UCOPIA Command Line Interface

Version 5.0

Table des matières

1 Introduction

This document describes the CLI (*Command Line Interface*) UCOPIA. The interface gives access to certain network, system, or advanced administration commands.

The CLI also lets you diagnose problems or malfunctions.

2 Accessing the CLI

The CLI can be accessed in different ways: from the Web administration tool, by connecting to the UCOPIA controller with a screen and keyboard, or even by connecting to the UCOPIA controller with SSH (on an administration VLAN).

To access the CLI from the Web administration tool, click on the "**CLI access**" item of the menu bar.
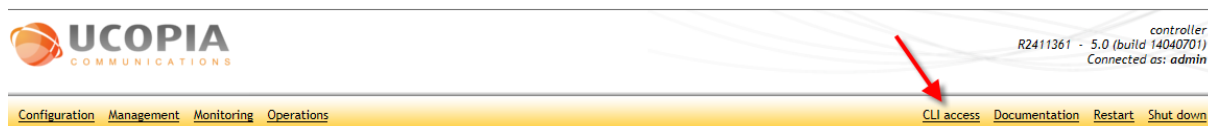
Figure 2.1. CLI access from the administration tool

The following page is displayed :
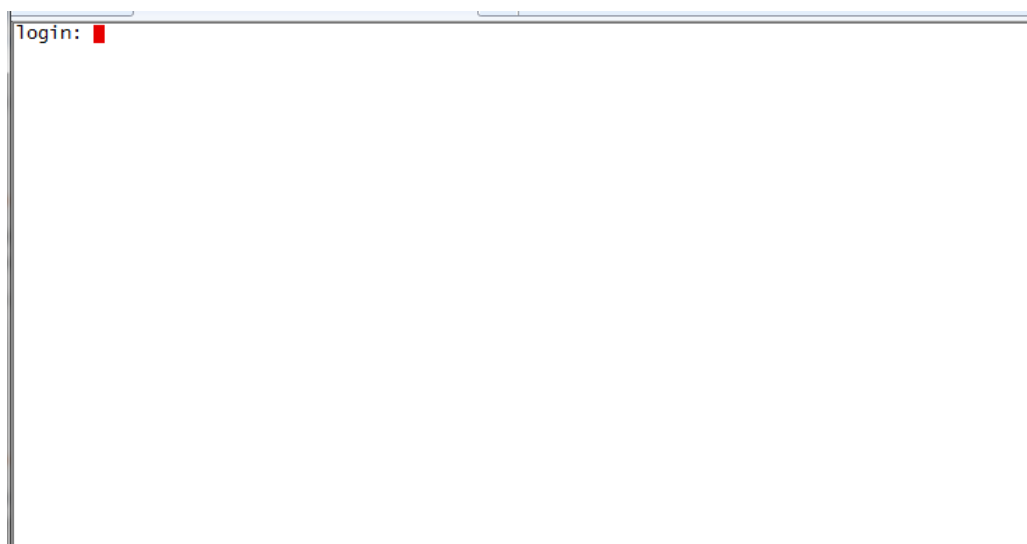
Figure 2.2. CLI Authentication page

Enter the login and the password. The prompt is displayed.

```
controller login: admin
admin@controller's password:

************************************************
              UCOPIA

        Production name    SV200
        Serial number      R2411361
        Production type    product
        Current version    5.0
        Current build      14040701
        Last upgrade       -

           Command Line Interface
************************************************

Welcome

Please use '?' to display context command help
Write the command name followed by '?' to display help about this command
Example : ping ?, will display Ping command help (The ? won't be visible)

> █
```

Figure 2.3. CLI Authentication

Note

Connection credentials (login/password) are admin/ bhu85tgb.

3 Commands

To display the list of available commands use the "**?**" key.

```
>

accessLimitationAdmin      List / Add / Remove limitations to access Web Administration Tools
activateLicense            Install license. In case of virtual appliance, the activation key must be specified
addSubnet                  Add incoming or outgoing subnet
adminInterface             Configure the network parameters of the admin interface
arp                        Show ARP cache
arping                     Send an ARP request to a neighbour host
autodisconnect             Sets the criteria for user automated disconnection
bzip2                      A block-sorting file compressor
delSubnet                  Delete a subnet
dhclient                   Get DHCP distributed IP address
dhcpLease                  manage DHCP leases
dnsRedirect                enable/disable DNS redirection
dnsspoofing                enable/disable DNS spoofing
enableLogLevel             Change logs level
exit                       Exit this CLI session
```

Figure 3.1. CLI command list

Commands can be classified in many categories:

3.1   Configuration

Command	Description
accessLimitationAdmin	List, add, delete access to the web administration tool
addSubnet	Add a new incoming or outgoing subnetwork
adminInterface	Configuration of administration parameters
autodisconnect	Configuration of the mechanism for automatically connecting users.
delSubnet	Deletion of an incoming or outgoing subnetwork
dhcpLease	Fixed DHCP leases management
dnsSpoofing	Activate or déactivate DNS spoofing
dnsRedirect	Activate or deactivate DNS redirection.
manageDhcpLease	Deletion and cleaning of DHCP leases
modifyNativeIP	Modification of IP address and netmask for incoming and outgoing VLANs

Command	Description
passwd	Modification the password for the administration tools and the CLI.
restoreConfiguration	Restore a remote configuration backup
service	Verification of the status of, and restart, UCOPIA services.
staticRoutes	Add, delete or list static routes
windowsDomainRegisteredMAC	Manage Registered MAC address used for devices authentication

3.1.1 accessLimitationAdmin

List, add, or delete access to the web administration tool.

```
> accessLimitationAdmin [-A action] [-I interface]
```

Options:

- *-A action (list | add | del)*: Action to perform (list, add, delete).
- *-I interface*: Interface on which it's possible to limit or authorize access to the web administration tool. For ingoing interfaces, use *out* or *outX* and for incoming interfaces, use *in* or *inY* (X and Y are the VLAN numbers)

Examples:

- To remove access on the incoming VLAN 1

```
> accesLimitationAdmin -A del -I in1
```

- To add access on the incoming VLAN 1

```
> accesLimitationAdmin -A add -I in1
```

3.1.2 addSubnet

Add incoming or outgoing subnetwork

```
> addSubnet <subnet_type> <subnet_ip> <netmask> [-i vlan_id] [-c controller_ip] [-g gateway] [-l label] [-k
comment] [-z zone] [-a admin_access] [-A deleg_access] [-d dhcp_state] [-r router] [-S dns] [-R range] [-N
ntp] [-w netbios] [-D domain]
```

- *subnet_type* : Type of the new subnetwork ((local-in | local-out | remote-in). local-in and local-out correspond to incoming and outgoing VLAN. remote-in corresponds to incoming subnetwork.
- *subnet_ip* : Subnetwork IP address.
- *netmask* : Subnetwork netmask.

Options :

- *-i vlan_id* : [Mandatory for: local-in / local-out] VAN ID for the incoming or outgoing VLANs []
- *-c controller_ip* : [Mandatory for : local-in / local-out] Controller IP address for incoming or outgoing subnetworks []
- *-g gateway* : [Mandatory for : local-out] Gateway address for outgoing subnetworks []
- *-l label* : [Mandatory for : local-out / remote-in] Label for incoming subnetworks (betweeb double quote, e.g. : -l « mt label » []
- *-k comment :* [Optional for : local-out / remote-in] A free comment describing the incoming subnetworks (between double quote, e.g. : -k « my comment » []
- *-z zone :* [Optional for : local-out / remote-in] An existing zone associated to the incoming subnetwork [Défault]
- *-a admin_acess :* [Optional for: all types] Administration tool access [y]
- *-A deleg_acess :* [Optional for : all types] Delegation portal access [y]
- *-A dhcp_state :* [Optional for : local-out / remote-in] DHCP on the incoming subnetwork (y/n) [n]
- *-r router :* [Mandatory if dhcp_state = y] The subnetwork gateway []
- *-S dns :* [Mandatory if dhcp_state = y] The subnetwork DNS []
- *-R range :* [Optional for dhcp_state] The IP address ranges for the subnetwork (separated by '-') []
- *-N ntp :* [Optional for dhcp_state] The NTP server associated to the subnetwork []
- *-w netbios :* [Optional for dhcp_state] The netbios (wins) server associated to the subnetwork []
- *-D domain* : [Optional for dhcp_state] The domain name for the subnetwork []

Example :

```
> addSubnet local-in 192.168.10.0 255.255.255.0 -i 10 -c 192.168.10.254 -l label10 -k "Commentaire" -a y -A y -d y -r
192.168.10.254 -S 192.168.10.254 -R 192.168.10.10-192.168.10.200 -N 192.168.10.254 -w 192.168.10.25

Checking values…

Adding local-in 'label10'…

Reconfiguring and restarting services…

Incoming subnet 'label10' added successfully
```

3.1.3 adminInterface

Allow to configure parameters for the administration interface

```
> adminInterface [action] [-a autonego] [-s speed] [d duplex] [-m mtu] [-i ip_address] [-n netmask]
```

■ action : get/set (display or modify the configuration) or 'help' (display help) [get]

Options :

■ *-a autonego* : Active or deactive auto-négociation (y|n) []
■ *-s speed* : Speed (10 or 100 Mb/s) []
■ *-d duplex* : Duplex (half or full) []
■ *-m mtu* : MTU []
■ *-i ip_address* : IP address of the controller []
■ *-n netmask* : Netmask for administration interface []

Example :

```
> adminInterface set -a y -s 100 -d full -m 1500 -i 192.168.0.1 -n 255.255.255.0

[NOTICE] admin interface available

[NOTICE] admin interface will be configured

[NOTICE] adding vlan in

[NOTICE] adding vlan out

[NOTICE] rewrite /etc/network/interfaces succeeded

[NOTICE] Configuring ifplugd daemon on interfaces 'eth0 eth1 eth2 eth3'

mounting admin
```

3.1.4 autodisconnect

Allow to configure the mechanism for disconnection of users.

```
> autodisconnect [action] [-p period] [-m mode] [-l layer] [-n nb_checks]
```

■ action : Possible values: 'get', 'set' (display or modify the configuration) or 'help' [get]

Options :

■ *-p period* : Period between checks of connected users (in seconds) []

- *-m mode* : Possible values: 'passive' or 'active'. In passive mode, network data available in the controller is used to detect the device's presence. In active mode, the device will be polled to detect its presence on the network. The active mode is allowed when using layer 2 only []
- *-l layer* : Possible values: 2 (Ethernet) or 3 (IP). The layer level at which the daemon will work []
- *-n nb_checks* : How many consecutive failed checks should trigger user disconnection []

Examples :

- Display the *auto-disconnect* configuration :

```
> autodisconnect get
period: 60
nb checks: 5
mode: passive
layer: 2
```

- Configuration of *auto-disconnect* to have user connection checked every 10 seconds and disconnected after 3 failed checks:

```
> > autodisconnect set -p 10 -n 3
Reloading autodisconnect service: autodisconnect.
```

3.1.5 delSubnet

Deletion of a subnetwork.

```
> delSubnet <subnet_type> [-i vlan_id] [-s subnet_ip] [-n netmask]
```

- *subnet_type* : Subnetwork type (local-in | local-out | remote-in)

Options :

- *vlan_id* : [Mandatory for : local-in / local-out] The name of the incoming or outgoing VLAN.
- *subnet_ip* : [Mandatory for : remote-in] The subnet IP address for the remote incoming network.
- *netmask* : [Mandatory for : remote-in] The netmask for the remote incoming network.

Example :

```
> delSubnet local-in -i 10 -s 192.168.10.0 -n 255.255.255.0

Deleting VLAN 10...

Reconfiguring and restarting services...

The incoming VLAN 10 deleted successfully.
```

3.1.6 dhcpLease

Fixed DHCP leases management

```
dhcp_lease [command] [name] [mac] [ip]
```

Options :

- *command* : Lists, add or removes a DHCP lease (list | add | delete) [list]
- *name* : Subnetwork name []
- *mac* : MAC address of the DHCP lease []
- *ip* : IP address of the DHCP lease []

Examples :

- Add a new fixed DHCP lease

```
> dhcpLease add in3 01:02:03:04:05:06 192.168.250.6
```

- Remove a fixed DHCP lease

```
> dhcpLease delete in3 01:02:03:04:05:06
```

- List all existing fixed DHCP leases

```
> dhcpLease list
### Subnet in3
 # MAC : 01:02:03:04:05:06, IP : 192.168.250.6
```

3.1.7 dnsSpoofing

Activate or deactivate DNS spoofing.

```
> dnsSpoofing <action>
```

- *action* : enable | disable

Examples :

- Activating DNS spoofing

```
> dnsSpoofing enable
Restarting DNS forwarder: dnsmasq.
```

- Deactivating DNS spoofing

```
> dnsSpoofing disable
Restarting DNS forwarder: dnsmasq.
```

3.1.8   dnsRedirect

Activates or deactivates DNS redirection.

```
> dnsRedirect <action>
```

■   *action* : enable | disable

Examples:

■   Activating DNS redirection

```
> dnsRedirect enable
```

■   Deactivating DNS redirection

```
> dnsRedirect disable
```

3.1.9   manageDhcpLeases

Deletion and cleaning of DHCP leases

```
manageDhcpLeases [-A action] [-I interface]
```

Options:

■   *-A action* : The intended action (clean | remove)   []
   ➤   *Clean* : All active DHCP leases are tested (arping on IP addresses) and removed if unused.
   ➤   *Remove* : Remove all DHCP leases.
■   *-I interface* : Type of the interface to be cleaned. Use **in** or **inX** (X being the VLAN number) (only for the "clean" action) []

Examples:

■   Clean DHCP leases for interface in3

```
>  manageDhcpLeases -A clean -I in3
!!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!!
! CAUTION: All modifications will be done at your own risks.
! Every active lease will be tested (arping on IP address) and removed if there is no
response.
! DHCP server will be down during this operation.
```

```
! Only in3 iface leases will be parsed.
!
! Do you wish to continue ? (y/N)
!!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!!
v
Stopping ISC DHCP server: dhcpd.
Starting ISC DHCP server: dhcpd.
Please check the DHCP logs with the CLI showLogs command.
+ Cleaning leases on in3             [ DONE ]
```

■ Remove all DHCP leases

```
>  manageDhcpLeases -A remove
!!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!!
! CAUTION: All modifications will be done at your own risks.
! All the DHCP leases will be removed.
! DHCP server will be down during this operation.
!
! Do you wish to continue ? (y/N)
!!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!! !!!!!
v
Stopping ISC DHCP server: dhcpd.
Starting ISC DHCP server: dhcpd.
Please check the DHCP logs with the CLI showLogs command.
+ Deleting leases on all incomming ifaces             [ DONE ]
```

3.1.10 modifyNativeIP

Allow to modify IP addresses and netmasks for incoming and outgoing networks.

```
> modifyNativeIP <vlan_type> <controller_ip> <netmask> [-g gateway]
```

■ *vlan_type* : VLAN type [in | out].

■ *controller_ip* : New IP address of the controller

■ *netmask* : New netmask of the controller.

Options:

■ *-g gateway* : IP address gateway for the native outgoing VLAN [].

Example :

```
> modifyNativeIP in 192.168.200.254 255.255.255.0

DHCP configuration will be deactivated on the native incoming network

Checking values...

Modifying native incoming network...

Reconfiguring and restarting services...

Native incoming network modified successfully.
```

3.1.11 passwd

Allow to modify the password for the administration tools and the CLI.

```
> passwd <type>
```

- *type* : web | cli (specify 'web' for the administration tool).

3.1.12 restoreConfiguration

Restore a remote configuration backup.

```
> restoreConfiguration <url> [-r recursive] [-c continue] [--no-check-certificate
no check certificate] [--passive-ftp passive ftp] [--no-proxy no proxy] [--proxy-ip
proxy ip] [--proxy-port proxy port] [--proxy-user username] [--proxy-passwd passwd] [--no-
reboot no_reboot]
```

- *url* : Address of the file containing the configuration backup

Options:
- *-r*: Turn on recursive retrieving [n]
- *-c*: Continue getting a partially-downloaded file [n]
- *--no-check-certificate*: Do not try to validate the server SSL certificate [n]
- *--passive-ftp*: FTP mode passive [n]
- *--no-proxy*: Do not use proxy [n]
- *--proxy-ip proxy_ip*: DNS or IP address of a proxy server []
- *--proxy-port proxy_port*: Port of a proxy server (1..65535) []
- *--proxy-user username*: Username for the authentication on a proxy server []
- *--proxy-passwd passwd*: Password for the authentication on a proxy server []
- *--no-reboot*: Don't reboot the server [n]

Warning

Reboot must be done to complete the restoration.

3.1.13 Service

Allow to verify the status of, and restart, UCOPIA services.

```
> service <service> <action>
```

- *action* : restart | status
- *service*: arp-protect, authserver, autodisconnect, dhcp, dns, high-availability, ldap, ldapmonitor, time-server, printers, printers-sharing, radius, sql, ssh, web-proxy, web-server, zeroconf-ip.

Example :

- Status of proxy services configured in the UCOPIA controller.

```
> service proxy status

privoxy (pid 3093) est en cours d'exécution...
squid (pid 3183) est en cours d'exécution...
```

3.1.14 staticRoutes

Add, delete or list static routes.

```
> staticRoutes [action] [-s subnet] [-n netmask] [-g gateway] [-i interface]
```

- *action* : add | del | list | help

Options :

- -s subnet : The remote subnet address []
- -n netmask: The remote subnet mask []
- -g gateway: The gateway that should be used to reach the remote subnet []
- *-i interface*: The interface that should be used by this static route []

Example :

```
> staticRoutes add -s 10.0.10.0  -n 255.255.255.0  -g 192.168.213.1  -i in
>
> staticRoutes list
Current routes:
    net 10.0.10.0  netmask 255.255.255.0  gw 192.168.213.1  dev in

>
> staticRoutes add -s 10.0.40.0  -n 255.255.255.0  -g 192.168.8.1  -i in800
>
> staticRoutes list
Current routes:
    net 10.0.10.0  netmask 255.255.255.0  gw 192.168.213.1  dev in
    net 10.0.40.0  netmask 255.255.255.0  gw 192.168.8.1  dev in800
>
> staticRoutes del -s 10.0.40.0  -n 255.255.255.0  -g 192.168.8.1  -i in800
>
> staticRoutes list
```

```
Current routes:
    net 10.0.10.0  netmask 255.255.255.0  gw 192.168.213.1  dev in
```

3.1.15 windowsDomainRegisteredMAC

Manage registered MAC address used for devices authentication.

```
> windowsDomainRegisteredMAC [-A action] [mac_address]
```

- *-A action*: The action to attempt. List : list all MAC addresses in list. Del : remove given MAC address from list. (list|del) [list]
- *mac_address*: Which MAC address to find/remove []

Example :

- Show currently registered MAC addresses

```
> windowsDomainRegisteredMAC
##### Display all MAC addresses list
00:1D:E0:62:D6:05
00:0c:f1:4c:2e:a9
```

3.2   FTP server

Command	Description
addFTPAccount	Add an FTP account
deleteFTPAccount	Delete an FTP account
modifyFTPAccount	Modify an FTP account
listFTPAccounts	Display all available FTP accounts

3.2.1   addFTPAccount

Add an FTP account.

```
> addFTPAccount <accountId> <host><port> <securedMode><communicationMode> <login> <path>
```

- *accountId*: Account name
- *host* : IP/DNS address of the FTP server
- *port* : Port number
- *secureMode* : Secure connection   (on|off)
- *communicationMode*: Communication mode (passive|active)
- *login* : Account login
- *path*: Account access path

Example :

```
> addFTPAccount account1 host1 21 on active login2 /updates
User password:
Account 'account1 added
```

.

3.2.2   deleteFTPAccount

Delete an FTP account.

```
> deleteFTPAccount <accountId>
```

- *accountId*: Accoun name

Example :

```
> deleteFTPAccount account1
Account 'account1 deleted
```

3.2.3 modifyFTPAccount

Modify an FTP account.

```
> modifyFTPAccount <accountId> <host> <port> <securedMode> <communicationMode> <login> <path>
```

- *accountId*: Account name
- *host* : IP/DNS of the FTP server
- *port* : Port number
- *secureMode* : secure connection (on|off)
- *communicationMode*: Communication mode (passive|active)
- *login* : Account login
- *path*: Account access path

Example :

```
> modifyFTPAccount account1 host1 21 off active login2 /updates
User password:
Account 'account2 modifed
```

3.2.4 listFTPAccounts

Display all available FTP accounts.

```
> listFTPAccounts [detailed]
```

Options

■   *detailed* : detailed list (y|n) [n]

Example :

```
> listFTPAccounts y
Available FTP accounts:
 Id: account1
   host: host1
   port: 21
   securedMode: off
   communicationMode: passive
   login: login2
   userPassword: *******
   path: /updates/
```

3.3   License

Command	Description
activateLicense	Install the license from the Services Management Platform.
installLicense	Download the license from an FTP server and install it.

3.3.1   activateLicense

Install the license from the Services Management Platform. This command needs an internet access in order to access the SMP. In case of virtual appliance, the activation key must be given.

```
> activateLicense <reseller_lastname> <reseller_firstname> <reseller_mail> <reseller_phone>
<customer_lastname> <customer_firstname> <customer_mail> <customer_phone> [-k activation_key]
```

- *reseller_lastname* : Reseller lastname
- *reseller_firstname* : Reseller firstname
- *reseller_mail* : Reseller email
- *reseller_phone* : Reseller phone number
- *customer_lastname* : Customer lastname
- *customer_firstname* : Customer firstname
- *customer_mail* : Customer email
- *customer_phone* : Customer phone number

Options :

- *-k activation_key* : Activation key for virtual appliance (ex: 876GT-564FR-5RDE3-GFDR4) []

3.3.2 installLicense

Download the license from an FTP server and install it.

```
> installLicense <license_link>
```

- *license_link* : Link to access the license

Example :

```
> installLicense ftp://user:password@host/license.tgz

INSTALLING LICENSE: OK
```

3.4 Certificate

Command	Description
installCertificate	Download a certificate (HTTPS or RADIUS) from an FP server and install it.
restoreCertificate	Restore a certificate

3.4.1   installCertificate

Download a certificate (HTTPS or RADIUS) from an FP server and install it.

---

installCertificate <type> <ca_file> <cert_file> <server_key>

- *type*: Certificate type (HTTPS/RADIUS)
- *ca_file* : Certificate of the certification Authority (CA)
- *cert_file* : Controller certificate
- *server_key*: Private key of the certificate controller

Examples :

> installCertificate https ftp://user:password@host/certs/cacert.pem

> ftp://user:password@host/certs/server.pem

> ftp://user:password@host/certs/server_key.pem

Private key password:

Downloading cacert.pem

Downloading server.pem

Downloading server_key.pem

Installing HTTPS certificates

> installCertificate radius

> ftp://user:password@host/certs/radius/ca.validation.lan.pem

> ftp://user:password@host/certs/radius/radius.validation.lan.pem

> ftp://user:password@host/certs/radius/radius_priv_key.pem

Private key password:

Downloading ca.validation.lan.pem

Downloading radius.validation.lan.pem

Downloading radius_priv_key.pem

Installing Radius certificates

3.4.2   restoreCertificate

Restore a certificate.

```
restoreCertificate <type>
```

■   *type*: Certificate type (HTTPS/RADIUS)

Example :

```
> restoreCertificate  https

Restoring https certificate

https certificates restored
```

3.5 Logs

Command	Description
enableLogLevel	Enable or disable system, PMS and RADIUS logs.
mysqlReadSessions	Displays user session tables.
showLogs	Displays internal logs.

3.5.1   enableLogLevel

Set the system, PMS and RADIUS log levels.

```
> enableLogLevel <type> <level>
```

■   `type`: core/pms/radius (UCOPIA controller core, PMS or RADIUS)
■   `level`: Debug level

Logs type	Level
core	EMERG	ALERT	CRIT	ERR	WARNING	NOTICE	INFO	DEBUG
pms	DEBUG	WARNING

Logs type	Level
radius	LOG_NOTICE	LOG_INFO	LOG_DEBUG

Example :

■ Setting the core log level to WARNING

```
> enableLogLevel core WARNING
```

3.5.2 mysqlReadSessions

Display user session tables.

```
> mysqlReadSessions
```

Example :

```
> mysqlReadSessions
192.168.14.234 00:27:10:0b:b9:74 seb MODAPPAPI UCOPIA 2014 1
192.168.14.237 3c:74:37:11:11:b3 pat MODAPPAPI UCOPIA 2014 1
192.168.14.232 E0:F8:47:56:A6:87 greg MODMAC  UCOPIA 2014 1
```

3.5.3 showLogs

Display internal logs.

```
> showLogs [type] [-n rows_number]
```

Options:

■ `type` : all | radius | core | autodisconnect | dhcp | pms [all]
■ `-n rows_number`: Number of line to display [1000]

Examples:

■ To display DHCP logs: showLogs dhcp

```
>
> showLogs dhcp
Apr 15 09:37:21 localhost dhcpd: Wrote 1 leases to leases file.
Apr 15 09:38:22 localhost dhcpd: DHCPREQUEST for 192.168.7.129 from 00:1d:e0:62:d6:05 (SEVEN-PC) via in214: wrong ne
Apr 15 09:38:22 localhost dhcpd: DHCPNAK on 192.168.7.129 to 00:1d:e0:62:d6:05 via in214
Apr 15 09:38:22 localhost dhcpd: DHCPDISCOVER from 00:1d:e0:62:d6:05 via in214
Apr 15 09:38:23 localhost dhcpd: DHCPOFFER on 192.168.214.1 to 00:1d:e0:62:d6:05 (SEVEN-PC) via in214
Apr 15 09:38:23 localhost dhcpd: execute_statement argv[0] = /usr/sbin/dhcp_event
Apr 15 09:38:23 localhost dhcpd: execute_statement argv[1] = commit
Apr 15 09:38:23 localhost dhcpd: execute_statement argv[2] = 192.168.214.1
Apr 15 09:38:23 localhost dhcpd: execute_statement argv[3] = 0:1d:e0:62:d6:5
Apr 15 09:38:23 localhost dhcpd: execute_statement argv[4] = 3600
Apr 15 09:38:23 localhost dhcpd: execute_statement argv[5] = SEVEN-PC
Apr 15 09:38:23 localhost dhcpd: DHCPREQUEST for 192.168.214.1 (192.168.214.253) from 00:1d:e0:62:d6:05 (SEVEN-PC) v
Apr 15 09:38:23 localhost dhcpd: DHCPACK on 192.168.214.1 to 00:1d:e0:62:d6:05 (SEVEN-PC) via in214
Apr 15 09:38:26 localhost dhcpd: DHCPINFORM from 192.168.214.1 via in214
Apr 15 09:38:26 localhost dhcpd: DHCPACK to 192.168.214.1 (00:1d:e0:62:d6:05) via in214
Apr 15 09:40:11 localhost dhcpd: DHCPINFORM from 192.168.214.1 via in214
Apr 15 09:40:11 localhost dhcpd: DHCPACK to 192.168.214.1 (00:1d:e0:62:d6:05) via in214

^C>
```

■ To display user connection logs: showLogs core

```
>
> showLogs core -n 10
Apr 15 09:41:02 localhost authserver [DEBUG] CoreUpdateServer: new connection, threadid=64751, total=1
Apr 15 09:41:02 localhost authserver [DEBUG] CoreUpdateServer: unRegister, total=0
Apr 15 09:41:02 localhost monitorscript.sh[19880]: connected users: 0 (local) / 0 (total) / 0 (authserver) / 0 (ipset); us/sy/id/wa=0/0/99/1; load=0.01, 0
ree_mem=66.5%; used_swap=0.0%; Tasks: 176 total, 1 running, 175 sleeping, 0 stopped, 0 zombie; uptime: 2 days, 20:30
Apr 15 09:41:07 localhost authserver [DEBUG] Accounting.java[39]: Current time = 1366011667113 -- lastUpdate = 0
Apr 15 09:41:08 localhost authserver [INFO] ThreadChecker: checking user threads

Apr 15 09:41:48 localhost authserver [DEBUG] CoreAuthServer: new thread 64778, total=1
Apr 15 09:41:48 localhost authserver [DEBUG] Core.authserver.CoreAuthClient[64778]: receivedMessage: 'nn#?/+,00:1D:E0:62:D6:05#?/+,192.168.214.1#?/+,EXT#?
RTAL#?/+,PW:nn#?/+,USERADIUS:#?/+,AUTODISCONNECT:#?/+,CALLED_STATION_ID:00-00-00-00-00#?/+,ENCODELOGINPASS:bm46bm4=#?/+,DEVICE_PORTAL_FORMAT:laptop#?/+,DE
CTOR_NAME:unknown#?/+,DEVICE_OPERATING_SYSTEM_NAME:Windows#?/+,DEVICE_OPERATING_SYSTEM_VERSION:7#?/+,DEVICE_BROWSER_NAME:MSIE#?/+,DEVICE_BROWSER_VERSION:9
CATION:LOCAL_NETMATCH#?/+,NETWORKID:in214'
Apr 15 09:41:48 localhost authserver [DEBUG] Core.authserver.CoreAuthClient[64778]: userAuthenticate (nn/00:1d:e0:62:d6:05/192.168.214.1): received parame
0:1d:e0:62:d6:05, IP: 192.168.214.1, userId: nn, keyword: MODPORTAL, noBind: 0, usergroup: null, useRadius: false, accessControlAttribute: null, ldapPos:
LL, userlastname: null, userfirstname: null, devicePortalFormat: laptop, deviceConstructorName: unknown, deviceOperatingSystemName: Windows, deviceOperati
ion: 7, deviceBrowserName: MSIE, deviceBrowserVersion: 9.0, calledStationId: 00-00-00-00-00, dryRun: false, incomingNetworkID: in214, ipLocation: LOCAL_NE
Apr 15 09:41:48 localhost authserver [NOTICE] Core.authserver.CoreAuthClient[64778]: userAuthenticate (nn/00:1d:e0:62:d6:05/192.168.214.1): new user nn
Apr 15 09:41:48 localhost authserver [NOTICE] Reading network 'in214' from ucopia.conf
Apr 15 09:41:48 localhost authserver [DEBUG] Core.authserver.CoreAuthClient[64778]: userAuthenticate (nn/00:1d:e0:62:d6:05/192.168.214.1): looking for use
 LDAP: local
Apr 15 09:41:48 localhost authserver [DEBUG] Core.authserver.CoreAuthClient[64778]: getAuthUser: /usr/sbin/extldapauth.sh -h ldap://127.0.0.1:389 -D cn=Ma
pia,dc=com -w --- -b uid=nn,ou=users,dc=ucopia,dc=com -f (&(objectclass=*)(!(banned=true))) -a dn|GroupObject|null|givenName|name|validitytype|validitydur
tytime|startdate|expirationdate|schedule|fieldtitle1|fieldvalue1|fieldtitle2|fieldvalue2|fieldtitle3|fieldvalue3|userEmail|phonenumber|pmsid|qosenable|flo
C|ForceDisconnectTimerEnable|ForceDisconnectTimer|MACAddressSettings|macauthenable|fieldid|fieldvalue|defaultLang|langId|text
Apr 15 09:41:48 localhost authserver [DEBUG] Core.authserver.CoreAuthClient[64778]: userAuthenticate (nn/00:1d:e0:62:d6:05/192.168.214.1): testing user's
on LDAP: local
Apr 15 09:41:48 localhost authserver [DEBUG] Core.authserver.CoreAuthClient[64778]: testing credential succeed for user: uid=nn,ou=users,dc=ucopia,dc=com
Apr 15 09:41:48 localhost authserver [DEBUG] Core.authserver.CoreAuthClient[64778]: userAuthenticate (nn/00:1d:e0:62:d6:05/192.168.214.1):  found and auth
 PORTAIL LDAP: local
Apr 15 09:41:48 localhost authserver [DEBUG] Core.authserver.CoreAuthClient[64778]: processCondition 'nn': user is connecting from network in214 - locatio
tid=in214,controllerId=local,siteId=local,ou=sites,dc=ucopia,dc=com, foundProfileName guest
Apr 15 09:41:48 localhost authserver [NOTICE] UcpLdapClient (ldapConnect): no LDAP connection, state is not connected. Trying to connect...
Apr 15 09:41:48 localhost authserver [NOTICE] UcpLdapClient (connect_and_bind): Successfully connected to 127.0.0.1 on port 389 with DN cn=Manager,dc=ucop
Apr 15 09:41:48 localhost authserver [DEBUG] Core.authserver.CoreAuthClient[64778]: userAuthenticate (nn/00:1d:e0:62:d6:05/192.168.214.1): no credit-time
Apr 15 09:41:48 localhost authserver [DEBUG] UcpLdapClient.LC.addUserMACAddress: user nn will be updated
```

■ To display disconnection logs: showLogs autodisconnect

```
> showLogs autodisconnect
Apr 15 09:37:17 localhost keepalived->notify_all_DISABLE[13060]: reloading autodisconnect
Apr 15 09:37:17 localhost autodisconnect[3248]: reloading configuration
Apr 15 09:37:17 localhost autodisconnect[3248]: using parameters: debug=LOG_DEBUG, period=60, nb_checks=5, kern_CONFIG_HZ=250, mode=passive, layers=3, pa
0.1
Apr 15 09:37:17 localhost autodisconnect[3248]: going to query SQL for connected users with parameters: host=127.0.0.1, port=3306, base=ucpdb, login=ucpr
Apr 15 09:37:17 localhost autodisconnect[3248]: SQL query succeeded, going to fill connected users array
Apr 15 09:37:17 localhost autodisconnect[3248]: sleeping before next loop iteration
Apr 15 09:37:19 localhost keepalived->notify_all_DISABLE[13060]: Reloading autodisconnect service: autodisconnect.
Apr 15 09:37:32 localhost authserver [DEBUG] reading config section:'conf' name:'autodisconnect'
Apr 15 09:38:17 localhost autodisconnect[3248]: going to query SQL for connected users with parameters: host=127.0.0.1, port=3306, base=ucpdb, login=ucpr
Apr 15 09:38:17 localhost autodisconnect[3248]: SQL query succeeded, going to fill connected users array
Apr 15 09:38:17 localhost autodisconnect[3248]: sleeping before next loop iteration
Apr 15 09:39:17 localhost autodisconnect[3248]: going to query SQL for connected users with parameters: host=127.0.0.1, port=3306, base=ucpdb, login=ucpr
Apr 15 09:39:17 localhost autodisconnect[3248]: SQL query succeeded, going to fill connected users array
Apr 15 09:39:17 localhost autodisconnect[3248]: sleeping before next loop iteration
Apr 15 09:40:17 localhost autodisconnect[3248]: going to query SQL for connected users with parameters: host=127.0.0.1, port=3306, base=ucpdb, login=ucpr
Apr 15 09:40:17 localhost autodisconnect[3248]: SQL query succeeded, going to fill connected users array
Apr 15 09:40:17 localhost autodisconnect[3248]: sleeping before next loop iteration
Apr 15 09:41:17 localhost autodisconnect[3248]: going to query SQL for connected users with parameters: host=127.0.0.1, port=3306, base=ucpdb, login=ucpr
Apr 15 09:41:17 localhost autodisconnect[3248]: SQL query succeeded, going to fill connected users array
Apr 15 09:41:17 localhost autodisconnect[3248]: sleeping before next loop iteration
```

■ To display RADIUS logs: showlogs RADIUS

```
Apr 15 10:11:49 localhost freeradius[20729]: Waking up in 0.9 seconds.
Apr 15 10:11:49 localhost freeradius[20729]: [<thread>] # Executing section authorize from file /etc/freeradius/sites-enabled/default
Apr 15 10:11:49 localhost freeradius[20729]: [pap] WARNING! No "known good" password found for the user.  Authentication may fail becaus
Apr 15 10:11:49 localhost freeradius[20729]: # Executing group from file /etc/freeradius/sites-enabled/default
Apr 15 10:11:49 localhost freeradius[20729]: [ldap1] performing user authorization for user
Apr 15 10:11:49 localhost freeradius[20729]:    [ldap1] ldap_get_conn: Checking Id: 0
Apr 15 10:11:49 localhost freeradius[20729]:    [ldap1] ldap_get_conn: Got Id: 0
Apr 15 10:11:49 localhost freeradius[20729]: [ldap1] No default NMAS login sequence
Apr 15 10:11:49 localhost freeradius[20729]: [ldap1] looking for check items in directory...
Apr 15 10:11:49 localhost freeradius[20729]:    [ldap1] userPassword -> User-Password == "u"
Apr 15 10:11:49 localhost freeradius[20729]: [ldap1] looking for reply items in directory...
Apr 15 10:11:49 localhost freeradius[20729]:    [ldap1] validitytype -> Ucopia-validitytype = "alwaysvalid"
Apr 15 10:11:49 localhost freeradius[20729]:    [ldap1] GroupObject -> Ucopia-Group = "GroupId=guest,dc=ucopia,dc=com"
Apr 15 10:11:49 localhost freeradius[20729]: [ldap1] Setting Auth-Type = LDAP1
Apr 15 10:11:49 localhost freeradius[20729]: [ldap1] user user authorized to use remote access
Apr 15 10:11:49 localhost freeradius[20729]:    [ldap1] ldap_release_conn: Release Id: 0
Apr 15 10:11:49 localhost freeradius[20729]: Exec-Program: returned: 0
Apr 15 10:11:49 localhost freeradius[20729]: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Apr 15 10:11:49 localhost freeradius[20729]: !!!    Replacing User-Password in config items with Cleartext-Password.    !!!
Apr 15 10:11:49 localhost freeradius[20729]: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Apr 15 10:11:49 localhost freeradius[20729]: !!! Please update your configuration so that the "known good"       !!!
Apr 15 10:11:49 localhost freeradius[20729]: !!! clear text password is in Cleartext-Password, and not in User-Password. !!!
Apr 15 10:11:49 localhost freeradius[20729]: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Apr 15 10:11:49 localhost freeradius[20729]: # Executing group from file /etc/freeradius/sites-enabled/default
Apr 15 10:11:49 localhost freeradius[20729]: [ldap1] login attempt by "user" with password "u"
Apr 15 10:11:49 localhost freeradius[20729]: [ldap1] user DN: uid=user,ou=users,dc=ucopia,dc=com
Apr 15 10:11:49 localhost freeradius[20729]:    [ldap1] (re)connect to ldap://127.0.0.1, authentication 1
Apr 15 10:11:49 localhost freeradius[20729]:    [ldap1] setting TLS Require Cert to never
Apr 15 10:11:49 localhost freeradius[20729]:    [ldap1] bind as uid=user,ou=users,dc=ucopia,dc=com/u to ldap://127.0.0.1
Apr 15 10:11:49 localhost freeradius[20729]:    [ldap1] waiting for bind result ...
Apr 15 10:11:49 localhost freeradius[20729]:    [ldap1] Bind was successful
Apr 15 10:11:49 localhost freeradius[20729]: [ldap1] user user authenticated succesfully
Apr 15 10:11:49 localhost freeradius[20729]: # Executing section post-auth from file /etc/freeradius/sites-enabled/default
Apr 15 10:11:49 localhost freeradius[20729]: Exec-Program-Wait: plaintext: Ucopia-Group=GroupId=guest,dc=ucopia,dc=com
Apr 15 10:11:49 localhost freeradius[20729]: Exec-Program: returned: 0
Apr 15 10:11:50 localhost freeradius[20729]: Waking up in 4.1 seconds.
Apr 15 10:11:54 localhost freeradius[20729]: Ready to process requests.
^C>
```

3.6   Update

Command	Description
applyUpdate	Download an update from an FTP server and apply it.
applyAllUpdates	Download all updates from an FTP server and apply it.
enableAutoUpdate	Enable or disable automatic update installation.
listUpdates	Display all available updates.

3.6.1   applyUpdate

Download an update from an FTP server and apply it.

```
> applyUpdate <ftp_accountId> <filename>
```

- ftp_*accountId*: FTP server name
- *filename*: Update name

Example :

```
> applyUpdate account1 update_5.0-b14061605.tar
```

> Applying: update_5.0-b14061605.tar. Please, wait.

3.6.2 applyAllUpdates

Download all updates from an FTP server and apply it.

> applyAllUpdates <ftp_accountId>

- ftp_*accountId*: FTP server name

Example :

```
> applyAllUpdates account1
Applying in order:
 update_5.0-b14061604.tar
 update_5.0-b14061605.tar


 ** Applying: update_5.0-b14061604.tar.  Please, wait.


 ** Applying: update_5.0-b14061605.tar.  Please, wait.
```

3.6.3 enableAutoUpdate

Enable or disable automatic update installation.

> enableAutoUpdate [enable]

Options

- *enable*: enable (y|n) [y]

Example :

```
> enableAutoUpdate n
Auto update process disabled
```

3.6.4   listUpdates

Display all available updates.

```
> listUpdates <ftp_accountId>
```

- ftp_*accountId*: FTP name server

Example :

```
> listUpdates test1
Available updates or upgrades:
 -> update_5.0-b14061604.tar
 -> update_5.0-b14061605.tar
```


3.7 Network troubleshooting

Command	Description
arp	Display the ARP cache of the controller.
arping	Send an ARP request.
dhclient	Retrieve an IP address distributed by DHCP.
host	Retrieve a host name or an IP address for a machine using the domain server.
interface	Display the configuration of network interfaces.
ipRouteGet	Display the route used to reach the destination address.
netstat	Display the set of connections opened by the UCOPIA controller.
nslookup	Querie a name server in order to obtain information about a domain.
ping	Send a Ping request.
showDhcpLeases	Display DHCP leases.
showRoute	Display the routing table.

Command	Description
tcpdump	Capture and analyze all packets passing through a network interface.
telnet	Start a telnet connection to another machine.
traceroute	Determine the path followed by a packet, letting you build a map of the routers present between source and target machines.
ssh	Start an SSH connection to another machine.
wget	Non-interactively downloads files.

3.7.1 arp

Display the ARP cache of the controller.

```
arp [-n do_not_resolve]
```

Options:

- *-n do_not _resolve* : Do not resolve IP addresses.

Example :

```
> arp
? (172.16.0.1) at 00:1b:21:04:f8:db [ether] on out
>
```

3.7.2 arping

Send an APR request..

```
arping [-c count] [-w timeout] [-I device] [-s source] <dest>
```

Options:

- *dest*: Destination IP address
- *-c count*: Number of packets to send
- *-w timeout*: Maximum time to wait for a response
- *-I device*: Choice of the interface to use
- *-s source*: Source IP address

Example:

```
> arping -c 3 -w 5 -I out -s 10.0.0.187 10.0.0.1
ARPING 10.0.0.1 from 10.0.0.187 out
Unicast reply from 10.0.0.1 [00:15:C5:F4:8D:E0]  0.786ms
Unicast reply from 10.0.0.1 [00:15:C5:F4:8D:E0]  0.734ms
```

```
Unicast reply from 10.0.0.1 [00:15:C5:F4:8D:E0]  0.753ms
Sent 3 probes (1 broadcast(s))
Received 3 response(s)
```

3.7.3   dhclient

Retrieve an IP address distributed by DHCP.

```
dhclient [interface]
```

- Interface : Choice of the interface

3.7.4   host

Retrieve a host name or an IP address for a machine using the domain server.

```
host <hostname> [-t querytype] [-v verbose] [-r recursion off] [-d debugging] [-l list] [-T
tcp] [server]
```

- *hostname*: Name of the server
- *server*: IP address of the name server

Options :

- *-t querytype*: Specify the type of query to search
- *-v verbose*: Verbose mode
- *-r recursion off*: Disable recursion in the query
- *-d debugging*: Enable debug mode
- *-l list*: Display the full domain
- *-T tcp*: Enable the TCP/IP mode

Example :

- Test resolution of the name oli1 by the 10.0.0.1 server

```
> host oli1 10.0.0.1
Using domain server:
Name: 10.0.0.1
Address: 10.0.0.1#53
Aliases:

oli1.ucopia.lan has address 10.0.0.132
```

3.7.5   interface

Display the configuration of network interfaces.

```
Interface [interface]
```

- *interface* : Interface for which the configuration is displayed

Example :

```
> interface eth0
3: eth0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:06:4f:90:50:aa brd ff:ff:ff:ff:ff:ff
    inet6 fe80::206:4fff:fe90:50aa/64 scope link
       valid_lft forever preferred_lft forever
```

3.7.6 ipRouteGet

Affiche la route utilisée pour atteindre l'adresse de destination.

```
ipRouteGet <dest_address> [from src_address]
```

- *from src_address* : Source IP address.
- *dest_address* : Destination IP address.

Example :

```
> iprouteget 10.0.0.23
10.0.0.23 via 172.16.0.1 dev out  src 172.16.0.200
    cache  mtu 1500 advmss 1460 hoplimit 64
```

3.7.7 netstat

Display the set of connections opened by the UCOPIA controller.

```
netstat [-r route] [-g groups] [-i interface] [-s statistics] [-n numeric] [-l listening]
        [-a all] [-e extend] [-I specified_iface]
```

Options :

- *-r route*: Display the routing table
- *-g groups*: Display information about multicast groups
- *-i interface*: Display a table of all network interfaces.
- *-s statistics*: Display a summary of statistics for each protocol.
- *-n numeric*: Display addresses in numeric format, instead of trying to determine the symbolic host, port, or user name.
- *-l listening*: Display only sockets that are listening.
- *-a all*: Display all sockets, whether they are listening or not.
- *-e extend*: Display extra information. Use this option twice to obtain maximum details.
- *-I specified_iface*: Display a table of all network interfaces.

Examples :

■ Verify the routing table

```
> netstat -r y
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.100.0   *               255.255.255.0   U        0 0          0 in
172.16.0.0      *               255.255.255.0   U        0 0          0 out
169.254.0.0     *               255.255.0.0     U        0 0          0 in
default         172.16.0.1      0.0.0.0         UG       0 0          0 out
```

■ Information related to statistics

```
> netstat -s y
Ip:
    134402 total packets received
    314 with invalid addresses
    0 forwarded
    0 incoming packets discarded
    76632 incoming packets delivered
    72986 requests sent out
    50 dropped because of missing route
Icmp:
    44 ICMP messages received
    0 input ICMP message failed.
    ICMP input histogram:
        destination unreachable: 33
        echo requests: 11
    45 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
        destination unreachable: 34
        echo replies: 11
IcmpMsg:
        InType3: 33
        InType8: 11
        OutType0: 11
        OutType3: 34
Tcp:
    1604 active connections openings
    1678 passive connection openings
    6 failed connection attempts
    872 connection resets received
    6 connections established
    65206 segments received
    64653 segments send out
    20 segments retransmited
    0 bad segments received.
    1175 resets sent
Udp:
    18487 packets received
    34 packets to unknown port received.
    0 packet receive errors
    8494 packets sent
UdpLite:
TcpExt:
    742 TCP sockets finished time wait in fast timer
    1101 delayed acks sent
    1 delayed acks further delayed because of locked socket
    Quick ack mode was activated 3 times
    7926 packets directly queued to recvmsg prequeue.
    86532 bytes directly received in process context from prequeue
    30366 packet headers predicted
    1636 packets header predicted and directly queued to user
    7915 acknowledgments not containing data payload received
    17642 predicted acknowledgments
    11 times recovered from packet loss by selective acknowledgements
```

```
    4 TCP data loss events
    TCPLostRetransmit: 1
    1 timeouts after SACK recovery
    13 fast retransmits
    6 other TCP timeouts
    3 DSACKs sent for old packets

[...]
```

3.7.8 nslookup

Allow to query a name server in order to obtain information about a host.

```
> nslookup <host>
```

- *host* : Host to find []

Note

The nslookup command enters interactive mode when no hosts are given.

Examples:

- Query for host www.ucopia.com

```
> nslookup www.ucopia.com
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
www.ucopia.com  canonical name = ucopia.com.
Name:   ucopia.com
Address: 93.88.249.185
```

- Query for host www.ucopia.com in interactive mode

```
> nslookup
> www.ucopia.com
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
www.ucopia.com  canonical name = ucopia.com.
Name:   ucopia.com
Address: 93.88.249.185
> exit
```

3.7.9 ping

Send a ping request

```
> ping [-c num_echos] [-t ttl] [-I interface] [-b broadcast] [-n numeric] [-R road] <dest>
```

- dest : Domain name or IP address.

Options:

- *-c num_echos:* Specify the number of "echo" requests to send
- *-t ttl*: iTme to live
- *-I interface*: Address of the interface
- *-b broadcast*: Broadcast address [n]
- *-n numeric*: Numeric output only [n]
- *-R road*: Record and display the route [n]

Example :

```
> ping 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp seq=1 ttl=64 time=2.80 ms
64 bytes from 10.0.0.10: icmp seq=2 ttl=64 time=0.363 ms
64 bytes from 10.0.0.10: icmp seq=3 ttl=64 time=0.295 ms
64 bytes from 10.0.0.10: icmp seq=4 ttl=64 time=0.367 ms

--- 10.0.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.295/0.958/2.809/1.069 ms
```

3.7.10 showDhcpLeases

Display DHCP leases.

```
> showDhcpLeases [pipe] [pipe_action] [grep_pattern]
```

Options:

- *pipe*: The pipe pattern '|'
- *pipe_action*: A piped action (less|grep)
- *grep_pattern*: A grep pattern (do not support ';' character)

Example :

```
> showDhcpLeases
DHCP Leases

lease 192.168.213.126 {
  starts 5 2011/08/26 13:47:51;
  ends 6 2011/08/27 13:47:51;
  binding state active;
  next binding state free;
  hardware ethernet 00:1d:e0:62:d6:05;
  uid "\001\000\035\340b\326\005";
  client-hostname "seven-PC";
}
lease 192.168.213.126 {
  starts 5 2011/08/26 13:56:47;
  ends 6 2011/08/27 13:56:47;
  binding state active;
```

```
  next binding state free;
  hardware ethernet 00:1d:e0:62:d6:05;
  uid "\001\000\035\340b\326\005";
  client-hostname "seven-PC";
}
```

3.7.11 showRoute

Display the routing table.

```
> showRoute [table]
```

- *table* : Name of the routing table to display.

Example :

```
> showRoute
0:      from all lookup local
101:    from 172.16.0.200 lookup 782
32766:  from all lookup main
32767:  from all lookup default
Table main:
192.168.100.0/24 dev in  proto kernel  scope link  src 192.168.100.254  metric 10
172.16.0.0/24 dev out  proto kernel  scope link  src 172.16.0.200  metric 10
169.254.0.0/16 dev in  scope link  metric 10
default via 172.16.0.1 dev out  metric 10
```

3.7.12 tcpdump

To capture and analyze all packets passing through a network interface.

```
tcpdump [expression] [-i interface] [-c count] [-D list_interface]
[-A ascii] [-e link level header] [-L data link types]
[-n numeric addresses] [-N no domain name] [-p no promiscuous]
[-q quick_outpput][-S sequence_absolute] [-v verbose] [-y data_link_type] [-T type]
```

- *expression* : See pcap-filter for the syntax of the expression (filter packets to be captured. If no expression is specified, all packets are captured).

Options:

- *-i interface* : Capture packets from a specific network interface[].
- *-c count* : Count the number of lines, at the end of the capture [].
- *-D list_interface* : The list of available interfaces [n].
- *-A ascii* : Display all packets using ASCII [n].
- *-e link_level_header* : Display the packet headers at the data link level [n].
- *-L data_link_types* : Display the list of known data link types [n].
- *-n numeric_addresses* : Do not convert addresses or ports in names [y].
- *-N no_domain_name* : Do not display the domain name of host names qualification [n].
- *-p no_promiscuous* : Accept all packets [n].
- *-q quick_outpput* : Stop capturing [n].

- *-S sequence_absolute* : Display the number of TCP sequence [n].

- *-v verbose* : Verbose mode [n].

- *-y data_link_type* : Define the type of data link to use during packet capture [].

- *-T type* : Force the type of packets selected by the expression to be interpreted by the specified type [].

Example :

```
> tcpdump -n v -e v -i out
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on out, link-type EN10MB (Ethernet), capture size 65535 bytes
22:31:59.050911 00:06:4f:90:50:aa > 00:1b:21:04:f8:db, ethertype IPv4 (0x0800), length 444:
172.16.0.200.222 >
 82.234.35.126.47800: Flags [P.], seq 2514509764:2514510142, ack 3884938946, win 1002,
options [nop,nop,TS val
 6334104 ecr 585395710], length 378
22:31:59.123248 00:1b:21:04:f8:db > 00:06:4f:90:50:aa, ethertype IPv4 (0x0800), length 940:
82.234.35.126.4780
0 > 172.16.0.200.222: Flags [P.], seq 1:875, ack 378, win 316, options [nop,nop,TS val
585395811 ecr 6334104],
 length 874
22:31:59.123781 00:06:4f:90:50:aa > 00:1b:21:04:f8:db, ethertype IPv4 (0x0800), length 556:
172.16.0.200.222 >
 82.234.35.126.47800: Flags [P.], seq 378:868, ack 875, win 1002, options [nop,nop,TS val
6334122 ecr 58539581
1], length 490
```

3.7.13 telnet

Start a telnet connection to another machine.

```
telnet <hostname> [-p port]
```

- *hostname* : Address of the machine to reach

Options:

- *-p* : Connection port

3.7.14 traceroute

Determines the path followed by a packet, letting you build a map of the routers present between source and target machines.

```
traceroute <host> [-g gateway] [-i interface] [-f first ttl] [-p port]
[-s src address] [-t tos] [-S packet length] [-F no_fragment] [-I icmp_echo]
[-n numerically] [-r bypass] [-v verbose output]
```

- *host* : Address of the machine to reach.

Options

- *-g gateway*: Specify the output gateway.

- `-i interface`: Specify the output interface.
- `-f first_ttl`: Parameterize the first hop of output packets.
- `-p port`: Parameterize the UDP base port number.
- `-s src_address`: Use this source IP address.
- `-t tos`: Parameterize the service type.
- `-S packet_length`: Specify packet size.
- `-F no_fragment`: Tell intermediate routers to avoid fragmenting packets.
- `-I icmp_echo`: Use ICMP ECHO instead of datagrams.
- `-n numerically`: Display numeric addresses.
- `-r Bypass`: Bypasses normal routing tables.
- `-v Verbose output`: Verbose mode.

.

3.7.15 ssh

Start an SSH connection to another machine.

```
ssh [-l login_name] [-p port] <hostname>
```

- `hotsname` : Address of the remote machine.

Options:

- `-l login_name` : User identifier (on the remote machine)
- `-p` : Connection port

3.7.16 wget

Non-interactively downloads files.

```
wget <url> [-r recursive] [-c continue]
--proxy-user  [--no-check-certificate no_check_certificate] [--passive-ftp passive_ftp][--
no-proxy no proxy]
 [--proxy-ip proxy_ip] [--proxy-port proxy_port] [--proxy-user username] [--proxy-passwd
passwd]
```

- `url` : Address of the ressource.

Options:

- `-r recursive`: Enable recursive mode.
- `-c continue`: Continue downloading a partially-downloaded file.
- `--no-check-certificate no_check_certificate`: Do not try to validate a server SSL certificate.
- `--passive-ftp`: FTP in passive mode.
- `--no-proxy`: Do not use a proxy.

- *--proxy-ip*: Host name or IP address of the proxy server.
- *--proxy-port*: Port of the proxy server.
- *--proxy-user Username*: user name for proxy authentication.
- *--proxy-passwd*: Password for proxy authentication.

Example :

- Test a connection to the site http://www.ucopia.com [http://www.ucopia.com/] by passing through a proxy.

```
> wget --proxy-ip 192.168.30.25 --proxy-port 3128 --proxy-user ucopia --proxy-passwd ucopia
http://www.ucopia.com
```

3.8   System troubleshooting

Command	Description
lsearch	Open a connection on an LDAP server and lets you perform a query on that server.
mysqlDbSize	Display the size of the SQL database.
mysqlCheck	Verify the status of the SQL database.
ps	Display active processes.
summary	Display the characteristics of the UCOPIA controller.

3.8.1   lsearch

Open a connection on an LDAP server and lets you perform a query on that server. This command in particular lets you display the UCOPIA user base.

```
lsearch [filter]
```

- *filter* : Search filter.

Example :

- Search for the user pborras: lsearch uid=pborras

```
> lsearch jdurand
# extended LDIF
#
```

```
# LDAPv3
# base <dc=ucopia,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: jdurand
#

# admin, administrators, ucopia.com
dn: administratorid=admin,ou=administrators,dc=ucopia,dc=com

# jdurand, users, ucopia.com
dn: uid=jdurand,ou=users,dc=ucopia,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

3.8.2   mysqlDbSize

Display the size of the SQL database

```
mysqlDbSize [table]
```

- ■   *table* : Name of the tale (sessions, stats, ulog, urls, users).

3.8.3   mysqlCheck

Check the status of the SQL database.

```
mysqlCheck
```

Example :

```
> mysqlCheck
ucpdb.controller_stats         OK
ucpdb.deleg_user    Table is already up to date
ucpdb.ihm OK
ucpdb.sessions      Table is already up to date
ucpdb.stats         Table is already up to date
ucpdb.ulogTable is already up to date
ucpdb.urlsOK
ucpdb.users         Table is already up to date
ucpdb.usrpaypal     Table is already up to date
```

3.8.4   ps

Display active processes

```
> ps [filter]
```

- ■   *filter* : The active process to display.

3.8.5 summary

Display the characteristics of the UCOPIA controller.

```
> summary
```

Example :

```
> summary
Maintenance validity.. 2014-08-30

***** ***** ***** ***** ***** *****
* Production name     SV150
* Serial number       R2791976
* Product type        Advance
* Current version     4.4
* Current build       13041001
* Last upgrade        -
***** ***** ***** ***** ***** *****
```

3.9   System commands

Commande	Description
bzip2	Compresse a file.
exit	Close the CLI session.
halt	Stop the controller.
help	Display help topics for the CLI.
keyboard	Change the keyboard layout
less	Allow to view a text file.
ls	Display a list of files and folders.
reboot	Restart the controller.
rm	Delete a file or a folder.
scp	Copy a file or a directory to a secured server

3.9.1   bzip2

Allow to compress a file.

```
bzip2 [-d decompress] [-k keep] <file>
```

■ *file* : File name.

Options :

- `-d decompress` : Force decompression.
- `-k keep` : Preserve input files.

3.9.2   Exit

Close the CLI session.

```
> exit
```

3.9.3   halt

Stop the controller.

```
> halt
```

3.9.4   help

Display help topics for the CLI.

```
> help
```

Example :

```
CONTEXT SENSITIVE HELP
[?] - Display context sensitive help. This is either a list of possible
      command completions with summaries, or the full syntax of the
      current command. A subsequent repeat of this key, when a command
      has been resolved, will display a detailed reference.

AUTO-COMPLETION
The following keys both perform auto-completion for the current command line.
 If the command prefix is not unique then the bell will ring and a subsequent
 repeat of the key will display possible completions.

[enter] - Auto-completes, syntax-checks then executes a command. If there is a
 syntax error then offending part of the command line will be highlighted and explained.

[Tab]   - Auto-completes, or if the command is already resolved inserts a space.

MOVEMENT KEYS
[CTRL-A] - Move to the start of the line
[CTRL-E] - Move to the end of the line.
[up]     - Move to the previous command line held in history.
[down]   - Move to the next command line held in history.
[left]   - Move the insertion point left one character.
[right]  - Move the insertion point right one character.

DELETION KEYS
[CTRL-C]    - Delete and abort the current line
[CTRL-D]    - Delete the character to the right on the insertion point.
[CTRL-K]    - Delete all the characters to the right of the insertion point.
[CTRL-U]    - Delete the whole line.
[backspace] - Delete the character to the left of the insertion point.

ESCAPE SEQUENCES
!! - Replaces the the last command line.
```

```
!N  - Replace the Nth command line (absolute as per 'history' command)
!-N - Replace the command line entered N lines before (relative)
```

3.9.5 keyboard

Change the keyboard layout.

```
> keyboard <layout>
```

■ *layout* : The keyboard layout (us|uk|fr|de)

Example :

■ Change keyboard layout to US

```
> kevboard fr
Loading /usr/share/keymaps/i386/qwerty/fr.kmap.gz
```

3.9.6 less

Allow to view a text file.

```
> less <file>
```

■ *file* : File name.

3.9.7 ls

Display a list of files and folders.

```
> ls [path]
```

■ *path* : Indicate the path of the file or directory to display.

3.9.8 reboot

Restart the controller.

```
> reboot
```

3.9.9 rm

Delete a file or a directory.

```
> rm <file> [-f force] [-r recursive]
```

■ file : nom du fichier à supprimer.

Options :

■ *-f force* : Force deletion.
■ *-r recursive* : Delete directories and their contents.

3.9.10 scp

Copy a file or a directory to a secured server.

```
> scp [-P port] [-r recursive] <file1> <file2>
```

■ file1, file2 : Files to copy.

Options :

■ *-P port* : Port to connect to the server.
■ *-r recursive* : Copy full directories.

Example :

```
> scp -p 22 -r UCOPIA/ pborras@10.0.0.1:/UCOPIA/
```

3.10 Maintenance

Command	Description
tunnel	Enable or diable a maintenance tunnel.

3.10.1 tunnel

Enable or disable a maintenance tunnel.

```
> tunnel <action> [emergency_host] [emergency_remote_port_ssh] [emergency_remote_port_https]
```

■ action: check_hosts |check_connection |automatic |enable_manual |disable_manual |keep_alive |check_all_status| enable_emergency |disable_emergency (mount/unmount the tunnel in automatic, manual or emergency mode, verifies status or connectivity of the tunnel)

Options :

Options concern only the tunnel in "emergency" mode.

■ *emergency_host*: Domain name or IP address of the emergency tunnel, by default [*services-management-platform.com*]

■ *emergency_remote_port_ssh*: SSH port for the emergency tunnel. Possible values: (1025..65535), by default it's 35500.

■ *emergency_remote_port_https*: HTTPS port for the emergency tunnel. Possible values: (1025..65535), by default it's 36500

Example :

■ Activate a manual maintenance tunnel on SSH port 35566 and HTTPS port 36566.

```
> tunnel enable_manual services-management-platform.com 35566 36566
```