LifeSize® Automation Command Line Interface

For LifeSize Phone Software Release v3.6

June 2008

Copyright Notice

©2008 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® is the trademark of LifeSize Communications Inc. All other trademarks are the property 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 Phone. The CLI allows you to:

- Retrieve configuration information about your LifeSize Phone. For example, you can retrieve information about the system software version or the serial numbers of the CPU and system boards.
- Apply new preferences to the configuration. For example, you can set the volume of the status tones played by the phone.
- Show the status of calls in the system. For example, you can show active calls or statistics for previous calls.
- Control aspects of the device. For example, you can add participants to an active call or reboot the device.

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 date`** Variable input appears in ***`Bold Italic`***, for example: **`set audio mute`** ***`off`*** Constant output appears in plain `monotype`, for example: `ok,00` Variable output appears in `monotype` *`Italic`*, for example: **`get locale country`** *`united-states`*
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}]`
	The following example restricts values to floating point numbers between -30.0 and 30.0 inclusive:
	`[-p {-30.0..30.0}]`

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 Phone as the ***auto*** user (default password ***lifesize***).

Help Mode

The CLI 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 28 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. Additionally, the argument **errors** returns a list of error message codes and their meanings. In normal mode, entering help produces an unsupported verb error.

Examples:

```
help

error,04

set help-mode on

ok,00

help
Possible verbs:
control
exit
get
help
history
set
status

ok,00

help set
Possible completions:
set admin password
set audio codecs
set audio mute
set config
set h323 alternate
set h323 extension
set h323 id
...
ok,00
```

Note: 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
```

Default Output Mode

The CLI 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 audio mute
```

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 editline library. These features operate in a similar manner to the GNU bash shell, including support for **`!n`**, **`!!`** and Emacs editing modes. History is limited to the last 100 commands.

Here Documents

The CLI supports a scripting feature known as a here document. When used in the CLI, a here document is a block of data that can be fed to certain commands that accept several lines of input (for example, uploading images or files to the system). Descriptions and examples in this manual indicate support for here documents when available for a command. Following is the syntax for specifying a here document in a CLI command:

command << ***TOKEN***

input_associated_with_command

TOKEN

where the here document consists of all text between the ***TOKEN*** document start symbol and the ***TOKEN*** document end symbol. The start symbol and end symbol must be identical. The input does not include the new line after the start symbol, but does include the new line immediately before the end symbol. The end symbol must start in the first column of a new line to be recognized. Here documents are generally used for sending scripts to the CLI through an SSH session. For example:

Manually enter an ssh key using a here document:

```
set ssh keys -i << EOF
ssh-rsa key_string user@lifesize.com
ssh-rsa key2_string user2@lifesize.com
EOF
```

Command Syntax

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

The general syntax 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.

[options] specifies arguments that may be passed in the command.

Note: Unless otherwise indicated, when specifying an argument that includes a text string with a space in the string, enclose the text in double quotes (for example, “QRB Meeting”).

Command Verbs

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.

Command Verb	Description
`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 language that appears in the interface).
`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. The exit verb has no arguments. *Example*: **`exit`** `ok,00` 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`** *`61,get locale country -V`* *`62,get locale language`* *`63,get locale language -V`* `ok,00` then, **`!61`** would execute command 61 (get locale country -V). 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. For example:

```
get system model -h
Usage: get system model [-?] [-D c] [-V] [-h]
-?          Display the column headers, even in terse mode
-D c        Specify an alternate delimiter character in
            terse mode (default is ',')
-V          Enable verbose output mode
-h          Produce this message

ok,00
```

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. For example:

```
get system model -D |
LifeSize|Phone

ok|00
```

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 ipv4
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 ipv4 -V
Mode     IP Address   Network Mask  Broadcast IP Gateway IP    MAC Address
static   10.10.100.5  255.255.255.0              10.10.100.1   00:13:fa:00:24:a1


    Hostname
    jsmith-ls
ok

get unknown-target

error 09 Target not recognized
```

Show Column Headings in Terse Mode: -?

To show column headings from verbose mode while in terse mode, specify the **-?** option to a command. In this mode, the column headings from verbose mode appear on the first line of output separated by commas, followed by terse mode output on the next line. For example:

```
get system model -?
OEM,Model
LifeSize,Phone

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) or the command timed out.
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.

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
Possible verbs:
control
exit
get
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 and set Verbs: Objects and Targets

This chapter identifies objects and targets that are applicable to the **`get`** and **`set`** verbs. Most of the objects and targets apply to both verbs. Where only one of the verbs applies, the description and examples specify the verb.

admin

The **`admin`** object controls configuration of administrator functions in the interface. This object applies to the **`set`** verb. The following targets apply to the **`admin`** object.

password

The **`password`** target sets the password for access to the administrator preferences. This target applies to the **`set`** verb.

Arguments:

`<value>`	The new administrator password. The password can be an empty string, the numbers 0-9 and/or the symbols * and #. The password 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

When used with the **`get`** verb, the **`codecs`** target retrieves the codec priority list. This list determines the order in which the audio codecs are used when connecting to other systems. When used with the **`set`** verb, this target changes the order in which the audio codecs are used when negotiating with a remote system. The list you specify is in highest priority to lowest priority order. For greatest compatibility, list all available codecs. Each codec may be listed only once.

get Arguments:

None

get Examples:

```
get audio codecs
aac-lc g.722.1c.48 g.722.1c.32 g.722.1c.24 g.722 g.729 g.711.u
   g.711.a

ok,00

get audio codecs -V
Codec Order
aac-lc g.722.1c.48 g.722.1c.32 g.722.1c.24 g.722 g.729 g.711.u
   g.711.a

ok
```

Following are the available codecs:

Codec	**CLI Name**
AAC Low Complexity	aac-lc
Polycom® Siren14™ (48 kb/s)	g.722.1c.48
Polycom® Siren14™ (32 kb/s)	g.722.1c.32
Polycom® Siren14™ (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

set 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 order of the audio codecs to use. List each codec only once. LifeSize recommends that you list each codec on the command line for greatest compatibility.

set Examples:

```
set audio codecs aac-lc g.722 g.722.1c.48 g.722.1c.32 g.722.1c.24
   g.711.a g.711.u g.729

ok,00
```

mute

When used with the **`get`** verb, the **`mute`** target retrieves the current setting of the local audio mute function. When used with **`set`** verb, this target controls whether or not the local audio inputs are muted.

get Arguments:

None

get Examples:

```
get audio mute
off

ok,00

get audio mute -V
State
on

ok
```

set Arguments:

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

set Examples:

```
set audio mute on

ok,00
```

config

When used with the **`get`** verb, the **`config`** target retrieves the current configuration for the system. This includes all saved parameters currently configurable by the CLI. The output is in the form of a script suitable for execution by the CLI. Before using the script with the **`set config`** command to restore the configuration of a system, you must edit the script as follows:

- Stored passwords are replaced by tokens surrounded by '###' characters (e.g., ###password###). Replace the these characters and tokens with the password.
- The system must be rebooted after the configuration is applied. Delete the trailing `ok,00` from the end of the script if it was captured. Append **`control reboot`** to the end of the script to effect a reboot.

When used with the **`set`** verb, this target allows reloading the system configuration from a script produced by **`get config`**. This is an alternate method to reading that script: it can also be fed directly to the CLI. The advantages are that the output of this command indicates the line numbers of failing commands in the script and the error codes of those commands and will exit with a return code indicating whether the entire script failed or succeeded. With the direct input method, the error messages for failing commands are mixed in with the output, and the exit code is that of the last command executed.

get Arguments:

`[-P]`	Export the file with all passwords except the shell password and snmp passwords. If you use this argument, all other passwords are visible without the pound (#) symbols.

get Examples:

```
sh% ssh auto@ip get config > codec_confg.as
sh%
```

To restore the configuration to a system using the direct input method:

```
sh% ssh auto@otherip < codec_config.as
command 1

ok,00
....
```

The output of the restore using the direct input method lists the executed commands followed by the return status of the command. Any command failures are indicated in the normal way. The script execution does not stop due to intermediate failures and the exit status of the script is the status of the final command that is executed.

set Arguments:

`[-i]`	Ignore errors in the script and execute to the end. The default executes up to the first error and then stops.

set Examples:

```
unix% ssh auto@ip get config > script
unix% vi script  # fix up passwords
unix% ssh auto@otherip set config -i -V < script
Line    Error    Note    Command
23      09       FIX     set admin password ###password###

error,09
```

h323

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

alternate

When used with the **`get`** verb, the **`alternate`** target retrieves the current settings for the alternate H.323 gatekeeper. When used with the **`set`** verb, this target configures the settings for the alternate H.323 gatekeeper when in manual mode.

get Arguments:

None

get Examples:

```
get h323 alternate
10.10.11.12.1719

ok,00

get h323 alternate -V
IP Address      Port
10.10.11.110    12345

ok
```

set 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.

***set Examples*:**

```
set h323 alternate 10.10.11.12

ok,00

set h323 alternate 10.10.11.12 1832

ok,00
```

extension

When used with the **`get`** verb, the **`extension`** target retrieves the H.323 extension associated with the endpoint. When used with the **`set`** verb, this target sets the extension to use when registering the device with the H.323 gatekeeper.

get Arguments:

None

get Examples:

```
get h323 extension
1188

ok,00

get h323 extension -V
Extension
1188

ok
```

set Arguments:

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

set Examples:

```
set h323 extension 1188

ok,00
```

id

When used with the **`get`** verb, the **`id`** target retrieves the H.323 gatekeeper ID. When used with the **`set`** verb, this target sets the H.323 gatekeeper ID.

get Arguments:

None

get Examples:

```
get h323 id
RADGK

ok,00

get h323 id -V
Gatekeeper ID
RADGK

ok
```

set Arguments:

`id`	Specify the gatekeeper ID.

set Examples:

```
set h323 id RADGK

ok,00
```

mode

When used with the **`get`** verb, the **`mode`** target retrieves the H.323 gatekeeper mode which indicates whether the gatekeeper is used at all or manually or automatically configured. When used with the **`set`** verb, this target configures the H.323 gatekeeper mode.

get Arguments:

None

get Examples:

```
get h323 mode
off

ok,00

get h323 mode -V
Mode
manual

ok
```

set Arguments:

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

set Examples:

```
set h323 mode auto

ok,00

set h323 mode off

ok,00
```

name

When used with the **`get`** verb, the **`name`** target retrieves the currently configured H.323 name for the device. When used with the **`set`** verb, this target sets the H.323 name for the device.

get Arguments:

None

get Examples:

```
get h323 name
LifeSize

ok,00

get h323 name -V
Name
LifeSize

ok
```

set Arguments:

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

set Examples:

```
set h323 name LifeSize

ok,00
```

primary

When used with the **`get`** verb, the **`primary`** target retrieves the configuration for the H.323 primary gatekeeper. When used with the **`set`** verb, this target configures the H.323 primary gatekeeper when the primary gatekeeper is in manual mode.

get Arguments:

None

get Examples:

```
get h323 primary
10.10.11.12,1719

ok,00

get h323 primary -V
IP Address      Port
10.10.11.110    12345

ok
```

set 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.

set Examples:

```
set h323 primary 10.10.11.12 1719

ok,00

set h323 primary 10.10.11.15

ok,00
```

register

When used with the **`get`** verb, the **`register`** target retrieves the current registration status of the H.323 gatekeeper. When used with the **`set`** verb, this target starts the registration process with the configured H.323 gatekeeper. Because registration may take an arbitrarily long time, the command returns immediately. Use the **`get h323 register`** command to check the status.

get Arguments:

None

`get Examples`:

```
get h323 register
registered

ok,00

get h323 register -V
Status
failed

ok
```

set Arguments:

None

set Examples:

```
set h323 register

ok,00
```

help-mode

When used with the **`get`** verb, the **`help-mode`** target retrieves the current setting for help mode. When used with the **`set`** verb, this target controls whether or not help is available. It also enables and disables the use of abbreviations for commands (abbreviating **`help-mode`** as just **`help`**). To avoid ambiguity in future software releases, LifeSize recommends that you do not use abbreviations in scripts.

get Arguments:

None

get Examples:

```
get help-mode
on

ok,00

get help-mode -V
Mode
off

ok
```

set Arguments:

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

set Examples:

```
set help-mode on

ok,00
```

http

When used with the **`get`** verb, the **`http`** target shows whether the web (http) service is enabled or disabled. When used with the **`set`** verb, this target controls whether the web (http) service is enabled or disabled.

get Arguments:

None

`get Examples:`

```
get http
on

ok,00

get http -V
Web (http) Service
off

ok
```

set Arguments:

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

set Examples:

```
set http on

ok,00
```

locale

The **`locale`** object controls location-specific information for a device. The following targets apply to the **`locale`** object.

country

When used with the **`get`** verb, the **`country`** target shows the current country setting for the system. When used with the **`set`** verb, this target configures the country code.

get Arguments:

None

get Examples:

```
get locale country
algeria

ok,00

get locale country -V
Country
uruguay

ok
```

set 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.

set Examples:

`set locale country` ***`algeria`***

```
ok,00
```

language

When used with the **`get`** verb, the **`language`** target shows the current language used for user interface prompts and messages. When used with the **`set`** verb, this target sets the language used for user interface prompts and messages. This setting does not affect the input or output of the CLI.

get Arguments:

None

get Examples:

```
get locale language
german

ok,00

get local language -V
GUI Language
traditional-chinese

ok
```

set Arguments:

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

set Examples:

```
set locale language brazilian-portuguese

ok,00
```

network

The **`network`** object controls the current network configuration. The following targets apply to the **`network`** object.

commit

The **`commit`** target commits the network settings and reboots the system. Network settings that are changed but not committed do not take effect until the next system reboot. This target applies to the **`set`** verb.

Arguments:

None

Examples:

```
set network commit

ok,00
```

dns

When used with the **`get`** verb, the **`dns`** target retrieves the current Directory Name Service settings. When used with the **`set`** verb, this target configures the Directory Name Service settings to allow the use of named hosts instead of IP addresses.

get Arguments:

None

get Examples:

```
get network dns
10.10.10.1,10.10.10.2,10.10.10.3,example.com

ok,00

get network dns -V
Primary DNS    Secondary DNS   Tertiary DNS   Search Domain List
10.10.10.1     10.10.10.2                     example.com

ok
```

set Arguments:

`[-i ipaddress]`	Specify an IP address for a DNS server. Up to 3 servers may be specified.
`[-r]`	Reset the DNS servers and search domains instead of appending additional servers/domains.
`[-s domain]`	Specify a search domain (used for unqualified hostname resolution).

set Examples:

```
set network dns -i 10.10.11.1 -i 10.10.11.2 -s ls.com -s cc.com

ok,00
```

hostname

The **`hostname`** target sets the network hostname for the system. Use a name that is similar or the same as the system name to avoid confusion. If DHCP is used for the network configuration, the hostname will be published to the DHCP server allowing name based lookups for the system. This target applies to the **`set`** verb.

Arguments:

`<hostname>`	Specify the hostname for the system

Examples:

```
set network hostname lifesize-phone

ok,00
```

ipv4

When used with the **`get`** verb, the **`ipv4`** target retrieves the current Internet Protocol Version 4 network configuration.

When used with the **`set`** verb, this target uses the **`dhcp`** and **`static`** targets to configure Internet Protocol Version 4 network parameters. The **`dhcp`** target configures the network to use Dynamic Host Control Protocol for the network settings. The **`static`** target controls the configuration of the network interface when you specify a static IP address for the device.

get Arguments:

None

get Examples:

```
get network ipv4
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 ipv4 -V
Mode  IP Address   Network Mask   Broadcast IP  Gateway IP   MAC Address
dhcp  10.10.100.5  255.255.255.0                10.10.100.1  00:13:fa:00:24:a1


   Hostname
   jsmith-ls
ok
```

set Arguments (dhcp target):

None

set Examples (dhcp target):

```
set network ipv4 dhcp

ok,00
```

set Arguments (static target):

`[-i ipaddr]`	Specify the IP address of the device.
`[-n netmask]`	Specify the network mask that defines the extent of the local network.
`[-g gateway]`	Specify the gateway address for routing traffic outside of the network defined by the IP address and network mask. The gateway device must be within the network.

set Examples (static target):

```
set network ipv4 static -i 10.10.11.12 -n 255.255.0.0 -g
   10.10.1.1

ok,00
```

ipv6

When used with the **`get`** verb, the **`ipv6`** target retrieves the current Internet Protocol Version 6 network configuration.

When used with the **`set`** verb, this target uses the following targets to configure Internet Protocol Version 6 (IPv6) networking parameters:

- **`auto`**

  The **`auto`** target enables the system to determine the networking parameters from the network without further user intervention.

- **`manual`**

  The **`manual`** target enables you to enter the IPv6 addresses of the system and the router manually.

- **`off`**

  The **`off`** target disables IPv6 networking.

get Arguments:

None

get Examples:

```
get network ipv6
yes,auto,yes,ipv6Address,

ok,00
get network ipv6 -V
Enabled   Mode   Active   IP Address                                Router
yes       auto   yes      ipv6Address

ok
```

set Arguments (auto target):

None

set Examples (auto target):

```
set network ipv6 auto

ok,00
```

set Arguments (manual target):

`[-i ipaddress]`	Specify the IPV6 address for the phone.
`[-r routerip]`	Specify the IPV6 address of the router.

set Examples (manual target):

```
set network ipv6 manual -i address

ok,00
```

set Arguments (off target):

None

set Examples (off target):

```
set network ipv6 off

ok,00
```

nat

When used with the **`get`** verb, the **`nat`** target retrieves the Network Address Translation settings for the system.

When used with the **`set`** verb, this target uses the **`disabled`** and **`enabled`** targets to configure Network Address Translation. The **`disabled`** target disables the use of NAT traversal on the device. The **`enabled`** target enables the use of NAT traversal on the device.

get Arguments:

None

get Examples:

```
get network nat

none,

ok,00

get network nat -V
Static NAT      Public IP
manual          10.10.11.111

ok
```

set Arguments (disabled target):

None

set Examples (disabled target):

```
set network nat disabled

ok,00
```

set Arguments (enabled target):

`<ipaddress>`	Specify the public IP address of the phone.

set Examples (enabled target):

```
set network nat enabled address

ok,00
```

ntp-server

When used with the **`get`** verb, the **`ntp-server`** target retrieves the address of the current Network Time Protocol server. When used with the **`set`** verb, this target specifies the Network Time Protocol server to use to keep the system clock synchronized with a common time source.

get Arguments:

None

get Examples:

```
get network ntp-server
10.10.11.10

ok,00

get network ntp-server -V
NTP Server
10.10.11.10

ok
```

set Arguments:

`<ntpserver>`	Specify the IP address of the NTP server

set Examples:

```
set network ntp-server 10.10.11.10

ok,00
```

qos

When used with the **`get`** verb, the **`qos`** target retrieves the configuration of the network Quality of Service options for the system.

When used with the **`set`** verb, this target uses the following targets to configure the Quality of Service (QoS) options for the system:

- **`diffserv`**

  The **`diffserv`** target configures the network QoS for DiffServ.

- **`intserv`**

  The **`intserv`** target configures the network QoS for IntServ (IP Precedence).

- **`none`**

  The **`none`** target disables network QoS.

Arguments:

None

Examples:

```
get network qos
DiffServ,46,34,46,

ok,00

get network qos -V
QoS Mode  Audio Priority  Video Priority  Data Priority  Type Of Service
IntServ   5               4               5              Minimize Cost

ok
```

set Arguments (diffserv target):

`[-a {0..63}]`	Specify the audio packet priority.
`[-d {0..63}]`	Specify the data packet priority.

set Examples (diffserv target):

```
set network qos diffserv -a 46 -d 46 -v 34

ok,00
```

set Arguments (intserv target):

`[-a {0..7}]`	Specify the audio packet priority.
`[-d {0..7}]`	Specify the data packet priority.
`[-t {none|min-delay|min-cost|max-rely|max-thru}]`	Specify the type of service used by your network: None, Minimize Delays, Minimize Cost, Maximize Reliability, or Maximize Throughput

set Examples (intserv target):

`set network qos intserv -a` ***`6`*** **`-d`** ***`4`*** **`-v`** ***`3`*** **`-t`** ***`min-delay`***

```
ok,00
```

set Arguments (none target):

None

set Examples (none target):

`set network qos none`

```
ok,00
```

reserved-ports

When used with the **`get`** verb, the **`reserved-ports`** target retrieves the configuration of ports reserved for use by the device. When used with the **`set`** verb, this target specifies the upper and lower bounds for the ports reserved for use by the device.

get Arguments:

None

get Examples:

```
get network reserved-ports
64000,64999,64000,64999

ok,00

get network reserved-ports -V
UDP Low Port     UDP High Port    TCP Low Port    TCP High Port
64000            64999            64000           64999

ok
```

set Arguments:

[-T {1025..65535}]	Specify the upper bound for TCP reserved ports
[-U {1025..65535}]	Specify the upper bound for UDP reserved ports
[-t {1025..65535}]	Specify the lower bound for TCP reserved ports
[-u {1025..65535}]	Specify the lower bound for UDP reserved ports

set Examples:

```
set network reserved-ports -t 30000 -T 40000

ok,00
```

speed

When used with the **get** verb, the **speed** target shows the actual speed of the network port. Possible values are `100-fd` (100Mbps, full duplex), `100-hd` (100Mbps, half duplex), `10-fd` (10Mbps, full duplex) and `10-hd` (10Mbps, half duplex). When used with the **set** verb, this target configures the network port default speed.

get Arguments:

None

get Examples:

```
get network speed
100-fd

ok,00

get network speed -V
Network Speed
10-hd

ok,00
```

set Arguments:

`<{auto|100-auto|100-fd|10-auto|10-fd}>`	Specify the network speed and duplex. `Auto` negotiates 10 or 100Mbps and full or half duplex. `100-auto` and `10-auto` negotiate only duplex. `100-fd` and `10-fd` do not negotiate at all. Set the speed to auto unless the remote networking equipment is incapable of auto negotiation.

set Examples:

```
set network speed auto

ok,00
```

status

The **`status`** target shows the current status of the network connection. This target applies to the **`get`** verb. Possible values include:

- `connected`
- `binding` (attempting to retrieve IP configuration)
- `no dhcp response` (dhcp server timed out)
- `unconnected`

Arguments:

None

Examples:

```
get network status
connected

ok,00

get network status -V
Network State
binding

ok
```

password

The **`password`** target changes the user's password while running the CLI. This target applies to the **`set`** verb. When used with the optional arguments, **`password`** behaves like other CLI targets. However, using the arguments is insecure: the **`set password`** command may show up in the saved history for the CLI across login sessions, allowing disclosure of the new password. Also, checks for minimum length and complexity are not performed on the password in this mode. For security reasons, LifeSize recommends that you use the interactive version.

Note: The interactive mode of this command does not conform to the standard output specification, because it uses the standard passwd utility to perform the change.

Arguments:

`[old-password]`	Specify the current password for the CLI.
`[new-password]`	Specify the new password for the CLI.

Examples:

Interactive:

```
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
```

Non-interactive:

```
set password lifesize 123ABC!@#abc

ok,00
```

prompt

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

Arguments:

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

Examples:

```
set prompt "% "

ok,00
% set prompt "-> "

ok,00
->
```

redial-list

The **`redial-list`** target retrieves the redial call list. This target applies to the **`get`** verb.

Arguments:

None

Examples:

```
get redial-list
1,Sunbob2,10.10.11.116,10.10.11.116,Audio,Outgoing,Yes,auto,auto
2,10.10.11.186,10.10.11.186,10.10.11.186,Audio,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      Training     10.10.11.116  10.10.11.116  Audio   Outgoing  No      h323      auto
2      LifeSize     10.10.11.186  10.10.11.186  Audio   Outgoing  No      h323      auto
3      Sunfish      10.10.11.116  10.10.11.116  Audio   Incoming  No      h323      auto

ok
```

The valid values for the `Type` column are *Audio*, *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. 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.

sip

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

authorization

When used with the **`get`** verb, the **`authorization`** target retrieves the user name for authorization with the SIP registrar. For security reasons, the associated password is not displayed. When used with the **`set`** verb, this target configures the user name used for authorization with the SIP registrar.

get Arguments:

None

get Examples:

```
get sip authorization
sipuser

ok,00

get sip authorization -V
Username
sipuser

ok
```

set Arguments:

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

set Examples:

```
set sip authorization sipuser sippassword

ok,00
```

proxy

When used with the **`get`** verb, the **`proxy`** target retrieves the SIP proxy settings. When used with the **`set`** verb, this target sets the SIP proxy configuration.

get Arguments:

None

get 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
```

set 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.

set 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

When used with the **`get`** verb, the **`register`** target shows the current registration status for SIP. When used with the **`set`** verb, this target registers the device with the configured SIP server or proxy. Use this command only after completing all other SIP configuration tasks. Since registration may take an arbitrarily long time, this command returns immediately. Use the **`get sip register`** command to retrieve the registration status.

get Arguments:

None

get Examples:

```
get sip register
unregistered

ok,00

get sip register -V
Status
registered

ok
```

set Arguments:

None

set Examples:

```
set sip register

ok,00
```

registrar

When used with the **`get`** verb, the **`registrar`** target retrieves the current SIP registrar settings. When used with the **`set`** verb, this target configures the SIP registrar settings.

get Arguments:

None

get 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
```

set 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.

set 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

When used with the **`get`** verb, the **`tcp`** target shows the configuration of the TCP options for SIP calls. When used with the **`set`** verb, this target configures the TCP options for SIP calls. If configuring the device for SIP calls, either the **`tcp`** or **`udp`** target must be enabled.

get Arguments:

None

get Examples:

```
get sip tcp
enabled,5060

ok,00

get sip tcp -V
State           Port
disabled        5060

ok
```

set 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.

set Examples:

```
set sip tcp enabled

ok,00

set sip tcp disabled

ok,00

set sip tcp enabled 5060

ok,00
```

udp

When used with the **`get`** verb, the **`udp`** target retrieves the configuration of the UDP options for SIP calls. When used with the **`set`** verb, this target configures the UDP options for SIP calls. If configuring the device for SIP calls, either the **`tcp`** or **`udp`** target must be enabled.

get Arguments:

None

get Examples:

```
get sip udp
disabled,5060

ok,00

get sip udp -V
State           Port
enabled         5060

ok
```

set 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.

set Examples:

```
set sip udp enabled

ok,00

set sip udp disabled

ok,00

set sip udp enabled 7000

ok,00
```

username

When used with the **`get`** verb, the **`username`** target retrieves the current SIP username. When used with the **`set`** verb, this target sets the SIP user name for the system.

get Arguments:

None

get Examples:

```
get sip username
lifesize

ok,00

get sip username -V
Name
lifesize

ok
```

set Arguments:

`name`	Set the user name for the system.

set Examples:

```
set sip username lifesize

ok,00
```

via-proxy

When used with the **`get`** verb, the **`via-proxy`** target shows whether SIP registration uses a proxy to connect to the registrar, or connects directly. When used with the **`set`** verb, target target controls whether SIP registration uses a proxy to connect to the registrar or connects directly.

get Arguments:

None

get Examples:

```
get sip via-proxy
proxy

ok,00

get sip via-proxy -V
State
direct

ok
```

set Arguments:

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

set 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.

community-name

When used with the **`get`** verb, the **`community-name`** target retrieves the SNMP community name.

Note: The SNMP server running on the device implements SNMP version 3. The **`set snmp community-name <communityName>`** command applies to SNMP v1 and v2 and is not supported in this release.

get Arguments:

None

get Examples:

```
get snmp community-name
foo

ok,00

get snmp community-name -V
SNMP Community Name
foo

ok
```

contact

When used with the **`get`** verb, the **`contact`** target retrieves the SNMP contact name. When used with the **`set`** verb, this target sets the SNMP contact name on the SNMP server running on the device.

get Arguments:

None

get Examples:

```
get snmp contact
Administrator

ok,00

get snmp contact -V
SNMP Contact
Administrator

ok
```

set Arguments:

<contactname>	Specify the contact name for the SNMP server. If the contact name contains more than one word separated by a space, enclose the name in quotes (“ “).

set Examples:

```
set snmp contact Administrator

ok,00
```

description

The **`description`** target retrieves the SNMP description of the system. This target applies to the **`get`** verb.

Arguments:

None

Examples:

```
get snmp description
LifeSize Room

ok,00

get snmp description -V
SNMP System Description
LifeSize Team

ok
```

enable

When used with the **`get`** verb, the **`enable`** target shows whether or not the SNMP service is enabled. When used with the **`set`** verb, this target enables or disables the SNMP service.

get Arguments:

None

get Examples:

```
get snmp enable
on

ok,00

get snmp enable -V
Value
off

ok
```

set Arguments:

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

set Examples:

```
set snmp enable on

ok,00
```

location

When used with the **`get`** verb, the **`location`** target shows the configured location for the SNMP service. When used with the **`set`** verb, this target sets the configured location for the SNMP service.

get Arguments:

None

get Examples:

```
get snmp location
Austin

ok,00

get snmp location -V
SNMP Location
Austin

ok
```

set Arguments:

`<location>`	Specify the location for the SNMP service. If the location contains more than one word separated by a space, enclose the location in quotes (“ “).

set Examples:

```
set snmp location Austin

ok,00
```

system-name

The **`system-name`** target retrieves the SNMP system name. This target applies to the **`get`** verb.

get Arguments:

None

get Examples:

```
get snmp system-name
foo

ok,00

get snmp system-name -V
SNMP System Name
foo

ok
```

user

When used with the **`get`** verb, the ***`user`*** target retrieves the SNMP user names. When used with the **`set`** verb, this target adds or deletes SNMP users.

get Arguments:

None

get Examples:

```
get snmp user
user1
Control
user2

ok,00

get snmp user -V
Username
user1
Control
user2

ok
```

Note: The Control user is a default user for use with LifeSize Control. You can delete this user if you are not using LifeSize Control or delete this user and create a different user for use with LifeSize Control. Use the **`set snmp user`** command to delete and create SNMP users.

set Arguments:

`-a`	Add the specified user (cannot be used with -d).
`-d`	Delete the specified user (cannot be used with -a).
`<username>`	Specify the user name. User names must not contain spaces.
`<password>`	Specify the password for the user. Required with -a. The password must be at least 8 characters in length and must not contain spaces.

set Examples:

Add a user:

`set snmp user -a` ***`username password`***

Delete a user:

`set snmp user -d` ***`username`***

v3trapdestination

When used with the **`get`** verb, the **`v3trapdestination`** shows the current version 3 SNMP trap destinations (where SNMP traps are sent). When used with the **`set`** verb, this target adds or removes entries from the list of version 3 SNMP trap destinations.

Note: The user's password is not displayed.

get Arguments:

None

get 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
```

set 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. User names must not contain spaces.
`<password>`	Specify the password for the user. Passwords must not contain spaces. Password must be at least 8 characters in length.
`<ipaddress>`	Specify the IP address of the trap destination. IP addresses must not contain spaces.

Note: The <username>, <password>, and <ipaddress> arguments are required with the -a and -d arguments. Either -a or -d must be specified. Users specified with this command appear in the output of the **`get snmp user`** command.

set 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
```

version

The **`version`** target retrieves the SNMP version number for the SNMP server running on the device. This target applies to the **`get`** verb.

Arguments:

None

Examples:

```
get snmp version
3

ok,00

get snmp version -V
SNMP Version
3

ok
```

ssh

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

keys

When used with the **get** verb, the **keys** target retrieves information about the currently installed ssh authorized keys. When used with the **set** verb, this target sets the ssh authorized keys for the **auto** user. Authorized keys allow the remote user to log into the system without using a password. Only one authorized ssh key is supported.

get Arguments:

None

get Examples:

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

ok,00

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

ok
```

set Arguments:

`[-c]`	Clear the keys file, removing all installed keys.
`[-i]`	Install a new ssh key. The key is read from standard input and must be less than 4096 characters in length. You may specify multiple keys on separate lines. A maximum of 64 keys are supported.
`[-r owner]`	Remove an existing ssh key. The owner field must match the prefix of the comment field in the key file, ignoring case (for example, an owner of "li" would match all comment fields beginning with "li" in any case).

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

set Examples:

Manually enter an ssh key using a here document or paste the key file into the command line:

```
set ssh keys -i << EOF
ssh-rsa key_string user@lifesize.com
ssh-rsa key2_string user2@lifesize.com
EOF

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: The key file must not have any text prior to the key data and the key type, value, and comment (owner) must be on a single line.

service

When used with the **`get`** verb, the **`service`** target shows whether the ssh service is enabled or disabled. When used with the **`set`** verb, this target enables or disables the ssh service. An ssh session in progress is not affected if the service is disabled.

Note: Be aware that if the device does not have a serial port and you disable the ssh service and then quit the session, you may need to use the user interface or web administration interface to re-enable the ssh service.

get Arguments:

None

get Examples:

```
get ssh service
on

ok,00

get ssh service -V
Secure Shell Service
off

ok
```

set Arguments:

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

set Examples:

```
set ssh service on

ok,00
```

system

The **`system`** object allows setting of certain system-specific parameters, for example, the system name and may be useful for tracking and monitoring inventory. The following targets are applicable to the **`system`** object.

clean

The **`clean`** target removes personally identifiable information from the system, including call history logs, directory entries, system identity data, IP addresses, and Redial list entries. Use the **`clean`** target, for example, when you wish to use a system for customer demonstrations or for other uses that require the removal of personally identifiable information. This target applies to the **`set`** verb.

Arguments:

`[-C]`	Clean the call history logs (status call history).
`[-a]`	Clean all data.
`[-c]`	Clean the corporate directory, disable LDAP and Auto Discovery (get directory corporate, get directory ldap, get directory auto).
`[-d]`	Clean all directories (equivalent to -c -l -m).
`[-i]`	Clean system identity data (get system name, get system number, get network hostname).
`[-l]`	Clean the local directory (get directory local).
`[-m]`	Clean the meetings directory (get directory meeting).
`[-n]`	Clean the network config (get network ipv4, get network ipv6). Network is set to IPv4 static with no address and IPv6 is disabled. This occurs on reboot.
`[-r]`	Clean the redial list (get redial-list).

Examples:

Clean everything:

```
set system clean -a

ok,00
```

Clean only the directories and redial list:

```
set system clean -d -r

ok,00
```

date

When used with the **`get`** verb, the **`date`** target shows the current system date and time in either the local time zone or as UTC time.

When used with the **`set`** verb, this target changes the system time and date. The value is always specified in terms of the local time zone.

get Arguments:

`[-u]`	Show the time in UTC instead of the local time zone.

get Examples:

```
get system date
2007,10,8,16,58,25

ok,00

get system date -u -V
Year     Month    Day     Hour     Minute     Second
2007     10       8       21       58         25

ok
```

set Arguments:

`[-H {0..23}]`	Specify the hour.
`[-M {0..59}]`	Specify the minute.
`[-S {0..59}]`	Specify the second.
`[-d {1..31}]`	Specify the day of month. February 31st is interpreted as March 2nd or 3rd depending on whether the year is a leap year or not.
`[-m {1..12}]`	Specify the month.
`[-y {2005..2025}]`	Specify the year.

set Examples:

```
# change only the time, not the day
set system date -H 4 -M 3 -S 0

ok,00

# change only the day of month
set system date -d 12

ok,00
```

lcd-contrast

When used with the **`get`** verb, the **`lcd-contrast`** target retrieves the current setting of the LifeSize Phone's LCD contrast. When used with the **`set`** verb, this target controls the current setting of the LifeSize Phone's LCD contrast.

get Arguments:

None

get Examples:

```
get system lcd-contrast
6

ok,00
get system lcd-contrast -V
Setting
12

ok
```

set Arguments:

`<{1..12}>`	Specify the contrast setting

set Examples:

```
set system lcd-contrast 7

ok,00
```

message

The **`message`** target specifies a pop-up dialog box with a message and button layout to appear in the user interface. The dialog box can be used to inform the users of impending system maintenance or other important news. This target applies to the **`set`** verb.

Arguments:

`[-b {yes|no|cancel|ok}]`	Specify the buttons to be present in the popup. A maximum of 3 buttons may be displayed, but each button can only appear once.
`[-e]`	Specify that the dialog use the error icon (red triangle with exclamation point). The default button layout for this dialog is the OK button.
`[-i]`	Specify that the dialog use the information icon (message page). This is the default dialog type and includes an OK button.
`[-q]`	Specify that the dialog use the question icon (a question mark). The default button layout includes the Yes and No buttons.
`[-t seconds]`	Specify the timeout interval (in number of seconds) for the dialog. The default timeout is 30 seconds.
`[-w]`	Specify that the dialog use the warning icon (yellow triangle with exclamation point). The default button layout for this dialog is the OK button.
`message`	Specify the message to place in the dialog. If including spaces, enclose the entire message in double quotes. To wrap the message at a specific point, insert `'\n'` at the desired location in the message.

Examples:

Shows an information dialog with the desired text:

```
set system message "Hello World"

ok,00
```

Shows an error dialog with a 45-second timeout period, ok, and cancel buttons:

```
set system message -b ok -b cancel -t 45 -e "Too Hot"

ok,00
```

The user response is available through the **`get system message-status`** command.

message-status

The **`message-status`** target retrieves the user response from the most recent popup message displayed. Results may include the following:

- `yes` (user pressed the dialog's yes or ok button)
- `no` (user pressed the dialog's no button)
- `cancel` (user pressed the dialog's cancel or the remote's back button)
- `timeout` (dialog timed out before the user responded)
- empty string (user has not yet responded and the dialog has not yet timed out)

This target applies to the **`get`** verb.

Arguments:

None

Examples:

```
get system message-status


ok,00

get system message-status -V
Result
cancel

ok
```

model

The **model** target shows the OEM and model name for the platform. This target applies to the **get** verb.

Arguments:

None

Examples:

```
get system model
LifeSize,Phone

ok,00

get system model -V
OEM         Model
LifeSize    Phone

ok
```

name

When used with the **get** verb, the **name** target shows the current name for the device. This is the same value that appears in the user interface and on a connected phone. When used with the **set** verb, this target sets the device name.

get Arguments:

None

get Examples:

```
get system name
LifeSize

ok,00

get system name -V
System Name
LifeSize

ok
```

set Arguments:

`<value>`	Specify the name for the system

set Examples:

```
set system name "LifeSize Phone"

ok,00
```

number

When used with the **`get`** verb, the **`number`** target retrieves the voice telephone number associated with the device. This appears in the phone display. When used with the **`set`** verb, this target sets the voice telephone number associated with the device.

get Arguments:

None

get Examples:

```
get system number
555-1212

ok,00

get system number -V
System Phone Number
555-1212

ok
```

set Arguments:

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

set Examples:

```
set system number 555-1212

ok,00
```

out-of-box

When used with the **`get`** verb, 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. When used with the **`set`** verb, this target runs the initial configuration process or cancels an already running initial configuration process.

get Arguments:

None

get Examples:

```
get system out-of-box
enabled

ok,00

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

ok
```

set Arguments:

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

set Examples:

```
set system out-of-box enabled

ok,00
```

pstn

The **`pstn`** target returns whether or not the system has the hardware necessary to make a Public Switched Telephone Network (PSTN) call. This target applies to the **`get`** verb.

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 retrieves the serial numbers of the CPU board and System board within the codec. This target applies to the **`get`** verb.

Arguments

None

Examples:

```
get system serial-number
BDxxxxxxxxxx,BBxxxxxxxxxx,BCxxxxxxxxxx

ok,00

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

ok
```

uptime

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

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. This target applies to the **`get`** verb.

Arguments:

None

Examples:

```
get system version
Software Version,LS_PH1_3.5.0 (0)
U-Boot Version,U-Boot 1.1.2 Lifesize Phone  1.7
Keyboard,003
ok,00

get system version -V
Software Version      Value
Software Version      LS_PH1_3.5.0 (0)
U-Boot Version        U-Boot 1.1.2 Lifesize Phone, 1.7
Keyboard              003

ok
```

telnet

When used with the **`get`** verb, the **`telnet`** target 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.

When used with the **`set`** verb, this object enables or disables the telnet service in real time. Active telnet sessions are disconnected if the service is stopped without closing the sessions first.

get Arguments:

None

get Examples:

```
get telnet
on

ok,00

get telnet -V
Telnet Service
off

ok
```

set Arguments:

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

set Examples:

```
set telnet on

ok,00
```

verbose-mode

When used with the **`get`** verb, the **`verbose-mode`** target retrieves the current setting for verbose mode. When used with the **`set`** verb, this target enables or disables verbose mode output. Verbose mode provides human readable output. Enabling verbose mode is equivalent to specifying -V with each command entered.

get Arguments:

None

get Examples:

```
get verbose-mode
on

ok,00

get verbose-mode -V
Mode
off

ok
```

set Arguments:

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

set Examples:

```
set verbose-mode on

ok

set verbose-mode off

ok,00
```

volume

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

dtmf

When used with the **`get`** verb, the **`dtmf`** target retrieves the current volume setting (using a scale of 0 to 10) for Dual Tone Multi Frequency (DTMF) tones when placing a call. When used with the **`set`** verb, this target controls the volume setting for Dual Tone Multi Frequency (DTMF) tones.

get Arguments:

None

get Examples:

```
get volume dtmf
5

ok,00

get volume dtmf -V
Volume
6

ok
```

set Arguments:

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

set Examples:

```
set volume dtmf 5

ok,00
```

ring-tone

When used with the **`get`** verb, the **`ring-tone`** target retrieves the current volume setting for the ring tone. When used with the **`set`** verb, this target controls the volume setting for the ring tone.

get Arguments:

None

get Examples:

```
get volume ring-tone
5

ok,00

get volume ring-tone -V
Volume
6

ok
```

set Arguments:

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

set Examples:

```
set volume ring-tone 5

ok,00
```

speaker

When used with the **`get`** verb, the **`speaker`** target retrieves the current volume setting for the system speaker (audio loudness). When used with the **`set`** verb, this target controls the volume of the system speaker.

get Arguments:

None

get Examples:

```
get volume speaker
50

ok,00

get volume speaker -V
Volume
70

ok
```

set Arguments:

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

set Examples:

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

status-tone

When used with the **`get`** verb, the **`status-tone`** target retrieves the current volume setting for the system status tones. When used with the **`set`** verb, this target controls the volume of the system status tones.

get Arguments:

None

get Examples:

```
get volume status-tone
3

ok,00

get volume status-tone -V
Volume
5

ok
```

set Arguments:

`<{0..10}>`	Specify the volume level (0 = off, 10 = max) for the status tones.

set Examples:

```
set volume status-tone 5

ok,00
```

status Verb: Object and Targets

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.

Examples:

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

ok,00
```

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

ok
```

```
status call active -d incoming
11,1,Connected,Yes,Audio,10.10.11.155,Jones
12,1,Connected,Yes,Audio,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 *`Audio`*, and *`Unknown`*.

history

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

Note: The output for fields identified as applicable to video contain *`0`*, *`No`*, *`None`*, or *`0.000`* for LifeSize Phone.

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.	Default
`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.	Default
`Local Name`	The name of the local system (the system providing the call history)	Default
`Local Number`	The number of the local system	Default
`Remote Name`	The name of the remote system (the other participant in the call)	Default
`Remote Number`	The number of the remote system	Default
`Dialed Digits`	The digits used to place the call	Default
`Start Time`	The time in ISO date format at which the call started	Default

Field Name	Description	Display Mode[a]
`End Time`	The time in ISO date format at which the call ended	Full
`Duration`	The length of the call in hours:minutes:seconds	Default
`Direction`	Indication of incoming or outgoing call	Default
`Protocol`	The communications protocol used for the call	Full
`Security`	The security protocol used for the call	Full
`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
`TxV1 Pct Loss`	Percent packet loss of primary video transmitted	Full
`RxV1 Pct Loss`	Percent packet loss for primary video received	Full

Field Name	Description	Display Mode[a]
`TxV1 Pkts Lost`	Number of packets lost for primary video transmitted	Full
`RxV1 Pkts Lost`	Number of packets lost for primary video transmitted.	Full
`TxV1 Avg Jitter`	Average jitter for primary video transmitted	Full
`RxV1 Avg Jitter`	Average jitter for primary video received	Full
`TxV1 Max Jitter`	Maximum jitter for primary video transmitted	Full
`RxV1 Max Jitter`	Maximum jitter for primary video received	Full
`TxA1 Pct Loss`	Percent packet loss for audio transmitted	Full
`RxA1 Pct Loss`	Percent packet loss for audio received	Full
`TxA1 Pkts Lost`	Number of packets lost for audio transmitted	Full
`RxA1 Pkts Lost`	Number of packets lost for audio received	Full
`TxA1 Avg Jitter`	Average jitter for audio transmitted	Full
`RxA1 Avg Jitter`	Average jitter for audio received	Full
`TxA1 Max Jitter`	Maximum jitter for audio transmitted	Full
`RxA1 Max Jitter`	Maximum jitter for audio received	Full
`TxV2 Pct Loss`	Percent packet loss for secondary video transmitted	Full
`TxV2 Pct Loss`	Percent packet loss for secondary video received	Full
`TxV2 Pkts Lost`	Number of packets lost for secondary video transmitted	Full
`RxV2 Pkts Lost`	Number of packets lost for secondary video transmitted	Full
`TxV2 Avg Jitter`	Average jitter for secondary video transmitted	Full

Field Name	Description	Display Mode[a]
`RxV2 Avg Jitter`	Average jitter for secondary video received	Full
`TxV2 Max Jitter`	Maximum jitter for secondary video transmitted	Full
`RxV2 Max Jitter`	Maximum jitter for secondary video received	Full

a. The `Default` 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
3,3,LifeSize,10.10.7.49,Training,10.10.11.213,10.10.11.213,
   2008-03-24 20:44:54,00:00:04,In
2,2,LifeSize,10.10.7.49,Training,10.10.11.213,10.10.11.213,
   2008-03-24 20:43:19,00:00:29,Out
1,1,LifeSize,10.10.7.49,,,,2008-03-24 20:42:52,00:00:00,Out

ok,00
```

Note: The output in the following example is split by column into tables 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
3      3      LifeSize       10.10.7.49       Training
2      2      LifeSize       10.10.7.49       Training
1      1      LifeSize       10.10.7.49

Remote Number    Dialed Digits  Start Time
10.10.11.213     10.10.11.213   2008-03-24 20:44:54
10.10.11.213     10.10.11.213   2008-03-24 20:43:19
                                2008-03-24 20:42:52

Duration  Direction
00:00:04  In
00:00:29  Out
00:00:00  Out

ok

status call history -f
3,3,LifeSize,10.10.7.49,Training,10.10.11.213,10.10.11.213,
   2008-03-24 20:44:54,2008-03-24 20:44:58,00:00:04,In,H.323,
   None,128,0,,AAC_LC,,,AAC_LC,,No,No,None,Normal,0.000,0.000,
   0,0,0.000,0.000,0,0,0.000,0.000,1,0,32.000,28.000,32,28,
   0.000,0.000,0,0,0.000,0.000,0,0
2,2,LifeSize,10.10.7.49,Training,10.10.11.213,10.10.11.213,
   2008-03-24 20:43:19,2008-03-24 20:43:48,00:00:29,Out,H.323,
   None,128,0,,AAC_LC,,,AAC_LC,,No,No,None,Normal,0.000,
   0.000,0,0,0.000,0.000,0,0,0.000,0.000,0,0,31.000,31.000,
   31,31,0.000,0.000,0,0,0.000,0.000,0,0
1,1,LifeSize,10.10.7.49,,,,2008-03-24 20:42:52,
   2008-03-24 20:42:52,00:00:00,Out,H.323,None,128,0,,,,,,,
   No,No,None,Normal,0.000,0.000,0,0,0.000,0.000,0,0,0.000,
   0.000,0,0,0.000,0.000,0,0,0.000,0.000,0,0,0.000,0.000,0,0

ok,00
```

The output in the following example is split by column into multiple tables 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
3	*3*	*LifeSize*	*10.10.7.49*	*Training*	*10.10.11.213*	*10.10.11.213*
2	*2*	*LifeSize*	*10.10.7.49*	*Training*	*10.10.11.213*	*10.10.11.213*
1	*1*	*LifeSize*	*10.10.7.49*	*Training*		

Start Time	End Time	Duration	Direction	Protocol	Security
2008-03-24 20:44:54	*2008-03-24 20:44:58*	*00:00:04*	*In*	*H.323*	*None*
2008-03-24 20:43:19	*2008-03-24 20:43:48*	*00:00:29*	*Out*	*H.323*	*None*
2008-03-24 20:42:52	*2008-03-24 20:42:52*	*00:00:00*	*Out*	*H.323*	*None*

Req Kibps	Act Kibps	TX Vid	TX Aud	TX Res	RX Vid	RX Aud	RX Res
128	*0*		*AAC_LC*			*AAC_LC*	
128	*0*		*AAC_LC*			*AAC_LC*	
128	*0*						

TX Pres	RX Pres	Pres Fmt	Term Code	TxV1 Pct Loss	RxV1 Pct Loss	TxV1 Pkts Lost
No	*No*	*None*	*Normal*	*0.000*	*0.000*	*0*
No	*No*	*None*	*Normal*	*0.000*	*0.000*	*0*
No	*No*	*None*	*Normal*	*0.000*	*0.000*	*0*

RxV1 Pkts Lost	TxV1 Avg Jitter	RxV1 Avg Jitter	TxV1 Max Jitter	RxV1 Max Jitter
0	*0.000*	*0.000*	*0*	*0*
0	*0.000*	*0.000*	*0*	*0*
0	*0.000*	*0.000*	*0*	*0*

TxA1 Pct Loss	RxA1 Pct Loss	TxA1 Pkts Lost	RxA1 Pkts Lost	TxA1 Avg Jitter
0.000	*0.000*	*1*	*0*	*32.000*
0.000	*0.000*	*0*	*0*	*31.00*
0.000	*0.000*	*0*	*0*	*0.000*

RxA1 Avg Jitter	TxA1 Max Jitter	RxA1 Max Jitter	TxV2 Pct Loss	RxV2 Pct Loss
28.000	*32*	*28*	*0.000*	*0.000*
31.00	*31*	*31*	*0.000*	*0.000*
0.000	*0*	*0*	*0.000*	*0.000*

```
TxV2 Pkts Lost  RxV2 Pkts Lost  TxV2 Avg Jitter  RxV2 Avg Jitter  TxV2 Max Jitter
0               0               0.000            0.000            0
0               0               0.000            0.000            0
0               0               0.000            0.000            0

RxV2 Max Jitter
0
0
0

ok
```

statistics

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

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.
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.

Field Name	Description
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.

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 -V
ID     ARX Codec Kibps   ATX Codec Kibps   ARX Jitter  ARX Pktps   ARX Pkt Loss
7      aac-lc    1.5     aac-lc    124.2   31          31          0
8      aac-lc    1.5     aac-lc    128     32          32          0
```

control Verb: Objects and Targets

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,Audio,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,Audio,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 reference number of the call.
3	Conference ID	<number>	Indicates the reference number of the conference managing this call.
4	State	On Hook Terminating Terminated Off Hook Valid Number Dialing Proceeding Ringing Answered Number Connected Notify Info Ring Incoming Caller ID Local Ring Back Off Far End Mute Far End Unmute Far End Hold Far End Resume	Phone is on hook. Call is terminating. 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. Call is connected. Notification of miscellaneous events. Incoming call received. Caller ID information. Local ringback is off. 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	The message pertains to an audio 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,Audio,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.

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.

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.

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:

`[-p {auto|h323|h323gw|sip}]`	Specify the protocol to use to connect the new party to the call. The default is auto.
`<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 1 555-1212

ok,00
CS,3,1,Ringing,Audio,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,Audio,Normal,10.10.11.10,LifeSize
Add 3rd redial entry as call:
% control call add-part 1 redial:3

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

Refer to “Asynchronous Messages” on page 90 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.
`<callHandle>`	Specify the handle of the incoming call to answer.

Examples:

```
set prompt "% "

ok,00
% control call answer 1

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

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

Refer to “Asynchronous Messages” on page 90 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,Audio,Normal,10.10.11.10,LifeSize
% control call del-part -V 2

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

Refer to “Asynchronous Messages” on page 90 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:

[-p {auto|h323|h323gw|sip}]	Specify the protocol to use to connect the new party to the call. The default is auto.
<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,Audio,Normal,10.10.11.10,LifeSize
% control call dial -V -p h323 10.10.11.11 -b 1024

ok
CS,1,2,Ringing,Audio,Normal,10.10.11.10,LifeSize
```

Dial using local directory entry:

```
% control call dial "local:john doe"

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

Refer to “Asynchronous Messages” on page 90 for a description of the response.

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,Audio,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,Audio,Normal,10.10.11.10,LifeSize
```

Refer to “Asynchronous Messages” on page 90 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|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,,
%
```

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.

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
```