COOLAUTOMATION

CoolMaster Edge Product Terminal Command-Line



This guide describes every command available in the ng_asciif command-line interface, the text-based console used to configure and operate CoolAutomation HVAC controllers. Commands are typed directly at the prompt by name — there is no need to prefix them with a menu or module name.

Some commands have one or more alternate names (aliases) that produce identical or very similarly formatted output; where relevant, the differences between aliases are called out in that command's entry.

Table of Contents

- General / System
  - `exit`
  - `date`
  - `reboot`
  - `boot`
  - `commit`
- Network Configuration
  - `ifconfig`
- GPIO
  - `gpio`
  - `gpio norm`
  - `gpio func`
- HVAC Line & Unit Control
  - `ls`
  - `stati`
  - `info`
  - `line`
  - `on`
  - `off`
  - `filt`
  - `mode`

    - `fan`
    - `dry`
    - `heat`
    - `cool`
    - `auto`
    - `fspeed`
    - `SP`
    - `RT`
    - `louver`
    - `lock`
    - `link`
- Device Settings
    - `set`
    - `set aserver`
    - `set baud`
    - `set defaults`
- Status Query
    - `query`
- Lutron Integration
    - `lutron`
    - `lutron discover`
    - `lutron pair`
    - `lutron bind`
    - `lutron unpair`
- Simulation
    - `simul`
- Status Reports
    - `istat`
    - `ostat`
    - `systat`
- Device Properties
    - `props`
- BACnet
    - `bacnet`
    - `bacnet ip`
    - `bacnet name`
    - `bacnet id`
    - `bacnet list`
- Cloud Connection
    - `coolhost clear`
    - `coolhost`
- Power
    - `power`
- Virtual Addressing
    - `va`
    - `va delall`
    - `va auto`
    - `va +`
    - `va -`

General / System

`exit`

Description: Ends the current CLI session.

Parameters: None

Example:

```
exit
```

`date`

Description: Prints the controller's current system date and time, followed by the standard "OK" status line.

Parameters: None

Example:

```
date
Sun Jul  5 17:32:10 UTC 2026
OK
```

`reboot`

Description: Immediately reboots the entire controller. Produces no console output before the connection drops.

Parameters: None

Example:

```
reboot
```

`boot`

Description: Applies all pending configuration changes made by other commands (such as `ifconfig`, `bacnet ip`, or `gpio func`) and restarts the affected services. The `boot_mode` parameter is accepted for compatibility but has no effect.

Parameters:

Name	Type	Required	Default	Description
`boot_mode`	text	No	2	Boot mode (accepted but currently has no effect)

Example:

```
boot
```

`commit`

Description: Same effect as `boot`: applies all pending configuration changes and restarts the affected services.

Parameters: None

Example:

```
commit
```

Network Configuration

`ifconfig`

Description: With no arguments, prints the LAN interface's MAC address, link state, IP/DHCP status, netmask, gateway, DNS servers, autoip state, and receive error count. With a sub-verb, it changes one aspect of the LAN network configuration (static/DHCP IP, admin up/down, autoip, gateway, netmask, or DNS).

Parameters:

Name	Type	Required	Default	Description
`subcom`	switch	No	(none — shows status)	Sub-verb selecting what to configure

Sub-verbs:

Sub-verb	Value	Description
`IP`	IP address	Set a static IP address, or `DHCP` to switch the interface to DHCP
`enable`	(none)	Bring the LAN interface up
`disable`	(none)	Bring the LAN interface down
`Autoip`	`0` or `1`	Enable (`1`) or disable (`0`) link-local (autoip) addressing
`Gateway`	IP address	Set the default gateway
`Netmask`	IP address	Set the subnet mask
`DNS1`	IP address	Set the primary DNS server
`DNS2`	IP address	Set the secondary DNS server

Example:

```
ifconfig
MAC    : AA:BB:CC:DD:EE:FF
Link   : Up
IP     : 192.168.1.50 (Static)
Netmask: 255.255.255.0
Gateway: 192.168.1.1
DNS1   : 8.8.8.8
DNS2   : 0.0.0.0
Autoip : No(0)
Rx err : 0

ifconfig IP 192.168.1.50
```

```
OK

ifconfig IP DHCP
OK

ifconfig enable
OK

ifconfig Gateway 192.168.1.1
OK
```

GPIO

`gpio`

Description: Prints the status of every configured GPIO line, showing its assigned function, current value, and normal state (Normally Open / Normally Closed).

Parameters: None

Example:

```
gpio
  A: OOS     (1), N.O.
  B: Unused  (0)
```

`gpio norm`

Description: Sets whether a GPIO line's normal (inactive) state is Normally Open or Normally Closed.

Parameters:

Name	Type	Required	Default	Description
`line`	text	Yes	—	GPIO line to configure: `A` or `B`
`norm`	text	Yes	—	Normal state: `O`/`o` = Normally Open, `C`/`c` = Normally Closed

Example:

```
gpio norm A o
OK, Boot Required!
```

`gpio func`

Description: Sets the function assigned to a GPIO line. The `ALL` function additionally takes an action verb (`OFF`/`ON`/`INH`) controlling how all indoor units behave when the line transitions between its Normal and Active states.

Parameters:

Name	Type	Required	Default	Description
`line`	text	Yes	—	GPIO line to configure: `1` or `2`

Name	Type	Required	Default	Description
`func`	text	Yes	—	Function: `Unused`, `ALL`, `OOS`, `FLRS`, `BI`, `BO`, `AI`
`action`	text (only when `func` is `ALL`)	Yes (if `func=ALL`)	—	`OFF`, `ON`, or `INH` — behavior applied to all indoor units on the state transition

Example:

```
gpio func 1 OOS
OK, Boot Required!

gpio func 2 ALL INH
OK, Boot Required!
```

HVAC Line & Unit Control

This is the largest command group, covering HVAC communication lines and indoor-unit control.

`ls`

Aliases: `ls2`, `stat`, `stat2`, `stat3`, `stat4` — same command, different output formatting: - `ls2`: setpoint and room temperature shown to 1 decimal place, full address, plus filter and therm-on flags. - `stat`: address printed without the line prefix, setpoint rounded to an integer, room temperature shown to 2 decimal places, no trailing flags. - `stat2`: same as `stat` but with a trailing filter flag appended. - `stat3`: setpoint and room temperature both rounded to integers, short address, trailing filter flag only. - `stat4`: setpoint and room temperature both rounded to integers, short address, trailing filter flag **and** therm-on flag.

Description: Queries the live status of one or more indoor units matching an address pattern, printing one line per unit showing on/off state, setpoint/room temperature, fan speed, mode, and failure code. If no units match, prints a “No UID” error.

Parameters:

Name	Type	Required	Default	Description
`dev_addr_ptrn`	text	No	*	Indoor unit address — * (all), `L1` (whole line), or `L1.100` (one unit). Wildcards within an address (e.g. `L1.*`) are **not** supported.

Example:

```
ls L1.100
L1.100 ON  30C 26C High Fan  OK   # 0
```

`stati`

Description: Fetches a short identifying string for indoor unit(s) and prints each device address followed by that string — for normal units an `ID:<hex>` value, for BACnet-origin units a `BACNET: <name> (...)` description.

Parameters: None

Example:

```
stati
L1.100 ID:D4664610
```

`info`

Description: Reports the on/off state and voltage of the 16V DC output line on line L2.

Parameters: None

Example:

```
info
L2 DC+ 16V
```

`line`

Description: Configures or reports on HVAC communication lines. With no subcommand, prints a summary of all HVAC lines (type, line-specific extra info such as Modbus address for CG5, unit/group counts, and Tx/Rx/timeout/checksum/collision/NAK/arbitration statistics). With a subcommand, updates a single setting of the given line and prints `OK, Boot Required!` on success, or an error code on invalid input.

Parameters:

Name	Type	Required	Default	Description
`subcom`	switch	No		Selects one of the sub-verbs below; omitting it prints the line summary

Example:

```
line
L1: DK U03/G03
    Tx:120 Rx:118 TO:0 CS:0 Col:0 NAK:0 ARB:0
L2: Unused
    Tx:0 Rx:0 TO:0 CS:0 Col:0 NAK:0 ARB:0
```

`line` sub-verbs:

Sub-verb	Parameters	Description	Example
`baud`	`line` (e.g. `L1`), `value` (baud rate, e.g. `9600..115200`), `optional_value` (optional serial format, e.g. `8N1`)	Sets the UART baud rate for a line; if `optional_value` is supplied it is appended with an underscore (e.g. `9600_8N1`).	`line baud L1 9600 8N1` → `OK, Boot Required!`
`version`	None	Prints the firmware/version string of the HVAC line	`line version` → `0.0.8`

Sub-verb	Parameters	Description	Example
		service — not a per-line setting.	
`polarity`	`line` (e.g. `L1`), `value` (`A|B`, `B|A`, or `auto`)	Sets the RS-485 wiring polarity for a line; rejects any other value with “Bad Parameter”.	`line polarity L1 auto` → `OK, Boot Required!`
`DCOUT`	`line` (e.g. `L1`), `value` (`-` or `+`)	Switches the line’s 16V DC output on (`+`) or off (`-`).	`line DCOUT L2 +` → `OK, Boot Required!`
`termination`	`line` (e.g. `L1`), `value` (`0` or `1`)	Enables/disables the RS-485 bus termination resistor for a line.	`line termination L1 1` → `OK, Boot Required!`
`type`	`line` (e.g. `L1`), `value` (e.g. `DK`, `GMV5SI`, `BACNET`, `MODBUS`, `Unused`)	Changes the line’s protocol/brand type. `unused` removes the line’s configuration entirely; `BAC`/`BACNET` also enables the corresponding BACnet MSTP interface (`L1`-`L3` only). Value is upper-cased before storage.	`line type L1 GMV5SI` → `OK, Boot Required!`
`simul`	`line` (e.g. `L1`), `value` (integer count)	Sets the number of simulated HVAC devices on a line (for bench/lab testing); rejects non-numeric values.	`line simul L1 5` → `OK, Boot Required!`
`myid`	`line` (e.g. `L1`), `value` (hex, up to 32 bits)	Sets the controller’s own bus address/ID on the given line; must be a valid hex value.	`line myid L1 0B` → `OK, Boot Required!`
`master`	`line` (e.g. `L1`), `value` (`0` or `1`)	Sets whether this controller acts as bus master (`1`) or slave (`0`) on the given line.	`line master L1 1` → `OK, Boot Required!`
`scan`	`line` (e.g. `L1`), `value` (`0` or `1`)	Enables/disables the initial device scan performed at boot for the given line.	`line scan L1 1` → `OK, Boot Required!`

Sub-verb	Parameters	Description	Example
`tpl`	`line` (e.g. `L1`), `value` (template name, e.g. `GMV5SI0`, `MESI0`)	Applies a predefined bundle of line/device settings to the line in one shot, replacing its existing configuration; unknown template names return “Out of Range”.	`line tpl L1 MESI0` → `OK, Boot Required!`

```
L1 TTM x:on
```

`on`

Aliases: `allon` — behaves identically to `on` (its help text says “Turn on all HVAC lines”, which is already the effective behavior of `on` when no pattern is given, since the default pattern is `*`).

Description: Turns on the HVAC line/indoor units matching the address pattern. If the pattern doesn’t already look like `L<n>` or `L<n>.<addr>`, it is automatically rewritten to match across all lines.

Parameters:

Name	Type	Required	Default	Description
`dev_addr_ptrn`	text	No	*	Indoor unit address pattern

Example:

```
on L1.100
OK
```

`off`

Aliases: `alloff` — identical behavior; only its help text (“Turn off all HVAC lines”) differs.

Description: Turns off the HVAC line/indoor units matching the address pattern.

Parameters:

Name	Type	Required	Default	Description
`dev_addr_ptrn`	text	No	*	Indoor unit address pattern

Example:

```
off L1.100
OK
```

`filt`

Description: Resets the filter sign on indoor unit(s) matching the address pattern. A bare pattern without an `L<n>` prefix is rewritten to match across all lines; omitting the address (or passing `*`) resets every unit.

Parameters:

Name	Type	Required	Default	Description
dev_addr_ptrn	text	No	*	Indoor unit address pattern

Example:

```
filt L1.102
OK
```

mode

Description: Sets the HVAC operating mode (e.g. Cool/Heat/Auto/Dry/Fan) for indoor units matching the address pattern.

Parameters:

Name	Type	Required	Default	Description
dev_addr_ptrn	text	No	*	Indoor unit address pattern
hvac_mode	text	No	None	HVAC mode

Example:

```
mode L1.100 Cool
OK
```

fan

Description: Convenience command that always sets the HVAC mode to `Fan` for units matching the address pattern.

Parameters:

Name	Type	Required	Default	Description
dev_addr_ptrn	text	No	*	Indoor unit address pattern

Example:

```
fan L1.100
OK
```

dry

Description: Convenience command that always sets the HVAC mode to `Dry` for matching units.

Parameters:

Name	Type	Required	Default	Description
dev_addr_ptrn	text	No	*	Indoor unit address pattern

Example:

```
dry L1.100
OK
```

heat

Description: Convenience command that always sets the HVAC mode to `Heat` for matching units.

Parameters:

Name	Type	Required	Default	Description
dev_addr_ptrn	text	No	*	Indoor unit address pattern

Example:

```
heat L1.100
OK
```

cool

Description: Convenience command that always sets the HVAC mode to `Cool` for matching units.

Parameters:

Name	Type	Required	Default	Description
dev_addr_ptrn	text	No	*	Indoor unit address pattern

Example:

```
cool L1.100
OK
```

auto

Description: Convenience command that always sets the HVAC mode to `Auto` for matching units.

Parameters:

Name	Type	Required	Default	Description
dev_addr_ptrn	text	No	*	Indoor unit address pattern

Example:

```
auto L1.100
OK
```

`fspeed`

Description: Sets the HVAC fan speed for units matching the address pattern. The single-letter shorthand `v/l/m/h/t/a` maps to `VLow/Low/Med/High/Top/Auto` respectively; any other value is passed through as-is.

Parameters:

Name	Type	Required	Default	Description
`dev_addr_ptrn`	text	No	*	Indoor unit address pattern
`fspeed_mode`	text	No	None	HVAC fan speed (`v`,`l`,`m`,`h`,`t`,`a` shorthand, or a full name)

Example:

```
fspeed L1.100 h
OK
```

`SP`

Aliases: `temp` — identical parameters/behavior.

Description: Sets the HVAC setpoint temperature for matching units. If the display unit setting is `F`, the entered value is converted from Fahrenheit to Celsius before being applied.

Parameters:

Name	Type	Required	Default	Description
`dev_addr_ptrn`	text	No	*	Indoor unit address pattern
`setpoint`	text	No	None	HVAC setpoint

Example:

```
SP L1.100 24
OK
```

`RT`

Aliases: `feed` — identical parameters/behavior.

Description: Sends a room-temperature feedback reading for matching units. If the display unit setting is `F`, the entered value is converted from Fahrenheit to Celsius first, same as `SP`.

Parameters:

Name	Type	Required	Default	Description
dev_addr_ptrn	text	No	*	Indoor unit address pattern
room_temp	text	No	None	HVAC room temperature

Example:

```
RT L1.100 24
OK
```

louver

Aliases: swing — **not** behaviorally identical: louver maps shorthand letters to full position names before sending, and applies the change as a louver position. swing sends the raw lower-cased value un-mapped and applies the change as a swing setting instead.

Description: Sets the louver/swing position for matching units. Maps the shorthand letters h/v/a/3/4/6/- to Horz/Vert/Swing/30/45/60/Stop respectively (when invoked as louver).

Parameters:

Name	Type	Required	Default	Description
dev_addr_ptrn	text	No	*	Indoor unit address pattern
state	text	No	None	Louver state (h,v,a,3,4,6,- shorthand, or a full name)

Example:

```
louver L1.100 h
OK
```

lock

Description: Shows or changes the HVAC keypad/remote lock flags for units matching the address pattern. If no lock_string is given, prints each matching device's current flags. If a lock_string is given, it is parsed as either a bare +/- (lock/unlock everything) or a sequence of +/--prefixed flag letters — o (on/off), m (mode), t (setpoint), n (on) — and the resulting change is applied.

Parameters:

Name	Type	Required	Default	Description
dev_addr_ptrn	text	No	*	Indoor unit address pattern
lock_string	text	No	None	+|-[o|m|t|n]: +/- = full lock/unlock; +o/-o = on-off; +m/-m = mode; +t/-t = setpoint; +n/-n = "on"

Example:

```
lock L1.100
L1.100: +o +m -t

lock L1.100 +m-t
OK
```

`link`

Description: Manages address-to-model link entries. With both arguments empty, lists all existing links (index, address, model). With both arguments given, adds a new link mapping `arg1` (device address, using `Y` in place of `L`, e.g. `Y1.100`) to `arg2` (model name). If only `arg1` is given and matches a negative-number pattern (e.g. `-2`), deletes the link entry at that index instead; any other combination has no effect.

NOTE: Currently applies to Toshiba

Parameters:

Name	Type	Required	Default	Description
`arg1`	text	No	None	Device address to link, `L` written as `Y` (e.g. `Y1.100`); or `-N` to delete link index `N`; omit with `arg2` to list all links
`arg2`	text	No	None	Model name to associate with `arg1` (only used when adding a link)

Example:

```
link
L000: Y1.100  >  SMMSu

link Y1.100 SMMSu
OK, Boot Required!

link -0
OK, Boot Required!
```

Device Settings

`set`

Description: With no arguments, prints the device's current settings snapshot (serial number, firmware version, build date, baud rate, echo/verbose flags, aserver port/enable, temperature unit, melody). With a sub-verb and value, updates the corresponding setting immediately — no separate `commit` command is required.

Parameters:

Name	Type	Required	Default	Description
`subcom`	switch	No	None	Sub-verb selecting which setting to change; omit both

Name	Type	Required	Default	Description
				`subcom` and `value` to just print current settings

Sub-verbs:

Sub-verb	Value	Description
`verbose`	`1` or `0`	Enable/disable verbose error-string output (vs. numeric error codes)
`deg`	`C` or `F`	Set the displayed temperature unit
`echo`	`1` or `0`	Enable/disable command echo
`melody`	`1` or `0`	Select the startup melody/sound

Example:

```
set
S/N          : 050048E3
version      : 0.9.2 (PCB Rev 5.7)
baud rate    : 9600
echo         : 1
verbose      : 0
deg C/F      : C
melody       : 1

set verbose 1
OK

set deg F
OK
```

`set aserver`

Description: Configures the ASCII/BACnet application server exposed over TCP. Applies immediately, so no separate `commit` command is needed.

Parameters:

Name	Type	Required	Default	Description
`subcom`	switch	Yes	None	Sub-verb selecting which aserver parameter to set

Sub-verbs:

Sub-verb	Value	Description
`port`	0..65535	TCP port the aserver listens on

Sub-verb	Value	Description
`enable`	`1` or `0`	Enable[1]/disable[0] the aserver

Example:

```
set aserver port 10102
OK

set aserver enable 1
OK
```

`set baud`

Description: Sets the RS-232/console baud rate. Applies immediately.

Parameters:

Name	Type	Required	Default	Description
`value`	`9600|19200|38400|115200`	Yes	None	New baud rate

Example:

```
set baud 115200
OK
```

`set defaults`

Description: Resets the device to factory defaults. Applies immediately — no `commit` is involved.

Parameters: None

Example:

```
set defaults
OK
```

Status Query

`query`

Description: Queries one or more indoor units matching a strict device-address pattern and prints a single value per matched unit, selected by the `value` letter. Degrees are converted to Fahrenheit first if the display unit setting is `F`. Prints “OK” after all matches, or “Bad Property” if the requested property can’t be read for a unit.

Parameters:

Name	Type	Required	Default	Description
`dev_addr_ptrn`	text	Yes	None	Device address pattern (e.g. `L1.001`)

Name	Type	Required	Default	Description
`value`	text	Yes	None	Property selector: `o`=on/off (1/0), `m`=mode (0=Cool..5=Fan), `f`=fan speed (0=Low..7=SSHi), `t`=setpoint temp (integer), `h`=setpoint temp (float, 2dp), `e`=failure code (`0` if OK), `a`=room temp (integer), `s`=louver/swing state

Example:

```
query L1.001 t
22
OK

query L1.001 o
1
OK
```

Lutron Integration

`lutron`

Description: Lists the Lutron bridges currently registered on the controller. For each registered bridge, prints the bridge ID, system type, and IP address; if none are registered, prints an error instead.

Parameters: None

Example:

```
lutron
L1: RA2 192.168.1.50
OK
```

`lutron discover`

Description: Searches the network for Lutron controllers, printing a status message while the search runs.

Parameters: None

Example:

```
lutron discover
Searching for Lutron devices on the network...
```

`lutron pair`

Description: Pairs the device with a Lutron controller discovered on the network.

Parameters:

Name	Type	Required	Default	Description
`id`	text	Yes	None	Lutron bridge Id

Example:

```
lutron pair id=abc123
Finding a device to pair with...
```

`lutron bind`

Description: Pairs the device with a Lutron controller directly by IP address, bypassing network discovery.

Parameters:

Name	Type	Required	Default	Description
`id`	text	Yes	None	Lutron bridge Id
`ip`	IP address	Yes	None	Lutron bridge IP address
`sys`	text	No	None	Lutron bridge system type

Example:

```
lutron bind id=abc123 ip=192.168.1.50 sys=RA2
Connecting 192.168.1.50...
```

`lutron unpair`

Description: Unpairs a registered Lutron controller. Passing `id=all` removes every registered controller instead of a single one.

Parameters:

Name	Type	Required	Default	Description
`id`	text	Yes	None	Lutron bridge Id (use `all` to remove every registered controller)

Example:

```
lutron unpair id=abc123
OK
```

Simulation

`simul`

Description: Runs a simulation sub-command for a given HVAC line — used for bench/lab testing without real HVAC hardware attached. When `command` is left empty, defaults to listing every active simulation patch

and its key/value fields; other sub-commands are accepted but produce no additional listing in the current implementation.

Parameters:

Name	Type	Required	Default	Description
`line`	text	No	*	Line number, e.g. `L1`; * targets all lines
`command`	text	No	“” (treated as listing active patches)	Simulation sub-command to run

Example:

```
simul L1
Patch 1
    dev  : L1.001
    prop : SP
    val  : 24
OK
```

Status Reports

`istat`

Description: Fetches indoor diagnostic parameters. With `arg` empty, queries live runtime data; with `arg` = `ch`, queries commissioning/history data instead. Any other `arg` value returns “Bad Parameter”. Note: the `line` parameter is accepted but currently has no filtering effect.

Parameters:

Name	Type	Required	Default	Description
`line`	text (e.g. `L1`)	No	None	Line number, e.g. `L1` (currently has no effect)
`arg`	text	No	“”	“” = live runtime data; `ch` = commissioning/history data; any other value errors

Example:

```
istat
X1.001 onoff: 1 SP: 21 RT: 24.500000 mode: "Auto" fspeed: "Med"
```

`ostat`

Description: Fetches outdoor diagnostic parameters, merged together with system-level data (the same data reported by `systat`). Same `arg`/data-source behavior as `istat`; the `line` parameter is likewise accepted but has no effect.

Parameters:

Name	Type	Required	Default	Description
`line`	text (e.g. `L1`)	No	None	Line number, e.g. `L1` (currently has no effect)
`arg`	text	No	“”	“” = live runtime data; `ch` = commissioning/history data; any other value errors

Example:

```
ostat
Y1.001 status: "OK" ambient: 32.500000
   Y1.100
      Y1.100 comp_hz: 55
```

`systat`

Description: Fetches system-level diagnostic parameters directly (the same data `ostat` merges in). Same `arg` behavior and the same non-functional `line` parameter as `istat`/`ostat`.

Parameters:

Name	Type	Required	Default	Description
`line`	text (e.g. `L1`)	No	None	Line number, e.g. `L1` (currently has no effect)
`arg`	text	No	“”	“” = live runtime data; `ch` = commissioning/history data; any other value errors

Example:

```
systat
Y1.001 status: "OK" ambient: 32.500000
```

Device Properties

`props`

Description: Prints a fixed-width table of every indoor unit’s UID, name, cool/heat setpoint limit ranges, and lock/limit flags (state/mode/setpoint/cool/heat lock indicators).

Parameters: None

Example:

```
props
  UID  |      Name      | Cool | Heat |   Elocks  |
-------+----------------+------------+-----------+
L1.001 | Living Room    | 18  30|16  32| o m t     |
OK
```

BACnet

`bacnet`

Description: Displays BACnet integration configuration. With no argument, prints every configured BACnet interface (`ip`, `mstp1`, `mstp2`, `mstp3`, …); with an interface argument, prints only that interface's parameters. Fails with "No BACnet License" if the BACnet feature is not licensed/available on this device.

Parameters:

Name	Type	Required	Default	Description
`interface`	text	No	(none)	BACnet interface to show — one of `ip`, `mstp1`, `mstp2`, `mstp3`; empty shows all

Example:

```
bacnet mstp1
    apdu_retries         : 3
    apdu_timeout_ms      : 6000
    datalink             : mstp
    device_id            : 42
    device_name          : CoolMasterEdge
    dnet                 : 1
    enable               : 0
    iface                : ubev1
    model_name           : CoolMaster EDGE
    mstp_baud            : 38400
    mstp_mac             : 42
    mstp_max_info_frames : 1
    mstp_max_master      : 127
    ubus_obj_name        : bacnet.mstp1
    vendor_id            : 1164
    whois_interval_ms    : 60000
OK
```

`bacnet ip`

Description: Configures the BACnet/IP datalink (interface, BBMD, UDP port, enable/disable). Always reports "OK, Boot Required!" since the change needs `commit`/`boot` to take effect.

Parameters:

Name	Type	Required	Default	Description
`subcom`	switch	Yes	—	Sub-verb selecting which BACnet/IP setting to change

Sub-verbs:

Sub-verb	Value	Description
`iface`	text	Network interface name to bind BACnet/IP to

Sub-verb	Value	Description
`BBMD`	`A.B.C.D:PORT`	BBMD address (or `0` to disconnect)
`port`	0..65535	UDP port for BACnet/IP
`enable`	`1` or `0`	Enable or disable BACnet/IP

Example:

```
bacnet ip enable 1
OK, Boot Required!

bacnet ip BBMD 192.168.1.10:47808
OK, Boot Required!
```

`bacnet name`

Description: Sets the BACnet device name for a BACnet interface. Requires a subsequent `boot`/`commit` to apply.

Parameters:

Name	Type	Required	Default	Description
`name`	text	Yes	—	New BACnet device name
`interface`	text	Yes	—	BACnet interface to rename — one of `ip`, `mstp1`, `mstp2`, `mstp3`

Example:

```
bacnet name CoolMasterEdge_42 ip
OK, Boot Required!
```

`bacnet id`

Aliases: `bacnet instance` — identical behavior; alternate wording for the same command.

Description: Sets the BACnet device instance ID for a BACnet interface. Requires a subsequent `boot`/`commit` to apply.

Parameters:

Name	Type	Required	Default	Description
`id`	number (0..4194303)	Yes	—	New BACnet device instance number
`interface`	text	Yes	—	BACnet interface to update — one of `ip`, `mstp1`, `mstp2`, `mstp3`

Example:

```
bacnet id 42 ip
OK, Boot Required!
```

`bacnet list`

Description: Fetches a list of known BACnet devices, units, or templates and pretty-prints the full raw response table (protocol envelope plus every field of each entry) — a debug-style dump, not a curated summary.

Parameters:

Name	Type	Required	Default	Description
`subcom`	switch	Yes	—	What to list: `devices`, `units`, or `templates`

Example:

```
bacnet list devices
{
  "type": "bacnetDevicesGetResponse"
  "rc": 0
  "devices":   {
    "826b8686-9d54-48c5-a6e5-27979a6b1a53":     {
      "name": "X1.013"
      "bacnetId": 2281
      "online": true
      ...
    }
    ...
  }
}
OK
```

Cloud Connection

`coolhost clear`

Description: Wipes cloud-related connection/registration state on the device. Prints “OK” on success or “Failed” on error.

Parameters: None

Example:

```
coolhost clear
OK
```

`coolhost`

Description: With no arguments, prints a formatted table of cloud connection parameters (server, ports, ping/reconnect counters, connection status, etc.). When invoked with a matching secret and a sub-verb, changes the cloud server, port, or connection security setting — any such change requires a reboot to take effect.

Parameters:

Name	Type	Required	Default	Description
`secret`	text	No	“” (empty)	Internal access code gating the sub-verbs below. Omit it entirely to just view cloud status; an incorrect non-empty value returns “Failed”
`subcom`	switch	No	—	Selects which cloud parameter to change once `secret` is valid: `server`, `port`, `ssl`

Sub-verbs (under `subcom`):

Sub-verb	Value param	Type	Description
`server`	`server_val`	text	New cloud server hostname; keeps the existing connection scheme/port
`port`	`port_val`	number (0-65535)	New cloud server TCP port
`ssl`	`ssl_val`	`0` or `1`	`0` selects a plain (unencrypted) connection, `1` selects a TLS-secured connection

Example:

```
coolhost
remote.access  : enabled
server         : cloud.example.com
port           : 10101
ping rate      : 120
ssl            : on
portS          : 443
...
status         : enabled V3
OK

coolhost <secret> server cloud2.example.com
OK, Boot Required!

coolhost wrong-secret server cloud2.example.com
Failed
```

Virtual Addressing

`va`

Description: Lists the current CA-to-VA (control address to virtual address) mapping, printing entries grouped under `INDOORS`, `PRO INDOORS`, and `PRO OUTDOORS` sections (each shown only if present).

Parameters: None

Example:

```
va
INDOORS
L1.001 --> 00001
L1.002 --> 00002
OK
```

`va delall`

Description: Clears the entire VA mapping, printing “Failed” in addition to an error message if the operation fails.

Parameters: None

Example:

```
va delall
OK
```

`va auto`

Description: Recreates the VA mapping automatically. On failure only the returned error message is printed (unlike the other `va` subcommands, no additional “Failed” line).

Parameters: None

Example:

```
va auto
OK
```

`va +`

Description: Adds a single CA-to-VA mapping entry.

Parameters:

Name	Type	Required	Default	Description
`ca`	text	Yes	None	Unit CA (e.g. `L1.001`)
`va`	number	Yes	None	Unit VA (e.g. `1`)

Example:

```
va + ca=L1.003 va=3
OK
```

`va -`

Description: Deletes a VA mapping entry; the single argument may be either a CA string or a VA number — the device automatically determines which it is.

Parameters:

Name	Type	Required	Default	Description
`arg`	text	Yes	None	Unit CA (e.g. `L1.001`) or unit VA (e.g. `1`)

Example:

```
va - arg=L1.003
OK
```