LifeSize® Support TechNote

Controlling LifeSize Video Systems from the CLI

Use the LifeSize command line interface (CLI) to automate access and control of LifeSize video communications systems and LifeSize Phone with software release 4.8. For information about CLIs for other LifeSize products, refer to the documentation for that product.

Logging In

Log in to your LifeSize system through an SSH or Telnet connection over the network as follows:

1. Open a client, such as Cygwin or Putty, and enter the IP address of your LifeSize system.
2. Log in to the system with the following credentials:

   Username: **auto**
   Password: **lifesize**

You can also manage the LifeSize system by embedding CLI scripts in devices such as control panels. Connect through SSH or Telnet, or attach the device directly to the codec in one of the following ways:

- Attach the device through the USB port on the codec's back panel. Read more at **Connecting through Serial Over USB**.
- *LifeSize Room 220 Only*: Attach the device through an RS-232 serial port on the codec's back panel. Read more at **Connecting through RS-232 Serial Ports**.

Sample Commands

The following table lists sample CLI tasks:

Retrieve configuration information.	Retrieve the system version number.	`get system version`
	Retrieve the camera's brightness setting.	`get camera brightness`
Apply new preferences to the system configuration.	Adjust the system speaker volume.	`set volume speaker 60`
	Set the time (in seconds) before the user interface fades out during an active call.	`set timer fadeout 20`
Show the status of calls.	Show active calls.	`status call active`
	Show statistics for previous calls.	`status call history`
Control the system.	Add participants to a call.	`control call add-part 1 10.10.10.10`
	Emulate remote control functionality.	`control remote1 left left ok zin`

March 14, 2011

Command Syntax

Use the following command syntax:

<verb> <object> <target> [options]

<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.

The following example retrieves the system's Internet Protocol Version 4 network configuration. Including the `-V` option displays the output in columns.

```
get network ipv4 –V
Mode IP Address   Network Mask   Broadcast IP  Gateway IP    MAC Address         Hostname
dhcp 10.10.10.10  255.255.255.0  10.10.10.255  10.10.10.1    00:00:aa:00:9a:00   LifeSize
```

Note: When specifying an argument that includes a text string with a space in the string, enclose the text in double quotes, as follows: `"QRB Meeting"`

Command Verbs

`get`	Retrieves preference configuration information and static state information (such as version number) from the system.
`set`	Applies new preferences to the system configuration.
`control`	Initiates an action on the system.
`status`	Retrieves system status information.
`exit`	Exits the shell prior to the end of output. You can also exit the shell by entering the end-of-file character (typically `^D`).
`help`	Lists the verbs available in the shell, but not the individual targets for the verbs. Entering `help` *`verb`* lists the targets for the verb. The `help` command is available only in help mode.
`history`	Lists the saved history of commands up to 100 lines. To limit the number of lines to fewer than 100, type `history` followed by the number of lines. To execute a command from the history list, type `!x`, where `x` is the number of the command in the list.

Help Mode

Enable help mode to learn how to use the CLI. With help mode enabled, help is available for verbs, objects, and targets. In this context, a complete command is defined as a verb followed by an object and complete target specification. For targets that include two words, you must specify both words to complete the command. If you specify an incomplete command, all possible completions for that command root appear in alphabetical order. Additionally, the command processor allows abbreviations of command targets and verbs to simplify usage and to allow for more descriptive targets.

By default, the CLI starts in help mode. If disabled, enable help mode as follows:

```
set help-mode on
```

Help Mode Examples

Use the `help` command to list available verbs:

```
help
Possible verbs:
control
get
history
set
status

ok,00
```

Use the `help` command to list the available objects and targets for a verb:

```
help set
Possible completions:
set admin password
set audio line-in
set audio mics
set audio video-output
set call auto-answer
set volume dtmf
set volume ring-tone
   :
   :
ok,00
```

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

Use the `errors` argument to return a list of error message codes and their meanings:

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

Using the -h Option

With help mode enabled, specify the `-h` option to provide basic usage information for interactive users.

Specifying `-h` after the completed command returns complete command usage:

```
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
```

When you specify `-h` at any level other than that of a completed command, a list of possible completions appears, as in the following example:

```
get system -h
Possible completions:
get system admcontrol
get system autoreboot
get system branding
get system corporate-dir-access
get system date
get system do-not-disturb
   :
   :
ok,00
```

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

Disabling Help Mode

Use the following command to disable help:

```
set help-mode off
```

LifeSize recommends that you run automated scripts or programs with the help mode disabled to prevent ambiguity from command abbreviations.

Because each new CLI session enables help, disable help before running scripts.

Output Modes

The CLI supports terse (the default) and verbose output modes.

Asynchronous status messages always print 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.

Terse Mode

Terse mode is the default output mode and is designed to be easily parsed by shell scripts and automated programs. The format of the output is rows of comma-separated text. 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
```

Changing the Column Delimiter

The default column delimiter in terse mode is the comma character. Use the `-D` option to change the delimiter to a 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. If both are specified, `-D` is ignored.

For example:

```
get system model -D |
LifeSize|Room 220

ok|00
```

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 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 of 00 indicates success of the command. Any other value indicates an error.

Showing Column Headings

To show column headings from verbose mode while in terse mode, specify the `-?` option. 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,Room 220

ok,00
```

Verbose Mode

Verbose mode is enabled by specifying the `-V` option to a command. Enable verbose globally for the session with the `set verbose-mode on` command. When creating scripts, you can manually parse the script in verbose mode. Verbose output separates data by a minimum of two spaces between columns. Verbose mode is not intended to be parsed by automated scripts.

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 help you select columns in terse mode. For example:

```
get network ipv4 –V
Mode  IP Address    Network Mask   Broadcast IP   Gateway IP   MAC Address        Hostname
dhcp  10.10.10.10   255.255.255.0  10.10.10.255   10.10.10.1   00:00:aa:00:9a:00  LifeSize

ok

get unknown-target

error 09 Invalid Command
```

Command Line History and Recall

The CLI 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. 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.

The following example shows how to 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
```

The following example shows how to upload a background image using a here document:

```
set video background image << EOF
<base 64 encoded data stream>
EOF
```

Command Line Arguments

Invoke a single command by specifying the command on the command line; for example:

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

In this example, the return code of the SSH command is the resulting code from the single command executed.

Connecting through Serial over USB

Attach devices through a serial over USB adapter to LifeSize codecs that have USB ports, including LifeSize Room 220, LifeSize Team 220, and LifeSize Express 220. Use the `set serial port3` command to set up these ports according to the adapter manufacturer's instructions.

Connecting through RS-232 Serial Ports on LifeSize Room 220

LifeSize Room 220 requires use of a standard null modem cable for interaction through the serial connection. These devices can also attach to LifeSize systems through a serial to USB adapter.

To connect through the rear panel serial ports on LifeSize Room 220, follow these steps:

1. Plug one end of the null modem cable into the RS-232 serial port.
2. Plug the other end of the cable into your PC serial port. Take note of which port you choose.
3. SSH into the system and set the serial port baud rate with the `set serial port` command:
   `set serial port`*`N`* `-b` *`baudRate`*
4. Depending on your operating system, do one of the following:
   - On Windows, start HyperTerminal:
     Start>All Programs>Accessories>Communications>HyperTerminal

     Configure the HyperTerminal for the serial port you selected on the PC, and set to the baud rate you specified in step 3. Press **Return** in HyperTerminal until `ok,00` appears.
   - On Linux, start Minicom; configure it for the serial port you selected on the PC; and set to the baud rate you specified in step 3. Press **Return** in Minicom until `ok,00` appears.

For more information about configuring the serial port, including USB ports, get help on the `serial` target.