LifeSize® Automation Command Line Interface

For LifeSize Video Communications Systems
Software Release v3.0

October 2007

Copyright Notice

© 2006-2007 LifeSize Communications Inc, and its licensors. All rights reserved.

LifeSize Communications has made every effort to ensure that the information contained in this document is accurate and reliable, but assumes no responsibility for errors or omissions. Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless noted. This document contains copyrighted and proprietary information, which is protected by United States copyright laws and international treaty provisions. No part of the document may be reproduced or transmitted, in any form or by any means, electronic or mechanical, for any purpose, without the express written permission from LifeSize Communications.

Trademark Acknowledgments

LifeSize® and the LifeSize logo are trademarks of LifeSize Communications Inc. All other trademarks are the properties of their respective owners.

Patent Notice

For patents covering LifeSize® products, refer to http://www.lifesize.com/support/legal.

Contacting Customer Support

If you have questions, concerns, or need assistance, contact your LifeSize Channel Partner.

Providing Customer Feedback

LifeSize Communications welcomes your comments regarding our products and services. If you have feedback about this or any LifeSize product, please send it to feedback@lifesize.com. Contact information for LifeSize Communications is as follows:

Method	*Address*
Internet	http://www.lifesize.com
E-mail	support@lifesize.com
Phone	(877) LIFESIZE or (877) 543-3749 (512) 347-9300
Fax	(512) 347-9301

Introduction

LifeSize® Automation Command Line Interface (CLI) provides a command line-based entry point for automating access and control of LifeSize video communications systems. The CLI allows you to:

- Retrieve configuration information about your LifeSize system. For example, you can get the system version number or the camera's brightness setting.
- Apply new preferences to the system configuration. For example, you can set the speaker volume or the fadeout timer.
- Show the status of calls in the system. For example, you can show active calls or statistics for previous calls.
- Control aspects of the system. For example, you can add participants to an active call or emulate remote control functionality.

This document contains information about using the CLI commands, their output, and generated return codes.

Notational Conventions

The following conventions are used in this document.

Convention	Description
`monotype` font	Monotype font reflects commands and the resulting output. Constant input appears in **`Bold`**, for example: **`get system uptime`** Variable input appears in ***`Bold Italic`***, for example: **`set audio mics`** ***`off`*** Constant output appears in plain `monotype`, for example: `ok,00` Variable output appears in `monotype` *`Italic`*, for example: **`get audio video-output`** *`phone`*
angle brackets <>	Required parameters are enclosed in angle brackets, for example: `<parameter>`

square brackets `[]`	Optional parameters are enclosed in square brackets, for example:
	`[parameter]`
	Similarly, optional options are enclosed in square brackets, for example:
	`[-p]`

Convention	Description

curly brackets `{}`	Parameters whose values are restricted are enclosed in curly brackets with discrete values separated by a pipe (`|`) symbol. The following example restricts the values to val1, val2, or val3:
	`<{val1|val2|val3}>`
	Parameters whose values are restricted to a range of values are enclosed in curly brackets and separated by a pair of periods (`..`). The following example restricts values to integers between 0 and 100, inclusive:
	`[{0..100}]`
	The following example restricts values to integers between -30 and 30, inclusive:
	`[-p {-30..30}]`

Fundamentals of the Command Line Interface

This chapter describes the fundamental concepts of the CLI, such as accessing the CLI, help and default output modes, command line syntax, and standard output format.

Accessing the Command Line Interface

The CLI is available through an ssh or telnet connection to your LifeSize video communications system as the ***auto*** user (default password ***lifesize***). For LifeSize Room, you can also access the command line interface through one of the two rear panel serial ports if they are configured for this shell. The LifeSize Room system requires use of a standard null modem cable for interaction through the serial connection. Refer to “Serial Cable Pin Assignment” on page 6.

Connecting through Serial Ports

To connect through the rear panel serial ports on LifeSize Room, follow these steps:

1. Plug one end of the null modem cable into either of the LifeSize Room serial ports, taking note of which port you choose.

   Note: If you are connecting through the serial port for the first time, LifeSize recommends that you use port 2 on the LifeSize Room codec. By default, port 2 is configured to connect at 9600 b/s and start the command line interface; port 1 is not configured. If you connect through port 1, you must perform additional steps to configure the port.

2. Plug the other end of the cable into your PC serial port, taking note of which port you choose.

3. If you are connecting through port 1 on the LifeSize Room codec for the first time, do the following. Otherwise, skip to step 4.

   a. ssh into the system as user ***auto***, default password ***lifesize***.

      Note: On Windows, you may need to download an ssh application, such as putty.

   b. Once connected, enter the following command to configure port 1 on LifeSize Room:

      ```
      set serial port1 -s auto
      ```

   c. To terminate an ssh session, enter the **exit** verb or press **Control-D**.

4. Depending on your operating system, do one of the following:

   - On Windows, start HyperTerminal (**Start>All Programs>Accessories>Communications>HyperTerminal**) and configure it for the serial port you selected on the PC. If you are connecting to port 2 on the LifeSize Room codec, set the speed to 9600 b/s, no parity, 8 data bits, 1 stop bit, no flow control. If you are connecting to port 1, set the speed to 38400 b/s. Press **Return** in HyperTerminal until you receive the `ok,00` message.
   - On Linux, start Minicom and configure it for the serial port you selected on the PC. Set the speed to 9600, no parity, 8 data bits, 1 stop bit, no flow control. If you are connecting to port 1, set the speed to 38400 b/s. Press **Return** in Minicom until you receive the `ok,00` message.

For more information about configuring the serial ports in LifeSize Room, see “serial” on page 81.

Serial Cable Pin Assignment

Following is the cable pinout for connecting the LifeSize Room serial port to another DTE device using a null modem serial cable. Refer to “Connecting through Serial Ports” on page 5 for more information.

LifeSize Codec		Other DTE Device	
Signal	**D-sub 9 pin**	**D-sub 9 pin**	**Signal**
Receive Data	2	3	Transmit Data
Transmit Data	3	2	Receive Data
System Ground	5	5	System Ground

Help Mode

The command line interface has two modes of operation: normal mode and help mode. By default, the CLI starts in help mode.

In help mode, full command help is available, as is abbreviation support. The help mode setting exists only for the duration of the current instance. It is not shared between multiple instances. Help mode is enabled by default, but can be controlled through the **`set help-mode`** command. Because help mode also enables command abbreviations, LifeSize recommends that you run automated scripts or programs with the help mode set to **`off`** to prevent using abbreviations in these types of situations. Refer to “help-mode” on page 75 for details about using this command.

All commands provide basic usage information when you specify the **`-h`** option to the command at any point in the argument list. **`Help`** followed by a verb produces the list of targets for that verb.

Default Output Mode

The command line interface supports a default output mode option. Like help mode, it exists only for the duration of the CLI instance and is not shared between instances. By default, terse output mode is enabled. You can change this mode using the **`set verbose-mode on`** command. This is equivalent to specifying the **`-V`** option to each command entered.

> ***Note:*** Asynchronous status messages are always printed in terse mode using the default delimiter, regardless of the current state of verbose mode or any delimiter option used on the command that caused the asynchronous message to occur.

Command Line Arguments

You can invoke a single command by specifying that command on the command line, for example:

```
ssh auto@lifesize get camera position
```

In this example, the return code of the ssh command is the result code from the single command executed.

Command Line History and Recall

The CLI also supports command line history, editing, and recall through the readline library. These features operate in an identical manner to the GNU bash shell, including support for **`!`*n***, **`!!`** and Emacs editing modes. History is limited to the last 100 commands.

Command Syntax

In general, the syntax is relatively rigid to ensure consistency across all commands that the shell supports.

Command Verbs

The general format of a command is <*verb*> <*object*> <*target*> [*options*] where:

<*verb*> defines the operation to perform.

<*object*> defines the subsystem on which the operation should be performed.

<*target*> identifies the specific parameter within the object.

The CLI verbs are **`get`**, **`set`**, **`control, history, status`**, **`exit`** and **`help`**.

If help mode is enabled, help is available for the verbs, objects, and targets. In this context, a complete command is defined as a verb followed by an object and complete target specification. (For two word targets, you must specify both to complete the command). If you specify an incomplete command, all possible completions for that command root are displayed in alphabetical order. Additionally, the command processor allows abbreviations of command targets and verbs to simplify usage and to allow for more descriptive targets.

LifeSize recommends you do not use abbreviations in shell scripts, because future releases may make the abbreviation ambiguous. To prevent such use, abbreviations are disabled when help mode is off.

`get`	The **`get`** verb retrieves preference configuration information from the system (for example, displaying the current IP configuration).
`set`	The **`set`** verb applies new preferences to the system configuration (for example, changing the camera position).
`control`	The **`control`** verb initiates an action on the system (for example, placing a call).
`status`	The **`status`** verb retrieves system status information (for example, call information).
`exit`	The **`exit`** verb exits the shell prior to the end of input. You can also exit the shell by entering the end-of-file character (generally **^D**).
`help`	The **`help`** verb is available only in help mode. It lists the verbs available in the shell (but does not list the individual targets for those verbs). **`Help`** followed by a verb produces the list of targets for that verb (as if just the verb had been entered on the command line).

`history`	The **`history`** verb lists the saved history of commands up to 100 lines. Blank and commented lines are not included. To limit the number of lines displayed to fewer than 100, type the verb followed by the number of lines to display. To execute a command from the history list, type ***`!X`***, where ***`X`*** is the number of the command. For example, if the **`history`** verb displayed the following history of commands: **`history`** *`1,control remote back back`* *`2,get camera position`* *`3,control call dial redial:1`* `ok,00` then, **`!2`** would execute command 2 (**`get camera position`**). The history is persistent across shell invocations.

Standard Options

All of the command verbs support a small set of standard command line options to provide a basic level of consistency.

Provide Help: -h

All commands provide basic usage information for interactive users. Specify the **`-h`** option in the command at any point in the argument list. When you specify **`-h`** at any level other than that of a completed command, a list of all possible completions appears (**`-h`** is ignored in this case). Command help is available only when help mode is enabled.

Enable Verbose Output: -V

By default, command output appears in terse format suitable for processing by scripts. If you specify **`-V`** , output appears in a tabular format with headers describing each column. A minimum of two spaces separate each column value. This format is suitable for human parsing and for use during prototyping. The order of the columns presented in verbose and terse modes is the same, so you can rely on the output in verbose mode to guide column selection in terse mode. To enable verbose permanently, set verbose-mode to **on**.

Set the Terse Mode Column Delimiter: -D <c>

The default column delimiter in terse mode is the comma (',') character. Use the **`-D`** option to change the delimiter to any single character other than space (ASCII 0x20) or newline (ASCII 0x0a) . The first character of the argument to **`-D`** is the new delimiter character. When outputting data in terse mode, any occurrence of the delimiter character in the output is replaced with the space character. The **`-D`** option and the **`-V`** option (or enabling verbose mode as a default) are mutually exclusive. In the event both are specified, **`-D`** is ignored.

Standard Output Format

All of the internal commands produce output in a specific format, based on the default output mode or the presence of the **-V** option.

Terse Mode Output

Terse mode is the default output mode. It is designed to be easily parsed by shell scripts and automated programs. The general format of the output is rows of comma-separated text. To change the separation character, specify the **-D** option. The completion code for the command is also sent to the output stream. For example:

```
get network
static,10.10.100.5,255.255.255.0,10.10.100.1,00:13:fa:00:24:a1,
   jsmith-ls

ok,00

get unknown-target

error,09
```

To allow differentiation between command output and the completion code output, a single newline is always inserted between the last line of command output and the completion code. Command output is not allowed to contain any blank lines. The completion code is printed as <*status*>,<*code*> where status is either *ok* or *error* and code is a two digit hexadecimal number. A code value of 00 indicates success of the command. Any other value indicates an error condition.

Verbose Mode Output

Verbose mode is enabled by specifying the **-V** option to a command. It may also be enabled globally by setting verbose mode to **on**. Verbose mode is designed for human parsing and is formatted in a tabular style. Verbose mode is not intended to be parsed by automated scripts. For example:

```
get network –V
Type    IP Address   Network Mask   Gateway       MAC Address
static  10.10.100.5  255.255.255.0  10.10.100.1   00:13:fa:00:24:a1

ok

get unknown-target

error 09 Target not recognized
```

Generating the Command Listing

To generate a complete list of the commands available in the CLI shell, execute the following:

```
% set help-mode on

ok,00
% help
control
get
exit
help
history
set
status

ok,00
% control
<control command list>

ok,00
% get
<get command list>
...
```

The result is a list of the supported commands available in the CLI with the exception of the help, history, and exit top level commands.

get Verb

The following objects and targets are applicable to the **`get`** verb.

audio

The following targets are applicable to the **`audio`** object.

codecs

The **`codecs`** target retrieves the codec priority list. This list determines the order in which the audio codes are used when connecting to other systems.

Arguments:

None

Examples:

```
get audio codecs
aac-lc g.722 g.722.1c.32 g.722.1c.24 g.729 g.711.u g.711.a

ok,00

get audio codecs -V
Codec Order
aac-lc g.722 g.722.1c.32 g.722.1c.24 g.729 g.711.u g.711.a

ok
```

Following are the available codecs:

Codec	Autosh Name
AAC Low Complexity	aac-lc
Polycom® Siren™ 14 (48 kb/s)	g.722.1c.48
Polycom® Siren™ 14 (32 kb/s)	g.722.1c.32
Polycom® Siren™ 14 (24 kb/s)	g.722.1c.24
G.722	g.722
G.729	g.729
G.711 μ-Law	g.711.u
G.711 a Law	g.711.a

eq

The **eq** target retrieves the bass and treble equalization settings.

Arguments:

None

Examples:

```
get audio eq
0,-3

ok,00

get audio eq -V
Bass     Treble
4        -5

ok
```

in-to-out

The **`in-to-out`** target shows the state of the line input to line output function. If on, the line input is routed to the line output.

Arguments:

None

Examples:

```
get audio in-to-out
off

ok,00

get audio in-to-out -v
State
on

ok
```

line-in

The **`line-in`** target retrieves the current setting of the line input. For LifeSize Room, this value is either on or off. For LifeSize Team systems, this value is either stereo, mono, or off.

Arguments:

None

Examples:

```
get audio line-in
on

ok,00

get audio line-in -V
State
stereo

ok
```

mics

The **`mics`** target retrieves the current setting of the microphones.

Arguments:

None

Examples:

```
get audio mics
off

ok,00

get audio mics -V
State
off

ok
```

mute

The **`mute`** target retrieves the current setting of the local audio mute function.

Arguments:

None

Examples:

```
get audio mute
off

ok,00

get audio mute -V
State
on

ok
```

video-output

The **`video-output`** target retrieves the current setting of the video call audio output.

Arguments:

None

Examples:

```
get audio video-output
phone

ok,00

get audio video-output -V
Destination
room

ok
```

call

The following targets are applicable to the **`call`** properties object.

auto-answer

The **`auto-answer`** target retrieves the current value of the single-way auto answer setting.

Arguments:

None

Examples:

```
get call auto-answer
on

ok,00

get call auto-answer -V
State
off

ok
```

auto-multiway

The **`auto-multiway`** target retrieves the current value of the multi-way auto answer setting.

Arguments:

None

Examples:

```
get call auto-multiway
on

ok,00

get call auto-multiway -V
State
off

ok
```

auto-mute

The **`auto-mute`** target retrieves the current value of the mute microphones on answer setting.

Arguments:

None

Examples:

```
get call auto-mute
on

ok,00

get call auto-mute -V
State
off

ok
```

max-redial-entries

The **`max-redial-entries`** target retrieves the maximum number of redial entries that may be stored in the directory.

Arguments:

None

Examples:

```
get call max-redial-entries
11

ok,00

get call max-redial-entries -V
Quantity
9

ok
```

max-speed

The **`max-speed`** target retrieves the current setting for the maximum incoming and outgoing bandwidth usable by a call. Speed is reported in kilobits per second.

Arguments:

None

Examples:

```
get call max-speed
1024,768

ok,00

get call max-speed -V
Incoming       Outgoing
512            1024

ok
```

max-time

The **`max-time`** target retrieves the maximum time that a call can be active.

Arguments:

None

Examples:

```
get call max-time
unlimited

ok,00

get call max-time -V
Time In Hours
4

ok
```

camera

The following targets are applicable to the **`camera`** object.

active

The **`active`** target retrieves the current active high definition camera. Only one HD camera can be active at a time. This command is supported only with LifeSize Room.

Arguments:

None

Examples:

```
get camera active
1

ok,00

get camera brightness -V
Camera
2

ok
```

anti-flicker

The **`anti-flicker`** target retrieves the current anti-flicker setting for the cameras.

Arguments:

None

Examples:

```
get camera anti-flicker
auto

ok,00

get camera anti-flicker -V
Mode
60Hz

ok
```

brightness

The **`brightness`** target retrieves the current brightness adjustment value for the camera. Negative numbers decrease the overall brightness, positive numbers increase brightness.

Arguments:

None

Examples:

```
get camera brightness
-20

ok,00

get camera brightness -V
Adjustment
25

ok
```

far-control

The **`far-control`** target retrieves the current state of the far control of the near camera setting.

Arguments:

None

Examples:

```
get camera far-control
enabled

ok,00

get camera far-control -V
State
disabled

ok
```

lock

The lock target retrieves the current state of the camera lock mechanism which reveals whether the camera motors are locked in place (*on*) or not (*off*).

Arguments:

None

Examples:

```
get camera lock
1,off
2,on

ok,00

get camera lock -V
Camera     Lock Mode
1          off
2          on

ok
```

pan-dir

The **`pan-dir`** target retrieves the current state of the camera pan direction setting. This setting affects the direction the camera moves when using the left and right buttons on the remote control, but does not affect the **`-l`** and **`-r`** arguments of the **`set camera position`** command.

Arguments:

None

Examples:

```
get camera pan-dir
perceived

ok,00

get camera pan-dir -V
Mode
reversed

ok
```

position

The **`position`** target retrieves the position of the indicated camera, 1, 2, or other (inactive). The default is to retrieve the position of the active camera.

Arguments:

`-N {1 | 2 | 0}`	Retrieves the position of the indicated camera, 1, 2, or other (inactive).

Note: The `-N` argument is only supported with LifeSize Room.

Examples:

```
get camera position
-30,5,15

ok,00

get camera position -D | -N 2
45|5|35

ok,00
```

```
get camera position -V
Pan      Tilt     Zoom
45       -5       30

ok
```

The ranges of the pan, tilt, and zoom values are shown in the following table. The camera may not be able to reach the maximum and minimum values for each parameter (ordering the camera to pan to -45 may return a value other than -45). This is normal. Due to sensor resolution limits, one of the camera position parameters may change slightly when another position is modified (for example, adjusting the tilt angle may slightly affect the pan position). The ranges allowed are designed to be compatible with any future version camera, therefore it is likely that your camera does not reach the limits.

	Pan	**Tilt**	**Zoom**
Minimum value	-180 degrees	-90 degrees	0
Maximum value	180 degrees	90 degrees	100

preset

The **`preset`** target retrieves the preset position information for the camera(s). The output may be restricted to a specific camera or a specific preset or set of presets. The preset position stores the associated camera and its pan, tilt, and zoom location.

Arguments:

`-N {1 | 2 | a | o}`	Retrieves the position of the indicated camera, 1, 2, or other (inactive). The default is to retrieve all presets. This argument is only supported with LifeSize Room.
`-P {0..19}`	Retrieves a specific preset, which may be specified multiple times to retrieve several preset positions. Presets are listed in the same order and displayed on the command line.

Examples:

```
get camera preset
0,any,0,0,13
1,1,0,0,50
2,2,-20,10,35
3,1,10,-5,0

ok,00
```

```
get camera preset -V -P 3 -P 2 -P 0
Preset   Camera      Pan   Tilt  Zoom
3        1           10    -5    0
2        2           -20   10    35
0        any         0     0     13

ok

get camera preset -N a
0,any,0,0,13
1,1,0,0,50
3,1,10,-5,0

ok,00
```

conference

The following targets are applicable to the **`conference`** object.

presence-mode

The **`presence-mode`** target reports whether the display shows all conference participants or only the most recent speaker.

Note: The presence-mode target is supported with LifeSize Room systems only.

Arguments:

None

Examples:

```
get conference presence-mode
continuous

ok,00

get conference presence-mode -V
Presence Mode
last-talker

ok
```

presentation

The **`presentation`** target reports whether the conference supports H.239 and is presentation capable.

Arguments:

`<conference>` The number of the conference to check.

Examples:

```
get conference presentation 1
off

ok,00

get conference presentation -V 1
Presentation Capable
on

ok
```

security

The **`security`** target reports whether encryption is required for conference calls.

Arguments:

None

Examples:

```
get conference security
off

ok,00

get conference security -V
H.235 Encryption
strict

ok
```

directory

The following targets are applicable to the **`directory`** object.

corporate | local

The **`corporate`** target retrieves the corporate directory entries and the **`local`** target retrieves the local directory entries. Otherwise, both commands are identical.

Arguments:

`[-1 number]`	Restrict the listing to those entries whose Number column starts with the specified ISDN number.
`[-2 x]`	This argument is deprecated in software release v3.0. The column labeled `B2 Number` in the output in previous releases appears with the column label `X` and is unused.
`[-B]`	Send the data in a format suitable for processing by the set directory local -B command.
`[-H hier]`	Restrict the listing to those entries whose Hierarchy column starts with the specified hierarchy grouping.
`[-a isdn-ac]`	Restrict the listing to those entries whose AC column starts with the specified ISDN area code.
`[-c isdn-cc]`	Restrict the listing to those entries whose CC column starts with the specified ISDN country code.
`[-i ip-addr]`	Restrict the listing to those entries whose IP Address column starts with the specified IP address.
`[-t phone]`	Restrict the listing to those entries whose Voice column starts with the specified voice telephone number.
`[-v video]`	Restrict the listing to those entries whose Video column starts with the specified video number.
`[prefix]`	Restrict the listing to those entries starting with the specified prefix (case insensitive.)

Examples:

```
get directory local
John Doe,,,10.10.11.254,,,,Video,auto,auto
Mary Jane,,,10.10.11.213,,,,Video,auto,auto
Noah James,,,10.10.11.116,,,,Video,1024,auto
Steve Jones,,,10.10.11.155,,,,Video,auto,auto
Test,,,555.1212,10.10.11.8,1.2.3.4,52,215,5550199,,Audio,auto,
   128

ok,00
```

```
get directory local –V
Name         Voice     Video       IP Address    CC   AC    Number   X   Hierarchy  IP BW  ISDN
                                                                                             BW

John Doe                           10.10.11.254                        Video      auto   auto

Mary Jane                          10.10.11.213                        Video      auto   auto

Noah James                         10.10.11.116                        Video      1024   auto

Steve Jones                        10.10.11.155                        Video      auto   auto

Test         555.1212  10.10.11.8  1.2.3.4       52   215   5550199      Audio      auto   128

ok
```

```
get directory local n
Noah James,,,10.10.11.116,,,,Video,1024,auto

ok,00
```

```
get directory local -B
John Doe|||10.10.11.254|||Video|auto|auto
Mary Jane|||10.10.11.213|||Video|auto|auto
Noah James|||10.10.11.116|||Video|1024|auto
Steve Jones|||10.10.11.155|||Video|auto|auto
Test|555.1212|10.10.11.8|1.2.3.4|52|215|5550199||Audio|auto|128

ok,00
```

Specification of multiple selection options results in output that is the logical and of all of the conditions (each line must match all conditions specified). Regular expressions in and logical ordering of the selection criteria are not supported.

ldap

The **`ldap`** target shows the configuration of the LDAP directory settings for the corporate directory.

Arguments:

None

Examples:

```
get directory ldap
ldap-server,user,OU=EndPoints DC=LifeSize DC=VideoCodecs,1-hour

ok,00
```

```
get directory ldap -V
Server       Username  Base                                        Filter  Refresh
ldap-server  user      OU=Endpoints,DC=LifeSize,DC=VideoCodecs           1-hour

ok
```

h323

The following targets are applicable to the **h323** object.

alternate

The **alternate** target retrieves the current settings for the alternate H.323 gatekeeper.

Arguments:

None

Examples:

```
get h323 alternate
10.10.11.12.1719

ok,00
```

```
get h323 alternate -V
IP Address      Port
10.10.11.110    12345

ok
```

extension

The **extension** target retrieves the H.323 extension associated with the endpoint.

Arguments:

None

Examples:

```
get h323 extension
1188

ok,00
```

```
get h323 extension -V
Extension
1188

ok
```

id

The **`id`** target retrieves the H.323 gatekeeper ID.

Arguments:

None

Examples:

```
get h323 id
RADGK

ok,00

get h323 id -V
Gatekeeper ID
RADGK

ok
```

mode

The **`mode`** target retrieves the H.323 gatekeeper mode which determines whether the gatekeeper is used at all or whether it is manually or automatically configured.

Arguments:

None

Examples:

```
get h323 mode
off

ok,00

get h323 mode -V
Mode
manual

ok
```

name

The **`name`** target retrieves the currently configured H.323 name for the endpoint.

Arguments:

None

Examples:

```
get h323 name
LifeSize

ok,00

get h323 name -V
Name
LifeSize

ok
```

primary

The **`primary`** target retrieves the configuration for the H.323 primary gatekeeper.

Arguments:

None

Examples:

```
get h323 primary
10.10.11.12,1719

ok,00

get h323 primary -V
IP Address     Port
10.10.11.110   12345

ok
```

register

The **`register`** target retrieves the current registration status on the H.323 gatekeeper.

Arguments:

None

Examples:

```
get h323 register
registered

ok,00

get h323 register -V
Status
failed

ok
```

help-mode

The **`help-mode`** target retrieves the current setting for help mode.

Arguments:

None

Examples:

```
get help-mode
on

ok,00

get help-mode -V
Mode
Off

ok
```

http

The **`http`** target shows whether the web (http) service is enabled or disabled.

Arguments:

None

Examples:

```
get http
on

ok,00

get http -V
Web (http) Service
off

ok
```

locale

The **`locale`** object shows the configuration options for setting the system's location specific information.

country

The **`country`** target shows the current country setting for the system. This setting is used by the PSTN interface to define how the PSTN connection should work.

Arguments:

None

Examples:

```
get locale country
algeria

ok,00

get locale country -V
Country
uruguay

ok
```

language

The **`language`** target shows the current language used for user interface prompts and messages. This setting does not affect the input or output of autosh.

Arguments:

None

Examples:

```
get locale language
german

ok,00

get local language -V
GUI Language
traditional-chinese

ok
```

timezone

The **`timezone`** target shows the current system timezone.

Arguments:

None

Examples:

```
get locale timezone
majuro

ok,00

get locale timezone -V
Time Zone
kirimati

ok
```

network

The **`network`** object retrieves the current network configuration.

Arguments:

None

Examples:

```
get network
static,10.10.100.1,255.255.255.0,10.10.100.254,00:13:fa:00:24:
   a1,jsmith-ls

ok,00
```

```
get network –V
Type    IP Address    Network Mask   Gateway      MAC Address        Hostname
dhcp    10.10.100.5   255.255.255.0  10.10.100.1  00:13:fa:00:24:a1  jsmith-ls

ok
```

redial-list

The **`redial-list`** target retrieves the redial call list.

Arguments:

None

Examples:

```
get redial-list
1,Sunbob2,10.10.11.116,10.10.11.116,Video,Outgoing,Yes,auto,auto
2,10.10.11.186,10.10.11.186,10.10.11.186,Video,Manual,No,
   auto,512
3,Sunbob2,10.10.11.116,10.10.11.116,Audio,Incoming,Yes,h323,auto
4,10.10.11.186,10.10.11.186,10.10.11.186,Audio,
   Multiway,No,auto,auto
5,10.10.11.155,10.10.11.155,10.10.11.155,Audio,
   Outgoing,No,auto,auto

ok,00
```

```
get redial-list -V
```

```
Index  Name          Number        IP Address    Type   Origin   Locked  Protocol  Bandwidth
1      Sunbob2       10.10.11.116  10.10.11.116  Video  Outgoing Yes     auto      auto
2      10.10.11.186  10.10.11.186  10.10.11.186  Video  Manual   No      auto      512
3      Sunbob2       10.10.11.116  10.10.11.116  Audio  Incoming Yes     h323      auto
4      10.10.11.186  10.10.11.186  10.10.11.186  Audio  Multiway No      auto      auto
5      10.10.11.155  10.10.11.155  10.10.11.155  Audio  Outgoing No      auto      auto
```

```
ok
```

The valid values for the `Type` column are *Audio*, *Video*, *Multiway*, and *Unknown* and the values for the `Origin` column are *Manual*, *Outgoing*, *Incoming*, *Multiway*, and *Unknown* where `Origin` refers to how the entry was placed into the redial list. Entries that are locked cannot be removed from the redial list with new entries. The protocol values are the same as for the ***`control call dial`*** command's **-p** argument, and the bandwidths are the same as for the **-b** argument.

serial

The **`serial`** object allows configuration of the serial ports on LifeSize Room.

port1 | port2

The **`port1`** and **`port2`** targets are identical in function except which port they affect. The port1 target affects the serial port marked "rs-232 1" and the port2 target affects the port marked "rs-232 2". By default, serial port 1 is set to 38400 b/s with no shell enabled. Serial port 2 is set to 9600 b/s with the CLI enabled.

Note: Serial port 1 is internal and unsupported on LifeSize Team. Serial port 2 is not present on LifeSize Team.

Arguments:

None

Examples:

```
get serial port1
38400,hardware,auto,delete

ok,00
```

```
get serial port2 -V
Speed    Flow Control   Shell    Erase Key
115200   software       none     backspace

ok
```

sip

The following targets are applicable to the **sip** object.

authorization

The **authorization** target retrieves the username for authorization with the SIP registrar. For security reasons, the associated password is not displayed.

Arguments:

None

Examples:

```
get sip authorization
sipuser

ok,00

get sip authorization -V
Username
sipuser

ok
```

proxy

The **proxy** target retrieves the SIP proxy settings.

Arguments:

None

Examples:

```
get sip proxy
disabled,proxy.example.com,5060

ok,00
```

```
get sip proxy -V
State      IP Address          Port
enabled    proxy.example.com   5060

ok
```

register

The **`register`** target shows the current registration status for SIP.

Arguments:

None

Examples:

```
get sip register
unregistered

ok,00

get sip register -V
Status
registered

ok
```

registrar

The **`registrar`** target retrieves the current SIP registrar settings.

Arguments:

None

Examples:

```
get sip registrar
disabled,sip.example.com,5060

ok,00

get sip registrar -V
State     IP Address         Port
enabled   sip.example.com    5060

ok
```

tcp

The **`tcp`** target shows the configuration of the TCP options for SIP calls.

Arguments:

None

Examples:

```
get sip tcp
disabled,5060

ok,00

get sip tcp -V
State      Port
enabled    5060

ok
```

udp

The **`udp`** target shows the configuration of the UDP options for SIP calls.

Arguments:

None

Examples:

```
get sip udp
disabled,5060

ok,00

get sip udp -V
State      Port
enabled    5060

ok
```

username

The **`username`** target shows the current SIP username.

Arguments:

None

Examples:

```
get sip username
lifesize

ok,00

get sip username -V
Name
lifesize

ok
```

via-proxy

The **`via-proxy`** target shows whether SIP registration will use the proxy to connect to the registrar, or will directly connect.

Arguments:

None

Examples:

```
get sip via-proxy
proxy

ok,00

get sip via-proxy -V
Registration Method
direct

ok
```

snmp

The following targets are applicable to the **`snmp`** server configuration object.

contact

The **`contact`** target retrieves the SNMP contact name.

Arguments:

None

Examples:

```
get snmp contact
Administrator

ok,00

get snmp contact -V
SNMP Contact
Administrator

ok
```

enable

The **`enable`** target shows whether the SNMP service is enabled.

Arguments:

None

Examples:

```
get snmp enable
on

ok,00

get snmp enable -V
Value
off

ok
```

location

The **`location`** target shows the configured location for the SNMP service.

Arguments:

None

Examples:

```
get snmp location
Austin

ok,00

get snmp location -V
SNMP Location
Austin

ok
```

v3trapdestination

The **`v3trapdestination`** shows the current version 3 SNMP trap destinations (where SNMP traps are sent).

Note: The user's password is not displayed.

Arguments:

None

Examples:

```
get snmp v3trapdestination
Control,10.10.11.12
joeuser,169.254.101.2

ok,00

get snmp v3trapdestination -V
Username        Host/IP Address
Control         10.10.11.12
joeuser         169.254.101.2

ok
```

ssh

The following targets are applicable to the **`ssh`** object.

keys

The **`keys`** target retrieves information about the currently installed ssh authorized keys. Authorized keys allow the remote user to log into the system without using a password. Currently, only one authorized ssh key is supported.

Arguments:

None

Examples:

```
get ssh keys
ssh-rsa,user@lifesize.com

ok,00

get ssh keys -V
Type     Owner
ssh-rsa  user@lifesize.com

ok
```

service

The **`service`** target shows whether the ssh service is enabled or disabled.

Arguments:

None

Examples:

```
get ssh service
on

ok,00

get ssh service -V
Secure Shell Service
off

ok
```

system

The following targets are applicable to the **`system`** object. The **`system`** object contains information that may be useful for tracking and monitoring inventory.

autoreboot

The **`autoreboot`** target shows the current setting of the nightly automatic reboot feature. If enabled, the system reboots itself nightly if the system is idle.

Arguments:

None

Examples:

```
get system autoreboot
off

ok,00

get system autoreboot -V
Nightly Reboot
on

ok
```

fans

The **`fans`** target shows the current speed of all system fans. The displayed speed is not recorded in a standard unit (such as revolutions per minute) but is such that the faster the fan spins, the higher the value reported.

Arguments:

None

Examples:

```
get system fans
125

ok,00

get system fans -V
Fan 1
128

ok
```

isdn

The **`isdn`** target shows the status of a connected LifeSize Networker device. If the device connection type is `Tethered` (connected to the codec), the PRI and BRI information is valid. The PRI and BRI fields indicate the number of connected ports of that type. The associated `Map` field indicates which ports are connected.

Arguments:

None

Examples:

```
get system isdn
No,None,,0,,0,

ok,00

get system isdn -V

ISDN    Type        IP Addr         PRI    Map    BRI    Map
Yes     Tethered    10.254.128.2    2      1,2    2      1,X,X,4

ok

get system isdn
Yes,Standalone,10.10.11.12,-1,,-1,

ok,00
```

model

The **`model`** target shows the OEM and model name for the platform.

Arguments:

None

Examples:

```
get system model
LifeSize,Room

ok,00
```

```
get system model -V
OEM         Model
LifeSize    Team

ok
```

name

The **name** target shows the current name for the system. This is the same value that shows in the user interface and on the connected phone.

Arguments:

None

Examples:

```
get system name
LifeSize

ok,00

get system name -V
System Name
LifeSize

ok
```

number

The **`number`** target retrieves the voice telephone number associated with the system. This is displayed in the interface and on the attached phone.

Arguments:

None

Examples:

```
get system number
555-1212

ok,00

get system number -V
System Phone Number
555-1212

ok
```

out-of-box

The **`out-of-box`** target shows the current state of the initial configuration process that starts when a system is installed or reset to its default configuration settings.

Arguments:

None

Examples:

```
get system out-of-box
enabled

ok,00

get system out-of-box -V
Out Of Box Setup
complete

ok
```

pstn

The **`pstn`** target returns whether the system has the hardware necessary to make a Public Switched Telephone Network (PSTN) call.

Note: It does not indicate whether there is an active phone line connected to the telephone jack.

Arguments:

None

Examples:

```
get system pstn
yes

ok,00

get system pstn -V
Public Switched Telephone Network Support
no

ok
```

serial-number

The **`serial-number`** target returns the serial numbers of the CPU board and System board within the codec.

Arguments

None

Examples:

```
get system serial-number
BKxxxxxxxxxx,BKxxxxxxxxxx

ok,00

get system serial-number -V
CPU Board        System Board
BKxxxxxxxxxx     BKxxxxxxxxxx

ok
```

temperatures

The **`temperatures`** target returns the current temperature readings within the system in degrees Celsius.

Arguments:

None

Examples:

```
get system temperatures
55,80,75,43

ok,00

get system temperatures -V
Board    Video In    Video Out   Ambient
55       80          75          43

ok
```

uptime

The **`uptime`** target returns the amount of time that the system has been up in days, hours, minutes, and seconds.

Arguments:

None

Examples:

```
get system uptime
5,21,13,20

ok,00

get system uptime -V
Days     Hours       Minutes     Seconds
5        21          13          40

ok
```

version

The **`version`** target returns the software version for all of the software loaded on the system.

Arguments:

None

Examples:

```
get system version
Software Version,LS_RM_2.1.0 (0)
Sysmon Version,SM_P_3 2.4 Jun 21 2006 14:37:22
U-Boot Version,U-Boot 1.1.2 LifeSize Room/Team 1.9a
Camera 0 Base,0x60525
Camera 0 Head,0x60605
Pixelworks Version,LS_QMBRom v1.6 May 23 2006 10:18:00
Video In FPGA,06061600
Video Out FPGA,05120400
Tethered Phone,LS_PH1_2.1.0 (0)
Phone Keyboard,003_000
Phone U-Boot,U-Boot 1.1.2 LifeSize Phone 1.1

ok,00

get system version -V
Software Version     Value
Software Version     LS_RM1_2.1.0 (0)
Sysmon Version       SM_P_3 2.4 Jun 21 2006 14:37:22
U-Boot Version       U-Boot 1.1.2.LifeSize Room/Team, 1.9a
Camera 0 Base        0x60525
Camera 0 Head        0x60605
Pixelworks Version   LS_QMBRom v1.6 May 23 2006 10:18:00
Video In FPGA        06061600
Video Out FPGA       051120400
Tethered Phone       LS_PH1_2.1.0 (0)
Phone Keyboard       003_000
Phone U-Boot         U-Boot 1.1.2 LifeSize Phone, 1.1

ok
```

video-number

The **`video-number`** target retrieves the video telephone number associated with the system. This number appears in the user interface.

Arguments:

None

Examples:

```
get system video-number
555-1213

ok,00

get system number -V
System Video Number
555-1213

ok
```

voltages

The **`voltages`** target reports the current system voltages in Volts. The order of the voltages and the voltages reported may change with each release. The **`voltages`** target is not supported with LifeSize Express.

Arguments:

None

Examples:

```
get system voltages
1.22,1.45,1.80,12.00,19.20,2.63,3.28,4.97

ok,00

get system voltages -V
1.2,1.5,1.8,12.0,19.0,2.5,3.3,5.0
1.22,1.45,1.80,12.00,19.20,2.63,3.28,4.97

ok
```

telnet

The **`telnet`** object retrieves the current state of telnet protocol support. LifeSize recommends that you disable **`telnet`**, because it is an insecure protocol. If you must use telnet, place the system behind a firewall or other external security device. By default, the telnet protocol service is disabled.

Arguments:

None

Examples:

```
get telnet
on

ok,00

get telnet -V
Telnet Service
off

ok
```

timer

The following targets are applicable to the **`timer`** object.

caller-id

The **`caller-id`** target retrieves the current setting of the caller ID display timeout.

Arguments:

None

Examples:

```
get timer caller-id
30

ok,00

get timer caller-id -V
Timeout in Seconds
on

ok
```

fadeout

The **`fadeout`** target retrieves the current setting for the user interface fadeout timer.

Arguments:

None

Examples:

```
get timer fadeout
6

ok,00

get timer fadeout -V
Timeout in Seconds
5

ok
```

screen-saver

The **`screen-saver`** target retrieves the current setting for the screen saver timer (the amount of idle time before the screen saver activates).

Arguments:

None

Examples:

```
get timer screen-saver
20

ok,00

get timer screen-saver -V
Timeout in Minutes
none

ok
```

sleep

The **`sleep`** target retrieves the current setting for the sleep timer (the amount of idle time after the screen saver activates and before the system enters sleep state).

Arguments:

None

Examples:

```
get timer sleep
30

ok,00

get timer sleep -V
Timeout in Minutes
10

ok
```

verbose-mode

The **`verbose-mode`** target retrieves the current setting for verbose mode.

Arguments:

None

Examples:

```
get verbose-mode
on

ok,00

get verbose-mode -V
Mode
off

ok
```

video

The following targets are applicable to the **`video`** object.

input-names

The **`input-names`** target retrieves the display names associated with the various video inputs. For LifeSize Room, names are returned for each of the 5 input sources (two HD cameras, document camera, VCR input, and VGA input). For LifeSize Team, names are returned for each of the three input sources (HD camera, document camera, and VGA input).

Arguments:

None

Examples:

```
get video input-names
HD Camera,Unused,Document Camera,VCR,PC

ok,00

get video input-names -V # LifeSize Room
HD 0        HD 1     SD 0              SD 1    VGA 0
HD Camera   Unused   Document Camera   VCR     PC

ok

get video input-names -V # LifeSize Team
HD 0        SD 0              VGA 0
HD Camera   Document Camera   PC

ok
```

pip-mode

The **`pip-mode`** target retrieves the current state of the PIP feature.

Arguments:

None

Examples:

```
get video pip-mode
auto

ok,00

get video pip-mode -V
State
on

ok
```

primary-input

The **`primary-input`** target retrieves the current setting for the primary input. For LifeSize Room, the valid values are hd0, hd1, sd0, sd1, and vga0. For LifeSize Team, the valid values are hd0, sd0, and vga0.

Arguments:

None

Examples:

```
get video primary-input
hd0
ok,00

get video primary-input -V
Input
hd0

ok
```

secondary-input

The **`secondary-input`** target retrieves the current setting for the secondary input. For LifeSize Room, the valid values are hd0, hd1, sd0, sd1, and vga0. For LifeSize Team, the valid values are hd0, sd0, and vga0.

Arguments:

None

Examples:

```
get video secondary-input
vga0

ok,00

get video secondary-input -V
Input
sd0

ok
```

volume

The following targets are applicable to the **`volume`** object.

aux-in

The **`aux-in`** target retrieves the volume setting for the auxiliary input on LifeSize Room systems. The scale is 0 to 10. This target is supported with LifeSize Room only.

Arguments:

None

Examples:

```
get volume aux-in
5

ok,00

get volume aux-in -V
Volume
6

ok
```

dtmf

The **`dtmf`** target retrieves the current volume setting for Dual Tone Multi Frequency (DTMF) tones when dialing (using a scale of 0 to 10).

Arguments:

None

Examples:

```
get volume dtmf
5

ok,00

get volume dtmf -V
Volume
6

ok
```

line-in

The **`line-in`** target retrieves the relative volume setting for the line input. The scale is 0 to 10.

Arguments:

None

Examples:

```
get volume line-in
5

ok,00

get volume line-in -V
Volume
6

ok
```

ring-tone

The **`ring-tone`** target retrieves the current volume setting for the ring tone.

Arguments:

None

Examples:

```
get volume ring-tone
5

ok,00

get volume ring-tone -V
Volume
6

ok
```

speaker

The **`speaker`** target retrieves the current volume setting for the system speaker (audio loudness).

Arguments:

None

Examples:

```
get volume speaker
50
ok,00

get volume speaker -V
Volume
70

ok
```

set Verb

The following objects and targets are applicable to the **`set`** verb.

admin

The **`admin`** object allows configuration of administrator functions in the interface.

password

The **`password`** target sets the password for access to the administrator preferences.

Arguments:

`<value>`	The new administrator password. The password must contain only the numbers 0-9 and/or the symbols * and #. It must be at least 1 character in length and is silently truncated to 16 characters.

Examples:

```
set admin password 12345*#

ok,00

set admin password -V abcdef

error  04  Invalid Parameter
```

audio

The following targets are applicable to the **`audio`** object.

codecs

The **`codecs`** target allows changing the order in which the audio codecs are used when negotiating with a remote system. The list is specified in highest priority to lowest priority order. For greatest compatibility, all available codecs should be listed. Each codec may be listed only once. Refer to **`get audio codecs`** on page 12 for a description of each of the arguments.

Arguments:

`{aac-lc | g.722.1c.24 | g.722.1c.32 | g.722.1c.48 | g.722 | g.729 | g.711.u | g.711.a`	Specify the audio codecs to use. List each codec only once on the command line for greatest compatibility.

Examples:

`set audio codecs` ***`aac-lc g.722 g.722.1c.32 g.722.1c.24 g.711.a g.711.u g.729`***

```
ok,00
```

eq

The **`eq`** target sets the bass and treble equalization parameters.

Arguments:

`[-b {-10..10}]`	Specify the bass equalization value.
`[-t {-10..10}]`	Specify the treble equalization value.

Examples:

`set audio eq -b` ***`-5`*** **`-t`** ***`6`***

```
ok,00
```

line-in

The **`line-in`** target controls whether the line input is enabled. On LifeSize Room, the line input is mono only so it can be either enabled or disabled. On LifeSize Team, the line input is stereo so it can be enabled in stereo mode, mono mode (left channel input) or disabled.

Arguments:

`<{on|off}>`	Enable or disable the line in input for LifeSize Room.
`<{stereo|mono|off}>`	Enable or disable the line in input for LifeSize Team.

Examples:

`set audio line-in` ***`on`***

```
ok,00
```

mics

The **`mics`** target controls whether the microphones are enabled.

Arguments:

`<{on|off}>`	Enable or disable microphone inputs.

Examples:

`set audio mics` ***`off`***

`ok,00`

mute

The **`mute`** target controls whether the local audio inputs are muted.

Arguments:

`<{on|off}>`	Mute or unmute the local audio inputs.

Examples:

`set audio mute` ***`on`***

`ok,00`

video-output

The **`video-output`** target controls whether the video call audio output is line out for the phone or the room.

Arguments:

`<{room|phone}>`	Choose the LifeSize Room or LifeSize Room line out jack or LifeSize Phone as the audio output device for video calls.

Examples:

`set audio video-output` ***`phone`***

`ok,00`

call

The following targets are applicable to the **`call`** properties object.

auto-answer

The **`auto-answer`** target controls whether the system automatically answers incoming single-way calls.

Arguments:

`<{on|off}>`	Enable or disable single-way call auto answer.

Examples:

`set call auto-answer` ***`on`***

`ok,00`

auto-multiway

The **`auto-multiway`** target controls whether the system automatically answers incoming multi-way calls.

Arguments:

`<{on|off}>`	Enable or disable multi-way call auto answer.

Examples:

`set call auto-multiway` ***`off`***

`ok,00`

auto-mute

The **`auto-mute`** target controls whether the system automatically mutes the microphone inputs when answering a call.

Arguments:

`<{on|off}>`	Enable or disable muting of the microphones when answering a call.

Examples:

`set call auto-mute` ***`off`***

`ok,00`

max-redial-entries

The **`max-redial-entries`** target controls the number of redial entries that may be stored in the redial list.

Arguments:

`<{5|7|9|11|13}>`	Specify the maximum number of entries in the redial list.

Examples:

`set call max-redial-entries` ***`11`***

`ok,00`

max-speed

The **`max-speed`** target controls the current settings for the maximum incoming and outgoing bandwidth usable by a call. Bandwidths greater than 2500 are not supported on LifeSize Team and will be rejected.

Arguments:

`[-i {128|192|256|320|384|512|768|1024|1152|1472|1728|1920|2500|3000|4000|5000|auto}]`	Specify the maximum incoming bandwidth to be used by a call in kilobits per second.
`[-o {128|192|256|320|384|512|768|1024|1152|1472|1728|1920|2500|3000|4000|5000|auto}]`	Specify the maximum outgoing bandwidth to be used by a call in kilobits per second.

Examples:

```
set call max-speed -i 512 -o auto

ok,00
```

max-time

The **`max-time`** target controls the maximum time that a call can be active.

Arguments:

`<{1|2|4|6|8|24|unlimited}>`	Specify the maximum time in hours that a call can be active.

Examples:

```
set call max-time unlimited

ok, 00
set call max-time 4

ok,00
```

camera

The following targets are applicable to the **`camera`** object.

active

The **`active`** target controls which of the high definition cameras is active. Only one HD camera may be active at a time. The active HD camera may also be affected by the set video primary-input and set video secondary-input commands. This command is supported with LifeSize Room only.

Arguments:

`<{1|2|0}>`	Specify the new active camera; either 1, 2, or the inactive camera.

Examples:

```
set camera active 1

ok,00

set camera active 0

ok,00
```

anti-flicker

The anti-flicker target controls the flicker avoidance setting of the cameras. If the local video image flickers, try the various arguments to see if the flicker goes away. Flicker is usually caused by florescent lighting.

Arguments:

`<{auto|50hz|60hz}>`	Specify the anti-flicker setting for the cameras.

Examples:

```
set camera anti-flicker 50hz

ok,00
```

brightness

The **`brightness`** target controls the brightness value for the camera's automatic iris function.

Arguments:

`<{-30..30}>`	Specify the brightness adjustment value. Negative numbers darken the image, positive numbers lighten it.

Examples:

```
set camera brightness 3

ok,00

set camera brightness -- -5

ok,00
```

Note: You must specify the double dash option (**`--`**) before ***`-5`*** to prevent the shell from interpreting ***`-5`*** as an option rather than as an argument.

far-control

The **`far-control`** target enables or disables the far control of the near camera function.

Arguments:

`<{enabled|disabled}>`	Specify whether far control of the near camera is enabled.

Examples:

```
set camera far-control disabled

ok,00
```

lock

The lock target enables or disables camera motor movement. You cannot move or zoom a camera that is locked.

Arguments:

`[-N <{1|2}>]`	Specify the camera to lock (LifeSize Room only). If not specified, all cameras are locked.
`<{on|off}>`	Enable or disable the lock.

Examples:

```
set camera lock on

ok,00

set camera lock -N 2 off

ok,00
```

position

The **`position`** target controls the current position of the near camera. Using the absolute position commands, you can recall a preset position. Using the motion commands, remote control of the camera is possible (through press and release style button mappings).

Arguments:

`[-p <{-180..180}>]`	Specify the absolute pan angle. The range is -180 to +180 and represents the degrees left (negative) or right (positive) of the center position that the camera should pan. Cannot be used with -r, -l, -u, -d, -n, -f, -s, or -c.
`[-t <{-90..90}>]`	Specify the absolute tilt angle. The range is -90 to +90 and represents the degrees below (negative) or above (positive) horizontal that the camera should tilt. Cannot be used with -r, -l, -u, -d, -n, -f, -s, or -c.
`[-z <{0..100}>]`	Specify the absolute zoom position. The range is 0 to 100 and represents the range of the zoom lens from widest angle (0) to narrowest angle (100). Cannot be used with -r, -l, -u, -d, -n, -f, -s, or -c.
`[-l]`	Specify that the camera pans to the left. Cannot be used with any other argument except -c.
`[-r]`	Specify that the camera pans to the right. Cannot be used with any other argument except -c.
`[-d]`	Specify that the camera tilts down. Cannot be used with any other argument except -c.
`[-u]`	Specify that the camera tilts up. Cannot be used with any other argument except -c.
`[-n]`	Specify that the camera zooms in (telephoto). Cannot be used with any other argument except -c.
`[-f]`	Specify that the camera zooms out (widens). Cannot be used with any other argument except -c.
`[-s]`	Specify that the camera stop all movements. Cannot be used with any other argument except -c.
`[-c n]`	Specify that the camera control operation is applied to the far camera for the specified call. Cannot be used with -p, -t, or -z.

`[-N <{1|2|o}>]`	Specify a specific camera to apply the change. Cannot be used with -r, -l, -u, -d, -n, -f, -s, or -c. By default, the active camera is controlled. The continuous motion commands cannot be used because the inactive camera's video stream is not available. This is applicable to LifeSize Room only.
`[-P <{0..19}>]`	Specify the preset to which to move. Can only be used with the -N option when the preset 0 is selected. This command moves the camera indicated by the preset to the position specified.

Examples:

Note: The double dash (--) in the following input is not required, because `-15` is an argument to the option `–t`.

`set camera position -p 34 -t -15`

`ok,00`

Start the camera panning left, then zoom in and stop all operations.

`set camera position -l`

`ok,00`

`set camera position -n`

`ok,00`

`set camera position -s`

`ok,00`

Start the far camera panning left in call 1.

`set camera position –r –c 1`

`ok,00`

Note: The command completes immediately and does not wait for the camera to complete the operation.

Move the inactive camera to a specific position.

`set camera position –N o -p 17 -t 19 -z 30`

`ok,00`

Move to a specific preset position.

`set camera position –P 12`

```
ok,00
```

preset

The **`preset`** target allows storing the preset positions for the camera in the codec's memory. Up to 19 different positions can be stored. Positions 1-9 are recallable via the remote control (in addition to the special position 0) and positions 10-19 are only recallable using the shell.

Arguments:

`[-N <{1|2|a|o}>]`	Specify the camera to which the preset will apply (1, 2, active, or inactive). LifeSize Room Only.
`-P <{1..19}>`	Specify the preset position to store.
`[-p<-180..180}>]`	Specify the pan position of the preset (default: current position of the selected camera).
`[-t<{-90..90}>]`	Specify the tilt position of the preset (default: current position of the selected camera).
`[-z<{0..100}>]`	Specify the zoom position of the preset (default: current position of the selected camera).

Examples:

Set preset 2 to the active camera's current position.

`set camera preset -P 2`

```
ok,00
```

Set preset 3 to the inactive camera's current position.

`set camera preset -P 3 -N o`

```
ok,00
```

Set preset 4 to a specific position for camera 1.

`set camera preset -P 4 -N 1 -p 28 -t 4 -z 20`

```
ok,00
```

conference

The following targets are applicable to the **`conference`** object.

presence-mode

The **`presence-mode`** target changes how video from connected parties appears in the display while in a conference call.

Note: The presence-mode target is supported with LifeSize Room systems only.

Arguments:

`<{continuous|last-talker}>`	Specify ***`continuous`*** to show all parties in the conference. Specify ***`last-talker`*** to show the most recent speaker.

Examples:

```
set conference presence-mode last-talker

ok,00
```

security

The **`security`** target enables you to change the use of encryption during conference calls.

Arguments:

`<{off|on|strict}>`	Specify ***`off`*** to disable encryption. Specify ***`on`*** to allow encryption. Specify ***`strict`*** to require encryption.

Examples:

```
set conference security strict

ok,00
```

directory

The following targets are applicable to the **`directory`** object.

ldap

The **`ldap`** target configures the LDAP directory server preferences.

Arguments:

`[-f filter]`	Specify the LDAP server search filter.
`[-p password]`	Specify the password used to access the LDAP server.
`[-r {1-minute|5-minutes| 10-minutes|20-minutes| 30-minutes|1-hour| 2-hours|3-hours| 6-hours|12-hours| 1-day|1-week|never}]`	Specify the time interval between LDAP server refreshes.
`[-s host-or-ip]`	Specify the hostname or IP address of the LDAP server.
`[-u username]`	Specify the username used to access the LDAP server.
`[dc-value]`	Specify the values of the DC fields in the order of use. The OU field is hard coded as `EndPoints`.

Examples:

`set directory ldap -u` ***`user`*** **`-p`** ***`password`*** **`-s`** ***`ldap-server LifeSize VideoCodec`***

```
ok,00
```

`set directory ldap -r` ***`1-hour`***

```
ok,00
```

local

The **`local`** target allows editing of the local directory.

Arguments:

`[-1 number]`	Set the ISDN number.
`[-2 x]`	This argument is deprecated in software release v3.0. In previous releases, this argument specified the ISDN B2 Number. The column labeled `B2 Number` in the output in previous releases appears with the column label `X` and is unused.
`[-B]`	Read data from standard input in batch mode. Accepts either the output of get local directory -B or the result of exporting the directory through the web UI. Cannot be used with any of the field set options.
`[-H hier]`	Specify the hierarchy for the entry. Use commas to delimit the hierarchy levels.
`[-K {auto|128|192|256| 320|384|512|768| 1024|1152|1472| 1728|1920}]`	Set the bandwidth limit in kbps for ISDN calls. Default is auto.
`[-a isdn-ac]`	Set the ISDN area code.
`[-c isdn-cc]`	Set the ISDN country code.
`[-d]`	Delete the specified entries. Only the name argument is used in single or batch mode. Matching entries are removed from the directory. Entries must match exactly (except for case).
`[-i ip-addr]`	Set the IP address for calls (used as backup if the video or telephone numbers are not specified.)
`[-k {auto|128|192|256| 320|384|512|768| 1024|1152|1472| 1728|1920|2500| 3000|4000|5000}]`	Set the bandwidth limit in kbps for video calls. Default is auto. Speeds above 2500 are not available with LifeSize Team.
`[-o]`	Overwrite existing entries. Default is to leave existing entries unchanged.
`[-t phone]`	Set the voice telephone number.

`[-u]`	Update existing entries by merging new data with old. The default is to leave existing entries unchanged.
`[-v video]`	Set the video telephone number.
`name`	Set the name of the entry. Case is preserved, but is not significant in locating a matching entry. Cannot be used in batch mode.

Examples:

Copy the local directory from one system to another.

```
sh$ ssh auto@room1 get directory local -B > localdir.txt
sh$ ssh auto@room2 set directory local -o -B < localdir.txt
```

Add an entry for “Sunbob” to the local directory stored hierarchically under Sun and then bob.

```
set directory local -i 10.10.10.11 -H Sun,bob Sunbob
```

```
ok,00
```

Update the sunbob entry to include a voice number.

```
set directory local -t 5551212 -u sunbob
```

```
ok,00
```

Overwrite the sunbob entry removing the voice number.

```
set directory local -i 10.10.10.11 -H Sun,bob -o Sunbob
```

```
ok,00
```

Alternative.

```
set directory local -t “” -u Sunbob
```

```
ok,00
```

Delete the sunbob entry.

```
set directory local -d Sunbob
```

```
ok,00
```

h323

The following targets are applicable to the **`h323`** object.

alternate

The **`alternate`** target configures the settings for the alternate H.323 gatekeeper when in manual mode.

Arguments:

`ipaddr`	Specify the IP address for the gatekeeper in manual mode.
`[port]`	Optional: Specify the port for the gatekeeper. The default is 1719 or the current setting.

Examples:

`set h323 alternate` ***`10.10.11.12`***

```
ok,00
```

`set h323 alternate` ***`10.10.11.12 1832`***

```
ok,00
```

extension

The **`extension`** target is used for H.323 calls to the endpoint.

Arguments:

`extension`	Specify the extension to use when registering with the H.323 gatekeeper.

Examples:

`set h323 extension` ***`1188`***

```
ok,00
```

id

The **`id`** target sets the H.323 gatekeeper ID.

Arguments:

`id`	Specify the gatekeeper ID.

Examples:

```
set h323 id RADGK

ok,00
```

mode

The **`mode`** target configures the H.323 gatekeeper mode. This determines whether the gatekeeper is used at all or whether it is manually or automatically configured.

Arguments:

`{off|manual|auto}`	Specify the gatekeeper mode. Off disables use of the H.323 gatekeeper, manual uses the primary and alternate settings, and auto determines the gatekeeper information automatically.

Examples:

```
set h323 mode auto

ok,00

set h323 mode off

ok,00
```

name

The **`name`** target sets the H.323 name for the endpoint.

Arguments:

`name`	Specify the name to use for the endpoint when registering with the H.323 gatekeeper.

Examples:

```
set h323 name LifeSize

ok,00
```

primary

The **`primary`** target configures the H.323 primary gatekeeper when in manual mode.

Arguments:

`ipaddr`	Specify the IP address for the gatekeeper in manual mode.
`[port]`	Optional: Specify the port for the gatekeeper. The default is 1719 or the current setting.

Examples:

```
set h323 primary 10.10.11.12 1719

ok,00

set h323 primary 10.10.11.15

ok,00
```

register

The **`register`** target starts the registration process with the configured H.323 gatekeeper. Because registration may take an arbitrarily long time, the command returns immediately. Status is available via the **`get h323 register`** command.

Arguments:

None

Examples:

```
set h323 register

ok,00
```

help-mode

The **`help-mode`** target controls whether help is available. It also enables and disables the ability to use abbreviations for commands (abbreviating help-mode as just help). Never use abbreviations in scripts because they may become ambiguous with future software releases.

Arguments:

`<{on|off}>`	Enable or disable help and abbreviation mode.

Examples:

```
set help-mode on

ok,00
```

http

The **`http`** target controls whether the web (http) service is enabled or disabled.

Arguments:

`<{off|on}>` Disable or enable the http service.

Examples:

```
set http on

ok,00
```

locale

The **`locale`** object enables you to configure options for setting the system's location-specific information.

country

The **`country`** target configures the country code used by the PSTN interface to define how the PSTN connection should work.

Arguments:

`<{algeria|argentina|australia|austria|bahrain|belarus|belgium|brazil|brunei|bulgaria|canada|chile|china|columbia|croatia|cyprus|czech-republic|denmark|ecuador|egypt|estonia|finland|france|germany|ghana|greece|hong-kong|hungary|india|indonesia|ireland|israel|italy|cote-d-ivoire|japan|jordan|kazakhstan|latvia|lebanon|lesotho|lithuania|luxembourg|malaysia|malta|mexico|morocco|netherlands|new-zealand|norway|oman|pakistan|paraguay|peru|philippines|poland|portugal|puerto-rico|qatar|romania|russia|singapore|slovakia|slovenia|south-africa|south-korea|spain|sri-lanka|sweden|switzerland|taiwan|thailand|tunisia|turkey|ukraine|united-arab-emirates|united-kingdom|united-states|uruguay|venezuela|vietnam|zambia}>`	Specify the country code to use.

Examples:

```
set locale country algeria

ok,00
```

language

The **`language`** target enables you to set the display language used for user interface prompts and messages. This setting does not affect the input or output of autosh.

Arguments:

`<{german|us-english|spanish|french|italian|japanese|korean|norwegian|brazilian-portuguese|russian|swedish|simplified-chinese|traditional-chinese}>`	Select the user interface language.

Examples:

```
set locale language brazilian-portuguese

ok,00
```

timezone

The **`timezone`** target changes the current system timezone.

Arguments:

`<{majuro|midway|honolulu|anchorage|los-angeles|vancouver|denver|edmonton|phoenix|austin|guatemala-city|manuaga|mexico-city|san-salvador|tegucigalapa|winnipeg|bogota|havana|indianapolis|kingston|lima|montreal|nassau|new-york|asuncion|caracas|halifax|la-paz|santiago|santo-domingo|san-juan|st-johns|sao-paulo|buenos-aires|montevideo|mid-atlantic|ponta-delgada|greenwich-mean-time|dublin|lisbon|london|reykjavik|amsterdam|berlin|brussels|budapest|copenhagen|madrid|oslo|paris|prague|rome|stockholm|vienna|warsaw|zagreb|zurich|athens|beirut|cairo|helsinki|istanbul|jerusalem|johannesburg|kyiv|baghdad|kuwait-city|moscow|riyadh|tehran|abu-dhabi|kabul|almaty|karachi|new-delhi|kathmandu|dhaka|yangon|bangkok|jakarta|beijing|kuala-lumpur|manila|perth|singapore-city|taipei|seoul|tokyo|darwin|adelaide|brisbane|sydney|vladivostok|suva|kamchatka|wellington|chatham-island|kiritimati}>`	Select the local timezone.

Examples:

`set locale timezone` ***`greenwich-mean-time`***

```
ok,00
```

password

The **`password`** target changes the user's password while running autosh.

Note: This function does not conform to the standard output specification because it uses the standard passwd utility to perform the change.

Arguments:

None

Examples:

```
set password
Changing password for auto
Old password:
Enter the new password (minimum of 5, maximum of 127 characters)
Please use a combination of upper and lower case letters and num-
   bers.
New password:
Re-enter password:
Password changed.

ok,00
```

prompt

The **`prompt`** target changes the default prompt ($) to any user specified string.

Arguments:

`<string>` Specify the new prompt string, use “” for an empty prompt.

Examples:

```
set prompt “% “

ok,00
% set prompt “-> “

ok,00
->
```

serial

The **`serial`** object allows configuration of the serial ports on LifeSize Room.

port1 and port2

The **`port1`** and **`port2`** targets are identical in function except which port they affect. The **`port1`** target affects the serial port marked “RS-232 1” and the **`port2`** target affects the port marked “RS-232 2”.

Note: Serial port 1 is internal and unsupported on LifeSize Team. Serial port 2 is not present on LifeSize Team.

Arguments:

`[-b {1200|2400|4800|9600| 19200|38400|57600|115200}]`	Specify the speed of the serial port.
`[-e {backspace|delete}]`	Specify the erase character to use.
`[-f {hardware|software|none}]`	Specify the flow control method to use. Hardware uses the RTS/CTS signal pins and software uses XON/XOFF (Ctrl-S/ Ctrl-Q).
`[-s {auto|none}]`	Specify the shell to run on the serial port. Auto uses the shell you are currently in and none disables the serial port.

Examples:

`set serial port1 -b` ***`115200`*** **`-e`** ***`backspace`*** **`-f`** ***`software`*** **`-s`** ***`auto`***

```
ok,00
```

`set serial port2 -b` ***`38400`***

```
ok,00
```

sip

The following targets are applicable to the **`sip`** object.

authorization

The **`authorization`** target configures the user name used for authorization with the SIP registrar.

Arguments:

`username`	Specify the username used for authorization.
`password`	Specify the password used for authorization.

Examples:

`set sip authorization` ***`sipuser sippassword`***

```
ok,00
```

proxy

The **`proxy`** target sets the SIP proxy configuration.

Arguments:

`{enabled|disabled}`	Enables or disables the use of the SIP proxy.
`[ip]`	Set the IP address or hostname of the SIP proxy. Only valid when enabled is chosen.
`[port]`	Optional: Specify the port to use on the proxy. The default is 5060 or the previously set value. Only valid when enabled is chosen.

Examples:

`set sip proxy enabled` ***`proxy.example.com`***

```
ok,00
```

`set sip proxy disabled`

```
ok,00
```

`set sip proxy enabled` ***`proxy.sip.com 6060`***

```
ok,00
```

register

The **`register`** target causes the system to register with the configured SIP server or proxy. It should be used after all other SIP configuration has been completed. Since registration may take an arbitrarily long time, this command returns immediately. Registration status is available via the **`get sip register`** command.

Arguments:

None

Examples:

```
set sip register

ok,00
```

registrar

The **`registrar`** target configures the SIP registrar settings.

Arguments:

`{enabled|disabled}`	Enables or disables the use of the SIP registrar.
`[ip]`	Set the IP address or hostname of the SIP registrar. Only valid when enabled is chosen.
`[port]`	Optional: Specify the port to use on the registrar. The default is 5060 or the previously set value. Only valid when enabled is chosen.

Examples:

```
set sip registrar enabled sip.example.com

ok,00

set sip registrar disabled

ok,00

set sip registrar enabled registrar.sip.com 6060

ok,00
```

tcp

The **`tcp`** target configures the TCP options for SIP calls. One of TCP or UDP should be enabled.

Arguments:

`{enabled|disabled}`	Enables or disables the use of TCP for SIP calls.
`[port]`	Optional: Specify the port to use for SIP calls. The default is 5060 or the previously set value. Only valid when enabled is chosen.

Examples:

`set sip tcp enabled`

```
ok,00
```

`set sip tcp disabled`

```
ok,00
```

`set sip tcp enabled` ***`7000`***

```
ok,00
```

udp

The **`udp`** target configures the UDP options for SIP calls. One of TCP or UDP should be enabled.

Arguments:

`{enabled|disabled}`	Enables or disables the use of UDP for SIP calls.
`[port]`	Optional: Specify the port to use for SIP calls. The default is 5060 or the previously set value. Only valid when enabled is chosen.

Examples:

`set sip udp enabled`

```
ok,00
```

`set sip udp disabled`

```
ok,00
```

`set sip udp enabled` ***`7000`***

`ok,00`

username

The **`username`** target sets the SIP username for the system.

Arguments:

`name`	Set the username for the system.

Examples:

`set sip username` ***`lifesize`***

`ok,00`

via-proxy

The **`via-proxy`** target controls whether SIP registration will use the proxy to connect to the registrar or will directly connect.

Arguments:

`{direct|proxy}`	Choose direct connection to the registrar or the proxy connection.

Examples:

`set sip via-proxy` ***`direct`***

`ok,00`

`set sip via-proxy` ***`proxy`***

`ok,00`

snmp

The following targets are applicable to the **`snmp`** server configuration object.

contact

The **`contact`** target sets the SNMP contact name.

Arguments:

<contactname>	Specify the contact name for the SNMP server.

Examples:

`set snmp contact` ***`Administrator`***

`ok,00`

enable

The **`enable`** target enables or disables the SNMP service.

Arguments:

`<{on|off}>`	Enable or disable the SNMP service.

Examples:

`set snmp enable` ***`on`***

`ok,00`

location

The **`location`** target sets the configured location for the SNMP service.

Arguments:

`<location>`	Specify the location for the SNMP service.

Examples:

`set snmp location` ***`Austin`***

`ok,00`

v3trapdestination

The **`v3trapdestination`** target adds or removes entries from the list of version 3 SNMP trap destinations (where SNMP traps are sent).

Arguments:

`-a`	Add the specified destination (cannot be used with -d).
`-d`	Delete the specified destination (cannot be used with -a).
`<username>`	Specify the user name associated with the trap destination.
`<password>`	Specify the password for the user. Passwords must not contain spaces.
`<ipaddress>`	Specify the IP address of the trap destination.

Examples:

set snmp v3trapdestination -a *user1 password 10.10.11.10*

ok,00

set snmp v3trapdestination -d *user1 password 10.10.11.10*

ok,00

Note: The <username>, <password>, and <ipaddress> arguments are required with -a and -d.

ssh

The following targets are applicable to the **ssh** object.

keys

The **keys** target sets the ssh authorized keys for the auto user. Authorized keys allow the remote user to log in to the system without a password.

Arguments:

[-i]	Install a new ssh key. The key is read from standard input and must be less than 4096 characters in length. Since only one key is currently supported, the key replaces any key presently installed.
[-r owner]	Remove an existing ssh key. Since only one key is currently supported, the owner parameter is ignored but required. This is reserved for future versions that may support multiple keys.

Note: Either **-i** or **-r** must be specified.

Examples:

Manually enter an ssh key (or paste into the terminal).

set ssh keys -i

ssh-rsa key_string user@lifesize.com

ok,00

Copy your own public key file to the auto user's authorized key file.

sh% cat *~/.ssh/id_rsa.pub* | ssh *auto@10.10.1.1* set ssh keys -i

ok,00

sh%

Remove the specified key.

set ssh keys -r *user@lifesize.com*

ok,00

Note: Currently, only one SSH authorized key is supported.

service

The **service** target enables or disables the ssh service. Already established ssh sessions will not be affected if the service is disabled.

Arguments:

<{off|on}> Disable or enable the ssh service.

Examples:

set ssh service *on*

ok,00

system

The **system** object allows setting of certain system-specific parameters such as the system name.

autoreboot

The **autoreboot** target controls whether the system will automatically reboot each evening.

Arguments:

<{on|off}> Specify ***on*** to enable the reboot feature.

Examples:

set system autoreboot *on*

ok,00

name

The **`name`** target enables you to set the system name, which shows in the user interface and on any connected phone.

Arguments:

`<value>`	Specify the name for the system

Examples:

```
set system name "LifeSize Room"

ok,00
```

number

The **`number`** target sets the voice telephone number associated with the system.

Arguments:

`<value>`	Specify the voice telephone number for the system.

Examples:

```
set system number 555-1212

ok,00
```

out-of-box

The **`out-of-box`** target runs the initial configuration process or cancels an already running initial configuration process.

Arguments:

<{enabled|complete}>	Specify enabled to rerun the initial configuration process or complete to disable an already running initial configuration process.

Examples:

```
set system out-of-box enabled

ok,00
```

video-number

The **`video-number`** target sets the video telephone number associated with the system.

Arguments:

`<value>`	Specify the new video telephone number for the system.

Examples:

```
set system video-number 555-1213

ok,00
```

telnet

The **`telnet`** object allows enabling and disabling the telnet service in real time. Active telnet sessions will be disconnected if the service is stopped without closing the sessions first.

Arguments:

`<{off|on}>`	Disable or enable the telnet service.

Examples:

```
set telnet on

ok,00
```

timer

The following targets are applicable to the **`timer`** object.

caller-id

The **`caller-id`** target controls the caller ID display fadeout timer.

Arguments:

`<{off|30|60|120|300|600|on}>`	Specify the time in seconds before the caller ID display fades out. Specifying off disables the caller ID display. Specifying on leaves the display on continuously.

Examples:

```
set timer caller-id 30

ok,00
```

fadeout

The **`fadeout`** target controls the user interface fadeout timer.

Arguments:

`<{5|10|20|30|60|120|300|600|never}>`	Specify the time in seconds before the user interface fades out during an active call. Specifying never disables the fadeout function.

Examples:

```
set timer fadeout 20

ok,00

set timer fadeout never

ok,00
```

screen-saver

The **`screen-saver`** target controls the screen saver activation timer (the amount of idle time before the screen saver activates).

Arguments:

`<{10|20|30|off}>`	Specify the time in minutes before the screen saver feature activates. Specifying off disables the screen saver function.

Examples:

```
set timer screen-saver 20

ok,00

set timer screen-saver off

ok,00
```

sleep

The **`sleep`** target retrieves the current setting for the sleep timer (the amount of idle time after the screen saver activates and before the system enters sleep state).

Arguments:

`<{10|20|30}>`	Specify the time in minutes before the system enters sleep state.

Examples:

```
set timer sleep 20

ok,00
```

user

The **`user`** object enables configuration of user functions in the user interface.

password

The **`password`** target enables you to set the password for access to the user preferences in the user interface.

Arguments:

`<value>`	The new user password. The password must contain only the numbers 0-9 and/or the symbols * and #. The length can be 0 to 16 characters. If more than 16 characters are specified, the password is silently truncated.

Examples:

```
set user password 12345*#

ok,00

set user password -V abcdef

error  04  Invalid Parameter
```

verbose-mode

The **`verbose-mode`** target controls the verbose mode setting. Verbose mode provides human readable output. Enabling verbose mode is equivalent to specifying -V with each command entered.

Arguments:

`<{on|off}>`	Enable or disable verbose mode output.

Examples:

```
set verbose-mode on

ok,00

set verbose-mode off

ok,00
```

video

The following targets are applicable to the **`video`** object.

input-names

The **`input-names`** target specifies the user friendly names of the various video inputs.

Arguments:

`<{hd0|hd1|sd0|sd1|vga0}>`	Specify the input to name. LifeSize Team does not support naming the hdl and sdl as these inputs are not available for this product.
`<value>`	The new name of the input. Enclose strings with spaces inside single or double quote characters.

Examples:

```
set video input-names hd0 "HD Camera"

ok,00

set video input-names vga0 PC

ok,00
```

pip-mode

The **`pip-mode`** target controls the default operation of the PIP feature during an active call. In auto mode, the window is visible only when the interface is visible and follows the fadeout timer settings. When on, the window is always visible; when off, it is never visible. This command cannot be used to change the PIP window state while a call is active, only while not in a call. To change the PIP window state while a call is active, use the **`pip-window`** command.

Arguments:

`<{on|off|auto}>`	Specify how the PIP window operates.

Examples:

```
set video pip-mode auto

ok,00
```

`set video pip-mode off`

`ok,00`

pip-window

The **`pip-window`** target controls the PIP display during a call. Turning the window on or off also changes the **`pip-mode`** setting to match the **`pip-window`** setting. This command shows the interface on the screen. If you set the PIP window to off, the window disappears when the interface fades out. If the interface fadeout timer is set to never, you cannot turn off the PIP window (since the interface never fades out). Turning the window on takes effect immediately.

Arguments:

`<{on|off}>` Turn the PIP window on or off.

Examples:

`set video pip-window on`

`ok,00`

`set video pip-window off`

`ok,00`

primary-input

The **`primary-input`** target controls what video source is associated with the primary input.

Arguments:

`<{hd0|hd1|sd0|sd1|vga0}>` Specify the new source for the primary video input.

Examples:

`set video primary-input hd0`

`ok,00`

`set video primary-input vga0`

`ok,00`

secondary-input

The **`secondary-input`** target controls what video source is associated with the secondary input

Arguments:

`<{hd0|hd1|sd0|sd1|vga0}>`	Specify the new source for the secondary video input. LifeSize Team does not support configuring the hdl and sdl as these inputs are not available for this product.

Examples:

```
set video secondary-input hd1

ok,00
```

volume

The following targets are applicable to the **`volume`** object.

dtmf

The **`dtmf`** target controls the volume setting for Dual Tone Multi Frequency (DTMF) tones when dialing (using a scale of 0 to 10).

Arguments:

`<{0..10}>`	Specify the volume level (0 = off, 10 = max) for DTMF tone generation.

Examples:

```
set volume dtmf 5

ok,00
```

ring-tone

The **`ring-tone`** target controls the volume setting for the ring tone.

Arguments:

`<{0..10}>`	Specify the volume level (0 = off, 10 = max) for ring tone generation.

Examples:

```
set volume ring-tone 5

ok,00
```

speaker

The **`speaker`** target controls the volume of the system speaker (audio loudness).

Arguments:

`<{0..100}>` Specify the volume level (0 = off, 100 = max) for system audio.

Examples:

```
set volume speaker 60
ok,00
```

status Verb

The following objects and targets are applicable to the **`status`** verb.

call

The following targets are applicable to the **`status call`** object.

active

The **`active`** target shows the status of all active calls in the system.

Arguments:

`[-c conference]`	Restrict output to the specified conference ID.
`[-C call]`	Restrict output to the specified call ID.
`[-d incoming|outgoing]`	Restrict output to the specified call direction.
`[-t audio|video]`	Restrict output to the specified call type.

Examples:

```
status call active
18,5,Ringback,Yes,Video,10.10.11.155,Jones
3,1,Connected,No,Audio,5551212,
4,2,Ringing,No,Video,10.10.11.110,LifeSize

ok,00
```

```
status call active –V –c 1
Call   Conf   State     Incoming   Type    Number        Name
11     1      Ringing   No         Video
12     1      Ringback  Yes        Video   10.10.11.155  Jones

ok
```

```
status call active -d incoming -t video
11,1,Connected,Yes,Video,10.10.11.155,Jones
12,1,Connected,Yes,Video,10.10.11.116,LifeSize

ok,00
```

Note: Valid values for the `State` field are *`Dialing`*, *`Ringing`*, *`Connected`*, *`Terminating`*, and *`Ringback`*. Valid values for the `Type` field are *`Video`*, *`Audio`*, and *`Unknown`*.

In software release v3.0, a LifeSize system that is a far end participant in a multiway call with a LifeSize system that is the MCU lists the virtual parties (the non-directly connected parties) of the call with *V* as the call ID. It is not possible to restrict output to only virtual parties.

history

The **`history`** target shows historical information on completed calls. Active calls are not shown. The information consists of the following fields:

Field Name	Description	Display Mode[a]
`ID`	The call identifier—a monotonically incrementing index for the call **Note**: This is not the call handle used during an active call.	All
`Conf`	The conference identifier— a monotonically incrementing index used to distinguish conference participants **Note**: This is not the conference handle used during an active conference.	All
`Local Name`	The name of the local system (the system providing the call history)	All
`Local Number`	The number of the local system	All
`Remote Name`	The name of the remote system (the other participant in the call)	All
`Remote Number`	The number of the remote system	All
`Dialed Digits`	The digits used to place the call	All
`Start Time`	The time in ISO date format at which the call started	All
`End Time`	The time in ISO date format at which the call ended	Full
`Duration`	The length of the call in hours:minutes:seconds	All
`Direction`	Indication of incoming or outgoing call	All
`Protocol`	The communications protocol used for the call	Full
`Security`	The security protocol used for the call	Full

Field Name	Description	Display Mode[a]
`Req Kibps`	Requested bit rate for the call	Full
`Act Kibps`	Actual bit rate for the call	Full
`TX Vid`	Transmit video codec used	Full
`TX Aud`	Transmit audio codec used	Full
`TX Res`	Transmit resolution used	Full
`RX Vid`	Received video codec used	Full
`RX Aud`	Received audio codec used	Full
`RX Res`	Received resolution	Full
`TX Pres`	Transmit presentation status—whether or not a presentation was transmitted	Full
`RX Pres`	Receive presentation status—whether or not a presentation was received	Full
`Pres Fmt`	Presentation format—the protocol used for the secondary video stream	Full
`Term Code`	Call termination code	Full

a. The `All` display mode indicates that the field always appears in the output. The `Full` display mode indicates that the field appears in the output only when you specify the `-f` option with the command. By default, only a limited set of statistics appear.

Arguments:

-U Show times as UTC instead of local time.

-f Enable full display mode showing all available statistics.

Examples:

`status call history`

```
39,10,lifesize,10.10.11.209,unknown,9710,1234567,2007-07-09 17:13:32,01:02:56,In
38,10,lifesize,10.10.11.209,unknown,1310,8901234,2007-07-09 16:14:03,00:00:00,In
37,9,lifesize,10.10.11.209,unknown,9710,5678901,2007-07-09 15:13:13,00:16:45,In

ok,00
```

Note: The output for the following example is split by column into two sections for visual clarity. The actual output is a single line for each call.

status call history -V

```
ID   Conf  Local Name  Local Number   Remote Name  Remote Number  Dialed Digits
39   10    lifesize    10.10.11.209   unknown      9710           1234567
38   10    lifesize    10.10.11.209   unknown      1310           8901234
37   9     lifesize    10.10.11.209   unknown      9710           5678901


Start Time          Duration   Direction
2007-07-09 17:13:32 01:02:56   In
2007-07-09 16:14:03 00:00:00   In
2007-07-09 15:13:13 00:16:45   In

ok
```

status call history -f

```
39,10,lifesize,10.10.11.209,unknown,9710,1234567,
   2007-07-09 17:13:32,2007-07-09,18:16:28,01:02:56,In,H.323,
   None,512,448,H.264,G.711Ulaw,HD,H.264,G.711Ulaw,HD,No,No,
   None,Normal

38,10,lifesize,10.10.11.209,unknown,1310,8901234,
   2007-07-09 16:14:03,2007-07-09 16:14:03,00:00:00,In,H.323,
   None,1152,0,,,,,,No,No,None,Normal

37,9,lifesize,10.10.11.209,unknown,9710,5678901,
   2007-07-09 15:13:13,2007-07-09 15:29:58,00:16:45,In,H.323,
   None,512,448,H.264,G.711Ulaw,HD,H.264,G.711Ulaw,HD,No,No,
   None,Normal

ok,00
```

The output for the following example is split by column into four sections for visual clarity. The actual output is a single line for each call.

status call history -V -f

```
ID  Conf  Local Name  Local Number   Remote Name  Remote Number   Dialed Digits
39  10    lifesize    10.10.11.209   unknown      9710            1234567
38  10    lifesize    10.10.11.209   unknown      1310            8901234
37  9     lifesize    10.10.11.209   unknown      9710            5678901


Start Time           End Time             Duration  Direction  Protocol  Security
2007-07-09 17:13:32  2007-07-09 18:16:28  01:02:56  In         H.323     None
2007-07-09 16:14:03  2007-07-09 16:14:03  00:00:00  In         H.323     None
2007-07-09 15:13:13  2007-07-09 15:29:58  00:16:45  In         H.323     None


Req Kibps  Act Kibps  TX Vid   TX Aud      TX Res   RX Vid     RX Aud      RX Res
512        448        H.264    G.711Ulaw   HD       H.264      G.711Ulaw   HD
1152       0
512        448        H.264    G.711Ulaw   HD       H.264      G.711Ulaw   HD


TX Pres   RX Pres  Pres Fmt   Term Code
No        No       None       Normal
No        No       None       Normal
No        No       None       Normal


ok
```

statistics

The **statistics** target shows bandwidth and codec statistics for active calls or a specific active call.

Note: The user interface shows statistics for the virtual parties in a virtual multiway call; autosh shows only statistics for the actual link.

Following are the complete set of fields that appear.

Field Name	Description
`ID`	The call handle
`ARX Codec`	Audio Receive Codec - shows the audio codec used by the remote transmitter.
`Kibps`	Kilo Bits per second - shows the bit rate divided by 1024 for the preceding column's codec.
`ATX Codec`	Audio Transmit Codec - shows the audio codec used by the local transmitter.
`VRX Codec`	Video Receive Codec - shows the video codec used by the remote transmitter.
`VTX Codec`	Video Transmit Codec - shows the video codec used by the local transmitter.
`ARX Jitter`	Audio Receive Jitter - shows the packet jitter from the remote audio transmission.
`ARX Pktps`	Audio Receive Packets per second - shows the received audio packet rate which is dependent on the bit rate and codec used.
`ARX Pkt Loss`	Audio Receive Packet loss - shows the instantaneous number of audio packets transmitted by the remote side that were never received (or received too late) at the local side.
`ARX Cumu Loss`	Audio Receive cumulative packet loss - shows the total number of remote transmitted audio packets that were lost.
`ARX % Loss`	Audio Receive percentage packet loss - shows the percent of the total remote transmitted audio packets that were lost.
`ATX Jitter`	Audio Transmit Jitter - shows the packet jitter from the local audio transmission.
`ATX Pktps`	Audio Transmit Packets per second - shows the transmitted audio packet rate which is dependent on the bit rate and codec used.
`ATX Pkt Loss`	Audio Transmit Packet loss - shows the instantaneous number of audio packets transmitted by the local side that were never received (or received too late) at the remote side.
`ATX Cumu Loss`	Audio Transmit cumulative packet loss - shows the total number of locally transmitted audio packets that were lost.
`ATX % Loss`	Audio Transmit percentage packet loss - shows the percent of the total locally transmitted audio packets that were lost.

Field Name	Description
`VRX Jitter`	Video Receive Jitter - shows the packet jitter from the remote video transmission.
`VRX Pkt Loss`	Video Receive Packet loss - shows the instantaneous number of video packets transmitted by the remote side that were never received (or received too late) at the local side.
`VRX Cumu Loss`	Video Receive cumulative packet loss - shows the total number of remote transmitted video packets that were lost.
`VRX % Loss`	Video Receive percentage packet loss - shows the percent of the total remote transmitted video packets that were lost.
`VRX Fps`	Video Receive Frames per second - shows the frame rate of the received video.
`VRX Res`	Video Receive Resolution - shows the resolution (width by height) of the received video.
`VTX Jitter`	Video Transmit Jitter - shows the packet jitter from the local video transmission.
`VTX Pkt Loss`	Video Transmit Packet loss - shows the instantaneous number of video packets transmitted by the local side that were never received (or received too late) at the remote side.
`VTX Cumu Loss`	Video Transmit cumulative packet loss - shows the total number of local transmitted video packets that were lost.
`VTX % Loss`	Video Transmit percentage packet loss - shows the percent of the total number of local transmitted video packets that were lost.
`VTX Fps`	Video Transmit Frames per second - shows the frame rate of the transmitted video.
`VTX Res`	Video Transmit Resolution - shows the resolution (width by height) of the transmitted video.

Arguments:

`[-C callHandle]` Specify that statistics for a specific call handle are desired. This argument cannot be used with **`-a`**.

`[-a]` The output produced by specifying the **`-a`** argument with this target for recent calls statistics is deprecated. The target accepts the **`-a`** argument, but ignores it. Specifying this argument produces statistics only for active calls.

Examples:

status call statistics

```
1,G722,78.1,AAC_LC,94.0,H264,924.6,H264,893.9,19,50,0,0,
   0.000000,19,50,0,0,0.000000,7,0,0,0.000000,30,
   1280 720,7,0,0,0.000000,30,1280 720
4,G711ULAW,62.4,G711ULAW,62.4,H264,1050.2,H264,1050.2,17,50,0,0,
   0.000000,17,50,0,4,0.000000,9,0,0,0.000000,30,
   1280 720,9,0,0,0.000000,30,1280 720

ok,00
```

The output for the following example is split by column into five sections for visual clarity. The actual output is a single line for each call.

status call statistics –v

```
ID  ARX Codec   Kibps   ATX Codec  Kibps   VRX Codec  Kibps   VTX Codec
1   G722        78.1    AAC_LC     94.0    H264       924.6   H264
4   G711ULAW    62.4    G711ULAW   62.4    H264       1050.2  H264


Kibps   ARX Jitter    ARX Pktps   ARX Pkt Loss  ARX Cumu Loss   ARX % Loss
893.9   19            50          0             0               0.000000
1050.4  17            50          0             0               0.000000


ATX Jitter  ATX Pktps ATX Pkt Loss  ATX Cumu Loss ATX % Loss  VRX Jitter
19          50        0             4             0.000000    7
17          50        0             4             0.000000    9


VRX Pkt Loss  VRX Cumu Loss VRX % Loss   VRX Fps   VRX Res   VTX Jitter
0             0             0.000000     30        1280 720  7
0             0             0.000000     30        1280 720  9


VTX Pkt LossVTX Cumu Loss VTX % Loss   VTX Fps  VTX Res
0           0             0.000000     30       1280 720
0           0             0.000000     30       1280 720


ok
```

```
status call statistics -C 1
1,G722,62.4,AAC_LC,93.7,H264,924.6,H264,894.8,19,50,0,0,
   0.000000,19,50,0,0,0.000000,7,0,0,0.000000,30,
   1280 720,7,0,0,0.000000,30,1280 720

ok,00
```

Presentation targets

The **`presentation`** object shows information about presentations.

statistics

The **`statistics`** target shows information about active presentations, either received or transmitted.

Arguments:

`[-c conference]` Restrict output to the specified conference ID.

Examples:

```
status presentation statistics
1,true,rx,dec2,H264,145.6,1280 720

ok,00

status presentation statistics -c 1 -V
Conf        Enabled     Type        Device      Codec       Kibps       Resolution
1           true        tx          sd0         H264        144.7       704 480


ok
```

control Verb

The following objects and targets are applicable to the **`control`** verb.

Asynchronous Messages

During normal operation, the system may receive asynchronous messages relating to call status changes, presentation status changes, or incoming call notifications. These messages are printed after a command completes between the ok or error message and the shell prompt, for example:

```
...
ok,00
CS,3,1,Connected,Video,Normal,10.10.11.10,LifeSize
<prompt>
```

In addition, if the shell detects that no input has been received since the prompt was printed, it may spontaneously print an asynchronous message by emulating the user having pressed return. In this way, asynchronous messages are delivered in a timely fashion while still guaranteeing that the messages do not interfere with processing the current command being executed.

Because asynchronous messages may be received at any time and to preserve the order in which messages arrive, some commands do not produce any synchronous output and instead produce only asynchronous output. Execution of these commands generally causes asynchronous messages (for example, placing a call, starting a presentation). Commands that operate in this fashion are indicated as doing so in the description of the command.

Call Status Messages

While a call is active, or as a response to the **`control call`**, **`add-part`**, **`answer`**, **`del-part, dial`** and **`hangup`** commands, the CLI produces status messages about the call. These messages use the Call Status (CS) format. For example:

```
...
ok,00
CS,3,1,Connected,Video,Normal,10.10.11.10,LifeSize
<prompt>
```

The meaning of the columns is as follows:

Col #	Meaning	Values	Description
1	Prefix	CS	CS indicates that this asynchronous event is a call status update.
2	Call ID	<number>	Indicates the number of the call.
3	Conference ID	<number>	Indicates the number of the conference managing this call.
4	State	On Hook Terminated Off Hook Valid Number Dialing Proceeding Ringing Answered Number Answered Consult Connected Ring Incoming Caller ID Local Ring Back Off Remote Pres Begin Remote Pres End Remote Pres Failed Far End Mute Far End Unmute Far End Hold Far End Resume	Phone is on hook. Call is terminated (but may still be off hook). Phone is off hook. Dialed number is valid. Dialing is proceeding. Call is proceeding. Call is ringing. Answered number information. When a call is answered in consult mode (private from main call). Call is connected. Incoming call received. Caller ID information. Local ringback is off. A remote presentation is beginning. A remote presentation is ending. The remote presentation has failed. The far end has muted the microphone. The far end has unmuted the microphone. The far end has placed the call on hold. The far end has resumed the call.
5	Type	Audio Video Unknown	The message pertains to an audio call. The message pertains to a video call. The message pertains to either type of call.

Col #	Meaning	Values	Description
6	Disconnect Reason	Normal Unknown Busy No Answer Bad Number Comm Failure Unreachable Rejected Max Calls Parse Error Enc Not Sup No Bandwidth Unreachable GK GK Resources GW Resources Invalid Addr Not Registered SIP 400 SIP 403 SIP 404 SIP 415 SIP 416 SIP 480 SIP 500 SIP 502 SIP 513 SIP 603 SIP 606	Normal disconnection. Unknown reason for disconnection. Remote end is busy. Remote end did not answer. Invalid number dialed. Communications failure. Remote end is unreachable. Remote end rejected the call. Simultaneous call limit reached. Parse error in called address. Encoder not supported. No bandwidth available for call. Gatekeeper is unreachable. Gatekeeper out of resources for call. Gateway out of resources for call. Invalid called address. Called address not registered. SIP Bad request. SIP Disallowed. SIP Remote party not in a domain. SIP Mismatched codec. SIP Unsupported address. SIP User temporarily unavailable. SIP Server error. SIP Bad gateway. SIP Server failed - request too large. SIP User declined call. SIP Service not acceptable.
7	Number	<IP or PSTN #>	The phone number of the remote side of the call.
8	Name	<string>	The assigned name of the remote system.

Incoming Call Messages

When an incoming call is received, a status message about the call is printed. These messages use mostly the same format as the call status messages previously described, but are prefixed with “IC” (incoming call) instead of “CS” and do not contain the disconnect reason field (row 6 in the previous example), for example:

```
...
ok,00
IC,16,1,Ringback,Video,10.10.11.155,Sunshine
<prompt>
```

Once an incoming call notice has been generated, further notices about that specific call are relayed through call status (“CS”) messages.

Presentation Status Messages

While a presentation is active, or as a response to the **`control call presentation`** command, the CLI produces status messages about the presentation. These messages use the PS (presentation status) format, for example:

```
...
ok,00
PS,15,1,Terminated,No,Rejected
<prompt>
```

The output columns for this command are as follows:

Col #	Meaning	Values	Description
1	Prefix	PS	PS indicates that this asynchronous event is a presentation status event.
2	Presentation ID	<number>	Indicates the number of the presentation.
3	Conference ID	<number>	Indicates the number of the conference running the presentation.
4	State	Initiated Terminated Relinquished	The presentation has started. The presentation has ended. The local presentation has been superseded by a remote one.
5	Remote	Yes No	The presentation message concerns a remote presentation. The presentation message concerns a local presentation.

Col #	Meaning	Values	Description
6	Disconnect Reason	None Normal Rejected Unknown	No disconnect has occurred. Normal disconnect occurred (phone on hook). The presentation was rejected. Unknown disconnect occurred.

Far Camera Control Messages

During an active call you may receive control messages for the local camera. The system normally handles these messages internally, but in the event that an external pan-tilt-zoom camera is being used, external control software can use these messages to determine what actions to take with that camera. These messages use the FC (far camera) format. For example:

```
...
ok,00
FC,1,Near,0,Move,Pan Left      # Begin moving camera left.
...
ok,00
FC,1,Near,0,Move,Continue      # Continue current camera motion.
...
ok,00
FC,1,Near,0,Stop,None          # Stop camera movement.
...
ok,00
FC,1,Near,sd0,None,None        # Change camera source to sd0.
```

Following are the output columns for this format:

Col #	Meaning	Values	Description
1	Prefix	FC	FC indicates this asynchronous event is a far camera control message.
2	Call ID	<number>	The call ID associated with this message.
3	Where	Near Far None	The local camera is the target of the operation. The far camera is the target of the operation. The message does not pertain to a camera.

Col #	Meaning	Values	Description
4	Target	0 hd0 hd1 sd0 sd1 vga0	The currently selected camera is the target. The primary high definition camera is the target. The secondary high definition camera is the target. The document camera is the target. The DVD/VCR input is the target. The VGA input is the target.
5	Operation	Move Stop None	The camera should begin or continue motion. The camera should stop all motion immediately. The message is not a motion control message.
6	Movement	Pan Left Pan Right Tilt Up Tilt Down Zoom In Zoom Out Focus In Focus Out None	Pan the camera to the left. Pan the camera to the right. Tilt the camera up. Tilt the camera down. Zoom the camera in (telephoto). Zoom the camera out (widen). Focus the camera in. Focus the camera out. Not a camera motion operation.

The movement messages generally do not indicate the camera being operated, so any control software must use **`get video primary-input`** to determine the camera to control if more than one PTZ camera is connected to the system. Monitoring for source change messages is insufficient, because the local user can change the camera source without causing a message to be generated.

Mute Status Messages

During normal call operation, the state of the remote side mute function is available through the call status messages. The local mute status is available through the Mute Status message. These messages use the `MS` prefix and are in direct response to the user pressing the local mute button on either the phone or the remote. The mute status messages use the following format:

```
...
ok,00
MS,true
...
ok,00
MS,false
```

When the second column is *true*, the local side is muted. When the second column is *false*, the local side is not muted. When an outgoing call is placed, the local mute status is false. When an incoming call is placed, the state of the auto-mute (see **get call auto-mute**) controls the initial state. The current status is available through the **get audio mute** command.

Video Capabilities Messages

The remote side of a call may support sending more than one video source. The video capabilities message provides a means to determine which sources are supported and what capabilities they provide. This message generally appears after a call is connected, but may also appear mid call if the remote codec supports hot-plugging of video sources. Video capabilities messages use the following format:

```
...
ok,00
VC,12,2,hd0,PTZF,sd0,---
...
ok,00
VC,12,1,hd0,PTZF
```

The output columns are as follows:

Col #	Meaning	Values	Description
1	Prefix	VC	VC indicates that this asynchronous event is a video capabilities message.
2	Call ID	<*number*>	The call ID associated with this message.
3	Count	<*number*>	The number of video sources available for this call. Each video source indicated in this count has two additional columns.
4, 6, ...	Source	hd0 hd1 sd0 sd1 vga0	Indicates that the primary Hi Def camera is available Indicates that the secondary Hi Def camera is available Indicates that the document camera is available Indicates that the DVD/VCR input is available Indicates that the VGA input is available
5, 7, ...	Capabilities	PTZF or ---- or a combination	Each character is a flag indicating a capability supported. P indicates support for panning, T indicates support for tilting, Z indicates support for zooming and F indicates support for focusing. A dash (-) indicates that the corresponding capability is not present.

Only the supported sources are reported. If a source is not listed, video is not available from that source.

call

The following targets are applicable to the **`call`** object.

For those targets that take a called address, the address may be specified as an IP address, a PSTN phone number, a URI, or a directory specification string. Following are the directory specification strings:

Form	Description
redial:<n>	Dial the indicated entry from the redial list. Entry 1 is at the top of the list, entry 2 is the entry immediately following it. The ordering of the redial list changes as calls are placed and received. For automation use, do not use this form unless the intent is to redial the last call.
redial:<string>	Dial the indicated entry from the redial list. The string is used as a case insensitive prefix to match the name stored in the redial list (the name that shows in the user interface). For example, the prefix “sun” matches the names “sunrise” and “SUNSET”, but not “summer” or “fun-in-the-sun”. For automation use, the prefix should completely specify the intended entry and that entry should be locked in the redial list.
local:<string>	Dial the indicated entry from the local directory. The string is used as a case insensitive prefix to match against the name stored in the local directory. For automation use, the string should completely specify the desired entry.

add-part

The **`add-part`** target adds a new participant to an existing conference call. This command produces only asynchronous messages as a response to ensure proper ordering of displayed call status.

Arguments:

`[-b {auto|128|192|256|320|384|512|768|1024|1152|1472|1728|1920|2500|3000|4000|5000}]`	Specify the maximum bandwidth to use for the call. Use auto to use the configured maximum bandwidth. The default is “auto”. On LifeSize Team, speeds greater than 2500 kb/s are not available. On LifeSize Express, speeds greater than 1472 kb/s are not available.
`[-p {auto|h323|h323gw|isdn|pstn|sip}]`	Specify the protocol to use to connect the new party to the call. The default is auto.
`[-t {audio|video}]`	Specify to add the participant as an audio or video call. The default is video.

`<confHandle>`	Specify the handle to the conference to which to add the participant.
`<number>`	Specify the phone number, IP address, or URI of the party to add.

Examples:

```
set prompt "% "

ok,00
% control call add-part 1 -p pstn -t audio 555-1212

ok,00
CS,3,1,Ringing,Video,Normal,10.10.11.10,LifeSize
% control call add-part -V 2 -p h323 10.10.11.11 -b 1024

ok
CS,5,2,Ringing,Video,Normal,10.10.11.10,LifeSize
% control call add-part 1 redial:3# Add 3rd redial entry as call

ok,00
CS,3,1,Ringing,Audio,Normal,1-512-555-1212,
%
```

Refer to "Asynchronous Messages" on page 106 for a description of the response.

answer

The **`answer`** target answers or rejects an incoming call. This command produces only asynchronous messages as a response to ensure proper ordering of displayed call status.

Arguments:

`[-r]`	Specify whether to reject the call. The default is to answer. Cannot be used with **`-t`**.
`[-t {audio|video}]`	Specify whether to answer the call as an audio or video call. Cannot be used with **`-r`**.
`<callHandle>`	Specify the handle of the incoming call to answer.

Examples:

```
set prompt "% "

ok,00
```

```
% control call answer 1 -t audio

ok,00
CS,1,2,Connected,Audio,Normal,10.10.11.10,LifeSize
% control call answer 2 -r -V

ok
CS,1,2,Terminated,Video,Rejected,10.10.11.10,LifeSize
%
```

Refer to “Asynchronous Messages” on page 106 for a description of the response.

del-part

The **`del-part`** target drops a participant from an existing conference call. This command produces only asynchronous messages as a response to ensure proper ordering of displayed call status.

Arguments:

`<callHandle>` Specify the handle to the call on which to drop the participant.

Examples:

```
set prompt “% “

ok,00
% control call del-part 3

ok,00
CS,3,2,Terminated,Video,Normal,10.10.11.10,LifeSize
% control call del-part -V 2

ok
CS,2,2,Terminated,Video,Normal,10.10.11.10,LifeSize
%
```

Refer to “Asynchronous Messages” on page 106 for a description of the response.

dial

The **`dial`** target initiates a new call. This command produces only asynchronous messages as a response to ensure proper ordering of displayed call status.

Arguments:

`[-b {auto|128|256|384|512|768|1024|1152|1472|1728|1920|2500|3000|4000|5000}]`	Specify the maximum bandwidth to use for the call. Use auto to use the configured maximum bandwidth. The default is auto. On LifeSize Team, speeds greater than 2500 kb/s are not available. On LifeSize Express, speeds greater than 1472 kb/s are not available.
`[-p {auto|h323|h323gw|isdn|pstn|sip}]`	Specify the protocol to use to connect the new party to the call. The default is auto.
`[-t {audio|video}]`	Specify to add the participant as an audio or video call. The default is video.
`<number>`	Specify the phone number, IP address, or URI of the party to dial.

Examples:

```
set prompt "% "

ok,00
% control call dial 10.10.11.155

ok,00
CS,1,2,Ringing,Video,Normal,10.10.11.10,LifeSize
% control call dial -V -p h323 -t video 10.10.11.11 -b 1024

ok
CS,1,2,Ringing,Video,Normal,10.10.11.10,LifeSize
% control call dial "local:john doe"    # Dial using local directory entry

ok,00
CS,3,1,Ringing,Video,Normal,192.168.168.203,John Doe
%
```

Refer to “Asynchronous Messages” on page 106 for a description of the response.

display

The **`display`** target controls what remote source is shown on the local display.

Arguments:

`[-s {main|aux|doc|auxdoc|playback}]`	Specify the far camera source, default is main.
`<callhandle>`	Specify the call whose camera source is to be changed.

Examples:

`control call display` ***`4`*** # Switch to the main video source

`ok,00`

`control call display -s` ***`doc 4`*** # Switch to the document camera source

`ok,00`

dtmf

The **`dtmf`** target allows sending DTMF tones inband in an active call (for example, to access remote menu systems).

Arguments

`<callHandle>`	Specify the handle of the call to which to send digits.
`<{0-9|A-D|a-d|*|#}>`	Specify the digits to dial. the digits may be strung together (for example, 5551212*#).

Examples:

`control call dtmf` ***`1 123456789abcd*#ABCD`***

`ok,00`

hangup

The **hangup** target disconnects from either a conference (multi-way call) or a single-way call. This command produces only asynchronous messages as a response to ensure proper ordering of displayed call status.

Arguments:

`[-a]`	Specify that all active calls be terminated (cannot be used with **`-c`** and `<handle>`)
`[-c]`	Terminate a conference. The default is a single call.
`<handle>`	Specify the call or conference handle to disconnect.

Examples:

```
set prompt “% “

ok,00
% control call hangup 1

ok,00
CS,1,2,Terminated,Video,Normal,10.10.11.10,LifeSize
% control call hangup -c 2 -V

ok
CS,1,2,Terminated,Audio,Normal,10.10.11.10,LifeSize
%
%control call hangup -a

ok,00
CS,1,2,Terminated,Video,Normal,10.10.11.10,LifeSize
```

Refer to “Asynchronous Messages” on page 106 for a description of the response.

> ***Note:*** The **hangup** command may occasionally report an error even though the call was actually hung up. To ensure the correct response for this command, refer to the asynchronous output that follows the command to determine the actual state of the command.

hook

The **hook** target enables control of the hook status of a PSTN call. The following operations are supported:

- **`off`**—takes the phone off hook in preparation to place a call
- **`flash`**—places the phone on hook for a short period and then takes it back off hook to allow access to features such as call waiting
- **on**—places the phone back on hook to hang up a call

Arguments:

`[-t {h323|h323gw|isdn|pstn|sip}]`	Specify the type of call on which to perform the hook operation. The default is pstn. The **`-t`** argument is valid only with **`off`** and **`flash`** operations.
`<flash|off|on>`	Specify the operation to perform.

`[<callHandle>]`	Specify the call on which to operate. The `[<callHandle>]` argument is valid only for the **on** operation.

Examples:

```
set prompt "%"

ok,00
% control call hook off

ok,00
CS,6,1,Dialing,Unknown,Normal,,
CS,6,1,Answered Number,Unknown,Normal,,
CS,6,1,Connected,Audio,Normal,,
% control call dial dtmf 6 5551212

ok,00
% control call hook flash

ok,00
% control call hook on 6

ok,00
CS,6,1,Terminated,Audio,Normal,,
%
```

presentation

The **`presentation`** target allows starting and stopping a presentation. All responses to this command are produced as asynchronous responses due to the interaction of local and remote presentations causing potential ordering issues with the output.

Arguments:

`[-t {slides}]`	Specify the source of the presentation. Slides indicates a PC based presentation connected through the vga0 input. The default is ***slides***. Specifying this argument is optional.
`<confHandle>`	Specify the conference to which to provide the presentation. In all cases the value is ***1***.
`<{start|stop}>`	Specify whether to start or stop the presentation.

Examples:

```
control call presentation 1 start
```

```
ok,00
PS,15,1,Initiated,No,None

ok,00
control call presentation 1 stop -V

ok
PS,15,1,Terminated,No,Rejected

ok,00
```

Refer to “Asynchronous Messages” on page 106 for a description of the response.

reboot

The **`reboot`** target causes the system to reboot.

Arguments:

`[seconds]` Delay the reboot for the indicated number of seconds.

Examples:

```
control reboot 60

ok,00
```

Note: Session terminated after 60 seconds.

remote

The **`remote`** target emulates the remote control by sending sequences of commands that replicate the functionality of the remote control.

Arguments:

`<call|tri|squ|cir|back|ok|left|right|up|down|vup|vdn|mute|zin|zout|near|far|1|2|3|4|5|6|7|8|9|0|*|#|home|dir>`	Specify the specific button to press. You can specify as many buttons on the command line as desired. Buttons are processed in the sequence given on the command line.

Note: The **`home`** and **`dir`** arguments do not correspond to any keys that are physically present on the remote, but serve as an aid to creating deterministic automated remote control sequences. The **`home`** argument goes to the main screen of the user interface. The **`dir`** argument goes to the directory screen from any other screen within the user interface.

Examples:

```
control remote left left ok zin 1 2 3 4 5 #

ok,00
```

reset

The **`reset`** target configures the system to return to default configuration settings after the next system reboot.

Note: The reset target does not reboot the system. To return the system to default configuration settings after entering **`control reset`**, enter **`control reboot`**.

Arguments:

None

Examples:

```
control reset

ok,00
```

sleep

The **`sleep`** target puts the system into sleep mode.

Arguments:

None

Examples:

```
control sleep
```

```
ok,00
```

wakeup

The **`wakeup`** target wakes the system up from the sleep state if it was previously sleeping. It has no effect if the system is already awake.

Arguments:

None

Examples:

```
control wakeup
```

```
ok,00
```

exit Verb

The **`exit`** verb terminates the shell immediately.

Arguments:

None

Examples:

```
exit
```

```
ok,00
```

Note: Session terminated.

help Verb

The **`help`** verb does not have any traditional targets. Its availability is controlled by the state of the help mode flag. If help mode is enabled, entering help produces a list of verbs that can be used with a short description of how to obtain additional help. Additionally, the argument **`errors`** returns a list of error message codes and their meanings. In normal mode, entering help produces an unsupported verb error.

Arguments:

None

Examples:

```
help

error,04

set help-mode on

ok,00

help
Possible verbs:
control
get
history
set
status

ok,00

help set
Possible completions:
set admin password
set audio line-in
set audio mics
set audio video-output
set call auto-answer
set volume dtmf
set volume ring-tone

ok,00
```

Note: The previous example shows only a subset of all possible completions.

```
help errors -V
Code    Description
00      Success
01      No Memory
02      File Error
03      Invalid Instance
04      Invalid Parameter
05      Argument is not repeatable
06      Invalid Selection Parameter Value
07      Missing Argument
08      Extra Arguments on Command Line
09      Invalid Command
0a      Ambiguous Command
0b      Conflicting Parameter
0c      Operational Error
0d      No Data Available
0e      Not In Call
0f      Interrupted
10      Ambiguous Selection
11      No Matching Entries
12      Not Supported

ok,00
```

Standard Return Codes

All CLI commands return a standard error code on completion. You can access the following table of return codes using the **`help errors`** command.

Return Code		Mnemonic	Description
Dec	**Hex**		
0	00	Ok	The command completed successfully.
1	01	NoMemory	The command failed due to a loss of memory.
2	02	IOError	The command failed due to a file read/write/open error.
3	03	InvalidInstance	The command failed due to data corruption.
4	04	InvalidParameter	An incorrect option or argument was specified on the command line.
5	05	Repeated	A non-repeatable option or argument was repeated.
6	06	NotInList	The specified option or argument value was not in the selection list.
7	07	Missing	A required option or argument was not specified.
8	08	TooMany	Too many arguments were specified.
9	09	InvalidCommand	The command entered was not found.
10	0a	AmbiguousCommand	The command entered is ambiguous.
11	0b	ParameterConflict	Two or more mutually exclusive options were specified.
12	0c	OperationalError	The command failed for unspecified reasons.
13	0d	NoData	No data is available for this operation (no active calls).
14	0e	NotInCall	The command requires an active call for operation.
15	0f	Interrupted	The command was interrupted.
16	10	Ambiguous	The directory selection is ambiguous (matches multiple entries).
17	11	NoMatch	The directory selection does not match any entries.
18	12	NotSupported	The far end of the call does not support presentations.