Route flag damping cutoff threshold, in minutes.             |

Syntax: `add_vr_bgp vr_id shelf resource bgp_id local_as flags cluster_id confed_id half_life max_suppress reuse suppress`

### 32. **add\_bgp\_peer**

Add/Modify BGP peer configuration to a virtual router. **Flags:**

```
ENABLE_PEER     | 0x1 # Set this to zero if you don't want this peer enabled.
PEER_CLIENT     | 0x2 # Sets corresponding Xorp flag in BGP Peer section.
PEER_CONFED_MEMBER | 0x4 # Sets corresponding Xorp flag in BGP Peer section.
PEER_UNICAST_V4 | 0x8 # Sets corresponding Xorp flag in BGP Peer section.
```

| Argument             | Description   |
|----------------------|---|
| 1. <i>vr_id</i>      | Name of virtual router. [R]                               |
| 2. <i>shelf</i>      | Shelf name/id. [R][D:1]                                   |
| 3. <i>resource</i>   | Resource number. [W]                                      |
| 4. <i>peer_index</i> | Peer index in this virtual router (0-7).                  |
| 5. <i>flags</i>      | Virtual router BGP Peer flags, see above for definitions. |

|                            |  |
|----------------------------|--|
| 6. <i>peer_id</i>          | BGP Peer Identifier: IPv4 Address          |
| 7. <i>as</i>               | BGP Peer Autonomous System number, 0-65535 |
| 8. <i>local_dev</i>        | BGP Peer Local interface.                  |
| 9. <i>nexthop</i>          | BGP Peer Nexthop, IPv4 Address.            |
| 10. <i>holdtime</i>        | BGP Peer hold-time.                        |
| 11. <i>delay_open_time</i> | BGP Peer delay open time.                  |
| 12. <i>nexthop6</i>        | BGP Peer IPv6 Nexthop address.             |

Syntax: `add_bgp_peer vr_id shelf resource peer_index flags peer_id as local_dev nexthop holdtime delay_open_time nexthop6`

### 33. **add\_vrcx**

Add or modify a Virtual Router Connection Endpoint. Virtual Router Connection Endpoints are used to logically connect two Virtual Routers with an emulated network link. Typically, 2 pairs of redirect virtual interfaces are bridged by a WanLink (which provides the network emulation.) The 'A' port in each pair of redirect devices is associated with one virtual router and has an IP address. Both endpoints should have the IP on the same subnet. The WanLink bridges the two 'B' sides of the redirect device pair. A pair of Connection Endpoint objects are required, with reversed values in their port configuration to make a connection. **Flags** can be entered in HEX if preceded by 0x. Add flags together to get desired options. Must use `apply_vr_cfg` for changes to take effect.

```

subnet_0 | 0x1    # Specify subnet 0
subnet_1 | 0x2    # Specify subnet 1
subnet_2 | 0x4    # Specify subnet 2
subnet_3 | 0x8    # Specify subnet 3
subnet_4 | 0x10   # Specify subnet 4
subnet_5 | 0x20   # Specify subnet 5
subnet_6 | 0x40   # Specify subnet 6
subnet_7 | 0x80   # Specify subnet 7
nat_enabled | 0x100 # This connection will NAT outgoing packets
dhcpd_enabled | 0x200 # Serve IPv4 DHCP on this interface
custom_dhcpd | 0x400 # Use custom DHCP config file
use_multicast | 0x800 # Use this interface for multicast and-rp
use_vrrp | 0x1000 # Use this interface for VRRP
ipv6_enabled | 0x2000 # Serve IPv6 DHCP on this interface

```

| Argument               | Description  |
|------------------------|--|
| 1. <i>shelf</i>        | Shelf name/id. [R][D:1]  |
| 2. <i>resource</i>     | Resource number. [W]   |
| 3. <i>vr-name</i>      | Virtual Router this endpoint belongs to. Use 'FREE_LIST' to add a stand-alone endpoint. [R][D:FREE_LIST] |
| 4. <i>local_dev</i>    | Name of port A, the local network device pair.   |
| 5. <i>local_dev_b</i>  | Name of port B for the local redirect device pair.   |
| 6. <i>remote_dev</i>   | Name the remote network device.  |
| 7. <i>remote_dev_b</i> | Name of port B for the remote network device.  |
| 8. <i>wanlink</i>      | The name of the WanLink that connects the two B ports.   |
| 9. <i>X</i>            | X coordinate to be used when drawn in the LANforge-GUI.  |
| 10. <i>Y</i>           | Y coordinate to be used when drawn in the LANforge-GUI.  |
| 11. <i>width</i>       | Width to be used when drawn in the LANforge-GUI.   |
| 12. <i>height</i>      | Height to be used when drawn in the LANforge-GUI.  |
| 13. <i>flags</i>       | Flags, specify if subnets 0-7 are in use, see above for others.  |
| 14. <i>subnets</i>     | Subnets associated with this link, format: 1.1.1.1/24,1.1.2.1/16...                                      |



|                            |  |
|----------------------------|--|
| 15. <i>nexthop</i>         | The next-hop to use when routing packets out this interface.                   |
| 16. <i>dhcp_lease_time</i> | DHCP Lease time (in seconds)   |
| 17. <i>dhcp_dns</i>        | IP Address of DNS server.  |
| 18. <i>dhcp_min</i>        | Minimum IP address range to serve.   |
| 19. <i>dhcp_max</i>        | Minimum IP address range to serve.   |
| 20. <i>dhcp_domain</i>     | DHCP Domain name to serve.   |
| 21. <i>interface_cost</i>  | If using OSPF, this sets the cost for this link (1-65535).                     |
| 22. <i>ospf_area</i>       | If using OSPF, this sets the OSPF area for this interface. Default is 0.0.0.0. |
| 23. <i>rip_metric</i>      | If using RIP, this determines the RIP metric (cost). (1-15, 15 is infinite).   |
| 24. <i>vrrp_ip</i>         | VRRP IPv4 address..ignored if not flagged for VRRP.                            |
| 25. <i>vrrp_ip_prefix</i>  | Number of bits in subnet mask, ie 24 for 255.255.255.0                         |
| 26. <i>vrrp_id</i>         | VRRP id, must be unique in this virtual router (1-255)                         |
| 27. <i>vrrp_priority</i>   | VRRP Priority (1-255, higher is more priority.)                                |
| 28. <i>vrrp_interval</i>   | VRRP broadcast message interval, in seconds (1-255)                            |
| 29. <i>dhcp_dns6</i>       | IPv6 Address of DNS server.  |
| 30. <i>dhcp_min6</i>       | Minimum IPv6 address to serve.   |
| 31. <i>dhcp_max6</i>       | Minimum IPv6 address to serve.   |

Syntax: `add_vrcx shelf resource vr-name local_dev local_dev_b remote_dev remote_dev_b wanlink X Y width height flags subnets nexthop dhcp_lease_time dhcp_dns dhcp_min dhcp_max dhcp_domain interface_cost ospf_area rip_metric vrrp_ip vrrp_ip_prefix vrrp_id vrrp_priority vrrp_interval dhcp_dns6 dhcp_min6 dhcp_max6`

#### 34. **add\_vrcx2**

Modify a Virtual Router Connection Endpoint. There were getting to be too many options to fit in the `add_vrcx` command, so this second command will need to be used for certain configuration.

| Argument                | Description   |
|-------------------------|---|
| 1. <i>shelf</i>         | Shelf name/id. [R][D:1]   |
| 2. <i>resource</i>      | Resource number. [W]  |
| 3. <i>vr-name</i>       | Virtual Router this endpoint belongs to. Use 'FREE_LIST' to add a stand-alone endpoint. [W][D:FREE_LIST]                          |
| 4. <i>local_dev</i>     | Name of port A for the connection.  |
| 5. <i>subnets6</i>      | IPv6 Subnets associated with this link, format: <code>aaaa:bbbb::0/64,cccc:ddd:eeee::0/64...</code>                               |
| 6. <i>nexthop6</i>      | The IPv6 next-hop to use when routing packets out this interface.   |
| 7. <i>dhcp_ignore1</i>  | MAC address and per 65535 chance MAC should be ignored by DHCPd, format: MAC-prcnt, example: <code>00:11:22:33:44:55-65535</code> |
| 8. <i>dhcp_ignore2</i>  | MAC address and per 65535 chance MAC should be ignored by DHCPd, format: MAC-prcnt, example: <code>00:11:22:33:44:55-65535</code> |
| 9. <i>dhcp_ignore3</i>  | MAC address and per 65535 chance MAC should be ignored by DHCPd, format: MAC-prcnt, example: <code>00:11:22:33:44:55-65535</code> |
| 10. <i>dhcp_ignore4</i> | MAC address and per 65535 chance MAC should be ignored by DHCPd, format: MAC-prcnt, example: <code>00:11:22:33:44:55-65535</code> |

Syntax: `add_vrcx2 shelf resource vr-name local_dev subnets6 nexthop6 dhcp_ignore1 dhcp_ignore2 dhcp_ignore3 dhcp_ignore4`

### 35. **set\_vrcx\_cost**

Modify a Virtual Router Connection interface cost. See 'add\_vrcx' for info on how to create a connection.

| Argument                 | Description  |
|--------------------------|--|
| 1. <i>shelf</i>          | Shelf name/id. [R][D:1]  |
| 2. <i>resource</i>       | Resource number. [W]   |
| 3. <i>vr-name</i>        | Virtual Router this endpoint belongs to. Use 'FREE_LIST' to add a stand-alone endpoint. [W][D:FREE_LIST] |
| 4. <i>local_dev</i>      | Name of port A for the local redirect device pair.   |
| 5. <i>local_dev_b</i>    | Name of port B for the local redirect device pair.   |
| 6. <i>remote_dev</i>     | Name of port B for the remote redirect device pair.  |
| 7. <i>remote_dev_b</i>   | Name of port B for the remote redirect device pair.  |
| 8. <i>wanlink</i>        | The name of the WanLink that connects the two B ports.   |
| 9. <i>interface_cost</i> | If using OSPF, this sets the cost for this link (1-65535).   |

Syntax: `set_vrcx_cost shelf resource vr-name local_dev local_dev_b remote_dev remote_dev_b wanlink interface_cost`

### 36. **add\_endp**

Add an endpoint to the LANforge Manager. The endpoint may then be added to a cross-connect. If the endpoint already exists, then this command may be used to update the values. Note that you can leave everything after 'port' off the command, and default values will be used. If you are configuring a TCP connection to make many connections, then use 0 (zero) for the IP Port so that the OS can choose a new one for each connection.

**Payload\_pattern** can be:

```

increasing      | # bytes start at 00 and increase, wrapping if needed
decreasing      | # bytes start at FF and decrease, wrapping if needed
random          | # generate a new random payload each time sent
random_fixed    | # means generate one random payload, and send it over and over again.
zeros           | # payload is all zeros (00)
ones            | # payload is all ones (FF)

PRBS_4_0_3      | # Use linear feedback shift register to generate pseudo random sequence.
                | # First number is bit-length of register, second two are
                | # TAPS (zero-based indexes). Seed value is always 1.

PRBS_7_0_6      | # PRBS (see above)
PRBS_11_8_10     | # PRBS (see above)
PRBS_15_0_14     | # PRBS (see above)
custom          | # Enter your own payload with the set_endp_payload cmd.

```

Endpoint **Types** can be of these types:

```

lf              | # LF protocol
lf_udp          | # UDP IPv4 connection
lf_udp6         | # UDP IPv6 connection
lf_tcp          | # TCP IPv4 connection
lf_tcp6         | # TCP IPv6 connection
custom_ether     | # LF frames with custom options, use with playback
custom_udp      | # LF UDP IPv4 frame with custom options
custom_tcp      | # LF TCP IPv4 frame with custom options
mc_udp          | # LF Multicast IPv4
custom_mc_udp   | # LF Multicast UDP IPv4
lf_sctp         | # SCTP IPv4 protocol
lf_sctp6        | # SCTP IPv6 protocol

```

#### Related Commands

| Argument           | Description           |
|--------------------|-----------------------|
| 1. <i>alias</i>    | Name of endpoint. [R] |
| 2. <i>shelf</i>    | Shelf name/id. [D:1]  |
| 3. <i>resource</i> | Resource number.      |

|                                     |   |
|-------------------------------------|---|
| 4. <i>port</i>                      | Port/Interface name or number.  |
| 5. <i>type</i>                      | Endpoint Type: See above.   |
| 6. <i>ip_port</i>                   | IP Port: IP port for layer three endpoints. Use -1 to let the LANforge server automatically configure the <i>ip_port</i> . Layer 2 endpoints will ignore this argument. Use 0 for 'ANY', and let the OS choose. |
| 7. <i>is_rate_bursty</i>            | Yes means bursty, anything else means NO.   |
| 8. <i>min_rate</i>                  | Minimum transmit rate (bps), or only rate if not bursty.  |
| 9. <i>max_rate</i>                  | Maximum transmit rate (bps), used if in bursty mode.  |
| 10. <i>is_pkt_sz_random</i>         | Yes means use random sized packets, anything else means NO.   |
| 11. <i>min_pkt</i>                  | Minimum packet size, including all headers. -1 means AUTO (5.3.2+) [D:-1]   |
| 12. <i>max_pkt</i>                  | Maximum packet size, including all headers. 0 means 'same', -1 means AUTO (5.3.2+) [D:0]  |
| 13. <i>payload_pattern</i>          | Payload pattern, see above.   |
| 14. <i>use_checksum</i>             | Yes means checksum the payload, anything else means NO.   |
| 15. <i>tll</i>                      | Time-to-live, used by UDP Multicast Endpoints only.   |
| 16. <i>send_bad_crc_per_million</i> | If NIC supports it, will randomly send X per million packets with bad ethernet Frame Check Sum.   |
| 17. <i>multi_conn</i>               | If > 0, will create separate process with this many connections per endpoint. See AUTO_HELPER flag  |

Syntax: `add_endp alias shelf resource port type ip_port is_rate_bursty min_rate max_rate is_pkt_sz_random min_pkt max_pkt payload_pattern use_checksum tll send_bad_crc_per_million multi_conn`

### 37. **add\_event**

#### Related Commands

| Argument           | Description  |
|--------------------|--|
| 1. <i>event_id</i> | Numeric ID for the event to modify, or 'new' if creating a new one. [W][D:new] |
| 2. <i>details</i>  | Event text description. Cannot include double-quote characters.                |
| 3. <i>priority</i> | See <code>set_event_priority</code> for available priorities.                  |
| 4. <i>name</i>     | Event entity name.   |

Syntax: `add_event event_id details priority name`

### 38. **add\_bond**

Add a Linux Bond Device. Specify one or more network devices to be added to the bonded interface.

| Argument           | Description                  |
|--------------------|------------------------------|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]       |
| 2. <i>resource</i> | Resource number. [W]         |
| 3. <i>port</i>     | Name of the bond device. [W] |

|                        |   |
|------------------------|---|
| 4. <i>network_devs</i> | Comma-separated list of network devices: eth1,eth2,eth3...<br>[W] |
|------------------------|---|

Syntax: `add_bond shelf resource port network_devs`

#### 39. **add\_br**

Add a Linux Bridge Device. Specify one or more network devices to be added to the bridge. This requires that the 'bridge-utils' package be installed on your Linux system. Most of the bridge settings are only used if spanning-tree is enabled. For more information on the spanning-tree values, see: `br_*` configuration is ignored. **br\_flags** can be:

```
none          | 0x0    # no features
stp_enabled | 0x1    # Enable Spanning Tree Protocol (STP)
```

| Argument                       | Description  |
|--------------------------------|--|
| 1. <i>shelf</i>                | Shelf number. [R][D:1]   |
| 2. <i>resource</i>             | Resource number. [W]   |
| 3. <i>port</i>                 | Name of the bridge device. [W]                                   |
| 4. <i>network_devs</i>         | Comma-separated list of network devices:<br>eth1,eth2,eth3...    |
| 5. <i>br_flags</i>             | Bridge flags, see above.   |
| 6. <i>br_priority</i>          | Bridge priority, 16-bit number.                                  |
| 7. <i>br_aging_time</i>        | MAC aging time, in seconds, 32-bit number.                       |
| 8. <i>br_max_age</i>           | How long until STP considers a non-responsive bridge dead.       |
| 9. <i>br_hello_time</i>        | How often does the bridge send out STP hello packets.            |
| 10. <i>br_forwarding_delay</i> | How long to wait until the bridge will start forwarding packets. |

Syntax: `add_br shelf resource port network_devs br_flags br_priority br_aging_time br_max_age br_hello_time br_forwarding_delay`

#### 40. **add\_mvlan**

Add a MAC based VLAN. This command requires that the designated machine support the macvlan kernel module. A MAC-VLAN interface is a light-weight virtual interface that is made unique by its MAC address. Do not add two MAC vlans with the same MAC to the same interface. In most cases, you do not want to duplicate a MAC at all! After creating the MAC-VLAN interface, you will need to configure its IP and other information. If you wish to create a MAC VLAN with a specific name, specify the index as well. If not specified, one will be automatically selected for you. For mac-address pattern, release 5.4.1 and higher also supports sub-byte randomizations. For instance, this will randomize just the low 4 bits of the second octet: `xx:xx:xx:xx:*:4:xx`

| Argument               | Description  |
|------------------------|--|
| 1. <i>shelf</i>        | Shelf number. [R][D:1]   |
| 2. <i>resource</i>     | Resource number. [W]   |
| 3. <i>port</i>         | Port number of an existing Ethernet interface. [W]   |
| 4. <i>MAC</i>          | The MAC address, can also use parent-pattern in 5.3.8 and higher: <code>xx:xx:xx:*:*:xx</code> |
| 5. <i>index</i>        | Optional: The index of the VLAN, (the 4 in <code>eth0#4</code> )                               |
| 6. <i>old_name</i>     | The temporary name, used for configuring un-discovered hardware.                               |
| 7. <i>report_timer</i> | Report timer for this port, leave blank or use NA for defaults.                                |
| 8. <i>flags</i>        | 0x1: Create admin-down.  |

Syntax: `add_mvlan shelf resource port MAC index old_name report_timer flags`

#### 41. **add\_rdd**

Add a Redirect-Device. This command requires that the designated machine support the `redirdev` kernel module. Redirect-Devices act like a pair of physical Ethernet interfaces connected externally by a loop-back cable, and are useful for creating virtual networks. Currently, the main reason to do this is to run LANforge ICE on a single interface in conjunction with routing. The basic idea is to create a pair of redirect devices. Give one an IP address that you want the local machine to have. The other redirect interface in the pair will not have an IP address and will be bridged by LANforge ICE (WanLink) to the real Ethernet interface, which also will not have an IP address. **It is possible to add 802.1Q and MAC-VLANs on top of redirect devices as well.**

To create an redirect-device pair, run this command twice, for example:

```
add_rdd 1 1 rdd0 rdd1
add_rdd 1 1 rdd1 rdd0
```

| Argument               | Description   |
|------------------------|---|
| 1. <i>shelf</i>        | Shelf number. [R][D:1]  |
| 2. <i>resource</i>     | Resource number. [W]  |
| 3. <i>port</i>         | Name of the Redirect Device to create. [W]                      |
| 4. <i>peer_ifname</i>  | The peer (other) RedirectDevice in this pair.                   |
| 5. <i>report_timer</i> | Report timer for this port, leave blank or use NA for defaults. |

Syntax: `add_rdd shelf resource port peer_ifname report_timer`

#### 42. **add\_gre**

Add a GRE Tunnel. These are point-to-point devices often used to connect to Cisco and similar routed networks.

| Argument                  | Description   |
|---------------------------|---|
| 1. <i>shelf</i>           | Shelf number. [R][D:1]  |
| 2. <i>resource</i>        | Resource number. [W]  |
| 3. <i>port</i>            | Name of the GRE to create, suggested to start with 'gre' [W]    |
| 4. <i>local_lower_ip</i>  | The local lower-level IP to use.                                |
| 5. <i>remote_lower_ip</i> | The remote lower-level IP to use.                               |
| 6. <i>report_timer</i>    | Report timer for this port, leave blank or use NA for defaults. |

Syntax: `add_gre shelf resource port local_lower_ip remote_lower_ip report_timer`

#### 43. **add\_sec\_ip**

Add or update secondary IP Address(es). Secondary IPs can be used to send and receive traffic, and are generally lighter weight than mac-vlans. They do share a network device (including routing table, MAC address, and network stats) with the base device, so they are not quite as flexible as mac-vlans and other virtual interfaces.

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]  |
| 2. <i>resource</i> | Resource number. [W]  |
| 3. <i>port</i>     | Name of network device (Port) to which these IPs will be added. [W] |
| 4. <i>ip_list</i>  | IP1/prefix,IP2/prefix,...IPZ/prefix. [W]                            |

Syntax: `add_sec_ip shelf resource port ip_list`

#### 44. **add\_vlan**

Add an 802.1Q VLAN. This command requires that the designated machine support the `8021q` kernel module. After creating the 802.1Q VLAN interface, you will need to configure its IP and other information.

| Argument               | Description  |
|------------------------|--|
| 1. <i>shelf</i>        | Shelf number. [R][D:1]   |
| 2. <i>resource</i>     | Resource number. [W]   |
| 3. <i>port</i>         | Port number of an existing Ethernet interface. [W]               |
| 4. <i>vid</i>          | The VLAN-ID for this 802.1Q VLAN interface. [W]                  |
| 5. <i>old_name</i>     | The temporary name, used for configuring un-discovered hardware. |
| 6. <i>report_timer</i> | Report timer for this port, leave blank or use NA for defaults.  |

Syntax: `add_vlan shelf resource port vid old_name report_timer`

#### 45. **add\_venue**

Add or modify a Venue. Venues are used to group WiFi stations and vAP, but unless you are using certain third party integrated tools, this will not have any affect on LANforge. If you are not sure what this is for, then it is not for you!

**freq\_24:** 16-bit number to specify 2.4Ghz channels to use. OR the values together to choose a list of available channels,

|      |  |        |   |           |
|------|--|--------|---|-----------|
| ALL  |  | 0xFFFF | # | ALL       |
| Ch 1 |  | 0x1    | # | Channel 1 |
| Ch 2 |  | 0x2    | # | Channel 2 |
| Ch 3 |  | 0x4    | # | Channel 3 |

**freq\_5:** See this page for cooresponding frequencies:  
[http://en.wikipedia.org/wiki/List\\_of\\_WLAN\\_channels](http://en.wikipedia.org/wiki/List_of_WLAN_channels)

|        |  |            |   |             |      |
|--------|--|------------|---|-------------|------|
| Ch 36  |  | 0x00000001 | # | Channel 36  | 5180 |
| Ch 38  |  | 0x00000002 | # | Channel 38  | 5190 |
| Ch 40  |  | 0x00000004 | # | Channel 40  | 5200 |
| Ch 42  |  | 0x00000008 | # | Channel 42  | 5210 |
| Ch 44  |  | 0x00000010 | # | Channel 44  | 5220 |
| Ch 46  |  | 0x00000020 | # | Channel 46  | 5230 |
| Ch 48  |  | 0x00000040 | # | Channel 48  | 5240 |
| Ch 52  |  | 0x00000080 | # | Channel 52  | 5260 |
| Ch 56  |  | 0x00000100 | # | Channel 56  | 5280 |
| Ch 60  |  | 0x00000200 | # | Channel 60  | 5300 |
| Ch 64  |  | 0x00000400 | # | Channel 64  | 5320 |
| Ch 100 |  | 0x00000800 | # | Channel 100 | 5500 |
| Ch 104 |  | 0x00001000 | # | Channel 104 | 5520 |
| Ch 108 |  | 0x00002000 | # | Channel 108 | 5540 |
| Ch 112 |  | 0x00004000 | # | Channel 112 | 5560 |
| Ch 116 |  | 0x00008000 | # | Channel 116 | 5580 |
| Ch 120 |  | 0x00010000 | # | Channel 120 | 5600 |
| Ch 124 |  | 0x00020000 | # | Channel 124 | 5620 |
| Ch 128 |  | 0x00040000 | # | Channel 128 | 5640 |
| Ch 132 |  | 0x00080000 | # | Channel 132 | 5660 |
| Ch 136 |  | 0x00100000 | # | Channel 136 | 5680 |
| Ch 140 |  | 0x00200000 | # | Channel 140 | 5700 |
| Ch 149 |  | 0x00400000 | # | Channel 149 | 5745 |
| Ch 153 |  | 0x00800000 | # | Channel 153 | 5765 |
| Ch 157 |  | 0x01000000 | # | Channel 157 | 5785 |
| Ch 161 |  | 0x02000000 | # | Channel 161 | 5805 |
| Ch 165 |  | 0x04000000 | # | Channel 165 | 5825 |

| Argument           | Description  |
|--------------------|--|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]                                       |
| 2. <i>resource</i> | Resource number. [W]   |
| 3. <i>venu_id</i>  | Number to uniquely identify this venue on this resource. [W] |
| 4. <i>x1</i>       | Floating point coordinate for lower-left corner.             |
| 5. <i>y1</i>       | Floating point coordinate for lower-left corner.             |
| 6. <i>x2</i>       | Floating point coordinate for upper-right corner.            |
| 7. <i>y2</i>       | Floating point coordinate for upper-right corner.            |
| 8. <i>freq_24</i>  | Frequency list for 2.4Ghz band, see above.                   |
| 9. <i>freq_5</i>   | Frequency list for 5Ghz band, see above.                     |

|                        |   |
|------------------------|---|
| 10. <i>description</i> | User-supplied description, ie: Big City Ball Park; 47-characters max. |
|------------------------|---|

Syntax: `add_venue shelf resource venu_id x1 y1 x2 y2 freq_24 freq_5 description`

#### 46. **add\_sta**

Add a WIFI Virtual Station (Virtual STA) interface. This command requires that the designated machine support LANforge driver for the Atheros brand WIFI NICs. A Virtual STA interface is a virtual interface that acts like a real wireless client. After creating the Virtual STA interface, you will need to configure its IP and other information. `NA` can be used for any values that you do not wish to modify.

Flags are currently defined as:

|                                    |                  |   |
|------------------------------------|------------------|---|
| <code>wpa_enable</code>            | 0x10             | # Enable WPA  |
| <code>custom_conf</code>           | 0x20             | # Use Custom wpa_supplicant config file.  |
| <code>wep_enable</code>            | 0x200            | # Use wpa_supplicant configured for WEP encryption.                             |
| <code>wpa2_enable</code>           | 0x400            | # Use wpa_supplicant configured for WPA2 encryption.                            |
| <code>ht40_disable</code>          | 0x800            | # Disable HT-40 even if hardware and AP support it.                             |
| <code>scan_ssid</code>             | 0x1000           | # Enable SCAN-SSID flag in wpa_supplicant.                                      |
| <code>passive_scan</code>          | 0x2000           | # Use passive scanning (don't send probe requests).                             |
| <code>disable_sgi</code>           | 0x4000           | # Disable SGI (Short Guard Interval).   |
| <code>lf_sta_migrate</code>        | 0x8000           | # OK-To-Migrate (Allow station migration between LANforge radios)               |
| <code>verbose</code>               | 0x10000          | # Verbose-Debug: Increase debug info in wpa-supPLICANT and hostapd logs.        |
| <code>80211u_enable</code>         | 0x20000          | # Enable 802.11u (Interworking) feature.  |
| <code>80211u_auto</code>           | 0x40000          | # Enable 802.11u (Interworking) Auto-internetworking feature. Always enable.    |
| <code>80211u_gw</code>             | 0x80000          | # AP Provides access to internet (802.11u Interworking)                         |
| <code>80211u_additional</code>     | 0x1000000        | # AP requires additional step for access (802.11u Interworking)                 |
| <code>80211u_e911</code>           | 0x2000000        | # AP claims emergency services reachable (802.11u Interworking)                 |
| <code>80211u_e911_unauth</code>    | 0x4000000        | # AP provides Unauthenticated emergency services (802.11u Interworking)         |
| <code>hs20_enable</code>           | 0x8000000        | # Enable Hotspot 2.0 (HS20) feature. Requires WPA-2.                            |
| <code>disable_gdafi</code>         | 0x10000000       | # AP: Disable DGAF (used by HotSpot 2.0).                                       |
| <code>8021x_radius</code>          | 0x20000000       | # Use 802.1x (RADIUS for AP).   |
| <code>80211r_pmska_cache</code>    | 0x40000000       | # Enable opportunistic PMSKA caching for WPA2 (Related to 802.11r).             |
| <code>disable_ht80</code>          | 0x80000000       | # Disable HT80 (for AC chipset NICs only)                                       |
| <code>ibss_mode</code>             | 0x200000000      | # Station should be in IBSS mode.   |
| <code>osen_enable</code>           | 0x400000000      | # Enable OSEN protocol (OSU Server-only Authentication)                         |
| <code>disable_roam</code>          | 0x800000000      | # Disable automatic station roaming based on scan results.                      |
| <code>ht160_enable</code>          | 0x1000000000     | # Enable HT160 mode.  |
| <code>disable_fast_reauth</code>   | 0x2000000000     | # Disable fast_reauth option for virtual stations.                              |
| <code>mesh_mode</code>             | 0x4000000000     | # Station should be in MESH mode.   |
| <code>power_save_enable</code>     | 0x8000000000     | # Station should enable power-save. May not work in all drivers/configurations. |
| <code>create_admin_down</code>     | 0x10000000000    | # Station should be created admin-down.   |
| <code>wds-mode</code>              | 0x20000000000    | # WDS station (sort of like a lame mesh), not supported on ath10k               |
| <code>no-supp-op-class-ie</code>   | 0x40000000000    | # Do not include supported-oper-class-IE in assoc requests. May work around.    |
| <code>txo-enable</code>            | 0x80000000000    | # Enable/disable tx-offloads, typically managed by set_wifi_txo command         |
| <code>use-wpa3</code>              | 0x100000000000   | # Enable WPA-3 (SAE Personal) mode.   |
| <code>use-bss-transition</code>    | 0x800000000000   | # Enable BSS transition.  |
| <code>disable-twt</code>           | 0x1000000000000  | # Disable TWT mode  |
| <code>disable-ofdma</code>         | 0x2000000000000  | # Disable OFDMA mode  |
| <code>disable-obss-scan</code>     | 0x4000000000000  | # Disable OBSS SCAN feature in supplicant.                                      |
| <code>ft-roam-over-ds</code>       | 0x8000000000000  | # Roam over DS when AP supports it.   |
| <code>rrm-ignore-beacon-req</code> | 0x10000000000000 | # Ignore (reject) RRM Beacon measurement request.                               |

To set any value to the default (or un-set), use `DEFAULT`. You may have to reboot the system to have the defaults take affect.

#### Rate configuration:

|                       |  |
|-----------------------|--|
| <code>DEFAULT</code>  | # Use maximum available speed  |
| <code>MCS0-76</code>  | # /n rates   |
| <code>[bitmap]</code> | # '0xff 00 ...' to directly specify the MCS bitmap.                    |
| <code>/b</code>       | # 1Mbps, 2Mbps, 5.5 Mbps, 11 Mbps                                      |
| <code>/a/g</code>     | # 6 Mbps, 9 Mbps, 12 Mbps, 18 Mbps, 24 Mbps, 36 Mbps, 48 Mbps, 54 Mbps |

#### Groups:

- o 802.11b
- o 802.11/a/g
- o 802.11/a/b/g
- o 1 Stream /n
- o 2 Streams /n
- o 3 Streams /n
- o v-1 Stream /AC
- o v-2 Streams /AC
- o v-3 Streams /AC

#### Mode

| Input                | : Enum | Val | : Shown by nc_show_ports |
|----------------------|--------|-----|--------------------------|
| <code>AUTO</code>    | 0      | #   | 802.11g                  |
| <code>802.11a</code> | 1      | #   | 802.11a                  |
| <code>b</code>       | 2      | #   | 802.11b                  |

```

g          | 3      # 802.11g
abg       | 4      # 802.11abg
abgn      | 5      # 802.11abgn
bgn       | 6      # 802.11bgn
bg        | 7      # 802.11bg
abgnAC    | 8      # 802.11abgn-AC
anAC      | 9      # 802.11an-AC
an        | 10     # 802.11an
bgnAC     | 11     # 802.11bgn-AC
abgnAX    | 12     # 802.11abgn-AX
          |        # a/b/g/n/AC/AX (dual-band AX) support
bgnAX     | 13     # 802.11bgn-AX
anAX      | 14     # 802.11an-AX
aAX       | 15     # 802.11a-AX (6E disables /n and /ac)
abgn7     | 16     # 802.11abgn-EHT
          |        # a/b/g/n/AC/AX/EHT (dual-band AX) support
bgn7      | 17     # 802.11bgn-EHT
an7       | 18     # 802.11an-EHT
a7        | 19     # 802.11a-EHT (6E disables /n and /ac)

```

### Related Commands

For mac-address pattern, release 5.4.1 and higher also supports sub-byte randomizations. For instance, this will randomize just the low 4 bits of the second octet: `xx:xx:xx:*4:xx`

| Argument                 | Description   |
|--------------------------|---|
| 1. <i>shelf</i>          | Shelf number. [R][D:1]  |
| 2. <i>resource</i>       | Resource number. [W]  |
| 3. <i>radio</i>          | Name of the physical radio interface, for example: <code>wiphy0</code>  |
| 4. <i>sta_name</i>       | Name for this Virtual STA, for example: <code>sta0</code> [W]   |
| 5. <i>flags</i>          | Flags for this interface (see above.)   |
| 6. <i>ssid</i>           | SSID for this Virtual STA. Use [BLANK] for empty SSID. Start with <code>0x</code> for HEX interpretation.                                     |
| 7. <i>nickname</i>       | Nickname for this Virtual STA. (No longer used)   |
| 8. <i>key</i>            | Encryption key (WEP, WPA, WPA2, WPA3, etc) for this Virtual STA. Prepend with <code>0x</code> for ascii-hex input.                            |
| 9. <i>AP</i>             | The Access Point BSSID this Virtual STA should be associated with (example: <code>00:11:22:33:4:55</code> , or <code>DEFAULT</code> for any). |
| 10. <i>wpa_cfg_file</i>  | WPA Supplicant config file.   |
| 11. <i>MAC</i>           | The MAC address, can also use parent-pattern in 5.3.8 and higher: <code>xx:xx:xx:*:*:xx</code>  |
| 12. <i>mode</i>          | WiFi mode: See above, use the numeric value (0 means AUTO, 1 means 802.11a, etc. [D:0])   |
| 13. <i>rate</i>          | Max rate, see help above.   |
| 14. <i>MAX_AMSDU</i>     | 1 == enabled, 0 == disabled, 0xFF == do not set.  |
| 15. <i>AMPDU_factor</i>  | 0-3, or 0xFF to not set.  |
| 16. <i>AMPDU_density</i> | 0-7, or 0xFF to not set.  |
| 17. <i>sta_br_IP</i>     | IP Address for station bridging. Set to 0.0.0.0 to use MAC bridging.  |
| 18. <i>flags_mask</i>    | If set, only these flags will be considered.  |
| 19. <i>ieee80211w</i>    | Management Frame Protection: 0: disabled, 1: optional, 2: Required.   |
| 20. <i>x_coord</i>       | Floating point number.  |
| 21. <i>y_coord</i>       | Floating point number.  |
| 22. <i>z_coord</i>       | Floating point number.  |

Syntax: `add_sta shelf resource radio sta_name flags ssid nickname key AP wpa_cfg_file MAC mode`



#### 47. add\_vap

Add a WIFI Virtual Access Point (VAP) interface. This command requires that the designated machine support the LANforge wifi driver for the Atheros brand WIFI NICs. A Virtual AP interface is a virtual interface that acts like a real Access Point. After creating the Virtual AP interface, you will need to configure it's IP and other information. 'NA' can be used for any values that you do not wish to modify.

AP flags are currently defined as:

|                    |                |   |
|--------------------|----------------|---|
| enable_wpa         | 0x10           | # Enable WPA  |
| hostapd_config     | 0x20           | # Use Custom hostapd config file.                                       |
| enable_80211d      | 0x40           | # Enable 802.11D to broadcast country-code & channels in VAPs           |
| short_preamble     | 0x80           | # Allow short-preamble  |
| pri_sec_ch_enable  | 0x100          | # Enable Primary/Secondary channel switch.                              |
| wep_enable         | 0x200          | # Enable WEP Encryption   |
| wpa2_enable        | 0x400          | # Enable WPA2 Encryption  |
| disable_ht40       | 0x800          | # Disable HT-40 (will use HT-20 if available).                          |
| verbose            | 0x10000        | # Verbose-Debug: Increase debug info in wpa-supPLICANT and hostapd logs |
| 80211u_enable      | 0x20000        | # Enable 802.11u (Interworking) feature.                                |
| 80211u_auto        | 0x40000        | # Enable 802.11u (Interworking) Auto-internetworking feature. Always en |
| 80211u_gw          | 0x80000        | # AP Provides access to internet (802.11u Interworking)                 |
| 80211u_additional  | 0x100000       | # AP requires additional step for access (802.11u Interworking)         |
| 80211u_e911        | 0x200000       | # AP claims emergency services reachable (802.11u Interworking)         |
| 80211u_e911_unauth | 0x400000       | # AP provides Unauthenticated emergency services (802.11u Interworking) |
| hs20_enable        | 0x800000       | # Enable Hotspot 2.0 (HS20) feature. Requires WPA-2.                    |
| disable_dgaf       | 0x1000000      | # AP Disable DGAF (used by HotSpot 2.0).                                |
| 8021x_radius       | 0x2000000      | # Use 802.1x (RADIUS for AP).   |
| 80211r_pmska_cache | 0x4000000      | # Enable oportunistic PMSKA caching for WPA2 (Related to 802.11r).      |
| disable_ht80       | 0x8000000      | # Disable HT80 (for AC chipset NICs only)                               |
| 80211h_enable      | 0x10000000     | # Enable 802.11h (needed for running on DFS channels) Requires 802.11d. |
| osen_enable        | 0x40000000     | # Enable OSEN protocol (OSU Server-only Authentication)                 |
| mcast_to_ucast     | 0x80000000     | # Request AP to translate multicats to unicast before sending to STAs   |
| ht160_enable       | 0x1000000000   | # Enable HT160 mode.  |
| create_admin_down  | 0x10000000000  | # Station should be created admin-down.                                 |
| use-wpa3           | 0x100000000000 | # Enable WPA-3 (SAE Personal) mode.                                     |
| use-bss-load       | 0x200000000000 | # Enable BSS Load IE in Beacons and Probe Responses (.11e).             |
| use-rrm-report     | 0x400000000000 | # Enable Radio measurements IE in beacon and probe responses.           |
| use-bss-transition | 0x800000000000 | # Enable BSS transition.  |

Mode options are below:

| Input   | : Enum | Val | : Shown by nc_show_ports                 |
|---------|--------|-----|--|
| AUTO    | 0      | #   | 802.11g                                  |
| 802.11a | 1      | #   | 802.11a                                  |
| b       | 2      | #   | 802.11b                                  |
| g       | 3      | #   | 802.11g                                  |
| abg     | 4      | #   | 802.11abg                                |
| abgn    | 5      | #   | 802.11abgn                               |
| bgn     | 6      | #   | 802.11bgn                                |
| bg      | 7      | #   | 802.11bg                                 |
| abgnAC  | 8      | #   | 802.11abgn-AC                            |
| anAC    | 9      | #   | 802.11an-AC                              |
| an      | 10     | #   | 802.11an                                 |
| bgnAC   | 11     | #   | 802.11bgn-AC                             |
| abgnAX  | 12     | #   | 802.11abgn-AX                            |
|         |        | #   | a/b/g/n/AC/AX (dual-band AX) support     |
| bgnAX   | 13     | #   | 802.11bgn-AX                             |
| anAX    | 14     | #   | 802.11an-AX                              |
| aAX     | 15     | #   | 802.11a-AX (6E disables /n and /ac)      |
| abgn7   | 16     | #   | 802.11abgn-EHT                           |
|         |        | #   | a/b/g/n/AC/AX/EHT (dual-band AX) support |
| bgn7    | 17     | #   | 802.11bgn-EHT                            |
| an7     | 18     | #   | 802.11an-EHT                             |
| a7      | 19     | #   | 802.11a-EHT (6E disables /n and /ac)     |

To set any value to the DEFAULT (or un-set), use DEFAULT. You may have to reboot the system to have the defaults take affect. For mac-address pattern, release 5.4.1 and higher also supports sub-byte randomizations. For instance, this will randomize just the low 4 bits of the second octet: xx:xx:xx:\*4:xx

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]  |
| 2. <i>resource</i> | Resource number. [W]  |
| 3. <i>radio</i>    | Name of the physical radio interface, for example: wiphy0 [W] |
| 4. <i>ap_name</i>  | Name for this Virtual AP, for example: vap0                   |

|                        |  |
|------------------------|--|
| 5. <i>flags</i>        | Flags for this interface (see above.)  |
| 6. <i>ssid</i>         | SSID for this Virtual AP.  |
| 7. <i>key</i>          | Encryption key for this Virtual AP. Prepend with 0x for ascii-hex representation.                |
| 8. <i>MAC</i>          | The MAC address, can also use parent-pattern in 5.3.8 and higher: xx:xx:xx:*:*:xx                |
| 9. <i>beacon</i>       | The beacon interval, in 1kus (1.024 ms), default 100, range: 15..65535                           |
| 10. <i>frag_thresh</i> | UN-USED, Was Fragmentation threshold, which is now set with <code>set_wifi_radio</code> , use NA |
| 11. <i>custom_cfg</i>  | Custom hostapd config file, if you want to craft your own config.                                |
| 12. <i>max_sta</i>     | Maximum number of Stations allowed to join this AP (1..2007)                                     |
| 13. <i>dtim_period</i> | DTIM period, range 1..255. Default 2.  |
| 14. <i>mode</i>        | WiFi mode: see table   |
| 15. <i>flags_mask</i>  | If set, only these flags will be considered.   |
| 16. <i>rate</i>        | Max rate, see help for <code>add_vsta</code>   |
| 17. <i>x_coord</i>     | Floating point number.   |
| 18. <i>y_coord</i>     | Floating point number.   |
| 19. <i>z_coord</i>     | Floating point number.   |
| 20. <i>ieee80211w</i>  | Management Frame Protection: 0: disabled, 1: optional, 2: Required.                              |

Syntax: `add_vap shelf resource radio ap_name flags ssid key MAC beacon frag_thresh custom_cfg max_sta dtim_period mode flags_mask rate x_coord y_coord z_coord ieee80211w`

#### 48. **add\_monitor**

Add a WIFI Monitor interface. These are useful for doing low-level wifi packet capturing. Flags are currently defined as:

```
disable_ht40 | 0x800 # Disable HT-40 even if hardware and AP support it.
disable_ht80 | 0x8000000 # Disable HT80 (for AC chipset NICs only)
ht160_enable | 0x10000000 # Enable HT160 mode.
```

| Argument             | Description  |
|----------------------|--|
| 1. <i>shelf</i>      | Shelf number. [R][D:1]   |
| 2. <i>resource</i>   | Resource number. [W]   |
| 3. <i>radio</i>      | Name of the physical radio interface, for example: <code>wiphy0</code> |
| 4. <i>ap_name</i>    | Name for this Monitor interface, for example: <code>moni0</code> [W]   |
| 5. <i>flags</i>      | Flags for this monitor interface.                                      |
| 6. <i>flags_mask</i> | Flags mask for this monitor interface.                                 |
| 7. <i>aid</i>        | AID, may be used when sniffing on /AX radios.                          |
| 8. <i>bssid</i>      | BSSID to use when sniffing on /AX radios, optional.                    |

Syntax: `add_monitor shelf resource radio ap_name flags flags_mask aid bssid`

#### 49. **add\_tm**

Create and add a new test manager to the system. A test manager is a collection of cross-connects that compose a test group. Users can be assigned to these groups and the groups can be password protected. This can be used to more easily share LANforge resources among several users. See Also: `tm_register`, `add_group`

| Argument       | Description  |
|----------------|--|
| 1. <i>name</i> | The name of the test manager. Must be unique across test managers. [R] |

Syntax: `add_tm name`

#### 50. **add\_group**

Create a new test group. Test groups are used to easily control and script collections of cross-connects. The CX types can be different within the group.

```
group_total_rates | 0x4 # Set rates as total for group.
```

See Also: `add_tgcx`

| Argument             | Description   |
|----------------------|---|
| 1. <i>name</i>       | The name of the test group. Must be unique across all groups. [R]         |
| 2. <i>flags</i>      | Flags for this group, see above.  |
| 3. <i>flags_mask</i> | Mask for flags that we care about, use 0xFFFFFFFF or leave blank for all. |

Syntax: `add_group name flags flags_mask`

#### 51. **add\_tgcx**

Adds CX to test group. See Also: `rm_tgcx`, `add_group`

| Argument             | Description                     |
|----------------------|---------------------------------|
| 1. <i>tgroupname</i> | The name of the test group. [R] |
| 2. <i>cxname</i>     | The name of the CX. [R]         |

Syntax: `add_tgcx tgroupname cxname`

#### 52. **add\_wl\_endp**

Add a WanLink (ICE) endpoint to the LANforge Manager. The endpoint may then be added to a cross-connect. If the endpoint already exists, then this command may be used to update the values. Note that you can leave everything after `port` off the command, and default values will be used.

For CPU thread, the value is only used on the A-endpoint. The B-endpoint is always on the same CPU as the A-endpoint.

```
SHOW_WP | 0x01 # Show WanPaths in wanlink endpoint table in GUI
```

| Argument              | Description  |
|-----------------------|--|
| 1. <i>alias</i>       | Name of endpoint. [R]  |
| 2. <i>shelf</i>       | Shelf name/id. [D:1]   |
| 3. <i>resource</i>    | Resource number.   |
| 4. <i>port</i>        | Port number or name.   |
| 5. <i>latency</i>     | The latency (ms) that will be added to each packet entering this WanLink.  |
| 6. <i>max_rate</i>    | Maximum transmit rate (bps) for this WanLink.                              |
| 7. <i>description</i> | Description for this endpoint, put in single quotes if it contains spaces. |
| 8. <i>cpu_id</i>      | The CPU/thread that this process should run on (kernel-mode only).         |
| 9. <i>wle_flags</i>   | WanLink Endpoint specific flags, see above.                                |

Syntax: `add_wl_endp alias shelf resource port latency max_rate description cpu_id wle_flags`

#### 53. **add\_wanpath**

Add a WanPath personality to a WanLink. The WanPath is like a virtual WanLink between a source

and destination IP or IP range. For instance, if you want communications between server A and client C to be different from communications between server B and client C, then you can set up two WanPaths to specify that behaviour. If the specified WanPath already exists, this command can be used to modify the existing values

| Argument                         | Description   |
|----------------------------------|---|
| 1. <i>wanlink</i>                | Name of WanLink to which we are adding this WanPath. [R]  |
| 2. <i>alias</i>                  | Name of WanPath. [R]  |
| 3. <i>speed</i>                  | The maximum speed this WanLink will accept (bps).   |
| 4. <i>latency</i>                | The base latency added to all packets, in milliseconds (or add 'us' suffix for microseconds)        |
| 5. <i>max_jitter</i>             | The maximum jitter, in milliseconds (or add 'us' suffix for microseconds)                           |
| 6. <i>extra_buffer</i>           | The extra amount of bytes to buffer before dropping pkts, in units of 1024, use -1 for AUTO. [D:-1] |
| 7. <i>reorder_freq</i>           | How often, out of 1,000,000 packets, should we make a packet out of order.                          |
| 8. <i>drop_freq</i>              | How often, out of 1,000,000 packets, should we purposefully drop a packet.                          |
| 9. <i>dup_freq</i>               | How often, out of 1,000,000 packets, should we purposefully duplicate a packet.                     |
| 10. <i>source_ip</i>             | Selection filter: Source IP.  |
| 11. <i>source_ip_mask</i>        | Selection filter: Source IP MASK.   |
| 12. <i>dest_ip</i>               | Selection filter: Destination IP.   |
| 13. <i>dest_ip_mask</i>          | Selection filter: Destination IP MASK.  |
| 14. <i>playback_capture</i>      | ON or OFF, should we play back a WAN capture file?  |
| 15. <i>playback_capture_file</i> | Name of the WAN capture file to play back.  |
| 16. <i>playback_loop</i>         | Should we loop the playback file, YES or NO or NA.  |
| 17. <i>ignore_bandwidth</i>      | Should we ignore the bandwidth settings from the playback file? YES, NO, or NA.                     |
| 18. <i>ignore_loss</i>           | Should we ignore the packet-loss settings from the playback file? YES, NO, or NA.                   |
| 19. <i>ignore_latency</i>        | Should we ignore the latency settings from the playback file? YES, NO, or NA.                       |
| 20. <i>ignore_dup</i>            | Should we ignore the Duplicate Packet settings from the playback file? YES, NO, or NA.              |
| 21. <i>jitter_freq</i>           | How often, out of 1,000,000 packets, should we apply random jitter.                                 |
| 22. <i>min_drop_amt</i>          | Minimum amount of packets to drop in a row. Default is 1. [D:1]                                     |
| 23. <i>max_drop_amt</i>          | Maximum amount of packets to drop in a row. Default is 1. [D:1]                                     |
| 24. <i>min_reorder_amt</i>       | Minimum amount of packets by which to reorder, Default is 1. [D:1]                                  |
| 25. <i>max_reorder_amt</i>       | Maximum amount of packets by which to reorder, Default is 10. [D:10]                                |
| 26. <i>drop_every_xth_pkt</i>    | YES to periodically drop every Xth pkt, NO to drop packets randomly.                                |

|                                  |   |
|----------------------------------|---|
| 27. <i>dup_every_xth_pkt</i>     | YES to periodically duplicate every Xth pkt, NO to duplicate packets randomly.        |
| 28. <i>reorder_every_xth_pkt</i> | YES to periodically reorder every Xth pkt, NO to reorder packets randomly.            |
| 29. <i>test_mgr</i>              | The name of the Test-Manager this WanPath is to use. Leave blank for no restrictions. |
| 30. <i>max_lateness</i>          | Maximum amount of un-intentional delay before pkt is dropped. Default is AUTO         |

Syntax: `add_wanpath wanlink alias speed latency max_jitter extra_buffer reorder_freq drop_freq dup_freq source_ip source_ip_mask dest_ip dest_ip_mask playback_capture playback_capture_file playback_loop ignore_bandwidth ignore_loss ignore_latency ignore_dup jitter_freq min_drop_amt max_drop_amt min_reorder_amt max_reorder_amt drop_every_xth_pkt dup_every_xth_pkt reorder_every_xth_pkt test_mgr max_lateness`

#### 54. **admin**

Various back-door commands. Current supported commands are:

##### **upgrade**

Upgrade lanforge using `lf_install` script currently installed on the LANforge system. Task executes in background and will reboot the LANforge when complete. First argument is resource-id (or ALL) Second argument is `lfver`, for instance: 5.4.6 Third argument is `kfver`, for instance: 5.19.17+ Fourth is extra arguments sent to `lf_install.pl`, leave blank if unsure.

##### **dhcpcd**

DHCPd event callback, creates event. First argument is network device name Second argument is `dhcpcd` message.

##### **mobile**

Mobile phone (hands free) script callback. First argument is endpoint name Second argument is call event type: answered or completed. Third argument is `Pesq` test file path or peer phone number Fourth argument is mob connection type: BT or cable

##### **resync\_clock**

Used on windows to force re-sync with the system clock.

##### **write\_xorp\_cfg [xorp-port]**

Re-write out the xorp-config file.

##### **ensure\_port [iface-name] [lanforge-iface-idx] [noprobe]**

Helper process only.

##### **scan\_complete [rsli-file-name] [request-key]**

Used by WiFi scan logic.

##### **probe\_complete [rsli-file-name] [request-key]**

Used by WiFi logic.

##### **ifup\_post\_complete [iface-name] [message]**

Tell LF that ifup script is complete.

##### **flush\_complete**

Tell resource all initial config has been sent from mgr.

##### **chamber [id] [angle] [flags] [table-speed-rpm]**

Chamber helper script callback. Angle is in 1/10 of a degree.

##### **req\_migrate [port-eid] [destination-radio] [mac-pattern]**

This will attempt to migrate a virtual station to a new radio. Any existing traffic connections will migrate with the station. The station may be re-named, but its MAC address and other configuration will remain the same (unless `mac-pattern`) is specified, in which case a new MAC will be created). If the `destination-radio` is not specified, then another radio will be chosen automatically. Example: `admin req_migrate 1.2.sta30 1.1.wiphy1 xx:xx:xx:xx*:xx`

##### **rfgen [message]**

API for the `rfgen` process to report status back to LANforge. Parsed messages are: starting, running, stopping, stopped, exiting Any other text will be treated as an error message to be delivered to the user(s).

##### **clean\_logs**

This will remove all LANforge related log files and restart logging with new log files. This will also reset WiFi radios so that related logs are restarted, so it is fairly disruptive.

##### **log\_complete**

An asynchronous log-gathering action has completed. Argument 1 is the name of the file.

##### **adb\_complete**

An asynchronous ADB command has completed. Argument 1 `adb-device`, arg2 is file-

name, arg3 is key

write\_xorp\_cfg only works on 'resource' processes.

| Argument       | Description   |
|----------------|---|
| 1. <i>cmd</i>  | Admin command:<br>resync_clock   write_xorp_cfg   scan_complete   ifup_post_complete   flush_complete   req_migrate   rfggen   chamber   clean_logs   upg |
| 2. <i>arg1</i> | Argument 1: xorp-port   scan-rslts-file   iface-name   iface-eid   rfggen-message   id   log_file_name  |
| 3. <i>arg2</i> | Argument 2: scan key   message   angle   dest-radio   adb-filename   lfver   event-id   |
| 4. <i>arg3</i> | Argument 3: noprobe   migrate-sta-mac-pattern   adb-key   kver   event-value-1  |
| 5. <i>arg5</i> | Argument 4: table-speed   extra-upgrade-args   event-value-2  |

Syntax: admin cmd arg1 arg2 arg3 arg5

#### 55. **apply\_vr\_cfg**

Apply all of the virtual routing settings for this Resource. This causes the routing tables to be created and configured properly for the specified configuration. This command should be run after making one or more changes to the virtual routers or virtual router connections. Please note that running this command when there are lots of virtual routers configured can take a long time. Check the status of the Card for percentage complete. Also, while this process is running, you will not be able to configure ports or virtual-router configuration.

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | The number of the shelf in question, or 'ALL'. [R][D:ALL] |
| 2. <i>resource</i> | The number of the resource in question, or 'ALL'. [W]     |

Syntax: apply\_vr\_cfg shelf resource

#### 56. **cancel\_vr\_cfg**

Setting up virtual router configurations can take a long time when there are lots of virtual routers. This command can cancel a configuration process before it is complete. Please note: the routing tables will be in an un-determined state after this, until you re-run the virtual router setup.

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | The number of the shelf in question, or 'ALL'. [R][D:ALL] |
| 2. <i>resource</i> | The number of the resource in question, or 'ALL'. [W]     |

Syntax: cancel\_vr\_cfg shelf resource

#### 57. **clear\_cx\_counters**

Clear counters for one or all cross-connects.

```
PORTS_TOO | 0x01 # Clear port counters this CX uses as well.  
SEND_EVENT | 0x02 # Send event when clearing counters.
```

| Argument              | Description  |
|-----------------------|--|
| 1. <i>cx_name</i>     | Name of Cross Connect, or 'all'. Null argument is same as 'all'.<br>[W][D:all] |
| 2. <i>clear_flags</i> | Optional argument to control clear logic.                                      |

Syntax: clear\_cx\_counters cx\_name clear\_flags

#### 58. **clear\_endp\_counters**

Clear counters for one or all endpoints. just\_lat: If YES, then just clear latency counters. just\_at: If RXGAP, then just clear the rxgap counters (5.4.2 and higher releases) Otherwise, all counters will be cleared.

```
PORTS_TOO | 0x01 # Clear this endpoint's port counters as well.  
SEND_EVENT | 0x02 # Send event when clearing counters.
```

| Argument | Description |
|----------|-------------|
|----------|-------------|

|                        |  |
|------------------------|--|
| 1. <i>endp_name</i>    | Name of Endpoint, or 'all'. Null argument is same as 'all'. [W]<br>[D:all]             |
| 2. <i>just_latency</i> | Enter 'YES' if you only want to clear latency counters, and see above for RXGAP.       |
| 3. <i>incr_seqno</i>   | Enter 'YES' if you want the target to increment the cfg-seq-no.                        |
| 4. <i>clear_flags</i>  | Optional argument to control clear logic. Ignored if <i>just_latency</i> is specified. |

Syntax: `clear_endp_counters endp_name just_latency incr_seqno clear_flags`

#### 59. **clear\_cd\_counters**

Clear counters for one or all Collision Domains.

| Argument          | Description   |
|-------------------|---|
| 1. <i>cd_name</i> | Name of Collision Domain, or 'all'. Null argument is same as 'all'.<br>[W][D:all] |

Syntax: `clear_cd_counters cd_name`

#### 60. **clear\_group**

Clears all cross-connects in a test group See Also: `add_group`, `add_tgcx`, `stop_group`

```
PORTS_TOO | 0x01 # Clear this endpoint's port counters as well.
SEND_EVENT | 0x02 # Send event when clearing counters.
```

| Argument       | Description                     |
|----------------|---------------------------------|
| 1. <i>name</i> | The name of the test group. [W] |

Syntax: `clear_group name`

#### 61. **clear\_port\_counters**

Clear counters on one or all ports on one or all resources. If `extra` is set to one of the below DHCP options, then counters will not be cleared, but the DHCP objects in question will be cleared. If DHCP is running, it will be stopped before clearing, and then restarted.

```
dhcp4_lease | # Remove dhcp lease files for IPv4 DHCP
dhcp6_lease | # Remove dhcp lease files for IPv6 DHCP
dhcp_leases | # Remove dhcp lease files for IPv4 and IPv6 DHCP
```

| Argument           | Description  |
|--------------------|--|
| 1. <i>shelf</i>    | The number of the shelf in question, or 'ALL'. [R][D:1]  |
| 2. <i>resource</i> | The number of the resource in question, or 'ALL'. [W]  |
| 3. <i>port</i>     | The number of the port in question, or 'ALL'. [W]  |
| 4. <i>extra</i>    | Clear something else instead: <code>dhcp4_lease</code>   <code>dhcp6_lease</code>   <code>dhcp_leases</code> |

Syntax: `clear_port_counters shelf resource port extra`

#### 62. **clear\_resource\_counters**

Clear counters on one or all resources.

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | The number of the shelf in question, or 'ALL'. [R][D:1] |
| 2. <i>resource</i> | The number of the resource in question, or 'ALL'. [W]   |

Syntax: `clear_resource_counters shelf resource`

#### 63. **clear\_wifi\_profiles**

This will clear wifi profiles from the device in question. Initial use is for ADB devices, probably will want to do similar for other 'real' devices in the future.

| Argument              | Description  |
|-----------------------|--|
| 1. <i>shelf</i>       | Shelf number, or ALL. [R][D:1]                                   |
| 2. <i>resource</i>    | Resource number, or ALL. [W]                                     |
| 3. <i>type</i>        | Object type: adb, or ALL.  |
| 4. <i>id</i>          | Object identifier: adb-id, or ALL.                               |
| 5. <i>except_ssid</i> | Do not delete profiles that reference this SSID, NA deletes all. |

Syntax: `clear_wifi_profiles shelf resource type id except_ssid`

#### 64. **clear\_wp\_counters**

Clear WanPath counters for one endpoint.

| Argument            | Description                   |
|---------------------|-------------------------------|
| 1. <i>endp_name</i> | Name of WanLink Endpoint. [W] |
| 2. <i>wp_name</i>   | Name of WanPath to clear.     |

Syntax: `clear_wp_counters endp_name wp_name`

#### 65. **discover**

Force discovery of nodes on the management network. Note that discovery runs automatically about every minute. option argument: Set to 'disconnect' to force disconnect to remote resource process. Set to 'adb' for ADB device discovery.

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf-ID, only used if discovering Attenuators. [R][D:1]        |
| 2. <i>resource</i> | Resource ID. Use if discovering Attenuators or ADB devices. [W] |
| 3. <i>option</i>   | See above.  |

Syntax: `discover shelf resource option`

#### 66. **diag**

This command prints out information that can be used by support staff to diagnose certain issues.

##### Diagnostic Types:

```

NA           |# everything (default)
alerts      |# alert messages
license     |# license contents
counters    |# endpoint counters
fds         |# file descriptors
clients     |# connected clients
endpoints   |# list of endpoints
shelf       |#
iobuffer    |#

```

| Argument       | Description  |
|----------------|--|
| 1. <i>type</i> | Default (blank) is everything, options: alerts, license, counters, fds, clients, endpoints, shelf, iobuffer. |
| 2. <i>arg1</i> | Optional: Endpoint name to diag.   |

Syntax: `diag type arg1`

#### 67. **notify\_dhcp**

Handle input from the DHCP client process. This should not normally be called by users, but only by other LANforge processes. This always assumes local shelf/card, so they are not specified.

| Argument         | Description  |
|------------------|--|
| 1. <i>cmd</i>    | set/down/timeout/info: What does DHCP want us to do? [W] |
| 2. <i>port</i>   | Interface name. [W]                                      |
| 3. <i>reason</i> | DHCP reason, informational mostly.                       |



|                      |  |
|----------------------|--|
| 4. <i>new_ip</i>     | New IP address.  |
| 5. <i>netmask</i>    | New subnet mask.   |
| 6. <i>new_mtu</i>    | New MTU.   |
| 7. <i>new_router</i> | One or more default routers. LANforge will only use the first one. |
| 8. <i>new_dns</i>    | New DNS server(s) for use by this interface.                       |
| 9. <i>new_ip6</i>    | New Global IPv6 address: ipv6/prefix                               |

Syntax: `notify_dhcp cmd port reason new_ip netmask new_mtu new_router new_dns new_ip6`

#### 68. **do\_pesq**

This command starts a PESQ calculation for the results saved by a VOIP endpoint. This command is usually used internally by LANforge so it is unlikely you will ever use it directly. The LANforge system will determine the source file (which must exist on the receiving machine in the same place it does on the transmitting machine) and send a request to the LANforge PESQ server to compare the source to the result file specified in this command.

The results will be associated with the VOIP endpoint and may be displayed with the `show_pesq` command

| Argument                   | Description  |
|----------------------------|--|
| 1. <i>endp_name</i>        | Name of Endpoint. [W]                              |
| 2. <i>result_file_name</i> | The name of the file received by the endpoint. [W] |

Syntax: `do_pesq endp_name result_file_name`

#### 69. **file**

Transfer files through LANforge API. This will include upload and download.

This command creates a prompt on the connected GUI. This command does not transfer files via JSON protocol.

```
UNLINK_WHEN_DL_COMPLETE | 0x01 # Remove the file once it has been downloaded.
```

| Argument            | Description  |
|---------------------|--|
| 1. <i>shelf</i>     | Shelf ID [R][D:1]  |
| 2. <i>card</i>      | Resource ID [W]  |
| 3. <i>cmd</i>       | Only 'Download' supported for now, 'Upload' reserved for future use. [W][D:Download] |
| 4. <i>filename</i>  | File to transfer. [W]  |
| 5. <i>req_id</i>    | Request identifier, uint32. Will be passed back in response frames.                  |
| 6. <i>client_id</i> | Internal use only.   |
| 7. <i>flags</i>     | Options for the file operation, see above.   |

Syntax: `file shelf card cmd filename req_id client_id flags`

#### 70. **gossip**

Send a message to everyone else logged in to the server.

| Argument          | Description  |
|-------------------|--|
| 1. <i>message</i> | Message to show to others currently logged on. Unescaped Value [W] |

Syntax: `gossip message`

#### 71. **getintxrate**

Get the tx rate (packets per second) over the last 3 seconds. Values will always be fresh (cached values are not used). Value will be an integer.

Response: InTxRate=INTEGER

| Argument       | Description   |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect or Test-Group name [W]                  |
| 2. <i>AorB</i> | For endpoint a, enter 'A', for endpoint b, enter 'B'. |

Syntax: `getintxrate CX AorB`

#### 72. **getinrxrate**

Get the rx rate (packets per second) over the last 3 seconds. Values will always be fresh (cached values are not used). Value will be an integer.

Response: InRxRate=INTEGER

| Argument       | Description   |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect or Test-Group name [W]                  |
| 2. <i>AorB</i> | For endpoint a, enter 'A', for endpoint b, enter 'B'. |

Syntax: `getinrxrate CX AorB`

#### 73. **getinrxbps**

Get the rx bits-per-second rate over the last 3 seconds. Values will always be fresh (cached values are not used). Value will be an integer.

Response: InRxBps=INTEGER

| Argument       | Description   |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect or Test-Group name [W]                  |
| 2. <i>AorB</i> | For endpoint a, enter 'A', for endpoint b, enter 'B'. |

Syntax: `getinrxbps CX AorB`

#### 74. **gettxpkts**

Get the total tx packets count. Values will always be fresh (cached values are not used). Value will be an integer.

Response: TxPkts=INTEGER

| Argument       | Description   |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect or Test-Group name [W]                  |
| 2. <i>AorB</i> | For endpoint a, enter 'A', for endpoint b, enter 'B'. |

Syntax: `gettxpkts CX AorB`

#### 75. **getrxpkts**

Get the total rx packets count. Values will always be fresh (cached values are not used). Value will be an integer.

Response: RxPkts=INTEGER

| Argument       | Description   |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect or Test-Group name [W]                  |
| 2. <i>AorB</i> | For endpoint a, enter 'A', for endpoint b, enter 'B'. |

Syntax: `getrxpkts CX AorB`

#### 76. **getpktdrops**

Get the total packets dropped. The drops will be detected by sequence number gaps, and will be based on packets RECEIVED by this endpoint. Values will always be fresh (cached values are not used). Value will be an integer.

Response: PktDrops=INTEGER

| Argument       | Description                               |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect or Test-Group name [W]      |
| 2. <i>AorB</i> | For AtoB, enter 'B', for BtoA, enter 'A'. |

Syntax: `getpktdrops CX AorB`

#### 77. **getavglatency**

Get the average latency (over the last 30 seconds) for packets received by and endpoint. Values will always be fresh (cached values are not used). Value will be an integer, units are milliseconds.

Response: AvgLatency=INTEGER

| Argument       | Description                               |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect or Test-Group name [W]      |
| 2. <i>AorB</i> | For AtoB, enter 'B', for BtoA, enter 'A'. |

Syntax: `getavglatency CX AorB`

#### 78. **getrxporterrpkts**

Get the total error packets detected on the receiving port (interface). The errors will be based on what is reported by the driver and/or hardware for this interface. Values will always be fresh (cached values are not used). Value will be an integer.

Response: RxPortErrPkts=INTEGER

| Argument       | Description                               |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect name [W]                    |
| 2. <i>AorB</i> | For AtoB, enter 'B', for BtoA, enter 'A'. |

Syntax: `getrxporterrpkts CX AorB`

#### 79. **getxendperpkts**

Get the total error packets detected on the receiving endpoint. The errors will be the sum of things like CRC errors, packets received on the wrong device, and any other errors we can detect for this particular endpoint. Values will always be fresh (cached values are not used). Value will be an integer.

Response: RxEndpErrPkts=INTEGER

| Argument       | Description                               |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect or Test-Group name [W]      |
| 2. <i>AorB</i> | For AtoB, enter 'B', for BtoA, enter 'A'. |

Syntax: `getxendperpkts CX AorB`

#### 80. **getipadd**

Get the IP for the endpoint. Value will be cached (but IP addresses do not often change, so the result should almost always be immediately correct.) Response: IPAdd=xxx.xxx.xxx.xxx

| Argument       | Description   |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect name [W]                                |
| 2. <i>AorB</i> | For endpoint a, enter 'A', for endpoint b, enter 'B'. |

Syntax: `getipadd CX AorB`

#### 81. **getmask**

Get the IP Mask for the endpoint. Value will be cached (but IP addresses do not often change, so the result should almost always be immediately correct.) Response: Mask=xxx.xxx.xxx.xxx

| Argument | Description |
|----------|-------------|
|----------|-------------|

|                |   |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect name                                    |
| 2. <i>AorB</i> | For endpoint a, enter 'A', for endpoint b, enter 'B'. |

Syntax: `getmask CX AorB`

82. **getmac**

Get the MAC address for the endpoint. Value will be cached (but IP addresses do not often change, so the result should almost always be immediately correct.) Response:  
MAC=aa:bb:cc:dd:ee:ff

| Argument       | Description   |
|----------------|---|
| 1. <i>CX</i>   | Cross-connect name [W]                                |
| 2. <i>AorB</i> | For endpoint a, enter 'A', for endpoint b, enter 'B'. |

Syntax: `getmac CX AorB`

83. **?**

Show help for commands(s). If no command is specified, then a brief listing of all commands will be printed out. If a command is specified, then a verbose printing of that command will be printed.

| Argument          | Description  |
|-------------------|--|
| 1. <i>command</i> | The command to get help for. Can be 'all', or blank. |

Syntax: `? command`

84. **init\_wiser**

Initialize the Wiser NCW/HNW module. This requires that one have the proper library installed. Contact sales@candelatech.com if you have questions. If the file\_name has spaces in it, be sure to enclose it in double quotes.

| Argument             | Description  |
|----------------------|--|
| 1. <i>shelf</i>      | The number of the shelf in question. [R][D:1]                                  |
| 2. <i>resource</i>   | The number of the resource in question. [W]                                    |
| 3. <i>file_name</i>  | The WISER file name for the desired emulation, or 'NA' for empty string.       |
| 4. <i>node_count</i> | The number of WISER nodes for the desired emulation, or 'NA' for empty string. |

Syntax: `init_wiser shelf resource file_name node_count`

85. **ios**

For IPC - SwiftUI sending information to BTSERVER

**Operations are defined as follows.** - setresourceinfo - devicestats - urlreport

| Argument       | Description                         |
|----------------|-------------------------------------|
| 1. <i>cmd</i>  | Operation that device is requesting |
| 2. <i>arg1</i> | arg1                                |
| 3. <i>arg2</i> | arg2                                |
| 4. <i>arg3</i> | arg3                                |
| 5. <i>arg4</i> | arg4                                |
| 6. <i>arg5</i> | arg5                                |
| 7. <i>arg6</i> | arg6                                |
| 8. <i>arg7</i> | arg7                                |
| 9. <i>arg8</i> | arg8                                |

|                  |       |
|------------------|-------|
| 10. <i>arg9</i>  | arg9  |
| 11. <i>arg10</i> | arg10 |
| 12. <i>arg11</i> | arg11 |
| 13. <i>arg12</i> | arg12 |
| 14. <i>arg13</i> | arg13 |
| 15. <i>arg14</i> | arg14 |
| 16. <i>arg15</i> | arg15 |
| 17. <i>arg16</i> | arg16 |
| 18. <i>arg17</i> | arg17 |
| 19. <i>arg18</i> | arg18 |
| 20. <i>arg19</i> | arg19 |
| 21. <i>arg20</i> | arg20 |

Syntax: `ios cmd arg1 arg2 arg3 arg4 arg5 arg6 arg7 arg8 arg9 arg10 arg11 arg12 arg13 arg14 arg15 arg16 arg17 arg18 arg19 arg20`

#### 86. licenses

Print out license information. See also: `set_license`

| Argument            | Description   |
|---------------------|---|
| 1. <i>popup</i>     | If 'popup', then cause a GUI popup msg, otherwise, just show text.        |
| 2. <i>show_file</i> | If 'yes', then show the license file, not the parsed license information. |

Syntax: `licenses popup show_file`

#### 87. load

This command will completely erase the current setup in memory and replace it with the database specified with this command. You must specify a database to be loaded, though note that if you specify a database that does not exist, and chose 'overwrite', you will effectively initialize the LANforge system to defaults. The default database is called: DFLT

| Argument                 | Description   |
|--------------------------|---|
| 1. <i>name</i>           | The name of the database to load. (DFLT is the default) [W]                                       |
| 2. <i>action</i>         | Should be 'append' or 'overwrite'. [W]  |
| 3. <i>clean_dut</i>      | If yes, then DUT will be cleaned up when overwrite is selected, otherwise they will be kept.      |
| 4. <i>clean_chambers</i> | If yes, then Chambers will be cleaned up when overwrite is selected, otherwise they will be kept. |
| 5. <i>clean_profiles</i> | If yes, then clean all profiles when overwrite is selected, otherwise they will be kept.          |

Syntax: `load name action clean_dut clean_chambers clean_profiles`

#### 88. login

If you are the first to use this name, a new client will be created for you. If this is an existing client account, then you take on the characteristics of that client. At this time, that is only a few flags. If the password is set for this client, and the password given here is invalid, the client will not be logged in as the new user. See `set_password` to modify the password.

| Argument       | Description  |
|----------------|--|
| 1. <i>name</i> | A single name with no white-spaces (15 characters or less) [W] |

|                    |   |
|--------------------|---|
| 2. <i>password</i> | Can be blank or 'NA' if no password is set, otherwise must be the password. |
|--------------------|---|

Syntax: login name password

### 89. **create\_client**

Create a new client (user).

| Argument             | Description   |
|----------------------|---|
| 1. <i>name</i>       | A single name with no white-spaces (15 characters or less) [W]  |
| 2. <i>password</i>   | Can be blank or 'NA' if no password is set, otherwise must be the password. Use IGNORE for no change. |
| 3. <i>super_user</i> | 1 If you want this user to have Administrative powers, 0 or blank otherwise.                          |

Syntax: create\_client name password super\_user

### 90. **log\_capture**

Save log files to a specified location, useful for gathering stuff for automated testing.

```
adb |#
journalctl |#
hostapd |#
lflogs |#
supplicant |#

async_feedback | %{user_key}
```

#### **adb**

Android ADB logs.

*identifier*: adb device id

*duration*: 'all' means entire file, a number X in seconds grabs last 'X' seconds of logs. This can take a while, so it is done asynchronously. A keyed message even will be sent when the log is complete.

#### **hostapd**

Logs from hostapd (VAP).

*identifier*: vap port name

*duration*: 'all' means entire file, a number X in seconds grabs last 'X' seconds of logs.

#### **journalctl**

System and kernel logs.

*identifier*: NA

*duration*: 'boot' means since boot, other values are passed to --since argument of journalctl, so use things like: '10 minutes ago'

#### **lflogs**

LANforge manager (resource 0) or resource (resource 1+) logs.

*identifier*: NA

*duration*: 'all' means entire file, a number X in seconds grabs last 'X' seconds of logs.

#### **supplicant**

Logs from wpa\_supplicant (STA ports)

*identifier*: port (wifi radio) name

*duration*: 'all' means entire file, a number X in seconds grabs last 'X' seconds of logs.

| Argument              | Description  |
|-----------------------|--|
| 1. <i>shelf</i>       | The number of the shelf in question. [R][D:1]  |
| 2. <i>resource</i>    | The number of the resource in question. [W]  |
| 3. <i>type</i>        | journalctl, supplicant, lflogs, adb, hostapd [W]   |
| 4. <i>identifier</i>  | port name or other identifier needed for some types, NA if not used.   |
| 5. <i>duration</i>    | For journalctl, seconds of logs to gather, or NA if not used.  |
| 6. <i>destination</i> | Where to save the file to on the LANforge resource. If 'stdout', then content will be passed back as a keyed text message. [R] |

|                    |  |
|--------------------|--|
| 7. <i>user_key</i> | Key to use for keyed-text-message response when using stdout destination [W] |
|--------------------|--|

Syntax: `log_capture shelf resource type identifier duration destination user_key`

### 91. **log\_level**

Sets the logging level for the primary log stream. The values are bit-fields: add them together to get the desired level. If you enter `log_level` by itself, then you can see the current level.

If the second argument exists, it will apply to the entity specified. Without an argument it just modifies the local server in general.

You can enter the value in HEX if you prefix it with `0x`.

|         |            |                            |         |
|---------|------------|----------------------------|---------|
| DIS     | 0x1        | # disasters                | (1)     |
| ERR     | 0x2        | # errors                   | (2)     |
| WRN     | 0x4        | # warnings                 | (4)     |
| INF     | 0x8        | # info                     | (8)     |
| TRC     | 0x10       | # function trace           | (16)    |
| DBG     | 0x20       | # debug                    | (32)    |
| SEC     | 0x40       | # log security violations  | (64)    |
| DB      | 0x80       | # Database related logging | (128)   |
| XMT     | 0x100      | # Output going to clients  | (256)   |
| SCRIPT  | 0x400      | # Scripting specific stuff | (1024)  |
| PARSE   | 0x800      | # PARSE specific           | (2048)  |
| DBG2    | 0x1000     | # very verbose logging     | (4096)  |
| LIO     | 0x2000     | # IO logging               | (8192)  |
| OUT1    | 0x4000     | # Some std-out logging     | (16384) |
| LL_PROF | 0x8000     | # Profiling information    | (32768) |
| CUST1   | 0x10000    | # Cust-1, latency info     | (65536) |
| ALL     | 0xFFFFFFFF | # Log everything           |         |

| Argument         | Description                                     |
|------------------|---|
| 1. <i>level</i>  | Integer corresponding to the logging flags. [W] |
| 2. <i>target</i> | Options: 'gnu'   [file-endp-name].              |

Syntax: `log_level level target`

### 92. **log\_msg**

Send an message to the LANforge log file.

| Argument          | Description                         |
|-------------------|-------------------------------------|
| 1. <i>message</i> | Message to log. Unescaped Value [W] |

Syntax: `log_msg message`

### 93. **motd**

This command prints out alerts and other info that may be useful for debugging LANforge configuration problems.

Syntax: `motd`

### 94. **nc\_show\_endpoints**

Show one or all endpoints. Will NOT use cached values. Some endpoint types take an extra argument to specify what to show more precisely: Generic endpoints check extra for 'history' and in that case they will report recent output, not just the last line of output.

| Argument           | Description                     |
|--------------------|---------------------------------|
| 1. <i>endpoint</i> | Name of endpoint, or 'all'. [W] |
| 2. <i>extra</i>    | See above.                      |

Syntax: `nc_show_endpoints endpoint extra`

### 95. **nc\_show\_pesq**

Show PESQ results for one or all VOIP endpoints. Will NOT use cached values.

| Argument           | Description                     |
|--------------------|---------------------------------|
| 1. <i>endpoint</i> | Name of endpoint, or 'all'. [W] |

Syntax: `nc_show_pesq endpoint`

### 96. `nc_show_ports`

Show one/all ports for one/all resources in one/all shelves. This command WILL NOT use cached values, so it will be a little slower. It is useful for scripts and situations where the 3-5 second caching is too slow to yield the results needed.

**Probe-Flags** options:

```
WIFI          | 0x1      # 1 include wifi stations
MII           | 0x2      # 2 include MII
ETHTOOL      | 0x4      # 4 include ethtool results
BRIDGE       | 0x8      # 8 include bridges
EASY_IP_INFO | 0x10     # 16 Everything but gateway information, which is expensive to probe.
GW           | 0x20     # 32 include gateway information
GW_FORCE_REFRESH | 0x40    # 64 Force GW (re)probe. Otherwise, cached values *might* be used.
```

| Argument              | Description   |
|-----------------------|---|
| 1. <i>shelf</i>       | Name/id of the shelf, or 'all'. [R][D:1]  |
| 2. <i>resource</i>    | Resource number, or 'all'. [W]  |
| 3. <i>port</i>        | Port number, or 'all'. [W]  |
| 4. <i>probe_flags</i> | See above, add them together for multiple probings. Leave blank if you want stats only. |

Syntax: `nc_show_ports shelf resource port probe_flags`

### 97. `c_show_ports`

Show one/all ports for one/all resources in one/all shelves. This command will ALWAYS use cached values, so it may return stale values. It is useful when the system cannot return non-cached values due to timeouts, and perhaps for configuration information that does not need to be probed.

**Probe-Flags** options:

```
WIFI          | 0x1      # 1 include wifi stations
MII           | 0x2      # 2 include MII
ETHTOOL      | 0x4      # 4 include ethtool results
BRIDGE       | 0x8      # 8 include bridges
EASY_IP_INFO | 0x10     # 16 Everything but gateway information, which is expensive to probe.
GW           | 0x20     # 32 include gateway information
GW_FORCE_REFRESH | 0x40    # 64 Force GW (re)probe. Otherwise, cached values *might* be used.
```

| Argument              | Description   |
|-----------------------|---|
| 1. <i>shelf</i>       | Name/id of the shelf, or 'all'. [R][D:1]  |
| 2. <i>resource</i>    | Resource number, or 'all'. [W]  |
| 3. <i>port</i>        | Port number, or 'all'. [W]  |
| 4. <i>probe_flags</i> | See above, add them together for multiple probings. Leave blank if you want stats only. |

Syntax: `c_show_ports shelf resource port probe_flags`

### 98. `nc_show_channel_groups`

Show one/all ChannelGroups for one/all resources in one/all shelves. An empty specifier will be treated as 'all'. Will always request the absolute latest information from the remote system(s)

| Argument               | Description                              |
|------------------------|--|
| 1. <i>shelf</i>        | Name/id of the shelf, or 'all'. [R][D:1] |
| 2. <i>resource</i>     | Resource number, or 'all'. [W]           |
| 3. <i>channel_name</i> | Name of the channel, or 'all'. [W]       |

Syntax: `nc_show_channel_groups shelf resource channel_name`

### 99. `nc_show_spans`

Show one/all Spans for one/all resources in one/all shelves. An empty specifier will be treated as 'all'. Will always request the absolute latest information from the remote system(s)



| Argument              | Description                              |
|-----------------------|--|
| 1. <i>shelf</i>       | Name/id of the shelf, or 'all'. [R][D:1] |
| 2. <i>resource</i>    | Resource number, or 'all'. [W]           |
| 3. <i>span_number</i> | Span-Number of the span, or 'all'. [W]   |

Syntax: `nc_show_spans shelf resource span_number`

#### 100. **nc\_show\_vr**

Show one/all Virtual Routers for one/all resources in one/all shelves. An empty specifier will be treated as 'all'. This command will always request the absolute latest information from the remote system(s)

| Argument           | Description                               |
|--------------------|---|
| 1. <i>shelf</i>    | Name/id of the shelf, or 'all'. [R][D:1]  |
| 2. <i>resource</i> | Resource number, or 'all'. [W]            |
| 3. <i>router</i>   | Name of the Virtual Router, or 'all'. [W] |

Syntax: `nc_show_vr shelf resource router`

#### 101. **nc\_show\_vrcx**

Show one/all Virtual Router Connections for one/all resources in one/all shelves. Only Connections on the 'free-list', those not associated with any Virtual Router will be shown with this command unless you exactly specify the VRCX Name. If the VRCX is in a virtual router, only cached results will be shown. Connections associated with routers will be shown with the 'show\_vr' command with the rest of the router information. This command will always request the absolute latest information from the remote system(s)

| Argument           | Description  |
|--------------------|--|
| 1. <i>shelf</i>    | Name/id of the shelf, or 'all'. [R][D:1]             |
| 2. <i>resource</i> | Resource number, or 'all'. [W]                       |
| 3. <i>cx_name</i>  | Name of the Virtual Router Connection, or 'all'. [W] |

Syntax: `nc_show_vrcx shelf resource cx_name`

#### 102. **nc\_show\_cd**

Show one/all Collision Domains for one/all resources in one/all shelves. An empty specifier will be treated as 'all'. This command will always request the absolute latest information from the remote system(s)

| Argument                   | Description                                 |
|----------------------------|---|
| 1. <i>shelf</i>            | Name/id of the shelf, or 'all'. [R][D:1]    |
| 2. <i>resource</i>         | Resource number, or 'all'. [W]              |
| 3. <i>collision-domain</i> | Name of the Collision Domain, or 'all'. [W] |

Syntax: `nc_show_cd shelf resource collision-domain`

#### 103. **nc\_show\_ppp\_links**

Show one/all PPP Links for one/all resources in one/all shelves. An empty specifier will be treated as 'all'.

| Argument           | Description                                |
|--------------------|--|
| 1. <i>shelf</i>    | Name/id of the shelf, or 'all'. [R][D:1]   |
| 2. <i>resource</i> | Resource number, or 'all'. [W]             |
| 3. <i>link_num</i> | Ppp-Link number of the span, or 'all'. [W] |

Syntax: `nc_show_ppp_links shelf resource link_num`

#### 104. **probe\_port**

This calls various command-line tools to probe the port and returns the results as a text message. This command will trigger a popup message in the LANforge client. To disable that popup, append the key `probe_port.quiet.[EID]` where EID is the Shelf, Resource, and ID of the port being probed. E.G.:

```
probe_port 1 1 br0 probe_port.quiet.1.1.3
```

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]  |
| 2. <i>resource</i> | Resource number. [W]  |
| 3. <i>port</i>     | Port number or name [W]   |
| 4. <i>key</i>      | Unique identifier for this request. Usually left blank. Use 'probe_port.quiet.[EID]' to suppress popup. |

Syntax: `probe_port shelf resource port key`

#### 105. **probe\_ports**

Check for the existence of new (virtual) interfaces.

| Argument           | Description                              |
|--------------------|--|
| 1. <i>shelf</i>    | Name/id of the shelf, or 'all'. [R][D:1] |
| 2. <i>resource</i> | Resource number, or 'all'. [W]           |

Syntax: `probe_ports shelf resource`

#### 106. **port\_reset\_completed**

Internal command used by port-reset script to notify LANforge the reset has completed. This is only valid for Resource processes.

| Argument        | Description  |
|-----------------|--|
| 1. <i>port</i>  | The port in question. [W]  |
| 2. <i>type</i>  | SUNOS, NORMAL, or SECIP..let us know what kind of reset completed. |
| 3. <i>extra</i> | IP for SECIP, blank for others.                                    |

Syntax: `port_reset_completed port type extra`

#### 107. **exit**

Log out of the LANforge control server.

Syntax: `exit`

#### 108. **report**

Configure server side reporting. This is useful if you want the LANforge-Manager to save reports instead of the LANforge-GUI.

| Argument                | Description  |
|-------------------------|--|
| 1. <i>rpt_dir</i>       | Directory in which reports should be saved. [W]              |
| 2. <i>reporting_on</i>  | Should we globally enable/disable reporting. (YES, NO or NA) |
| 3. <i>save_endps</i>    | Should we save endpoint reports or not. (YES, NO or NA)      |
| 4. <i>save_resource</i> | Should we save Resource reports or not. (YES, NO or NA)      |
| 5. <i>save_ports</i>    | Should we save Port reports or not. (YES, NO or NA)          |

Syntax: `report rpt_dir reporting_on save_endps save_resource save_ports`

#### 109. **reset\_port**

This command will cause the driver on the selected ports to reset the driver (admin down, admin up). It will also re-initialize all of the routing information for that interface. This command will disrupt

traffic, but it can be useful if the port locks up or if you wish to restart higher level services (such as dhcp and supplicant for wifi). See the user-guide section on setting up IP addresses and routing for more information.

Do not override the default of YES for `reset_ospf` unless you are certain that is the right thing to do.

The `pre_ifdown` field controls portal login/logout activity and may not actually cause the lower-level driver information to be reset. If left blank or set to NA, then the port will be reset as described above (and any existing ifdown/up scripts will be aborted), and the portal logout script will not be called. Basic options are as follows:

```
YES      | # (include logout) Call portal-bot.pl ... --logout before going down.
P-OUT   | # Only call the portal logout (do not reset drivers/supplicant/dhcp)
P-IN    | # Only call the portal login (do not reset drivers/supplicant/dhcp)
```

| Argument             | Description   |
|----------------------|---|
| 1. <i>shelf</i>      | Shelf number, or ALL. [R][D:1]  |
| 2. <i>resource</i>   | Resource number, or ALL. [W]  |
| 3. <i>port</i>       | Port number to reset, or ALL. [W]   |
| 4. <i>reset_ospf</i> | If set to 'NO' or 'NA', then OSPF will not be updated. Otherwise, it will be updated. |
| 5. <i>pre_ifdown</i> | See above. Leave blank or use NA if unsure.   |

Syntax: `reset_port shelf resource port reset_ospf pre_ifdown`

#### 110. **reset\_serial\_span**

This command will cause the Serial Span (T1, etc) driver to be reloaded. This may help work around bugs in the T1 driver and/or hardware.

| Argument           | Description                      |
|--------------------|----------------------------------|
| 1. <i>shelf</i>    | Shelf number [R][D:1]            |
| 2. <i>resource</i> | Resource (machine) number. [W]   |
| 3. <i>span</i>     | Serial-Span number to reset. [W] |

Syntax: `reset_serial_span shelf resource span`

#### 111. **reboot\_os**

This will reboot the Operating System on the resource specified. All processes will be killed on that resource, of course. Upon reboot, server processes will be re-started, including the LANforge server. See also: `reboot_OS`

| Argument           | Description                    |
|--------------------|--------------------------------|
| 1. <i>shelf</i>    | Shelf number, or ALL. [R][D:1] |
| 2. <i>resource</i> | Resource number, or ALL. [W]   |

Syntax: `reboot_os shelf resource`

#### 112. **rm\_attenuator**

| Argument           | Description                                 |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number, usually 1 [R][D:1]            |
| 2. <i>resource</i> | Resource number [W]                         |
| 3. <i>serno</i>    | Serial number for requested Attenuator. [W] |

Syntax: `rm_attenuator shelf resource serno`

#### 113. **rm\_chamber**

| Argument | Description |
|----------|-------------|
|----------|-------------|

|                   |                            |
|-------------------|----------------------------|
| 1. <i>chamber</i> | Chamber name, or 'ALL' [W] |
|-------------------|----------------------------|

Syntax: `rm_chamber chamber`

#### 114. **rm\_chamber\_path**

Remove one or all chamber paths from a chamber.

| Argument          | Description                                   |
|-------------------|---|
| 1. <i>chamber</i> | Chamber Name. [W]                             |
| 2. <i>path</i>    | Path Name, use 'ALL' to delete all paths. [W] |

Syntax: `rm_chamber_path chamber path`

#### 115. **rm\_dut**

| Argument        | Description            |
|-----------------|------------------------|
| 1. <i>shelf</i> | DUT name, or 'ALL' [W] |

Syntax: `rm_dut shelf`

#### 116. **rm\_rfgn**

| Argument           | Description                      |
|--------------------|----------------------------------|
| 1. <i>shelf</i>    | Shelf number, usually 1 [R][D:1] |
| 2. <i>resource</i> | Resource number [W]              |

Syntax: `rm_rfgn shelf resource`

#### 117. **rm\_cd**

Remove a Collision Domain. Any endpoints still associated with this CD will be gracefully removed from the CD, but will not otherwise be affected.

| Argument     | Description                   |
|--------------|-------------------------------|
| 1. <i>cd</i> | Name of Collision Domain. [W] |

Syntax: `rm_cd cd`

#### 118. **rm\_cd\_endp**

Remove an Endpoint from a Collision Domain.

| Argument       | Description                   |
|----------------|-------------------------------|
| 1. <i>cd</i>   | Name of Collision Domain. [W] |
| 2. <i>endp</i> | Endpoint name/id. [W]         |

Syntax: `rm_cd_endp cd endp`

#### 119. **rm\_cd\_vr**

Remove a Virtual Router from a Collision Domain.

| Argument       | Description                   |
|----------------|-------------------------------|
| 1. <i>cd</i>   | Name of Collision Domain. [W] |
| 2. <i>endp</i> | Virtual-Router name/id. [W]   |

Syntax: `rm_cd_vr cd endp`

#### 120. **rm\_endp**

Remove an endpoint. 'YES\_ALL' for endp-name will delete all endpoints.

#### Related Commands

| Argument            | Description                             |
|---------------------|---|
| 1. <i>endp_name</i> | Name of the endpoint, or 'YES_ALL'. [W] |

Syntax: `rm_endp endp_name`

#### 121. **rm\_channel\_group**

Remove a channel group, or set of groups.

| Argument               | Description                              |
|------------------------|--|
| 1. <i>shelf</i>        | Name/id of the shelf, or 'all'. [R][D:1] |
| 2. <i>resource</i>     | Resource number, or 'all'. [W]           |
| 3. <i>channel_name</i> | Name of the channel, or 'all'. [W]       |

Syntax: `rm_channel_group shelf resource channel_name`

#### 122. **rm\_event**

| Argument           | Description                    |
|--------------------|--------------------------------|
| 1. <i>event_id</i> | Numeric event-id, or 'all' [W] |

Syntax: `rm_event event_id`

#### 123. **rm\_group**

Deletes a test group. See Also: `add_group`, `rm_tgcx`

| Argument       | Description                     |
|----------------|---------------------------------|
| 1. <i>name</i> | The name of the test group. [W] |

Syntax: `rm_group name`

#### 124. **rm\_profile**

Remove Device Profile configuration.

| Argument       | Description                |
|----------------|----------------------------|
| 1. <i>name</i> | Profile name, or 'ALL' [W] |

Syntax: `rm_profile name`

#### 125. **rm\_text\_blob**

Remove Text Blob.

| Argument       | Description                  |
|----------------|------------------------------|
| 1. <i>type</i> | Text Blob type, or 'ALL' [W] |
| 2. <i>name</i> | Text Blob Name, or 'ALL' [W] |

Syntax: `rm_text_blob type name`

#### 126. **rm\_traffic\_profile**

Remove Traffic Profile configuration.

| Argument       | Description                |
|----------------|----------------------------|
| 1. <i>name</i> | Profile name, or 'ALL' [W] |

Syntax: `rm_traffic_profile name`

#### 127. **rm\_threshold**

Remove a threshold-alert for a particular endpoint.

| Argument | Description |
|----------|-------------|
|----------|-------------|

|                     |  |
|---------------------|--|
| 1. <i>endp</i>      | Endpoint name or ID. [W]                             |
| 2. <i>thresh_id</i> | Threshold ID to remove. Use 'all' to remove all. [W] |

Syntax: `rm_threshold endp thresh_id`

128. **rm\_tgcx**

Removes CX from test group. See Also: `add_tgcx`, `add_group`

| Argument         | Description                     |
|------------------|---------------------------------|
| 1. <i>tgroup</i> | The name of the test group. [W] |
| 2. <i>cxname</i> | The name of the CX. [W]         |

Syntax: `rm_tgcx tgroup cxname`

129. **rm\_venue**

Remove a venue

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]  |
| 2. <i>resource</i> | Resource number, or 'ALL' [W]   |
| 3. <i>venu_id</i>  | Number to uniquely identify this venue on this resource, or 'ALL' [W] |

Syntax: `rm_venue shelf resource venu_id`

130. **rm\_vr**

Remove one or all Virtual Routers.

| Argument              | Description                              |
|-----------------------|--|
| 1. <i>shelf</i>       | Name/id of the shelf, or 'all'. [R][D:1] |
| 2. <i>resource</i>    | Resource number, or 'all'. [W]           |
| 3. <i>router_name</i> | Virtual Router name, or 'all'. [W]       |

Syntax: `rm_vr shelf resource router_name`

131. **rm\_vrcx**

Remove one or all Virtual Router Connections on the free-list. Underlying objects will be deleted if they were auto-created to begin with unless you specify the last argument as 'vrcx\_only'.

| Argument                  | Description   |
|---------------------------|---|
| 1. <i>shelf</i>           | Name/id of the shelf, or 'all'. [R][D:1]  |
| 2. <i>resource</i>        | Resource number, or 'all'. [W]  |
| 3. <i>connection_name</i> | Virtual Router Connection name, or 'all'. [W]   |
| 4. <i>vrcx_only</i>       | If we should NOT delete underlying auto-created objects, enter 'vrcx_only' here, otherwise leave blank or use NA.     |
| 5. <i>vr_id</i>           | If not removing from the free-list, then supply the virtual-router name/ID here. Leave blank or use NA for free-list. |

Syntax: `rm_vrcx shelf resource connection_name vrcx_only vr_id`

132. **rm\_span**

Remove a Serial Span (T1, etc), or a set of spans.

| Argument        | Description                              |
|-----------------|--|
| 1. <i>shelf</i> | Name/id of the shelf, or 'all'. [R][D:1] |

|                    |   |
|--------------------|---|
| 2. <i>resource</i> | Resource number, or 'all'. [W]            |
| 3. <i>span_num</i> | Span-Number of the channel, or 'all'. [W] |

Syntax: `rm_span shelf resource span_num`

### 133. **rm\_ppp\_link**

Remove a PppLink.

| Argument           | Description                                    |
|--------------------|--|
| 1. <i>shelf</i>    | Name/id of the shelf. [R][D:1]                 |
| 2. <i>resource</i> | Resource number that holds this PppLink. [W]   |
| 3. <i>unit_num</i> | Unit-Number for the PppLink to be deleted. [W] |

Syntax: `rm_ppp_link shelf resource unit_num`

### 134. **rm\_client**

Delete a stored client profile. The client cannot be logged on currently. Changes will not be permanent until you write out the database. The client will be removed from all test managers as well.

| Argument                  | Description   |
|---------------------------|---|
| 1. <i>client_name</i>     | Name of the client profile you wish to remove. [W]  |
| 2. <i>client_password</i> | Client password. Not required if we are super-user. |

Syntax: `rm_client client_name client_password`

### 135. **rm\_cx**

Delete a cross-connect from the system.

#### Related Commands

| Argument           | Description                              |
|--------------------|--|
| 1. <i>test_mgr</i> | Name of test-mgr, or 'all'. [W]          |
| 2. <i>cx_name</i>  | Name of the cross-connect, or 'all'. [W] |

Syntax: `rm_cx test_mgr cx_name`

### 136. **rm\_wanpath**

Remove one or all wanpaths from an endpoint.

| Argument            | Description               |
|---------------------|---------------------------|
| 1. <i>endp_name</i> | Name of the endpoint. [W] |
| 2. <i>wp_name</i>   | Name of the wanpath. [W]  |

Syntax: `rm_wanpath endp_name wp_name`

### 137. **rm\_db**

Delete a database.

| Argument          | Description                         |
|-------------------|-------------------------------------|
| 1. <i>db_name</i> | Name of the database to delete. [W] |

Syntax: `rm_db db_name`

### 138. **rm\_resource**

Remove a phantom Resource and all of its configuration.

| Argument        | Description            |
|-----------------|------------------------|
| 1. <i>shelf</i> | Shelf number. [R][D:1] |

|                    |                      |
|--------------------|----------------------|
| 2. <i>resource</i> | Resource number. [W] |
|--------------------|----------------------|

Syntax: `rm_resource shelf resource`

### 139. **rm\_sec\_ip**

Remove secondary IP Address(es).

#### Related Commands

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]  |
| 2. <i>resource</i> | Resource number. [W]  |
| 3. <i>port</i>     | Name of network device (Port) from which these IPs will be removed. [W] |
| 4. <i>ip_list</i>  | IP1/prefix,IP2/prefix,...IPZ/prefix, or ALL [W]                         |

Syntax: `rm_sec_ip shelf resource port ip_list`

### 140. **rm\_vlan**

Remove an 802.1Q VLAN or MAC-VLAN.

#### Related Commands

| Argument           | Description                                       |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]                            |
| 2. <i>resource</i> | Resource number. [W]                              |
| 3. <i>port</i>     | Port number or name of the virtual interface. [W] |

Syntax: `rm_vlan shelf resource port`

### 141. **rm\_test\_mgr**

Remove a test manager. Cross-connects will not be directly affected. There is no need to unregister clients first: This command will take care of that for you.

| Argument           | Description                                 |
|--------------------|---|
| 1. <i>test_mgr</i> | Name of the test manager to be removed. [W] |

Syntax: `rm_test_mgr test_mgr`

### 142. **save**

This command allows you to save the current test configuration, including all Endpoints, and all TestManagers. You may then use the 'load' command to initialize the LANforge Manager with the previously saved database. If you do not specify a name, it will be saved as the default database (DFLT), and will be automatically loaded at startup.

| Argument          | Description  |
|-------------------|--|
| 1. <i>db_name</i> | The name the backup shall be saved as (blank means dflt) |

Syntax: `save db_name`

### 143. **scan\_wifi**

Scan for WiFi access points. Only works for WiFi Virtual Station Interfaces (Virtual STA). The `extra` argument allows some control over how the scan is done:

```
NA | # (or left blank) the system does a full scan
dump | # then only cached values are returned
trigger freq [freq] | # scan exactly those frequencies
```

Example of scanning multiple frequencies:

```
scan 1 1 sta1 NA 'trigger freq 5180 5300'
```

| Argument | Description |
|----------|-------------|
|----------|-------------|



|                    |   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]                                  |
| 2. <i>resource</i> | Resource number. [W]                                    |
| 3. <i>port</i>     | Port number or name of the virtual interface. [W]       |
| 4. <i>key</i>      | Unique identifier for this request. Usually left blank. |
| 5. <i>extra</i>    | Extra arguments to the scan script, see above.          |

Syntax: `scan_wifi shelf resource port key extra`

#### 144. **set\_arm\_info**

Set Armageddon Endpoint configuration. You may enter `AUTO` for any value that you wish LANforge to calculate for you or set to defaults. Note that randomizing many of these values will mean packets may not be received on the receiving port due to routing or switching issues.

If `multi_pkts` is set to a value greater than 1, that number of identical packets will be sent before creating a new packet. This can significantly increase performance, but at the cost of not having as much accuracy when calculating latency values. It will also cause the 'duplicate packet' to increment.

**Armageddon-flags** are as follows:

```
rel_tstamp      | 0x400      # Use Relative Timestamps. This will increase performance
                # but can only work if the 'TSC' clock is stable and both
                # endpoints are on the same machine. It is difficult for
                # the code to know if the TSC is stable or not, so we cannot
                # verify this for you at this time.

use_gw_mac      | 0x1000     # Use default gateway's MAC for destination MAC.
                # Dest-MAC must also be set to 'DEFAULT' for
                # this option to take effect.

slow_start      | 0x2000     # Use slow-start logic. This ramps up
                # the speed a bit slower when
                # starting the endpoint and after a clear of its stats.
                # With this disabled (the default value), the endpoint
                # may over-shoot the desired bandwidth for a fraction
                # of a second causing un-expected stress on the
                # network under test.

udp_checksum    | 0x4000     # Use UDP Checksums.
use_tcp         | 0x8000     # Use TCP instead of UDP protocol. (Note this is NOT stateful TCP!)
random_payload  | 0x10000    # Use random payload sizes instead of linear increase
                # between min and max (release 5.3.6+)
```

| Argument                 | Description  |
|--------------------------|--|
| 1. <i>name</i>           | Name of the Endpoint we are setting. [R]                           |
| 2. <i>min_pkt_size</i>   | Minimum packet size, including all Ethernet headers (but not CRC). |
| 3. <i>max_pkt_size</i>   | Maximum packet size, including all Ethernet headers (but not CRC). |
| 4. <i>udp_src_min</i>    | Minimum source UDP port.   |
| 5. <i>udp_src_max</i>    | Maximum source UDP port.   |
| 6. <i>udp_dst_min</i>    | Minimum destination UDP port.                                      |
| 7. <i>udp_dst_max</i>    | Minimum destination UDP port.                                      |
| 8. <i>ip_src_min</i>     | Minimum source IP address to use.                                  |
| 9. <i>ip_src_max</i>     | Maximum source IP address to use.                                  |
| 10. <i>ip_dst_min</i>    | Minimum destination IP address to use.                             |
| 11. <i>ip_dst_max</i>    | Maximum destination IP address to use.                             |
| 12. <i>src_mac_count</i> | How many source MACs to iterate through.                           |
| 13. <i>dst_mac_count</i> | How many destination MACs to iterate through.                      |
| 14. <i>src_mac</i>       | The source MAC address.  |

|                         |  |
|-------------------------|--|
| 15. <i>dst_mac</i>      | The destination MAC address.   |
| 16. <i>multi_pkts</i>   | The number of identical packets to send before creating a new one.   |
| 17. <i>pkts_to_send</i> | The number of packets to send. Set to zero for infinite.   |
| 18. <i>arm_flags</i>    | Armageddon-related flags, see above for details.   |
| 19. <i>burst</i>        | Burst amount, can significantly improve throughput with some modern drivers, similar to 'multi_pkts', and uses the 'xmit_more' linux skb option. |

Syntax: `set_arm_info name min_pkt_size max_pkt_size udp_src_min udp_src_max udp_dst_min udp_dst_max ip_src_min ip_src_max ip_dst_min ip_dst_max src_mac_count dst_mac_count src_mac dst_mac multi_pkts pkts_to_send arm_flags burst`

#### 145. **set\_attenuator**

Set attenuation value on specified attenuator module. Units are 1/10 of a dB (ddB). To start/stop the Attenuator, which really only makes sense when using scripts on the Attenuator, set attenuator-index to 'all', and 'val' to START or STOP

##### Attenuator Mode:

```
0 |# Normal
1 |# Pulse mode (API Tech 4205A modules directly connected via USB only)
```

| Argument                    | Description  |
|-----------------------------|--|
| 1. <i>shelf</i>             | Shelf number, usually 1. [R][D:1]  |
| 2. <i>resource</i>          | Resource number. [W]   |
| 3. <i>serno</i>             | Serial number for requested Attenuator, or 'all'. [W]  |
| 4. <i>atten_idx</i>         | Attenuator index, or 'all'. [W]  |
| 5. <i>val</i>               | Requested attenuation in 1/10ths of dB (ddB).  |
| 6. <i>mode</i>              | 0 == normal attenuator, 1 == pulse mode (API Tech 4205A modules directly connected via USB only) |
| 7. <i>pulse_width_us5</i>   | Pulse width in units of 1/2 micro second. So, if you want 1.5us, use value 3 (0-60000)           |
| 8. <i>pulse_interval_ms</i> | Time between pulses, in milli-seconds (0-60000).   |
| 9. <i>pulse_count</i>       | Number of pulses (0-255)   |
| 10. <i>pulse_time_ms</i>    | Time interval between pulse groups in milliseconds (1-60000)                                     |
| 11. <i>atten_count</i>      | For cases where we are creating/setting a phantom attenuator.                                    |
| 12. <i>ip_addr</i>          | IP address, in case this Attenuator is to be managed over TCP.                                   |

Syntax: `set_attenuator shelf resource serno atten_idx val mode pulse_width_us5 pulse_interval_ms pulse_count pulse_time_ms atten_count ip_addr`

#### 146. **set\_rfggen**

Set RF Noise-generator (RADAR) config.

```
running      | 0x2      # Should we start the RF Generator or not?
one-burst    | 0x8      # Run for about 1 second and stop. Uses 5-sec sweep time for single pulse train.
```

| Argument               | Description   |
|------------------------|---|
| 1. <i>shelf</i>        | Shelf number, usually 1. [R][D:1]                               |
| 2. <i>resource</i>     | Resource number. [W]  |
| 3. <i>id</i>           | RF Generator ID, not used at this time, enter 'NA' or 0. [D:NA] |
| 4. <i>rfggen_flags</i> | RF Generator flags, see above.                                  |

|                             |  |
|-----------------------------|--|
| 5. <i>rfgen_flags_mask</i>  | Mask of what flags to set, see above.              |
| 6. <i>pulse_width_us</i>    | Requested pulse width, units are in micro-seconds. |
| 7. <i>pulse_interval_us</i> | Time between pulses, in micro-seconds.             |
| 8. <i>pulse_count</i>       | Number of pulses (0-255)                           |
| 9. <i>sweep_time_ms</i>     | Time interval between pulse groups in milliseconds |
| 10. <i>freq_khz</i>         | Center frequency in Khz                            |
| 11. <i>gain</i>             | Main TX/RX Amp, 0 or 14 (dB), default is 14        |
| 12. <i>if_gain</i>          | Fine-tune TX/RX Gain, 0 - 40 dB                    |
| 13. <i>bb_gain</i>          | RX Gain, 0 - 62 in 2dB steps                       |

Syntax: `set_rfgen shelf resource id rfgen_flags rfgen_flags_mask pulse_width_us pulse_interval_us pulse_count sweep_time_ms freq_khz gain if_gain bb_gain`

#### 147. **blink\_attenuator**

Visually identify attenuator by blinking LEDs or changing LCD colors or similar.

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number, usually 1. [R][D:1]                     |
| 2. <i>resource</i> | Resource number. [W]                                  |
| 3. <i>serno</i>    | Serial number for requested Attenuator, or 'all'. [W] |

Syntax: `blink_attenuator shelf resource serno`

#### 148. **flash\_attenuator**

Upload new software image to specified attenuator.

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number, usually 1. [R][D:1]                     |
| 2. <i>resource</i> | Resource number. [W]                                  |
| 3. <i>serno</i>    | Serial number for requested Attenuator, or 'all'. [W] |
| 4. <i>filename</i> | File to use when uploading to attenuator.             |

Syntax: `flash_attenuator shelf resource serno filename`

#### 149. **set\_chamber**

| Argument               | Description   |
|------------------------|---|
| 1. <i>chamber</i>      | Chamber name [W]  |
| 2. <i>turntable</i>    | Turn-table address, for instance: 192.168.1.22:3001   |
| 3. <i>speed_rpm</i>    | Speed in rpm (floating point number is accepted)  |
| 4. <i>position</i>     | Absolute position in degrees.   |
| 5. <i>tilt</i>         | Absolute tilt in degrees.   |
| 6. <i>cur_rotation</i> | Primarily used to store the last known rotation for turntables that do not report absolute position. Use NA or leave blank if unsure. |

Syntax: `set_chamber chamber turntable speed_rpm position tilt cur_rotation`

#### 150. **set\_cx\_report\_timer**

You must be registered with the Test-Manager(s) in order for this operation to succeed. The timer should be  $\geq 500$ ms. This command will also cause the LANforge Resources to report to the LANforge Manager on a similar time interval.

| Argument               | Description  |
|------------------------|--|
| 1. <i>test_mgr</i>     | Name of the test manager, or 'all'. [W]  |
| 2. <i>cx_name</i>      | Name of cross-connect, or 'all'. [W]   |
| 3. <i>milliseconds</i> | Report timer length in milliseconds. [W,250-60000][D:5000]   |
| 4. <i>CXONLY</i>       | If you want to set the timer for ONLY the CX, and not the endpoints, enter 'cxonly'. Otherwise, leave it blank.. |

Syntax: `set_cx_report_timer test_mgr cx_name milliseconds CXONLY`

#### 151. **set\_endp\_proxy**

This is only used when using proxy IP & Port with Layer-3 connections.

| Argument                | Description                              |
|-------------------------|--|
| 1. <i>endp_name</i>     | Name of endpoint. [W]                    |
| 2. <i>enabled</i>       | YES or NO to enable or disable proxying. |
| 3. <i>proxy_ip</i>      | Proxy IP Address.                        |
| 4. <i>proxy_ip_port</i> | Proxy IP Port.                           |

Syntax: `set_endp_proxy endp_name enabled proxy_ip proxy_ip_port`

#### 152. **set\_endp\_report\_timer**

The timer should be greater or equal to 500ms. This will cause the LANforge-GUI to request reports at the specified interval. For large numbers of entities, it is suggested to use longer report times to decrease load on the GUI.

| Argument               | Description  |
|------------------------|--|
| 1. <i>endp_name</i>    | Name of endpoint. [R]                                      |
| 2. <i>milliseconds</i> | Report timer length in milliseconds. [W,250-60000][D:5000] |

Syntax: `set_endp_report_timer endp_name milliseconds`

#### 153. **set\_cx\_state**

Set the state of the Cross-Connect(s). Valid states are:

```

RUNNING | # Sets the CX(s) in the running state.
SWITCH  | # Sets the CX(s) in the running state, stopping any conflicting tests.
QUIESCE | # Stop transmitting and gracefully stop cross-connect.
STOPPED | # Sets the CX(s) in the stopped state.
DELETED | # Deletes the CX(s).

```

SWITCH only works on WanLink cross-connects at this time.

#### Related Commands

| Argument           | Description  |
|--------------------|--|
| 1. <i>test_mgr</i> | Name of the test-manager, or 'all'. [W]                    |
| 2. <i>cx_name</i>  | Name of the cross-connect, or 'all'. [W]                   |
| 3. <i>cx_state</i> | One of: RUNNING, SWITCH, QUIESCE, STOPPED, or DELETED. [W] |

Syntax: `set_cx_state test_mgr cx_name cx_state`

#### 154. **set\_l4\_endp**

Set some extra layer-4 endpoint configuration.

```

Media source:
  DASH           1
  SMOOTH_STREAMING 2
  HLS            3
  PROGRESSIVE    4
  RTSP           5

Media quality:
  4K             0
  8K             1

```

|       |   |
|-------|---|
| 1080p | 2 |
| 720p  | 3 |
| 360p  | 4 |

| Argument                     | Description                                    |
|------------------------------|--|
| 1. <i>alias</i>              | Name of endpoint. [R]                          |
| 2. <i>media_source</i>       | Specify media source, see above                |
| 3. <i>media_quality</i>      | Specify media quality, see above               |
| 4. <i>media_playbacks</i>    | Maximum number of media playbacks              |
| 5. <i>media_random_seeks</i> | Maximum number of media random seeks           |
| 6. <i>duration_min</i>       | Minimum duration of media playback, in seconds |
| 7. <i>duration_max</i>       | Maximum duration of media playback, in seconds |

Syntax: `set_l4_endp alias media_source media_quality media_playbacks media_random_seeks duration_min duration_max`

#### 155. **set\_license**

Install license keys on the manager machine. Enter the license keys as a single command. LANforge will break them into separate lines internally.

| Argument           | Description  |
|--------------------|--|
| 1. <i>licenses</i> | License keys all appended into a single line. <code>Unescaped Value</code> [W] |

Syntax: `set_license licenses`

#### 156. **set\_password**

Set the password for the current client (if client is not specified), or the specified client if we are logged in as 'admin'.

| Argument               | Description   |
|------------------------|---|
| 1. <i>old_password</i> | Old password, or 'NA' for blank password. [W]               |
| 2. <i>new_password</i> | New password, or 'NA' for blank password. [W]               |
| 3. <i>client</i>       | Specify the client. If left blank, will use current client. |

Syntax: `set_password old_password new_password client`

#### 157. **set\_ppp\_link\_state**

Set the state of the PPP Link(s). Valid states are: RUNNING -- Sets the PPP Link(s) in the running state. STOPPED -- Sets the PPP Link(s) in the stopped state. DELETED -- Deletes the PPP Link(s).

| Argument            | Description                                |
|---------------------|--|
| 1. <i>shelf</i>     | Name of the Shelf, or 'all'. [R][D:1]      |
| 2. <i>resource</i>  | Number of the Resource, or 'all'. [W]      |
| 3. <i>link</i>      | Unit Number of the PPP Link, or 'all'. [W] |
| 4. <i>ppp_state</i> | One of: RUNNING, STOPPED, or DELETED. [R]  |

Syntax: `set_ppp_link_state shelf resource link ppp_state`

#### 158. **set\_resource**

Set some options for resources (clients)

##### **Max staged bringup**

is for all interfaces on a resource.

##### **Max trying ifup**

is the maximum amount of IP/Route configuration scripts that can be running concurrently.

##### **Max station bringup**

is maximum amount of stations that can be brought up per radio per 'tick'.

A tick is a minimum of 0.25 seconds, and may be longer on slower or more heavily loaded systems.

In general, you would want **max-station-bringup** to be less than **max-staged-bringup** so that multiple radios could bring up stations concurrently.

Device profiles specify high-level behaviour. The value is set as a list of device profile names, each of which may have an optional number of traffic profile names appended with a '.' separating them. For example, this would specify two device profiles: '5-STA-AC 5-STA-N' and this would specify each with one or more traffic-profiles: '5-STA-AC.tcp-dl.udp-slow-bi 5-STA-n.slow-tcp-dl'

```
skip_load_db_on_start | 0x1 # Should we skip loading the DB on start?
```

| Argument                       | Description   |
|--------------------------------|---|
| 1. <i>shelf</i>                | Name of the Shelf, or all. [R][D:1]   |
| 2. <i>resource</i>             | Number of the Resource, or all. [W]   |
| 3. <i>max_staged_bringup</i>   | Maximum amount of interfaces attempting to come up at once. Default is 50             |
| 4. <i>max_trying_ifup</i>      | Maximum amount of interfaces running the network config 'ifup' logic. Default is 15   |
| 5. <i>max_station_bringup</i>  | Maximum amount of stations to bring up per radio per tick. Default is 12.             |
| 6. <i>device_profiles</i>      | List of profiles, see above   |
| 7. <i>top_left_x</i>           | X Location for Chamber View.  |
| 8. <i>top_left_y</i>           | X Location for Chamber View.  |
| 9. <i>max_helper_count</i>     | Maximum number of helper traffic generation processes. 0 means CPU-core-count (AUTO). |
| 10. <i>resource_flags</i>      | System wide flags, often requires a reboot for changes to take effect.                |
| 11. <i>resource_flags_mask</i> | What flags to change. If unset, default is all.                                       |
| 12. <i>user_name</i>           | Store user-name configured for this Resource. Only settable during DB load.           |

Syntax: `set_resource shelf resource max_staged_bringup max_trying_ifup max_station_bringup device_profiles top_left_x top_left_y max_helper_count resource_flags resource_flags_mask user_name`

### 159. **set\_script**

Add or modify a script for a particular endpoint, Test-Group, or Attenuator. Script types supported are currently:

```
NONE | # Delete any existing script.
Script2544 | # For RFC 2544 type testing.
ScriptHunt | # Hunt for maximum speed with constraints.
ScriptWL | # For iterating through WanLink settings
ScriptAtten | # For Attenuators only.
```

Flags are defined as:

```
SCR_STOPPED | 0x1 # Script should NOT have any affect on the endpoint.
SCR_NO_KEYED_RPT | 0x2 # Script should NOT send reports to the CLI/GUI.
SCR_SYMMETRIC | 0x4 # This script should apply settings to the peer endpoing as well.
SCR_HIDE_ITER_DETAILS | 0x8 # Hide iteration detail reports.
SCR_HIDE_LEGEND | 0x10 # Don't print the legend in the report.
SCR_HIDE_CSV | 0x20 # Don't print the CSV data in the report.
SCR_RUN_ON_MGR | 0x40 # Set automatically by LANforge.
SCR_COMPLETED | 0x80 # Set automatically by LANforge.
SCR_LOOP | 0x100 # Loop script until manually stopped.
SCR_SHOW_DUPS | 0x200 # Report duplicate packets.
SCR_SHOW_OOO | 0x400 # Report out-of-order packets.
SCR_HIDE_HUNT | 0x800 # Hide the individual hunt steps..just show results.
SCR_HIDE_LAT | 0x1000 # Hide latency distribution reports.
SCR_HIDE_CONSTRAINTS | 0x2000 # Hide constraints messages.
SCR_SHOW_ATTENUATION | 0x4000 # Show attenuation packet stats.
SCR_USE_MSS | 0x8000 # When setting packet size, set TCP MSS instead if endpoint supports that.
SCR_SHOW_GOLDEN_LF | 0x10000 # Add 'golden' LANforge graph for comparison (where available).
SCR_SHOW_GOLDEN_3P | 0x20000 # Add 'golden' third-party AP graph for comparison (where available).
```

#### Script2544

- Private data syntax:

```
run_duration pause_duration rates_a payload_sizes_a constraints report rates_b sizes_b attenuations attenuator
```

- `rates_*` and `payload_sizes_*` are comma-separated-strings, e.g.:  
60,128,256,1472
- The interval durations are in milliseconds.
- Constraints syntax:

```
drops,jitter_us,latency_us,max_tx_slowdown,max_failed_ok
```

- Report syntax: (read-only, use `NA` when configuring)

```
steps_completed,steps_failed
```

### ScriptHunt

- Private data syntax:

```
run_duration pause_duration constraints payload_sizes_a payload_sizes_b attenuations attenuator
```

- Constraints syntax:

```
drops,jitter_us,latency_us,max_steps,start_rate,accuracy,is_bps,max_tx_slowdown
```

### ScriptWL

- Private data syntax:

```
run_duration rates latencies jitter drops
```

- Rates, latencies, jitter and drops are comma-separated-strings, e.g.: 60,128,256,1472
- Default units for latencies and jitter is milliseconds
- Use the suffix 'us' for micro-second precision.
- The interval duration is in milliseconds.

### ScriptAtten

- Private data syntax:

```
run_duration attenuations
```

- `run_duration` is in milliseconds
- `attenuations` is a comma-separated range.

Use `NA` for no changes to existing config, and use `NONE` if you want the value to be blank.

| Argument               | Description   |
|------------------------|---|
| 1. <i>endp</i>         | Endpoint, Test Group or Attenuator name or ID. [R]          |
| 2. <i>name</i>         | Script name. [W]  |
| 3. <i>flags</i>        | See above for description of the defined flags.             |
| 4. <i>type</i>         | One of: NONE, Script2544, ScriptHunt, ScriptWL, ScriptAtten |
| 5. <i>private</i>      | Private encoding for the particular script.                 |
| 6. <i>group_action</i> | How to handle group script operations: ALL, Sequential      |
| 7. <i>loop_count</i>   | How many times to loop before stopping (0 is infinite).     |

Syntax: `set_script endp name flags type private group_action loop_count`

### 160. set\_test\_id

Set the test ID on specified resource(s). Currently this is only used by the Android app, and other resources will ignore the request. Test-ID can be up to 15 characters in length.

| Argument           | Description                                       |
|--------------------|---|
| 1. <i>shelf</i>    | Name of the Shelf, or <code>all</code> . [R][D:1] |
| 2. <i>resource</i> | Number of the Resource, or <code>all</code> . [W] |
| 3. <i>test_id</i>  | Up to 15 character identifier.                    |

Syntax: `set_test_id shelf resource test_id`

### 161. rpt\_script

| Argument               | Description   |
|------------------------|---|
| 1. <i>endp</i>         | Endpoint name or ID. [W]                                |
| 2. <i>name</i>         | Script name. [W]  |
| 3. <i>flags</i>        | See above for description of the defined flags.         |
| 4. <i>type</i>         | One of: NONE, Script2544, ScriptHunt, ScriptWL          |
| 5. <i>private</i>      | Private encoding for the particular script.             |
| 6. <i>group_action</i> | All or Sequential.                                      |
| 7. <i>loop_count</i>   | How many times to loop before stopping (0 is infinite). |

Syntax: `rpt_script endp name flags type private group_action loop_count`

## 162. **add\_threshold**

Add or modify a threshold-alert for a particular endpoint. Threshold Types are defined as:

```
TX_BPS_RATE_OOR_3S | 0 # tx-bps over last 3 seconds is out of range.
RX_BPS_RATE_OOR_3S | 1 # rx-bps over last 3 seconds is out of range.
TX_BPS_RATE_OOR_30S | 2 # tx-bps over last 30 seconds is out of range.
RX_BPS_RATE_OOR_30S | 3 # rx-bps over last 30 seconds is out of range.
TX_BPS_RATE_OOR_1m | 4 # tx-bps over last 1 minute is out of range.
RX_BPS_RATE_OOR_1m | 5 # rx-bps over last 1 minute is out of range.
NO_RX_SINCE | 6 # Have not received any bytes/packets in specified time.
TT_RX_LAT_OOR | 7 # Latency running-average out of range.
TT_RX_DROP_OOR | 8 # RX Drop percentage is out of range (per-million).
```

Use `NA` for no changes to existing config, and use `NONE` if you want the value to be blank.

Special `thresh_id` values to help with flushing entire list of thresholds to remote:

```
Mark All | -2 # Mark all
Delete Marked | -3 # Delete any marked.
```

Setting a threshold will clear the mark.

| Argument              | Description  |
|-----------------------|--|
| 1. <i>endp</i>        | Endpoint name or ID. [R]   |
| 2. <i>thresh_id</i>   | Threshold ID. If adding new threshold, use -1, otherwise use correct ID. [W] |
| 3. <i>thresh_type</i> | Threshold type, integer, (see above).  |
| 4. <i>thresh_min</i>  | Minimum acceptable value for this threshold.                                 |
| 5. <i>thresh_max</i>  | Maximum acceptable value for this threshold.                                 |

Syntax: `add_threshold endp thresh_id thresh_type thresh_min thresh_max`

## 163. **set\_wifi\_radio**

Modify a WIFI Radio Interface (such as `phy0` or `wiphy0`). This command requires that the designated machine support the LANforge wifi driver for the Atheros brand WIFI NICs. The radio interface holds common configuration for the Virtual WIFI interfaces. `NA` can be used for any values that you do not wish to modify.

**NOTE:** this command is also used for radios dedicated for *radar emulation*. For adjusting vAPs, only **channel**, **NSS**, and **txpower** are commonly used.

To set any option to default (or un-set value, use `DEFAULT`. You may have to reboot the system to have the defaults take affect.

Mode options are below:

| Input   | : Enum | Val | : Shown by | nc_show_ports |
|---------|--------|-----|------------|---------------|
| AUTO    |        | 0   | #          | 802.11g       |
| 802.11a |        | 1   | #          | 802.11a       |
| b       |        | 2   | #          | 802.11b       |
| g       |        | 3   | #          | 802.11g       |
| abg     |        | 4   | #          | 802.11abg     |
| abgn    |        | 5   | #          | 802.11abgn    |
| bgn     |        | 6   | #          | 802.11bgn     |
| bg      |        | 7   | #          | 802.11bg      |
| abgnAC  |        | 8   | #          | 802.11abgn-AC |
| anAC    |        | 9   | #          | 802.11an-AC   |



|        |    |  |
|--------|----|--|
| an     | 10 | # 802.11an                                 |
| bgnAC  | 11 | # 802.11bgn-AC                             |
| abgnAX | 12 | # 802.11abgn-AX                            |
|        |    | # a/b/g/n/AC/AX (dual-band AX) support     |
| bgnAX  | 13 | # 802.11bgn-AX                             |
| anAX   | 14 | # 802.11an-AX                              |
| aAX    | 15 | # 802.11a-AX (6E disables /n and /ac)      |
| abgn7  | 16 | # 802.11abgn-EHT                           |
|        |    | # a/b/g/n/AC/AX/EHT (dual-band AX) support |
| bgn7   | 17 | # 802.11bgn-EHT                            |
| an7    | 18 | # 802.11an-EHT                             |
| a7     | 19 | # 802.11a-EHT (6E disables /n and /ac)     |

Antenna settings determine number of active antennae:

|               |   |
|---------------|---|
| Diversity/All | 0 |
| Fixed-A (1x1) | 1 |
| AB (2x2)      | 4 |
| ABC (3x3)     | 7 |
| ABCD (4x4)    | 8 |
| 8x8 (8x8)     | 9 |

Flags are currently defined as:

|                |            |  |
|----------------|------------|--|
| hw_sim         | 0x1        | # Create hw-sim virtual radio if radio does not already exist.                   |
| no_scan_share  | 0x00040    | # Disable sharing scan results.  |
| verbose        | 0x10000    | # Verbose-Debug: Increase debug info in wpa-supPLICANT and hostapd logs.         |
| no_sw_crypt    | 0x20000    | # Disable software-crypt for this radio. Disables some virtual-station features. |
| ct-sta-mode    | 0x40000    | # Enable CT-STA mode if radio supports it. Efficiently replaces sw-crypt in some |
| firmware_cfg   | 0x80000    | # Apply firmware config.   |
| ignore_radar   | 0x200000   | # Ignore RADAR events reported by firmware.                                      |
| allow_all_mcs  | 0x400000   | # Enable MCS otherwise disabled by firmware (ath10k only).                       |
| no_runtime_pm  | 0x800000   | # Disable runtime deep sleep mode (mtk7921k only at current)                     |
| extra_txstatus | 0x1000000  | # Enable increased packet tx-stats. May decrease performance. MTK radios only.   |
| extra_rxstatus | 0x2000000  | # Enable increased packet rx-stats. May decrease performance. MTK 7915 radio on. |
| ofdma_stats    | 0x4000000  | # Enable increased OFDMA statistics. May decrease performance. MTK 7915 radios   |
| use_syslog     | 0x20000000 | # Put supplicant logs in syslog instead of a file.                               |

The `firmware_cfg` flag is not saved in LANforge databases, so if you are reloading databases, you may have to manually re-apply the firmware settings. The config data for the last apply is stored on local disk and used by the driver when it loads on bootup.

`const_tx`: This is only supported on carl9170 adapters with modified firmware. Contact your supplier if you want more info on this feature.

### Related Commands

|                          |   |
|--------------------------|---|
| <code>max_amsdu</code> : | Number of frames for ath10k radios, but for ax200/ax210 radios, the values are: |
| 100:                     | Default (4k for ax200 in current driver)  |
| 104:                     | 2Kb   |
| 101:                     | 4Kb   |
| 102:                     | 8Kb   |
| 103:                     | 12Kb  |

| Argument              | Description  |
|-----------------------|--|
| 1. <i>shelf</i>       | Shelf number. [R][D:1]   |
| 2. <i>resource</i>    | Resource number. [W]   |
| 3. <i>radio</i>       | Name of the physical radio interface, for example: wiphy0 [W]  |
| 4. <i>mode</i>        | RF Pattern Generator: WiFi mode for radar emulation, see table. Do not use for vAPs.   |
| 5. <i>channel</i>     | Channel number for this radio device. Frequency takes precedence if both are set to non-default values. 0xFFFF, AUTO or DEFAULT means ANY. |
| 6. <i>country</i>     | Country number for this radio device.  |
| 7. <i>frequency</i>   | Frequency for this radio. 0xFFFF, AUTO or DEFAULT means ANY.   |
| 8. <i>frag_thresh</i> | Fragmentation Threshold (256 - 2346, 2346 == disabled).  |
| 9. <i>rate</i>        | No longer used, specify the rate on the virtual station(s) instead.  |
| 10. <i>rts</i>        | The RTS Threshold for this radio (off, or 1-2347).   |

|                               |   |
|-------------------------------|---|
| 11. <i>txpower</i>            | The transmit power setting for this radio. (AUTO for system defaults)                                 |
| 12. <i>mac</i>                | Used to identify when name cannot be trusted (2.6.34+ kernels).                                       |
| 13. <i>antenna</i>            | Antenna configuration: 0 Diversity/All, 1 Fixed-A (1x1), 4 AB (2x2), 7 ABC (3x3), 8 ABCD (4x4), 9 8x8 |
| 14. <i>flags</i>              | Flags for this interface (see above.)   |
| 15. <i>flags_mask</i>         | If set, only these flags will be considered.  |
| 16. <i>const_tx</i>           | RF Pattern Generator: encoded as a single 32-bit integer. See above.                                  |
| 17. <i>pulse_width</i>        | RF Pattern generator: pulse width in usecs.   |
| 18. <i>pulse_interval</i>     | RF Pattern generator: interval between pulses in usecs.   |
| 19. <i>vdev_count</i>         | Configure radio vdev count.   |
| 20. <i>peer_count</i>         | Number of peer objects for this radio.  |
| 21. <i>stations_count</i>     | Number of stations supported by this radio.   |
| 22. <i>rate_ctrl_count</i>    | Number of rate-ctrl objects for this radio.   |
| 23. <i>fwname</i>             | Firmware name (for example: firmware-5.bin)   |
| 24. <i>fwver</i>              | Firmware API version (for example, 5 if firmware is based on firmware-5.bin)                          |
| 25. <i>txdesc_count</i>       | Transmit descriptor count for this radio.   |
| 26. <i>tids_count</i>         | TIDs count for this radio.  |
| 27. <i>skid_limit</i>         | Firmware hash-table Skid Limit for this radio.  |
| 28. <i>active_peer_count</i>  | Number of locally-cached peer objects for this radio.   |
| 29. <i>tx_pulses</i>          | Number of pattern pulses per burst for RF noise generator.  |
| 30. <i>pulse2_interval_us</i> | Pause between pattern burst for RF noise generator.   |
| 31. <i>max_amsdu</i>          | Maximum number of frames per AMSDU that may be transmitted. See above.                                |
| 32. <i>pref_ap</i>            | Preferred AP BSSID for all station vdevs on this radio.   |
| 33. <i>ampdu_factor</i>       | ax200/ax210 only, currently. Requires module reload. OS Default: 0xFF                                 |

Syntax: `set_wifi_radio shelf resource radio mode channel country frequency frag_thresh rate rts txpower mac antenna flags flags_mask const_tx pulse_width pulse_interval vdev_count peer_count stations_count rate_ctrl_count fwname fwver txdesc_count tids_count skid_limit active_peer_count tx_pulses pulse2_interval_us max_amsdu pref_ap ampdu_factor`

#### 164. **set\_wifi\_extra**

This configures WiFi ports with advanced features. Not all combinations are valid..contact support and/or see wpa\_supplicant & hostapd configuration documentation for details. Most values will default to sane values if left blank. To clear a text value, set it to '[BLANK]'

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]  |
| 2. <i>resource</i> | Resource number. [W]  |
| 3. <i>port</i>     | WiFi interface name or number. [W]  |
| 4. <i>key_mgmt</i> | Key management: WPA-PSK, WPA-EAP, IEEE8021X, NONE, WPA-PSK-SHA256, WPA-EAP-SHA256 or combo. |
| 5. <i>pairwise</i> | Pairwise ciphers: CCMP, TKIP, NONE, or combination.   |

|                               |  |
|-------------------------------|--|
| 6. <i>group</i>               | Group cyphers: CCMP, TKIP, WEP104, WEP40, or combination.  |
| 7. <i>psk</i>                 | WPA(2) pre-shared key. If unsure, use this field for any password entry. Prepend with 0x for ascii-hex representation. |
| 8. <i>key</i>                 | WEP key0. This should be entered in ascii-hex. Use this only for WEP.  |
| 9. <i>ca_cert</i>             | CA-CERT file name.   |
| 10. <i>eap</i>                | EAP method: MD5, MSCHAPV2, OTP, GTC, TLS, PEAP, TTLS.  |
| 11. <i>identity</i>           | EAP Identity string.   |
| 12. <i>anonymous_identity</i> | Anonymous identity string for EAP.   |
| 13. <i>phase1</i>             | Outer-authentication, ie TLS tunnel parameters.  |
| 14. <i>phase2</i>             | Inner authentication with TLS tunnel.  |
| 15. <i>password</i>           | EAP Password string.   |
| 16. <i>pin</i>                | EAP-SIM pin string. (For AP, this field is HS20 Operating Class)   |
| 17. <i>pac_file</i>           | EAP-FAST PAC-File name. (For AP, this field is the RADIUS secret password)   |
| 18. <i>private_key</i>        | EAP private key certificate file name. (For AP, this field is HS20 WAN Metrics)  |
| 19. <i>pk_passwd</i>          | EAP private key password. (For AP, this field is HS20 connection capability)   |
| 20. <i>hessid</i>             | 802.11u HESSID (MAC address format) (or peer for WDS stations).  |
| 21. <i>realm</i>              | 802.11u realm: mytelco.com   |
| 22. <i>client_cert</i>        | 802.11u Client cert file: /etc/wpa_supplicant/ca.pem   |
| 23. <i>imsi</i>               | 802.11u IMSI: 310026-000000000   |
| 24. <i>milenage</i>           | 802.11u milenage:<br>90dca4eda45b53cf0f12d7c9c3bc6a89:cb9cccc4b9258e6dca4760379fb82                                    |
| 25. <i>domain</i>             | 802.11u domain: mytelco.com  |
| 26. <i>roaming_consortium</i> | 802.11u roaming consortium: 223344 (15 characters max)   |
| 27. <i>venue_group</i>        | 802.11u Venue Group, integer. VAP only.  |
| 28. <i>venue_type</i>         | 802.11u Venue Type, integer. VAP only.   |
| 29. <i>network_type</i>       | 802.11u network type, integer, VAP only.   |
| 30. <i>ipaddr_type_avail</i>  | 802.11u network type available, integer, VAP only.   |
| 31. <i>network_auth_type</i>  | 802.11u network authentication type, VAP only.   |
| 32. <i>anqp_3gpp_cell_net</i> | 802.11u 3GCPP Cellular Network Info, VAP only.   |

Syntax: `set_wifi_extra shelf resource port key_mgmt pairwise group psk key ca_cert eap identity anonymous_identity phase1 phase2 password pin pac_file private_key pk_passwd hessid realm client_cert imsi milenage domain roaming_consortium venue_group venue_type network_type ipaddr_type_avail network_auth_type anqp_3gpp_cell_net`

#### 165. **set\_wifi\_extra2**

This configures WiFi ports with advanced features. Not all combinations are valid. Contact support and/or see wpa\_supplicant & hostapd configuration documentation for details. Most values will default to sane values if left blank. To clear a text value, set it to [BLANK].

freq\_24 and freq\_5 are used to configure a subset of available channels that can be used. See add\_venue for syntax definition.

For stations, the behaviour is thus:

- If the parent radio has a VAP, then the configured frequency for the radio will be used.
- Else if the user has configured freq\_24 or freq\_5, that will be used.
- However, if the mode specifies a frequency range (ie /b or /g), then frequencies outside of the selected band will still not be allowed.

| Argument                        | Description  |
|---------------------------------|--|
| 1. <i>shelf</i>                 | Shelf number. [R][D:1]   |
| 2. <i>resource</i>              | Resource number. [W]   |
| 3. <i>port</i>                  | WiFi interface name or number. [W]   |
| 4. <i>req_flush</i>             | Set to 1 if you wish to flush changes to kernel now.   |
| 5. <i>ignore_probe</i>          | Per-million: AP ignore probe percentage.   |
| 6. <i>ignore_auth</i>           | Per-million: AP ignore auth request percentage.  |
| 7. <i>ignore_assoc</i>          | Per-million: AP ignore assoc request percentage.   |
| 8. <i>ignore_reassoc</i>        | Per-million: AP ignore re-assoc request percentage.  |
| 9. <i>corrupt_gtk_rekey_mic</i> | Per-million: AP corrupts GTK Rekey MIC.  |
| 10. <i>radius_ip</i>            | RADIUS server IP Address (AP Only)   |
| 11. <i>radius_port</i>          | RADIUS server IP Port (AP Only)  |
| 12. <i>freq_24</i>              | Frequency list for 2.4Ghz band, see above.   |
| 13. <i>freq_5</i>               | Frequency list for 5Ghz band, see above.   |
| 14. <i>post_ifup_script</i>     | Script name with optional args, will run after interface comes up and gets IP.                         |
| 15. <i>ocsp</i>                 | OCSP settings: 0=disabled, 1=try, but to not require response, 2=require valid OCSP stapling response. |
| 16. <i>venue_id</i>             | Venue-ID for this wifi device. VAP in same venue will share neigh reports as appropriate.              |
| 17. <i>sae_pwe</i>              | Set SAE-PWE, 0 == hunting-and-pecking, 1 == hash-to-element, 2 allow both.                             |
| 18. <i>initial_band_pref</i>    | Initially connect on this band, if available in scan. 0=ignore, 2=2ghz, 5=5ghz, 6=6ghz.                |

Syntax: `set_wifi_extra2 shelf resource port req_flush ignore_probe ignore_auth ignore_assoc ignore_reassoc corrupt_gtk_rekey_mic radius_ip radius_port freq_24 freq_5 post_ifup_script ocsp venue_id sae_pwe initial_band_pref`

#### 166. **set\_wifi\_txo**

This allows one to configure a VAP or Station wifi device to override the normal rate-control and send Data and QOS Data frames at the exact rates specified. This may only work on certain radios/firmware. It can be used to do packet-error testing and other testing where controlling the TX rate is important. `txo_txpower` notes: For ath10k wifi-5 radios, this is a power setting, for MTK radios, this is an adjustment from default, where 8 is default, less than 8 is reducing power, and more than 8 is increasing power above default.

| Argument              | Description   |
|-----------------------|---|
| 1. <i>shelf</i>       | Shelf number. [R][D:1]  |
| 2. <i>resource</i>    | Resource number. [W]  |
| 3. <i>port</i>        | WiFi interface name or number. [W]  |
| 4. <i>txo_enable</i>  | Set to 1 if you wish to enable transmit override, 0 to disable.                     |
| 5. <i>txo_txpower</i> | Configure TX power in db. Use 255 for system defaults. See notes above.             |
| 6. <i>txo_pream</i>   | Select rate preamble: 0 == OFDM, 1 == CCK, 2 == HT, 3 == VHT, 4 == HE_SU.           |
| 7. <i>txo_mcs</i>     | Configure the MCS (0-3 for CCK, 0-7 for OFDM, 0-7 for HT, 0-9 for VHT, 0-11 for HE) |
| 8. <i>txo_nss</i>     | Configure number of spatial streams (0 == nss1, 1 == nss2, ...).                    |

|                        |   |
|------------------------|---|
| 9. <i>txo_bw</i>       | Configure bandwidth: 0 == 20, 1 == 40, 2 == 80, 3 == 160, 4 == 80+80. |
| 10. <i>txo_retries</i> | Configure number of retries. 0 or 1 means no retries).                |
| 11. <i>txo_sgi</i>     | Should rates be sent with short-guard-interval or not?                |

Syntax: `set_wifi_txo shelf resource port txo_enable txo_txpower txo_pream txo_mcs txo_nss txo_bw txo_retries txo_sgi`

### 167. **set\_wifi\_corruptions**

This lets one configure a station or AP to purposely corrupt, delay, and drop various management frames. To disable a corruption, set it to 0. To have corruption always happen, set to maximum value (1000000).

For the delay options, this will effectively delay the response by a random number of milliseconds between the configured min and max.

This command is primarily for WiFi stations at this time. For AP devices, see the `set_wifi_extra2` command.

To specify which packet types are to be affected, set the **Corrupt Flags** accordingly:

```
MSG_TYPE_EAPOL | 0x0001 # Any EAPOL message
MSG_TYPE_DEAUTH | 0x0002 # de-authentication message
MSG_TYPE_EAPOL_1_OF_4 | 0x0004 # EAPOL message 1/4
MSG_TYPE_EAPOL_2_OF_4 | 0x0008 # EAPOL message 2/4
MSG_TYPE_EAPOL_3_OF_4 | 0x0010 # EAPOL message 3/4
MSG_TYPE_EAPOL_4_OF_4 | 0x0020 # EAPOL message 4/4
MSG_TYPE_EAPOL_1_OF_2 | 0x0040 # EAPOL message 1/2
MSG_TYPE_EAPOL_2_OF_2 | 0x0080 # EAPOL message 2/2
MSG_TYPE_EAPOL_KEY_REQ | 0x0100 # EAP Key Request (not sure if this works properly)
MSG_TYPE_EAPOL_ASSOC | 0x0200 # EAP Association
MST_TYPE_EAPOL_ID_REQ | 0x0400 # EAP Identity request
MST_TYPE_EAPOL_ID_RESP | 0x0800 # EAP Identity response
MST_TYPE_EAPOL_OTHER_REQ | 0x1000 # EAP Requests that do not match other things.
MST_TYPE_EAPOL_OTHER_RESP | 0x2000 # EAP Responses that do not match other things.
```

| Argument                  | Description   |
|---------------------------|---|
| 1. <i>shelf</i>           | Shelf number. [R][D:1]  |
| 2. <i>resource</i>        | Resource number. [W]  |
| 3. <i>port</i>            | WiFi interface name or number. [W]  |
| 4. <i>req_flush</i>       | Set to 1 if you wish to flush changes to kernel now.                            |
| 5. <i>ignore_per_mil</i>  | Per-million: Station to randomly ignore selected message types by this amount.  |
| 6. <i>ignore_flags</i>    | Specify packet types to ignore (see flags above).                               |
| 7. <i>corrupt_per_mil</i> | Per-million: Station to randomly corrupt selected message types by this amount. |
| 8. <i>corrupt_flags</i>   | Specify packet types to corrupt (see flags above).                              |
| 9. <i>delay_min</i>       | milliseconds: Station to randomly delay processing received messages, min time  |
| 10. <i>delay_max</i>      | milliseconds: Station to randomly delay processing received messages, max time  |
| 11. <i>delay_flags</i>    | Specify packet types to delay (see flags above).                                |
| 12. <i>dup_flags</i>      | Specify packet types to duplicate (see flags above).                            |
| 13. <i>dup_per_65535</i>  | Percentage, represented as x per 65535 of packets we should duplicate.          |

Syntax: `set_wifi_corruptions shelf resource port req_flush ignore_per_mil ignore_flags corrupt_per_mil corrupt_flags delay_min delay_max delay_flags dup_flags dup_per_65535`

### 168. **set\_wifi\_custom**

This text will be added to the end of the `hostapd` config file for virtual APs, and to the `wpa_supplicant` config file for virtual stations. This can be used for experimental work and for cases where LANforge does not support all of the desired features through normal means. The text must be entered one line at a time, primarily due to CLI parsing limitations. NOTE: You have to

manually reset the interface to have the new changes take effect.

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]  |
| 2. <i>resource</i> | Resource number. [W]  |
| 3. <i>port</i>     | WiFi interface name or number. [W]  |
| 4. <i>type</i>     | NA for now, may specify specific locations later. [D:NA]                                      |
| 5. <i>text</i>     | [BLANK] will erase all, any other text will be appended to existing text. Unescaped Value [W] |

Syntax: `set_wifi_custom shelf resource port type text`

#### 169. **set\_ifup\_script**

Set the IF-UP script for a port. The `post_ifup_script` argument does not need to use single quotes, since all tokens after the port-id will be considered part of the script variable. This script needs to be on the LANforge machine of the resource number. While the default directory is often `/home/lanforge`, it is worth setting this in case you are on non-standard hardware.

| Argument                   | Description  |
|----------------------------|--|
| 1. <i>shelf</i>            | Shelf number. [R][D:1]   |
| 2. <i>resource</i>         | Resource number. [W]   |
| 3. <i>port</i>             | WiFi interface name or number. [W]   |
| 4. <i>flags</i>            | Currently un-defined, use NA   |
| 5. <i>post_ifup_script</i> | Script name with optional args, will run after interface comes up and gets IP. Use [BLANK] to clear. Unescaped Value [W] |

Syntax: `set_ifup_script shelf resource port flags post_ifup_script`

#### 170. **set\_endp\_addr**

Set the MAC, IP, and Port addresses for an UN\_MANAGED endpoint. The endpoint must be created as UN\_MANAGED, and you must set its addresses before you can start it. The syntax for addresses is:

- MAC addresses is: `01:BB:CC:DD:EE:FF`.
- IP addresses should be entered in dot notation, eg: `172.4.1.1`.
- and port is the IP port (1-65534).

#### Related Commands

| Argument           | Description   |
|--------------------|---|
| 1. <i>name</i>     | The name of the endpoint we are configuring. [R]              |
| 2. <i>mac</i>      | The MAC address. Only needed for LANforge protocol Endpoints. |
| 3. <i>ip</i>       | The IP Address. Used for TCP/IP and UDP/IP protocols.         |
| 4. <i>min_port</i> | The Minimum IP Port. Used for TCP/IP and UDP/IP protocols.    |
| 5. <i>max_port</i> | The Maximum IP Port. Used for TCP/IP and UDP/IP protocols.    |

Syntax: `set_endp_addr name mac ip min_port max_port`

#### 171. **set\_endp\_payload**

Set the payload type, and potentially the payload for a particular Endpoint. To enter an actual payload, use space separated Hexadecimal. For example: `00 00 01 04 bb de ad be ef`. The payload must be entered all at once on one line. The payload cannot be longer than 2048 bytes (though when represented as ASCII HEX, the actual input can be longer than that.)

Possible values for **payload type**:

```
increasing    |# bytes start at 00 and increase, wrapping if needed.
decreasing    |# bytes start at FF and decrease, wrapping if needed.
```

```

random      |# generate a new random payload each time sent.
random_fixed|# means generate one random payload, and send it over and over again.
zeros      |# Payload is all zeros (00).
ones       |# Payload is all ones (FF).

PRBS_4_0_3 |# Use linear feedback shift register to generate pseudo random sequence.
           |# First number is bit-length of register, second two are TAPS (zero-based indexes)
           |# Seed value is always 1.

PRBS_7_0_6 |# PRBS (see above)
PRBS_11_8_10|# PRBS (see above)
PRBS_15_0_14|# PRBS (see above)
custom     |# Enter your own payload with the set_endp_payload

```

### Related Commands

| Argument               | Description  |
|------------------------|--|
| 1. <i>name</i>         | The name of the endpoint we are configuring. [R]                                 |
| 2. <i>payload_type</i> | The payload type. See help for add_endp. [W]<br>[D:increasing]                   |
| 3. <i>payload</i>      | For custom payloads, enter the payload in hex, up to 2048 bytes. Unescaped Value |

Syntax: `set_endp_payload name payload_type payload`

### 172. set\_endp\_details

Modify TCP window sizes. The `rcvbuf_size` will be passed to:

```
setsockopt(desc, SOL_SOCKET, SO_RCVBUF, &size, sizeof(size))
```

and the `sndbuf` will be set similarly:

```
setsockopt(desc, SOL_SOCKET, SO_SNDBUF, &size, sizeof(size))
```

See the socket man page: `man socket` for more detailed information about what this means.

`conn_timer` is used to create TCP connections of short duration. If this is set to some value other than `0xFFFFFFFF`, then the connection will be closed and reopened at that duration. Set to a low value for testing firewalls and devices that are interested in connections-per-second.

`dst_mac` is used for custom-ethernet endpoints that are replaying pkts and my want to re-write the DST MAC as we replay.

### Related Commands

| Argument                   | Description   |
|----------------------------|---|
| 1. <i>name</i>             | The name of the endpoint we are configuring. [R]  |
| 2. <i>rcvbuf_size</i>      | The receive buffer (window) size. Zero for AUTO   |
| 3. <i>sndbuf_size</i>      | The sending buffer (window) size. Zero for AUTO   |
| 4. <i>min_conn_timer</i>   | The minimum duration (in ms) this connection should run before re-establishing.   |
| 5. <i>pkts_to_send</i>     | Number of packets to send before stopping. 0 means infinite.  |
| 6. <i>dst_mac</i>          | Destination MAC address, used for custom Ethernet replays.  |
| 7. <i>max_conn_timer</i>   | The maximum duration (in ms) this connection should run before re-establishing.   |
| 8. <i>min_reconn_pause</i> | The minimum time between re-connects, in ms.  |
| 9. <i>max_reconn_pause</i> | The maximum time between re-connects, in ms.  |
| 10. <i>max_ip_port</i>     | The maximum IP Port value. (The value for min ip port is set through the add_endp/ip_port parameter.) If greater than min, each connection will use a random value between min and max. |
| 11. <i>conn_timeout</i>    | For TCP, the max time in milliseconds to wait for connection to establish.  |

|                            |   |
|----------------------------|---|
| 12. <i>tcp_mss</i>         | TCP Maximum Segment Size, affects packet size on the wire (88 - 32767). |
| 13. <i>tcp_min_delack</i>  | NA: No longer supported.  |
| 14. <i>tcp_max_delack</i>  | NA: No longer supported.  |
| 15. <i>tcp_delack_segs</i> | NA: No longer supported.  |
| 16. <i>mcast_src_ip</i>    | Multicast source address (used in SSM mode, multicast endpoints only)   |
| 17. <i>mcast_src_port</i>  | Multicast source address (used in SSM mode, multicast endpoints only)   |

Syntax: `set_endp_details name rcvbuf_size sndbuf_size min_conn_timer pkts_to_send dst_mac max_conn_timer min_reconn_pause max_reconn_pause max_ip_port conn_timeout tcp_mss tcp_min_delack tcp_max_delack tcp_delack_segs mcast_src_ip mcast_src_port`

### 173. `set_event_interest`

Set event interest. If flags and val1 are left blank, then the current settings will be displayed.

**ei\_flags:**

```
CLEAR | 0 # will clear interest
SET | 0x1 # set interest flag
```

**events1 values:**

```
Link-Down | 0x000001 # Notify when Interface Link goes DOWN.
Link-Up | 0x000002 # Notify when Interface Link goes UP.
Custom | 0x000004 # Custom event (generated by USER in GUI or CLI).
Resource-Down | 0x000008 # Resource has crashed, rebooted, etc.
Resource-Up | 0x000010 # Resource has connected to manager.
Endp-Stopped | 0x000020 # Endpoint stopped for some reason.
Endp-Started | 0x000040 # Endpoint was started.
Disconnect | 0x000080 # WiFi interface disconnected from AP.
Connect | 0x000100 # WiFi interface connected to AP.
Logout | 0x000200 # CLI/GUI user disconnected from LANforge.
Login | 0x000400 # CLI/GUI user connected to LANforge.
Stop-Reports | 0x000800 # Stop saving report data files (CSV).
Start-Reports | 0x001000 # Start saving report data files (CSV).
Cleared | 0x002000 # Counters were cleared for some entity.
Link-Errors | 0x004000 # Port shows low-level link errors.
DHCP-Fail | 0x008000 # DHCP Failed, maybe out of leases?
DHCP-Timeout | 0x010000 # Timed out talking to DHCP server.
DHCP4-Error | 0x020000 # DHCP gave out duplicated IP address.
DHCP6-Error | 0x040000 # DHCPv6 gave out duplicated IPv6 address.
WiFi-Config | 0x080000 # WiFi Configuration Error.
Bad-MAC | 0x100000 # Invalid MAC address configured.
Migrated | 0x200000 # Port (station network interface) migrated.
BAD-TOS | 0x400000 # Endpoint has bad ToS values configured.
NO-RX-SINCE | 0x800000 # Endpoint threshold alert.
NO-RX-SINCE-CLEARED | 0x1000000 # Endpoint threshold alert cleared.
RX-BPS-OOR-3S | 0x2000000 # Endpoint threshold alert.
RX-BPS-OOR-3S-CLEARED | 0x4000000 # Endpoint threshold alert cleared.
RX-BPS-OOR-30S | 0x8000000 # Endpoint threshold alert.
RX-BPS-OOR-30S-CLEARED | 0x10000000 # Endpoint threshold alert cleared.
RX-BPS-OOR-1M | 0x20000000 # Endpoint threshold alert.
RX-BPS-OOR-1M-CLEARED | 0x40000000 # Endpoint threshold alert cleared.
TX-BPS-OOR-3S | 0x80000000 # Endpoint threshold alert.
```

**events2 values:**

```
TX-BPS-OOR-3S-CLEARED | 0x1 # Endpoint threshold alert cleared.
TX-BPS-OOR-30S | 0x2 # Endpoint threshold alert.
TX-BPS-OOR-30S-CLEARED | 0x4 # Endpoint threshold alert cleared.
TX-BPS-OOR-1M | 0x8 # Endpoint threshold alert.
TX-BPS-OOR-1M-CLEARED | 0x10 # Endpoint threshold alert cleared.
RX-LAT-OOR | 0x20 # Endpoint threshold alert.
RX-LAT-OOR-CLEARED | 0x40 # Endpoint threshold alert cleared.
RX-DROP-OOR-3S | 0x80 # Endpoint threshold alert.
RX-DROP-OOR-3S-CLEARED | 0x100 # Endpoint threshold alert cleared.
RX-DROP-OOR-1M | 0x200 # Endpoint threshold alert.
RX-DROP-OOR-1M-CLEARED | 0x400 # Endpoint threshold alert cleared.
FW-CRASH | 0x800 # Firmware for entity has crashed.
FW-FAIL | 0x1000 # Firmware failed powerup, may require reboot.
IFUP-FAIL | 0x2000 # IFUP-POST Script returned error code.
IFUP-OK | 0x4000 # IFUP-POST Script completed successfully.
IFDOWN-FAIL | 0x8000 # IFDOWN-PRE Script (ifup --logout) returned error code.
IFDOWN-OK | 0x10000 # IFDOWN-PRE Script (ifup --logout) completed successfully.
```

**events3-4** are currently un-used.

**Var1:** Currently un-defined.



| Argument            | Description                          |
|---------------------|--------------------------------------|
| 1. <i>ei_flags</i>  | Event Interest flags, see above. [W] |
| 2. <i>events1</i>   | See description for possible values. |
| 3. <i>events2</i>   | See description for possible values. |
| 4. <i>events3</i>   | See description for possible values. |
| 5. <i>events4</i>   | See description for possible values. |
| 6. <i>var1</i>      | Currently un-used.                   |
| 7. <i>event_cnt</i> | Maximum number of events to store.   |

Syntax: `set_event_interest ei_flags events1 events2 events3 events4 var1 event_cnt`

#### 174. **set\_event\_priority**

Set event priority. If flag an priority are left blank, then the current settings will be displayed.  
Events:

```
Link-Down | 0 # Notify when Interface Link goes UP.
Link-Up | 1 # Notify when Interface Link goes DOWN.
Custom | 2 # Custom event (generated by USER in GUI or CLI).
Resource-Down | 3 # Resource has crashed, rebooted, etc.
Resource-Up | 4 # Resource has connected to manager.
Endp-Stopped | 5 # Endpoint stopped for some reason.
Endp-Started | 6 # Endpoint was started.
Disconnect | 7 # WiFi interface disconnected from AP.
Connect | 8 # WiFi interface connected to AP.
Logout | 9 # CLI/GUI user disconnected from LANforge.
Login | 10 # CLI/GUI user connected to LANforge.
Stop-Reports | 11 # Stop saving report data files (CSV).
Start-Reports | 12 # Start saving report data files (CSV).
Cleared | 13 # Counters were cleared for some entity.
Link-Errors | 14 # Port shows low-level link errors.
DHCP-Fail | 15 # DHCP Failed, maybe out of leases?
DHCP-Timeout | 16 # Timed out talking to DHCP server.
DHCP4-Error | 17 # DHCP gave out duplicated IP address.
DHCP6-Error | 18 # DHCPv6 gave out duplicated IPv6 address.
WiFi-Config | 19 # WiFi Configuration Error.
Bad-MAC | 20 # Invalid MAC address configured.
Migrated | 21 # Port (station network interface) migrated.
```

Priorities:

```
AUTO | 0 # Let event creator decide the priority.
DEBUG | 1 #
INFO | 2 #
WARNING | 3 #
CRITICAL | 4 #
FATAL | 5 #
```

| Argument           | Description                                       |
|--------------------|---|
| 1. <i>event</i>    | Number or name for the event, see above. [R,0-21] |
| 2. <i>priority</i> | Number or name for the priority. [R,0-5]          |

Syntax: `set_event_priority event priority`

#### 175. **set\_mc\_endp**

| Argument                  | Description   |
|---------------------------|---|
| 1. <i>name</i>            | The name of the endpoint we are configuring. [R]          |
| 2. <i>tfl</i>             | Time to live for the multicast packets generated.         |
| 3. <i>mcast-group</i>     | Multicast group IP, ie: 224.1.1.2 IPv6 supported as well. |
| 4. <i>mcast-dest-port</i> | Multicast destination IP Port, for example: 55000         |
| 5. <i>rcv_mcast</i>       | Should we attempt to receive? Values: Yes or No           |

Syntax: `set_mc_endp name tfl mcast-group mcast-dest-port rcv_mcast`

#### 176. **show\_adb**

Show one or all ADB (Android) devices. See 'discover' command for how to request discovery of

devices.

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number or alias, can be 'all'. [R][D:1]         |
| 2. <i>resource</i> | Resource number, or 'all'. [W]                        |
| 3. <i>serno</i>    | Serial number for requested ADB device, or 'all'. [W] |

Syntax: `show_adb shelf resource serno`

#### 177. **show\_chamber**

Show one or all Chambers. If the name is 'ALL', or no name is specified then all are shown, otherwise only the single requested Chamber is shown.

| Argument       | Description                       |
|----------------|-----------------------------------|
| 1. <i>name</i> | Chamber Name or 'ALL'. [W][D:ALL] |

Syntax: `show_chamber name`

#### 178. **show\_dut**

Show one or all Devices Under Test (DUT). If the name is 'ALL', or no name is specified then all are shown, otherwise only the single requested DUT is shown.

| Argument       | Description                   |
|----------------|-------------------------------|
| 1. <i>name</i> | DUT Name or 'ALL'. [W][D:ALL] |

Syntax: `show_dut name`

#### 179. **show\_events**

Show recent events of interest. To filter on certain events, specify the entity in question. Otherwise, use `all` or leave blank to match all events.

##### Event types:

|                 |   |
|-----------------|---|
| All             | # |
| Shelf           | # |
| Card            | # |
| Port            | # |
| Endp            | # |
| CX              | # |
| Test_Mgr        | # |
| Span            | # |
| Channel_Group   | # |
| PPP_Link        | # |
| PESQ            | # |
| CollisionDomain | # |

| Argument        | Description                                     |
|-----------------|---|
| 1. <i>type</i>  | Event type filter. [R]                          |
| 2. <i>shelf</i> | Event shelf filter.                             |
| 3. <i>card</i>  | Event resource filter.                          |
| 4. <i>port</i>  | Event port filter (can be port name or number). |
| 5. <i>endp</i>  | Event endpoint filter.                          |
| 6. <i>extra</i> | Extra filter, currently ignored.                |

Syntax: `show_events type shelf card port endp extra`

#### 180. **show\_alerts**

Show active Alerts of interest. To filter on certain alerts, specify the entity in question. Otherwise, use 'all' or leave blank to match all events.

##### Alert types:

|       |   |
|-------|---|
| All   | # |
| Shelf | # |
| Card  | # |
| Port  | # |

```

Endp          |#
CX            |#
Test_Mgr     |#
Span         |#
Channel_Group |#
PPP_Link     |#
PESQ        |#
CollisionDomain |#

```

| Argument        | Description                                     |
|-----------------|---|
| 1. <i>type</i>  | Alert type filter. [R]                          |
| 2. <i>shelf</i> | Alert shelf filter.                             |
| 3. <i>card</i>  | Alert resource filter.                          |
| 4. <i>port</i>  | Alert port filter (can be port name or number). |
| 5. <i>endp</i>  | Alert endpoint filter.                          |
| 6. <i>extra</i> | Extra filter, currently ignored.                |

Syntax: `show_alerts type shelf card port endp extra`

#### 181. **show\_event\_interest**

Display Event settings.

Syntax: `show_event_interest`

#### 182. **show\_err**

Send an error message to everyone else logged in to the server.

| Argument          | Description  |
|-------------------|--|
| 1. <i>message</i> | Message to show to others currently logged on. Unescaped Value [R] |

Syntax: `show_err message`

#### 183. **start\_endp**

Start and endpoint. This command is only valid for Multicast endpoints, which are NOT managed by a cross-connect like the rest of the endpoints. See Also: `set_cx_state`

##### Related Commands

| Argument            | Description                              |
|---------------------|--|
| 1. <i>endp_name</i> | Name of the cross-connect, or 'all'. [R] |

Syntax: `start_endp endp_name`

#### 184. **show\_profile**

Show one or all Device Profiles. If the name is 'ALL', or no name is specified then all are shown, otherwise only the single requested Profile is shown.

| Argument       | Description                |
|----------------|----------------------------|
| 1. <i>name</i> | Profile Name or 'ALL'. [R] |

Syntax: `show_profile name`

#### 185. **show\_text\_blob**

Show one or all Text Blobs. If the name is 'ALL', or no name is specified then all are shown, otherwise only the single requested blob is shown.

| Argument       | Description                  |
|----------------|------------------------------|
| 1. <i>type</i> | Text Blob type or 'ALL'. [R] |
| 2. <i>name</i> | Text Blob Name or 'ALL'. [R] |

|                 |   |
|-----------------|---|
| 3. <i>brief</i> | Set to 'brief' for a brief listing of all text blobs. |
|-----------------|---|

Syntax: `show_text_blob type name brief`

186. **show\_traffic\_profile**

Show one or all Traffic Profiles. If the name is 'ALL', or no name is specified then all are shown, otherwise only the single requested Profile is shown.

| Argument       | Description                |
|----------------|----------------------------|
| 1. <i>name</i> | Profile Name or 'ALL'. [R] |

Syntax: `show_traffic_profile name`

187. **start\_group**

Starts all cross-connects in a test group See Also: `add_group`, `add_tgcx`

**Related Commands**

| Argument       | Description                     |
|----------------|---------------------------------|
| 1. <i>name</i> | The name of the test group. [R] |

Syntax: `start_group name`

188. **start\_ppp\_link**

Start a PppLink.

| Argument           | Description                                    |
|--------------------|--|
| 1. <i>shelf</i>    | Name/id of the shelf. [R][D:1]                 |
| 2. <i>resource</i> | Resource number that holds this PppLink. [W]   |
| 3. <i>unit_num</i> | Unit-Number for the PppLink to be started. [R] |

Syntax: `start_ppp_link shelf resource unit_num`

189. **stop\_endp**

Stop an endpoint. This command is only valid for Multicast endpoints, which are NOT managed by a cross-connect like the rest of the endpoints. See Also: `set_cx_state`

| Argument            | Description                         |
|---------------------|-------------------------------------|
| 1. <i>endp_name</i> | Name of the endpoint, or 'all'. [R] |

Syntax: `stop_endp endp_name`

190. **quiesce\_endp**

Quiesce an endpoint. This command is only valid for Multicast endpoints, which are NOT managed by a cross-connect like the rest of the endpoints. See Also: `set_cx_state`

| Argument            | Description                         |
|---------------------|-------------------------------------|
| 1. <i>endp_name</i> | Name of the endpoint, or 'all'. [R] |

Syntax: `quiesce_endp endp_name`

191. **stop\_group**

Stops all cross-connects in one or all test groups See Also: `add_group`, `add_tgcx`, `start_group`

**Related Commands**

| Argument       | Description                              |
|----------------|--|
| 1. <i>name</i> | The name of the test group, or 'all' [R] |

Syntax: `stop_group name`

192. **quiesce\_group**

Quiesces all cross-connects one or all test groups See Also: `add_group`, `add_tgcx`, `stop_group`

### Related Commands

| Argument       | Description                              |
|----------------|--|
| 1. <i>name</i> | The name of the test group, or 'all' [R] |

Syntax: `quiesce_group name`

#### 193. **stop\_ppp\_link**

Stop a PppLink.

| Argument           | Description                                    |
|--------------------|--|
| 1. <i>shelf</i>    | Name/id of the shelf. [R][D:1]                 |
| 2. <i>resource</i> | Resource number that holds this PppLink. [W]   |
| 3. <i>unit_num</i> | Unit-Number for the PppLink to be stopped. [W] |

Syntax: `stop_ppp_link shelf resource unit_num`

#### 194. **set\_endp\_tos**

Set the IP Type of Service (TOS) byte for this Endpoint. Only valid for TCP/IP and UDP/IP based endpoint types. You should consult RFC-791, RFC-1349 and RFC-2474 for ideas of what this value can and should be.

RFC 1394 standard TOS settings can be entered by name:

```
LOWDELAY      |#  
THROUGHPUT   |#  
RELIABILITY   |#  
LOWCOST      |#
```

You may also instruct the Endpoint to NOT set any TOS with the TOS keyword: `DONT-SET`. This will make the Endpoint use the kernel defaults. If you have already set the TOS, then you must stop and restart the Endpoint to have the new default values take affect.

For **Priority**, please read the Linux socket man page: `man 7 socket`

| Argument           | Description                                      |
|--------------------|--|
| 1. <i>name</i>     | The name of the endpoint we are configuring. [R] |
| 2. <i>TOS</i>      | The Type of Service, can be HEX, see above.      |
| 3. <i>priority</i> | The socket priority, can be any positive number. |

Syntax: `set_endp_tos name TOS priority`

#### 195. **set\_endp\_quiesce**

Set the quiesce timer. This determines how long an endpoint will wait in a quiet state before stopping the test. This is good for gracefully finishing the last transaction and allowing all the packets in flight to be received by the receiving end (which continues to function as normal during the quiesce.) Use `set_cx_state` to actually put the endpoint in quiesce state.

| Argument          | Description  |
|-------------------|--|
| 1. <i>name</i>    | The name of the endpoint we are configuring. [R]                         |
| 2. <i>quiesce</i> | The number of seconds to quiesce this endpoint when told to quiesce. [R] |

Syntax: `set_endp_quiesce name quiesce`

#### 196. **set\_endp\_pld\_bounds**

Set the min/max payload size bounds for an endpoint. If the endpoint payload size is set to 'random', then the actual sizes will vary with an even distribution between the min and max. If the payload size is not random, it will always be the minimum payload size, as set here.

| Argument       | Description                                      |
|----------------|--|
| 1. <i>name</i> | The name of the endpoint we are configuring. [R] |

|                        |   |
|------------------------|---|
| 2. <i>min_pld_size</i> | The minimum payload size, in bytes.                   |
| 3. <i>max_pld_size</i> | The maximum payload size, in bytes.                   |
| 4. <i>is_random</i>    | YES if random, anything else for NO.                  |
| 5. <i>use_checksum</i> | YES if use checksum on payload, anything else for NO. |

Syntax: `set_endp_pld_bounds name min_pld_size max_pld_size is_random use_checksum`

#### 197. **set\_endp\_tx\_bounds**

Set the min/max transmit rate bounds for an endpoint. If the endpoint transmit rate is set to 'bursty', then the actual rates will vary between the min and max in a bursty fashion. If the rate is not bursty, it will always be the minimum rate, as set here.

| Argument              | Description  |
|-----------------------|--|
| 1. <i>name</i>        | The name of the endpoint we are configuring. [R]     |
| 2. <i>min_tx_rate</i> | The minimum transmit rate, in bits per second (bps). |
| 3. <i>max_tx_rate</i> | The maximum transmit rate, in bits per second (bps). |
| 4. <i>is_bursty</i>   | YES if bursty, anything else for NO.                 |

Syntax: `set_endp_tx_bounds name min_tx_rate max_tx_rate is_bursty`

#### 198. **set\_fe\_info**

Set read/write size and other file information for File Endpoints. You can also enter 'NA' for any value you do not wish to change. The quiesce-after-files option allows one to configure the test to automatically stop after completing a certain number of file reads or writes. The default is zero (0), which means run forever until stopped by user.

| Argument                       | Description   |
|--------------------------------|---|
| 1. <i>name</i>                 | The name of the file endpoint we are configuring. [R]                   |
| 2. <i>min_rw_sz</i>            | Minimum read/write size, in bytes.                                      |
| 3. <i>max_rw_sz</i>            | Maximum read/write size, in bytes.                                      |
| 4. <i>num_files</i>            | Number of files to create when writing.                                 |
| 5. <i>min_file_size</i>        | The minimum file size, in bytes.  |
| 6. <i>max_file_size</i>        | The maximum file size, in bytes.  |
| 7. <i>directory</i>            | The directory to read/write in. Absolute path suggested.                |
| 8. <i>prefix</i>               | The prefix of the file(s) to read/write.                                |
| 9. <i>io_direction</i>         | Should we be reading or writing: options: read, write                   |
| 10. <i>quiesce_after_files</i> | If non-zero, quiesce test after this many files have been read/written. |

Syntax: `set_fe_info name min_rw_sz max_rw_sz num_files min_file_size max_file_size directory prefix io_direction quiesce_after_files`

#### 199. **set\_gen\_cmd**

Set command that will be executed when this generic endpoint is started. Example:

```
set_gen_cmd fio-endp bonnie++ -f -d /mnt/test_fs/ -q
```

| Argument          | Description   |
|-------------------|---|
| 1. <i>name</i>    | The name of the file endpoint we are configuring. [R]       |
| 2. <i>command</i> | The rest of the command line arguments. Unescaped Value [R] |

Syntax: `set_gen_cmd name command`

#### 200. **set\_endp\_flag**

This command allows you to modify certain Endpoint specific options, including Unmanaged. Different endpoint types will support different options. To get a full listing of options, use the command without specifying a flag. Example:

```
[default@btbits] set_endp_flag t0100-A
Endpoint: Shelf: 1, Card: 1 Port: 6 Endpoint: 1 Type: LANFORGE_UDP
Unmanaged(0) DoChecksum(0) KernelMode(0)
ClearPortOnStart(0) EnableRndSrcIP(0)
EnableLinearSrcIP(0) EnableConcurrentSrcIP(0)
UseAutoNAT(0) EnableLinearSrcIPPort(0)
QuiesceAfterRange(0) QuiesceAfterDuration(0)
```

Notice how you need to place endpoint-A or endpoint-B to use command.

```
Unmanaged          |# Set endpoint unmanaged
DoChecksum         |# Enable checksumming
KernelMode        |# Enable kernel mode
ClearPortOnStart  |# clear stats on start
EnableRndSrcIP    |# randomize source IP
EnableLinearSrcIP |# linearized source IPs
EnableConcurrentSrcIP |# Concurrent source IPs?
UseAutoNAT        |# NAT friendly behavior
EnableLinearSrcIPPort |# linearized IP ports
QuiesceAfterRange |# quiesce after range of bytes
QuiesceAfterDuration |# quiesce after time period
AutoHelper        |# Automatically run on helper process
```

| Argument       | Description                                      |
|----------------|--|
| 1. <i>name</i> | The name of the endpoint we are configuring. [R] |
| 2. <i>flag</i> | The name of the flag. [R]                        |
| 3. <i>val</i>  | Either 1 (for on), or 0 (for off). [R,0-1]       |

Syntax: `set_endp_flag name flag val`

#### 201. set\_flag

This command allows you to modify certain client specific options, including the brevity of the output. Some useful flags are:

```
brief              |# Request more abbreviated output to various commands.
                  |# If enabled, the 'RSLT>>' CLI response will be hidden, for example.

push_endp_rpts    |# If enabled, server will send endpoint reports without
                  |# being asked. This may be more information than you want!

push_all_rpts     |# If enabled, server will send port, endpoint, and other
                  |# reports without being asked. This can flood scripts if
                  |# they are not expecting the input.

prompt_newlines   |# Add a newline after every prompt. Can help with scripts
                  |# that want to handle line-based input.

stream_events     |# Normally the CLI will not show Events (as seen in the Event
                  |# tab in the GUI) as they happen to keep the text output
                  |# cleaner. But, this option can be enabled by toggling the
                  |# stream_events flag on.

request_keyed_text |# Normally most keyed-text events are only sent to the GUI
                  |# (binary) clients. Enable 'request_keyed_text' to have these
                  |# events sent to the CLI session as well.
```

To get a full listing of options, use the `set_flag` command without any arguments.

| Argument         | Description   |
|------------------|---|
| 1. <i>flag</i>   | The name of the flag. [R]   |
| 2. <i>val</i>    | Either 1 (for on), or 0 (for off). [R,0-1]                                  |
| 3. <i>client</i> | Specify the user, if it is not the current user. Requires admin privileges. |

Syntax: `set_flag flag val client`

#### 202. set\_gps\_info

This command sets the position of the device: latitude, longitude, and altitude. You can manually enter the value for stationary equipment, or you can hook your LANforge device up to a GPS receiver for real-time updates. The values come from the \$GPGGA line, as defined by the NMEA protocol. Shelf can be 'SELF' when talking to data-generators, and it will set itself.

| Argument            | Description   |
|---------------------|---|
| 1. <i>shelf</i>     | Shelf number for the port to be modified, or SELF. [R][D:1] |
| 2. <i>resource</i>  | Resource number for the port to be modified. [W]            |
| 3. <i>latitude</i>  | The latitude, as read from a GPS device.                    |
| 4. <i>ns</i>        | North or South (Latitude).                                  |
| 5. <i>longitude</i> | The longitude, as ready from a GPS device.                  |
| 6. <i>ew</i>        | East or west (Longitude).                                   |
| 7. <i>altitude</i>  | Altitude, assumes units are Meters.                         |

Syntax: `set_gps_info shelf resource latitude ns longitude ew altitude`

### 203. **set\_poll\_mode**

When set to polling mode, LANforge will not generate reports unless asked. This is more efficient for very large numbers of connections and works fine for smaller configurations too. Non-polling mode works fine up to about 500 cross-connects on high-end hardware.

Polling Modes:

```
polling |#
push    |#
```

| Argument       | Description              |
|----------------|--------------------------|
| 1. <i>mode</i> | 'polling' or 'push'. [R] |

Syntax: `set_poll_mode mode`

### 204. **set\_port**

This command allows you to modify attributes on an Ethernet port. These options includes the IP address, netmask, gateway address, MAC, MTU, and TX Queue Length.

In order for this command to succeed the Endpoints which are using the port must not be running. Endpoints which use IP will be updated automatically with the appropriate information if the port is modified.

If you do not wish to modify one or more of the settings, enter 'NA' instead of a real value.

For the flags entries, add up as many flags as you wish to set, and enter the sum. For example, if you want to set flag 1, 2, and 8, then enter: 11, or 0xB.

When setting the link speed with **current\_flags**, use one of the Fixed flags and don't set auto-negotiate for fixed mode, or set as many of the advert flags as you wish and set auto-negotiate for auto-negotiate mode.

Normally, you will advertise everything your resource is capable of.

**current\_flags** can be:

```
if_down          | 0x1          # Interface Down
fixed_10bt_hd    | 0x2          # Fixed-10bt-HD (half duplex)
fixed_10bt_fd    | 0x4          # Fixed-10bt-FD
fixed_100bt_hd   | 0x8          # Fixed-100bt-HD
fixed_100bt_fd   | 0x10         # Fixed-100bt-FD
auto_neg         | 0x100        # auto-negotiate
adv_10bt_hd      | 0x1000000    # advert-10bt-HD
adv_10bt_fd      | 0x2000000    # advert-10bt-FD
adv_100bt_hd     | 0x4000000    # advert-100bt-HD
adv_100bt_fd     | 0x8000000    # advert-100bt-FD
adv_flow_ctl     | 0x8000000    # advert-flow-control
promisc          | 0x100000000  # PROMISC
use_dhcp         | 0x800000000  # USE-DHCP
adv_2.5g_fd      | 0x400000000  # advert-2.5G-FD
adv_10g_fd       | 0x800000000  # advert-10G-FD
tso_enabled      | 0x1000000000 # TSO-Enabled
lro_enabled      | 0x2000000000 # LRO-Enabled
gro_enabled      | 0x4000000000 # GRO-Enabled
ufo_enabled      | 0x8000000000 # UFO-Enabled
gso_enabled      | 0x10000000000 # GSO-Enabled
use_dhcpv6       | 0x20000000000 # USE-DHCPv6
rxfcfs           | 0x40000000000 # RXFCFS
no_dhcp_rel      | 0x80000000000 # No-DHCP-Release
staged_ifup      | 0x100000000000 # Staged-IFUP
http_enabled     | 0x200000000000 # Enable HTTP (nginx) service for this port.
ftp_enabled      | 0x400000000000 # Enable FTP (vsftpd) service for this port.
aux_mgt          | 0x800000000000 # Enable Auxillary-Management flag for this port.
```



```

no_dhcp_restart | 0x10000000000000 # Disable restart of DHCP on link connect (ie, wifi).
# This should usually be enabled when testing wifi
# roaming so that the wifi station can roam
# without having to re-acquire a DHCP lease each
# time it roams.

ignore_dhcp | 0x20000000000000 # Don't set DHCP acquired IP on interface,
# instead print CLI text message. May be useful
# in certain wifi-bridging scenarios where external
# traffic-generator cannot directly support DHCP.

no_ifup_post | 0x40000000000000 # Skip ifup-post script if we can detect that we
# have roamed. Roaming is considered true if
# the IPv4 address has not changed.

radius_enabled | 0x20000000000000 # Enable RADIUS service (using hostapd as radius server)
ipsec_client | 0x40000000000000 # Enable client IPSEC xfrm on this port.
ipsec_concentrator | 0x80000000000000 # Enable concentrator (upstream) IPSEC xfrm on this port.
service_dns | 0x10000000000000 # Enable DNS (dnsmasq) service on this port.
adv_5g_fd | 0x40000000000000 # Advertise 5Gbps link speed.

```

cmd\_flags can be:

```

reset_transceiver | 0x1 # Reset transceiver
restart_link_neg | 0x2 # Restart link negotiation
force_MII_probe | 0x4 # Force MII probe
no_hw_probe | 0x8 # Don't probe hardware
probe_wifi | 0x10 # Probe WIFI
new_gw_probe | 0x20 # Force new GW probe
new_gw_probe_dev | 0x40 # Force new GW probe for ONLY this interface
from_user | 0x80 # from_user (Required to change Mgt Port config
# (IP, DHCP, etc)

skip_port_bounce | 0x100 # skip-port-bounce (Don't ifdown/up
# interface if possible.)

from_dhcp | 0x200 # Settings come from DHCP client.
abort_if_scripts | 0x400 # Forceably abort all ifup/down scripts on this Port.
use_pre_ifdown | 0x800 # Call pre-ifdown script before bringing interface down.

```

The **interest** flags are normally not needed by casual users. They are used to ignore certain arguments or flags.

interest flag values are:

```

command_flags | 0x1 # apply command flags
current_flags | 0x2 # apply current flags
ip_address | 0x4 # IP address
ip_mask | 0x8 # IP mask
ip_gateway | 0x10 # IP gateway
mac_address | 0x20 # MAC address
supported_flags | 0x40 # apply supported flags
link_speed | 0x80 # Link speed
mtu | 0x100 # MTU
tx_queue_length | 0x200 # TX Queue Length
promisc_mode | 0x400 # PROMISC mode
internal_use_1 | 0x800 # (INTERNAL USE)
alias | 0x1000 # Port alias
rx_all | 0x2000 # Rx-ALL
dhcp | 0x4000 # including client-id.
rpt_timer | 0x8000 # Report Timer
bridge | 0x10000 # BRIDGE
ipv6_addr | 0x20000 # IPv6 Address
bypass | 0x40000 # Bypass
gen_offload | 0x80000 # Generic offload flags, everything but LRO
cpu_mask | 0x100000 # CPU Mask, useful for pinning process to CPU core
lro_offload | 0x200000 # LRO (Must be disabled when used in Wanlink,
# and probably in routers)

sta_br_id | 0x400000 # WiFi Bridge identifier. 0 means no bridging.
ifdown | 0x800000 # Down interface
dhcpv6 | 0x1000000 # Use DHCPv6
rxfcs | 0x2000000 # RXFCS
dhcp_rls | 0x4000000 # DHCP release
svc_httpd | 0x8000000 # Enable/disable HTTP Service for a port
svc_ftpd | 0x10000000 # Enable/disable FTP Service for a port
aux_mgt | 0x20000000 # Enable/disable Auxillary-Management for a port
no_dhcp_conn | 0x40000000 # Enable/disable NO-DHCP-ON-CONNECT flag for a port
no_apply_dhcp | 0x80000000 # Enable/disable NO-APPLY-DHCP flag for a port
skip_ifup_roam | 0x100000000 # Enable/disable SKIP-IFUP-ON-ROAM flag for a port

```

flags2 values are:

```

use_stp | 0x1 # Use Spanning Tree Protocol
supports_bypass | 0x2 # Support Bypass Devices
bypass_enabled | 0x10 # Enable Bypass Device
bypass_power_down | 0x80 # Should bypass be on when we shutdown or loose power?
bypass_power_on | 0x100 # Should bypass be on when we first power up?

```

```
bypass_disconnect | 0x200 # Logically disconnect the cable (link-down)
```

IPv6 Address format is: `addr/prefix` Scope is implied by the position (first address is global, etc).

NOTE: You may create custom `dhclient` config files if you need more flexibility than the built-in features LANforge supports.

#### dhcp\_vendor\_id:

```
NA | # Do not change from current value.
NONE | # Do not use dhcp vendor ID
[string] | # Use the string for the vendor ID.
```

#### dhcp\_hostname:

```
NA | # Do not change from current value.
NONE | # Do not use dhcp Hostname
__EID__ | # Use hostname 'CT-[resource-id].[port-name]'
__ALIAS__ | # Use alias if set, or EID behaviour if alias is not set..
[string] | # Use the string for the Hostname.
```

#### dhcp\_client\_id:

```
NA | # Do not change from current value.
NONE | # Do not use dhcp client ID.
__MAC__ | # Use interface's MAC address for the client ID.
__DEVNAME__ | # Use the interface's name as the client ID.
[string] | # Use the string for the client ID.
```

#### Related Commands

| Argument                 | Description  |
|--------------------------|--|
| 1. <i>shelf</i>          | Shelf number for the port to be modified. [R][D:1]   |
| 2. <i>resource</i>       | Resource number for the port to be modified. [W]   |
| 3. <i>port</i>           | Port number for the port to be modified. [W]   |
| 4. <i>ip_addr</i>        | IP address for the port, or NA.  |
| 5. <i>netmask</i>        | Netmask which this port should use, or NA.   |
| 6. <i>gateway</i>        | IP address of the gateway device - used for IP routing, or NA.   |
| 7. <i>cmd_flags</i>      | Command Flags: See above, or NA.   |
| 8. <i>current_flags</i>  | See above, or NA.  |
| 9. <i>MAC</i>            | MAC address to set this port to, or leave blank to not set it, or NA.                                  |
| 10. <i>MTU</i>           | Maximum Transmit Unit (MTU) for this interface. Can be blank or NA.                                    |
| 11. <i>tx_queue_len</i>  | Transmit Queue Length for this interface. Can be blank or NA.  |
| 12. <i>alias</i>         | A user-defined name for this interface. Can be BLANK or NA.  |
| 13. <i>interest</i>      | Which things are we really interested in setting. Can over-ride defaults based on the other arguments. |
| 14. <i>report_timer</i>  | How often, in milliseconds, should we poll stats on this interface?                                    |
| 15. <i>flags2</i>        | Bridge & other flags, see above.   |
| 16. <i>br_priority</i>   | Bridge priority, 16-bit number.  |
| 17. <i>br_aging_time</i> | MAC aging time, in seconds, 32-bit number (or peer IP for GRE).  |
| 18. <i>br_max_age</i>    | How long until STP considers a non-responsive bridge dead.   |
| 19. <i>br_hello_time</i> | How often does the bridge send out STP hello packets.  |

|                                |  |
|--------------------------------|--|
| 20. <i>br_forwarding_delay</i> | How long to wait until the bridge will start forwarding packets.   |
| 21. <i>br_port_cost</i>        | STP Port cost for a port (this applies only to NON-BRIDGE interfaces).   |
| 22. <i>br_port_priority</i>    | STP Port priority for a port (this applies only to NON-BRIDGE interfaces).   |
| 23. <i>IPv6_addr_global</i>    | Global scoped IPv6 address.  |
| 24. <i>IPv6_addr_link</i>      | Link scoped IPv6 address.  |
| 25. <i>IPv6_dflt_gw</i>        | IPv6 default gateway.  |
| 26. <i>bypass_wdt</i>          | Watch Dog Timer (in seconds) for this port. Zero (0) to disable.   |
| 27. <i>cpu_mask</i>            | CPU Mask for CPUs that should service this interface. Zero is don't set (let OS make the decision). This value will be applied to the proper /proc/irq/[irq-num]/smp_affinity file by the pin_irq.pl script. |
| 28. <i>dns_servers</i>         | DNS servers for use by traffic on this port, comma-separated list, BLANK means zero-length string.   |
| 29. <i>sta_br_id</i>           | WiFi STATION bridge ID. Zero means none.   |
| 30. <i>dhcp_client_id</i>      | Optional string of up to 63 bytes in length to be passed to the dhclient process. See above.   |
| 31. <i>current_flags_msk</i>   | This sets 'interest' for flags 'Enable RADIUS service' and higher. See above, or NA.   |
| 32. <i>dhcp_vendor_id</i>      | Optional string of up to 63 bytes in length to be passed to the dhclient process. See above.   |
| 33. <i>ipsec_concentrator</i>  | IP Address of IPsec concentrator.  |
| 34. <i>ipsec_passwd</i>        | Password for IPsec, for pubkey, use: pubkey:[pem-file-name], for instance: pubkey:station.pem  |
| 35. <i>ipsec_local_id</i>      | Local Identifier for this IPsec tunnel.  |
| 36. <i>ipsec_remote_id</i>     | Remote Identifier for this IPsec tunnel.   |
| 37. <i>dhcp_hostname</i>       | Optional string of up to 63 bytes in length to be passed to the dhclient process. Option 12, see above.  |

Syntax: `set_port shelf resource port ip_addr netmask gateway cmd_flags current_flags MAC MTU tx_queue_len alias interest report_timer flags2 br_priority br_aging_time br_max_age br_hello_time br_forwarding_delay br_port_cost br_port_priority IPv6_addr_global IPv6_addr_link IPv6_dflt_gw bypass_wdt cpu_mask dns_servers sta_br_id dhcp_client_id current_flags_msk dhcp_vendor_id ipsec_concentrator ipsec_passwd ipsec_local_id ipsec_remote_id dhcp_hostname`

#### 205. **set\_port2**

Set additional port configuration for existing port. The dhclient\_50 syntax must be like this, including spaces: 1, 2, 3, 4 Surround it with single quotes when sending through LANforge CLI.

| Argument              | Description   |
|-----------------------|---|
| 1. <i>shelf</i>       | Shelf number for the port to be modified. [R][D:1]                    |
| 2. <i>resource</i>    | Resource number for the port to be modified. [W]                      |
| 3. <i>port</i>        | Port identifier. [R]  |
| 4. <i>dhclient_50</i> | Set DHCP Client option-50 text. DEFAULT means do not use this option. |

Syntax: `set_port2 shelf resource port dhclient_50`

#### 206. **set\_port\_alias**

Set the alias for a virtual interface specified by MAC or 802.1Q VLAN-ID. This command is designed to make it easier to script MAC an 802.1Q VLANs

## Related Commands

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number for the port to be modified. [R][D:1]                    |
| 2. <i>resource</i> | Resource number for the port to be modified. [W]                      |
| 3. <i>port</i>     | Physical Port identifier that owns the virtual interface. [R]         |
| 4. <i>vport</i>    | Virtual port identifier. MAC for MAC-VLANs, VLAN-ID for 802.1Q vlans. |
| 5. <i>alias</i>    | New alias to assign to this virtual interface. [W]                    |

Syntax: `set_port_alias shelf resource port vport alias`

## 207. set\_sec\_ip

Set a new list secondary IP Address(es). Only makes necessary incremental changes to have the requested configuration.

### Related Commands

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]  |
| 2. <i>resource</i> | Resource number. [W]  |
| 3. <i>port</i>     | Name of network device (Port) to which these IPs will be added. [R] |
| 4. <i>ip_list</i>  | IP1/prefix,IP2/prefix,...IPZ/prefix. [W]                            |

Syntax: `set_sec_ip shelf resource port ip_list`

## 208. set\_voip\_info

Set various VOIP endpoint related values. Use this to enable behaviour different from the defaults (see `add_voip_endp`, and `set_endp_flag`). If the min and max values are different, a random value in that range will be chosen. Any values can be 'NA' and they will be ignored. If `min/max_call_duration` is less than the length of the wave file multiplied by the number of times to play the wave file, then the `max_call_duration` will determine the call length. If Min/Max call duration are not the same, a random value between the min and max will be chosen each time a call is started. Otherwise, the call will be determined by the wave file size & repetition. The registration expire timer affects the sip messaging protocol: The default of 300 is fine in most cases. The `sound_dev` determines which sound device to play the received RTP stream on. Usually `/dev/dsp` or `/dev/audio` is the correct value.

| Argument                      | Description  |
|-------------------------------|--|
| 1. <i>name</i>                | The name of the endpoint we are configuring. [R]   |
| 2. <i>first_call_delay</i>    | How long to wait before making first call, in seconds.   |
| 3. <i>min_inter_call_gap</i>  | Minimum time to wait between calls, in seconds.  |
| 4. <i>max_inter_call_gap</i>  | Maximum time to wait between calls, in seconds.  |
| 5. <i>reg_expire_timer</i>    | SIP Registration expire timer, in seconds.   |
| 6. <i>codec</i>               | Codec to use for the voice stream, supported values: G711U, G711A, SPEEX, g726-16, g726-24, g726-32, g726-40, g729a. |
| 7. <i>messaging_protocol</i>  | Messaging protocol, supported values: SIP.   |
| 8. <i>loop_call_count</i>     | How many calls to make, zero means infinite.   |
| 9. <i>loop_wavefile_count</i> | How many times to play the wave file, zero means infinite.   |
| 10. <i>min_call_duration</i>  | How long should the call be, in seconds.   |
| 11. <i>max_call_duration</i>  | How long should the call be, in seconds.   |

|                               |   |
|-------------------------------|---|
| 12. <i>sound_dev</i>          | Which sound device should we play sound to. (see <code>set_endp_flags</code> ).         |
| 13. <i>ringing_timer</i>      | How long (milliseconds) to wait in the ringing state before flagging call as no-answer. |
| 14. <i>local_sip_port</i>     | Local SIP UDP port. Default is <code>min_rtp_port + 2</code> .                          |
| 15. <i>PESQ_server_IP</i>     | LANforge PESQ server IP address.  |
| 16. <i>PESQ_server_port</i>   | LANforge PESQ server port, default is 3998.   |
| 17. <i>PESQ_server_passwd</i> | LANforge PESQ server password. Default is to use no authentication (blank entry).       |
| 18. <i>jitter_buffer_sz</i>   | The size of the jitter buffer in packets. Default value is 8.                           |

Syntax: `set_voip_info name first_call_delay min_inter_call_gap max_inter_call_gap reg_expire_timer codec messaging_protocol loop_call_count loop_wavfile_count min_call_duration max_call_duration sound_dev ringing_timer local_sip_port PESQ_server_IP PESQ_server_port PESQ_server_passwd jitter_buffer_sz`

### 209. **set\_wanpath\_filter**

Set the filter type for the WanPath. If the filter is set to MAC, then it will match based on the source and/or destination MAC address.

- o If IP is chosen, it will match on the source and destination IP addresses and masks. Default behaviour is to match on the IP address.
- o MAC syntax is: `00:11:22:33:44:55`
- o IP Syntax is: `a.b.c.d/24` or `a.b.c.d/255.255.255.0`
- o PCAP syntax is same as for `tcpdump`. Use 'man `tcpdump`' on Linux, or see this page: [http://www.tcpdump.org/tcpdump\\_man.html](http://www.tcpdump.org/tcpdump_man.html)
- o The 'passive' argument is to allow you to set the pcap filter, but not actually use it (perhaps you are using IP filtering, but we want to remember the pcap filter for later.)

| Argument              | Description  |
|-----------------------|--|
| 1. <i>wl_name</i>     | The name of the WanLink endpoint we are configuring. [R]         |
| 2. <i>wp_name</i>     | The name of the WanPath we are configuring. [R]                  |
| 3. <i>filter_type</i> | The filter type, one of: MAC, IP, PCAP.                          |
| 4. <i>src_filter</i>  | The source MAC or IP/Mask. For PCAP, this is the only filter.    |
| 5. <i>dst_filter</i>  | The destination MAC or IP/Mask, 'NA' for PCAP.                   |
| 6. <i>reverse</i>     | If you want the logic reversed, use 'ON', otherwise set to 'OFF' |
| 7. <i>defer_flush</i> | Enter 'YES' if you do NOT want this flushed to the remote.       |
| 8. <i>passive</i>     | Enter 'YES' if you do NOT want to use this filter currently.     |

Syntax: `set_wanpath_filter wl_name wp_name filter_type src_filter dst_filter reverse defer_flush passive`

### 210. **set\_wanpath\_running**

Set the Running state of the WanPath. If the state is set to:

```
AS_PARENT | # then it will be started and stopped as the parent WanLink is.
STOPPED   | # then it will not be running at any time.
RUNNING   | # then it will be running at all times
```

Though, due to implementation, it may not actually pass any traffic if the parent WanLink is not running.

| Argument          | Description  |
|-------------------|--|
| 1. <i>wl_name</i> | The name of the WanLink endpoint we are configuring. [R] |
| 2. <i>wp_name</i> | The name of the WanPath we are configuring. [R]          |
| 3. <i>running</i> | The state, one of: AS_PARENT, RUNNING, STOPPED. [R]      |

Syntax: `set_wanpath_running wl_name wp_name running`

## 211. set\_wanpath\_corruption

Set a corruption for a WanPath. Corruptions include random and fixed over-write of a byte in the Ethernet frame, as well as random bit-flips and bit transposes. Up to 6 corruptions are supported per WanLink. If the 'chain' flag is set on a corruption, then if that corruption is chosen to be applied, the next corruption will always be applied. The 'byte' specifies the byte to write into the frame, if OVERWRITE\_FIXED flag is chosen. The min and max offset determine the possible position of the byte to be modified. If min is less than max, a random byte between min and max will be modified. The offset is from the beginning of the Ethernet header. The 'rate' specifies how often, per million, the corruption will be applied. This is flat-random distribution. The flags are defined as:

```

OVERWRITE_RANDOM | 1      # Write a random value to a byte.
OVERWRITE_FIXED  | 2      # Write a fixed value to a byte.
BIT_FLIP         | 4      # Flip a random bit in a byte.
BIT_TRANSPOSE    | 8      # Transpose two side-by-side bits in a byte.
DO_CHAIN_ON_HIT  | 16     # Do next corruption if this corruption is applied.
RECALC_CSUMS    | 32     # Attempt to re-calculate UDP and TCP checksums.
    
```

The RECALC\_CSUMS option will ONLY work if the UDP or TCP packet spans a single Ethernet frame.

Only one of the first 4 bits should be selected. Add flag values together to set multiple flags.

| Argument             | Description   |
|----------------------|---|
| 1. <i>name</i>       | WanLink name [R]  |
| 2. <i>path</i>       | WanPath name [R]  |
| 3. <i>index</i>      | The corruption to modify (0-5). [R,0-5]                                       |
| 4. <i>flags</i>      | The flags for this corruption.  |
| 5. <i>byte</i>       | The byte to use for OVERWRITE_FIXED (or NA).                                  |
| 6. <i>min_offset</i> | The minimum offset from start of Ethernet packet for the byte to be modified. |
| 7. <i>max_offset</i> | The maximum offset from start of Ethernet packet for the byte to be modified. |
| 8. <i>rate</i>       | Specifies how often, per million, this corruption should be applied.          |

Syntax: `set_wanpath_corruption name path index flags byte min_offset max_offset rate`

## 212. set\_wanlink\_info

Set the WanLink information for an endpoint. You can set the Latency, MaxJitter, and reorder characteristics here. Special attention should be paid to extra\_buffer. This setting should be zero, or a small number, if you are doing latency-sensitive testing. Use -1 if you want LANforge to automatically configure a proper extra\_buffer size based on your maximum bandwidth. The server will add the extra\_buffer size to a calculated buffer size based on the maximum jitter and latency specified in the WanLink endpoint. If you wish to drop bursts of packets, then set the min\_drop\_amt and max\_drop\_amt. When LANforge determines that a packet drop should occur (based on the drop\_freq), then it will also pick a random value between the min and max drop\_amt and drop that many packets in a row. The value of all attributes other than the name can be 'NA', which means do not change the current value.

| Argument               | Description  |
|------------------------|--|
| 1. <i>name</i>         | The name of the endpoint we are configuring. [R]   |
| 2. <i>speed</i>        | The maximum speed of traffic this endpoint will accept (bps).                                |
| 3. <i>latency</i>      | The base latency added to all packets, in milliseconds (or add 'us' suffix for microseconds) |
| 4. <i>max_jitter</i>   | The maximum jitter, in milliseconds (or ad 'us' suffix for microseconds)                     |
| 5. <i>reorder_freq</i> | How often, out of 1,000,000 packets, should we make a packet out of order.                   |
| 6. <i>extra_buffer</i> | The extra amount of bytes to buffer before dropping pkts, in units of 1024. Use -1 for AUTO. |
| 7. <i>drop_freq</i>    | How often, out of 1,000,000 packets, should we purposefully drop a packet.                   |

|                                 |   |
|---------------------------------|---|
| 8. <i>dup_freq</i>              | How often, out of 1,000,000 packets, should we purposefully duplicate a packet. |
| 9. <i>playback_capture_file</i> | Name of the WAN capture file to play back.                                      |
| 10. <i>jitter_freq</i>          | How often, out of 1,000,000 packets, should we apply jitter.                    |
| 11. <i>min_drop_amt</i>         | Minimum amount of packets to drop in a row. Default is 1.                       |
| 12. <i>max_drop_amt</i>         | Maximum amount of packets to drop in a row. Default is 1.                       |
| 13. <i>min_reorder_amt</i>      | Minimum amount of packets by which to reorder, Default is 1.                    |
| 14. <i>max_reorder_amt</i>      | Maximum amount of packets by which to reorder, Default is 10.                   |
| 15. <i>max_lateness</i>         | Maximum amount of un-intentional delay before pkt is dropped. Default is AUTO   |

Syntax: `set_wanlink_info name speed latency max_jitter reorder_freq extra_buffer drop_freq dup_freq playback_capture_file jitter_freq min_drop_amt max_drop_amt min_reorder_amt max_reorder_amt max_lateness`

### 213. **set\_wanlink\_pcap**

Set the WanLink packet capture file name, and whether or not the system should actually capture the packets. The generated files for both WanLink endpoints can then be played back across a network using the LANforge playback features. The capture will start and stop with the endpoint, and it will write over any existing file so be careful. To mitigate the risk, if the path is absolute, it must start with /tmp or /home/lanforge. To effectively store files elsewhere, you can set up soft-links to directories within one of these directory trees.

Capture Options:

```
ON    |# start capturing
OFF   |# stop capturing
```

| Argument            | Description   |
|---------------------|---|
| 1. <i>name</i>      | The name of the endpoint we are configuring. [R]                  |
| 2. <i>capture</i>   | Should we capture or not? ON or OFF. [R]                          |
| 3. <i>directory</i> | The directory name in which packet capture files will be written. |

Syntax: `set_wanlink_pcap name capture directory`

### 214. **set\_wl\_corruption**

Set a corruption for WanLink. Corruptions include random and fixed over-write of a byte in the Ethernet frame, as well as random bit-flips and bit transposes. Specific rules apply:

- Up to 6 corruptions are supported per WanLink.
- If the `chain` flag is set on a corruption, then if that corruption is chosen to be applied, the next corruption will always be applied.
- If `OVERWRITE_FIXED` flag is chosen, the 'byte' specifies the byte to write into the frame.

The `min_offset` and `max_offset` determine the possible position of the byte to be modified. If min is less than max, a random byte between min and max will be modified. The offset is from the beginning of the Ethernet header.

The `rate` specifies how often, per million, the corruption will be applied. This is flat-random distribution.

The **flags** are defined as:

```
OVERWRITE_RANDOM | 1    # Write a random value to a byte.
OVERWRITE_FIXED  | 2    # Write a fixed value to a byte.
BIT_FLIP         | 4    # Flip a random bit in a byte.
BIT_TRANSPOSE    | 8    # Transpose two side-by-side bits in a byte.
DO_CHAIN_ON_HIT  | 16   # Do next corruption if this corruption is applied.
RECALC_CSUMS     | 32   # Attempt to re-calculate UDP and TCP checksums.
```

The `RECALC_CSUMS` option will ONLY work if the UDP or TCP packet spans a single Ethernet frame.

Only one of the first 4 bits should be selected. Add flag values together to set multiple flags.

| Argument             | Description   |
|----------------------|---|
| 1. <i>name</i>       | WanLink name [R]  |
| 2. <i>index</i>      | The corruption to modify (0-5). [R,0-5]                                       |
| 3. <i>flags</i>      | The flags for this corruption.  |
| 4. <i>byte</i>       | The byte to use for OVERWRITE_FIXED (or NA).                                  |
| 5. <i>min_offset</i> | The minimum offset from start of Ethernet packet for the byte to be modified. |
| 6. <i>max_offset</i> | The maximum offset from start of Ethernet packet for the byte to be modified. |
| 7. <i>rate</i>       | Specifies how often, per million, this corruption should be applied.          |

Syntax: `set_wl_corruption name index flags byte min_offset max_offset rate`

#### 215. **set\_wl\_qdisc**

Set a Queuing Discipline on the WanLink.

```
FIFO | # is the default queuing discipline, no arguments
WRR, [queue, queue, ...] | # Weighted Round Robbin is also available
```

For WRR you must specify the weights (and in doing so, the number of queues):

```
set_wl_qos [wanlink] WRR,10000,10000,10000,10000,500000,600000,600000
```

The packet priority will be mapped directly onto the queues. If the packet priority cannot be queried from the OS, the 3 IP ToS bits will be used for priority, so we recommend 7 queues for WRR QDiscs.

Others queuing disciplines may be added in the future.

| Argument        | Description                     |
|-----------------|---------------------------------|
| 1. <i>name</i>  | WanLink name [R]                |
| 2. <i>qdisc</i> | FIFO, WRR,a,b,c,d,e,f,g etc [R] |

Syntax: `set_wl_qdisc name qdisc`

#### 216. **set\_endp\_file**

Set the file name for an endpoint. In the future, this may affect various endpoint types differently, but for now it is only used to set the capture file that a Custom Ethernet endpoint can 'play back'. To use this feature, first use a WanLink connection to capture packets flowing across a network. The WanLink connections can be configured to save all incoming packets to a file. The Customer Ethernet connection can then be configured with one of the capture files associated with each endpoint. During playback, each endpoint will play back the packet stream as it arrived, inserting pauses between the packets, and ensuring that packets are placed on the wire in the same order that they were received. file can be blank or NA if you wish to only turn playback on or off.

Playback options:

```
ON | # on
OFF | # off
```

| Argument           | Description   |
|--------------------|---|
| 1. <i>name</i>     | The name of the endpoint we are configuring. [R]      |
| 2. <i>playback</i> | Should we playback the capture or not? ON or OFF. [R] |
| 3. <i>file</i>     | The file name to read the playback packets from.      |

Syntax: `set_endp_file name playback file`

#### 217. **show\_attenuators**

Show Attenuator information.

| Argument | Description |
|----------|-------------|
|----------|-------------|



|                    |   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number or alias, can be 'all'. [R][D:1]         |
| 2. <i>resource</i> | Resource number, or 'all'. [W]                        |
| 3. <i>serno</i>    | Serial number for requested Attenuator, or 'all'. [W] |

Syntax: [show\\_attenuators shelf resource serno](#)

#### 218. **show\_rfggen**

Show RF-Generators configured and/or discovered.

| Argument           | Description                                   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number or alias, can be 'all'. [R][D:1] |
| 2. <i>resource</i> | Resource number, or 'all'. [W]                |

Syntax: [show\\_rfggen shelf resource](#)

#### 219. **show\_resources**

Show one or all resources for one or all shelves.

| Argument           | Description                                   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number or alias, can be 'all'. [R][D:1] |
| 2. <i>resource</i> | Resource number, or 'all'. [W]                |

Syntax: [show\\_resources shelf resource](#)

#### 220. **show\_clients**

Show all unique clients that have registered in the past. Using login, you can become any client on the list, and take on the values of that client. Multiple users can login as the same client, if desired.

Syntax: [show\\_clients](#)

#### 221. **show\_cx**

Show one or all cross-connects for one or all test managers.

| Argument                | Description                                    |
|-------------------------|--|
| 1. <i>test_mgr</i>      | Specify test-mgr to act on, or 'all'. [R]      |
| 2. <i>cross_connect</i> | Specify cross-connect to act on, or 'all'. [W] |

Syntax: [show\\_cx test\\_mgr cross\\_connect](#)

#### 222. **show\_cxe**

Show one or all cross-connects and their endpoints for one or all test managers. Please note that as of Release 5.2.4 (and earlier), this only returns cached Endpoint values. This means if the GUI is not running or if endpoint results are not otherwise being queried, the returned stats will not be accurate.

| Argument                | Description                                  |
|-------------------------|--|
| 1. <i>test_mgr</i>      | Specify test-mgr to use, or 'all'. [R]       |
| 2. <i>cross_connect</i> | Specify cross-connect to show, or 'all'. [W] |

Syntax: [show\\_cxe test\\_mgr cross\\_connect](#)

#### 223. **show\_cd**

Show one/all Collision Domains for one/all resources in one/all shelves. An empty specifier will be treated as 'all'. This command will always request the absolute latest information from the remote system(s)

| Argument        | Description                              |
|-----------------|--|
| 1. <i>shelf</i> | Name/id of the shelf, or 'all'. [R][D:1] |

|                            |   |
|----------------------------|---|
| 2. <i>resource</i>         | Resource number, or 'all'. [W]              |
| 3. <i>collision-domain</i> | Name of the Collision Domain, or 'all'. [W] |

Syntax: [show\\_cd shelf resource collision-domain](#)

#### 224. **show\_rt**

Show a Virtual Router's routing table.

| Argument                 | Description   |
|--------------------------|---|
| 1. <i>shelf</i>          | Shelf number. [R][D:1]                                  |
| 2. <i>resource</i>       | Resource number. [W]                                    |
| 3. <i>virtual-router</i> | Name of the virtual router. [W]                         |
| 4. <i>key</i>            | Unique identifier for this request. Usually left blank. |

Syntax: [show\\_rt shelf resource virtual-router key](#)

#### 225. **show\_vr**

Show one/all Virtual Routers for one/all resources in one/all shelves. An empty specifier will be treated as 'all'. May use cached values if the values are fresh enough.

| Argument           | Description                               |
|--------------------|---|
| 1. <i>shelf</i>    | Name/id of the shelf, or 'all'. [R][D:1]  |
| 2. <i>resource</i> | Resource number, or 'all'. [W]            |
| 3. <i>router</i>   | Name of the Virtual Router, or 'all'. [W] |

Syntax: [show\\_vr shelf resource router](#)

#### 226. **show\_vrcx**

Show one/all Virtual Router Connections for one/all resources in one/all shelves. Only Connections on the 'free-list', those not associated with any Virtual Router will be shown with this command unless the VRCX is specified by name. If the VRCX is in a virtual router, only cached results will be shown. Connections associated with routers will be shown with the 'show\_vr' command with the rest of the router information. Cached values may be used if they are recent enough.

| Argument           | Description  |
|--------------------|--|
| 1. <i>shelf</i>    | Name/id of the shelf, or 'all'. [R][D:1]             |
| 2. <i>resource</i> | Resource number, or 'all'. [W]                       |
| 3. <i>cx_name</i>  | Name of the Virtual Router Connection, or 'all'. [W] |

Syntax: [show\\_vrcx shelf resource cx\\_name](#)

#### 227. **show\_dbs**

Show all available databases that may be loaded.

Syntax: [show\\_dbs](#)

#### 228. **show\_endpoints**

Show one or all endpoints. Some endpoint types take an extra argument to specify what to show more precisely: Generic endpoints check extra for 'history' and in that case they will report recent output, not just the last line of output.

| Argument           | Description                     |
|--------------------|---------------------------------|
| 1. <i>endpoint</i> | Name of endpoint, or 'all'. [R] |
| 2. <i>extra</i>    | See above.                      |

Syntax: [show\\_endpoints endpoint extra](#)

#### 229. **show\_script\_results**

Show results of last script run for one or all endpoints. If using 'all', results will be for all endpoints and test-groups will be skipped entirely.

| Argument           | Description   |
|--------------------|---|
| 1. <i>endpoint</i> | Name of endpoint, test-group, or 'all'. [R]             |
| 2. <i>key</i>      | Optional 'key' to be used in keyed-text message result. |

Syntax: `show_script_results endpoint key`

### 230. **show\_pesq**

Show PESQ results for one or all VOIP endpoints.

| Argument           | Description                     |
|--------------------|---------------------------------|
| 1. <i>endpoint</i> | Name of endpoint, or 'all'. [R] |

Syntax: `show_pesq endpoint`

### 231. **show\_endp\_payload**

Show the payloads for one or all endpoints. The results will be shown in HEX. You may specify the number of bytes to print out, or you can just use the default value of 128 by not entering the length. You should not specify a very large length and also use 'ALL' for your endpoint, or you may over-run internal buffers can cause your message to be truncated.

| Argument            | Description  |
|---------------------|--|
| 1. <i>name</i>      | The name of the endpoint we are configuring. [R]                         |
| 2. <i>max_bytes</i> | The max number of payload bytes to print out, default is 128. [R][D:128] |

Syntax: `show_endp_payload name max_bytes`

### 232. **show\_files**

Show files in a particular directory. All paths are relative to the LANforge base directory (usually `/home/lanforge/`). You can also add a filter, such as `*.txt` If `key` is specified, it will be returned as the first line in the response. Directory and filter do not have to be specified, or can be NA to be left at defaults.

```

SORT_BY_TIME | 1 # Sort by date/time

```

| Argument            | Description  |
|---------------------|--|
| 1. <i>shelf</i>     | The virtual shelf to search in. Use 0 for manager machine. [R,0-1] |
| 2. <i>resource</i>  | The machine to search in. [W]                                      |
| 3. <i>key</i>       | A special key, can be used for scripting.                          |
| 4. <i>directory</i> | The sub-directory in which to list.                                |
| 5. <i>filter</i>    | An optional filter, as used by the 'ls' command.                   |
| 6. <i>dir_flags</i> | Determines format of listing, see above.                           |

Syntax: `show_files shelf resource key directory filter dir_flags`

### 233. **show\_ports**

Show one/all ports for one/all resources in one/all shelves.

#### Probe-Flags:

```

WIFI | 1 # show wifi ports
MII | 2 # show MII ports
ETHTOOL | 4 # ethtool results
BRIDGE | 8 # show bridge ports
EASY_IP_INFO | 16 # show Everything but gateway, which is expensive to probe.
GW | 32 # show gateway
GW_FORCE_REFRESH | 64 # Force GW (re)probe. (Otherwise, cached values *might* be used.)
DHCP_KEYED_MSG | 128 # Show (only) the HANDLE_DHCP- keyed message.

```

| Argument              | Description   |
|-----------------------|---|
| 1. <i>shelf</i>       | Name/id of the shelf, or 'all'. [R][D:1]  |
| 2. <i>resource</i>    | Resource number, or 'all'. [W]  |
| 3. <i>port</i>        | Port number, or 'all'. [W]  |
| 4. <i>probe_flags</i> | See above, add them together for multiple probings. Leave blank if you want stats only. |

Syntax: `show_ports shelf resource port probe_flags`

#### 234. **show\_channel\_groups**

Show one/all ChannelGroups for one/all resources in one/all shelves. An empty specifier will be treated as 'all'.

| Argument               | Description                              |
|------------------------|--|
| 1. <i>shelf</i>        | Name/id of the shelf, or 'all'. [R][D:1] |
| 2. <i>resource</i>     | Resource number, or 'all'. [W]           |
| 3. <i>channel_name</i> | Name of the channel, or 'all'. [W]       |

Syntax: `show_channel_groups shelf resource channel_name`

#### 235. **show\_spans**

Show one/all Spans for one/all resources in one/all shelves. An empty specifier will be treated as 'all'.

| Argument              | Description                              |
|-----------------------|--|
| 1. <i>shelf</i>       | Name/id of the shelf, or 'all'. [R][D:1] |
| 2. <i>resource</i>    | Resource number, or 'all'. [W]           |
| 3. <i>span_number</i> | Span-Number of the span, or 'all'. [W]   |

Syntax: `show_spans shelf resource span_number`

#### 236. **show\_ppp\_links**

Show one/all PPP Links for one/all resources in one/all shelves. An empty specifier will be treated as 'all'.

| Argument           | Description                                |
|--------------------|--|
| 1. <i>shelf</i>    | Name/id of the shelf, or 'all'. [R][D:1]   |
| 2. <i>resource</i> | Resource number, or 'all'. [W]             |
| 3. <i>link_num</i> | Ppp-Link number of the span, or 'all'. [W] |

Syntax: `show_ppp_links shelf resource link_num`

#### 237. **show\_tm**

Show one or all test managers.

| Argument           | Description                                |
|--------------------|--|
| 1. <i>test_mgr</i> | Can be name of test manager, or 'all'. [R] |

Syntax: `show_tm test_mgr`

#### 238. **show\_group**

Show one or all Test Groups.

| Argument        | Description   |
|-----------------|---|
| 1. <i>group</i> | Can be name of test group. Use 'all' or leave blank for all groups. |

Syntax: `show_group group`

239. **show\_venue**

Show one or more venues

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]  |
| 2. <i>resource</i> | Resource number, or 'ALL' [W]   |
| 3. <i>venu_id</i>  | Number to uniquely identify this venue on this resource, or 'ALL' [W] |

Syntax: `show_venue shelf resource venu_id`

240. **show\_wps**

Show one or all WanPaths for one or all WanLink Endpoints.

| Argument           | Description                     |
|--------------------|---------------------------------|
| 1. <i>endpoint</i> | Name of endpoint, or 'all'. [W] |
| 2. <i>wanpath</i>  | Name of wanpath, or 'all'. [W]  |

Syntax: `show_wps endpoint wanpath`

241. **shutdown**

Restart the LANforge Manager server. Restarting the manager will cause interruption to all of the Resource processes as well. If you want to restart all LANforge processes on the Manager machine, enter 'YES' for the third argument (and probably 'NA' for the second)

| Argument            | Description  |
|---------------------|--|
| 1. <i>really</i>    | Must be 'YES' for command to really work.  |
| 2. <i>chdir</i>     | Directory to cd to before dying. Only useful when using gprof to debug, or 'NA' to ignore.   |
| 3. <i>serverctl</i> | Enter 'YES' to do a <code>./serverctl.bash restart</code> to restart all LANforge processes. |

Syntax: `shutdown really chdir serverctl`

242. **shutdown\_resource**

This will restart the LANforge processes on the resource specified. This will cause all tests that are utilizing that resource to be destroyed. Depending on how the system is set up, the remote resource will probably be restarted in about 5 seconds.

| Argument           | Description                    |
|--------------------|--------------------------------|
| 1. <i>shelf</i>    | Shelf number, or ALL. [R][D:1] |
| 2. <i>resource</i> | Resource number, or ALL. [W]   |

Syntax: `shutdown_resource shelf resource`

243. **shutdown\_os**

This will bring down the Operating System on the resource specified, including all processes running on it. Only a power-cycle will bring it back up again. This command should be used before powering down the LANforge resources. Wait about 1 minute before shutting off the power to allow the OS to bring itself down gracefully. See also: `reboot_OS`

| Argument           | Description                    |
|--------------------|--------------------------------|
| 1. <i>shelf</i>    | Shelf number, or ALL. [R][D:1] |
| 2. <i>resource</i> | Resource number, or ALL. [W]   |

Syntax: `shutdown_os shelf resource`

244. **sniff\_port**

This will attempt to launch the Wireshark program on the specified port's machine and display Wireshark to the specified X server. Wireshark will be tried first, but if it is not found, Ethereal will be

attempted. You must be running X, and have allowed other machines to connect to your X server. If you do not specify the `DISPLAY`, LANforge will attempt to guess it based on your connecting IP address.

For PCs, you can use the [exceed program from Hummingbird software](#).

To enable X access on Unix/Linux, run this command:

```
xhost +
```

This can open your machine to security threats, so read up on `xhost` before you run this command on a mission critical machine not protected by a good firewall!

Flags are defined as follows.

```
TSHARK          | 0x1  # Use command-line tshark instead of wireshark
DUMPCAP         | 0x2  # Use command-line dumpcap, more efficient than tshark
MATE_TERMINAL   | 0x4  # Launch tshark/dumpcap in mate-terminal
MATE_XTERM      | 0x8  # Launch tshark/dumpcap in xterm
MATE_KILL_DUMPCAP | 0x10 # Kill last dumpcap
```

Learn more about [Wireshark program](#).

For questions specific to LANforge, you should contact Candela Technologies.

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]  |
| 2. <i>resource</i> | Resource number. [W]  |
| 3. <i>port</i>     | The port we are trying to run the packet sniffer on. [R]  |
| 4. <i>display</i>  | The DISPLAY option, for example: 192.168.1.5:0.0. Will guess if left blank.                                   |
| 5. <i>flags</i>    | Flags that control how the sniffing is done.  |
| 6. <i>outfile</i>  | Optional file location for saving a capture.  |
| 7. <i>duration</i> | Duration for doing a capture (in seconds). Default is 5 minutes for dumpcap/tshark, and forever for wireshark |
| 8. <i>snaplen</i>  | Amount of each packet to store. Default is to store all of it.  |

Syntax: `sniff_port shelf resource port display flags outfile duration snaplen`

#### 245. **tail**

Deal with 'tailing' a file. This is usually going to be a log file.

This displays a GUI popup. This does not stream text to JSON. If you need to see the end of a log file, use [logfile](#)

| Argument           | Description   |
|--------------------|---|
| 1. <i>shelf</i>    | Shelf that holds the resource that holds the file. [R][D:1]             |
| 2. <i>resource</i> | Resource that holds the file. [W]                                       |
| 3. <i>cmd</i>      | Command: start, stop, results   |
| 4. <i>key</i>      | File-name that we should be tailing.                                    |
| 5. <i>message</i>  | The contents to display (for results only) <code>Unescaped Value</code> |

Syntax: `tail shelf resource cmd key message`

#### 246. **tm\_register**

When a client is registered with a test manager, the manager will send the client reports at specified intervals (see `set_tm_rpt`).

| Argument              | Description   |
|-----------------------|---|
| 1. <i>test_mgr</i>    | Name of test manager (can be all.) [R]                        |
| 2. <i>client_name</i> | Name of client to be registered. (dflt is current client) [W] |

Syntax: `tm_register test_mgr client_name`

247. **tm\_unregister**

The client will receive no more un-requested reports from the test manager(s).

| Argument              | Description   |
|-----------------------|---|
| 1. <i>test_mgr</i>    | Name of test manager (can be all.) [R]                            |
| 2. <i>client_name</i> | Name of client to be un-registered. (dfllt is current client) [W] |

Syntax: `tm_unregister test_mgr client_name`

248. **version**

Print out the version of the LANforge server.

Syntax: `version`

249. **wiser\_reset**

This command will reset the WISER library on the specified machine. This is only useful when running with the Telcordia WISER module.

| Argument           | Description                    |
|--------------------|--------------------------------|
| 1. <i>shelf</i>    | Shelf number, or ALL. [R][D:1] |
| 2. <i>resource</i> | Resource number, or ALL. [W]   |

Syntax: `wiser_reset shelf resource`

250. **who**

Show who is currently logged into the system.

Syntax: `who`

251. **wifi\_event**

This is used internally by LANforge to listen for WiFi events.

| Argument         | Description                          |
|------------------|--------------------------------------|
| 1. <i>device</i> | Interface or PHY in most cases. [R]  |
| 2. <i>event</i>  | What happened. [R]                   |
| 3. <i>status</i> | Status on what happened.             |
| 4. <i>msg</i>    | Entire event in human readable form. |

Syntax: `wifi_event device event status msg`

252. **wifi\_cli\_cmd**

LANforge WiFi station interfaces are controlled by the `wpa_supplicant` process, which can be directly manipulated with the `wpa_cli` command. For normal LANforge use, users will not need to deal directly with `wpa_supplicant` or `wpa_cli`. For more advanced features, such as roaming, users will need to use `wpa_cli` commands directly. This LANforge API makes that easier to accomplish.

Example:

```
wifi_cli_cmd 1 1 sta1 'roam 00:00:01:01:01:02'
```

NOTE: These commands will be queued if the interface is phantom, but otherwise the commands will be sent to the `wpa_cli` command immediately. This can collide with automated LANforge actions such as automatically re-associating and interface that was dropped by the AP. Any configuration changes made by this method will not be saved through restarts of LANforge or even through network interface resets.

If the port is a VAP, then the command will be passed to the `hostapd_cli` process in a similar manner.

| Argument        | Description            |
|-----------------|------------------------|
| 1. <i>shelf</i> | Shelf number. [R][D:1] |

|                       |  |
|-----------------------|--|
| 2. <i>resource</i>    | Resource number. [W]   |
| 3. <i>port</i>        | Name of the WiFi station or AP interface to which this command will be directed. [R] |
| 4. <i>wpa_cli_cmd</i> | Command to pass to wpa_cli or hostap_cli. This must be single-quoted. [R]            |

Syntax: `wifi_cli_cmd shelf resource port wpa_cli_cmd`

### 253. **xorpsh**

Connect to a Virtual Router's xorpsh shell. The **cmd** parameter value `display` determines what X11 screen the terminal will appear on. (You must be running X windows on the target DISPLAY system.) For `display` mode, the argument is the display address and screen number found in typical DISPLAY environment variables for this instance of xorpsh. Example:

```
DISPLAY=localhost:10.0
```

For 'run\_cmd' mode, it is the command (in single quotes) to pass to the xorpsh process.

| Argument           | Description  |
|--------------------|--|
| 1. <i>shelf</i>    | Shelf number. [R][D:1]   |
| 2. <i>resource</i> | Resource number. [W]   |
| 3. <i>router</i>   | Name of the virtual router. [R]  |
| 4. <i>cmd</i>      | Determines action, current commands: <code>display</code> , <code>run_cmd</code> |
| 5. <i>arg</i>      | See above.   |

Syntax: `xorpsh shelf resource router cmd arg`