REFERENCE GUIDE

3.2.0 | August 2020 | 3725-86004-004A

Poly G7500

Command-Line API

GETTING HELP

For more information about installing, configuring, and administering Poly/Polycom products or services, go to the Poly Online Support Center.

Plantronics, Inc. (Poly – formerly Plantronics and Polycom)

345 Encinal Street

Santa Cruz, California

95060

Contents

Before You Begin

This guide helps audio/video (AV) integrators, developers, and administrators use the Poly system command-line API.

Audience, Purpose, and Required Skills

This guide is written for a technical audience. You must be familiar with the following concepts before beginning:

- Current telecommunications practices, protocols, and principles
- Telecommunication basics, video teleconferencing, and voice or data equipment
- Open SIP networks and VoIP endpoint environments

Related Poly and Partner Resources

See the following sites for information related to this release.

- The Poly Online Support Center is the entry point to online product, service, and solution support information including Video Tutorials, Documents & Software, Knowledge Base, Community Discussions, Poly University, and additional services.
- The Polycom Document Library provides support documentation for active products, services, and solutions. The documentation displays in responsive HTML5 format so that you can easily access and view installation, configuration, or administration content from any online device
- The Polycom Community provides access to the latest developer and support information. Create an account to access Poly support personnel and participate in developer and support forums. You can find the latest information on hardware, software, and partner solutions topics, share ideas, and solve problems with your colleagues
- The Polycom Partner Network are industry leaders who natively integrate the Poly standards-based RealPresence Platform with their customers' current UC infrastructures, making it easy for you to communicate face-to-face with the applications and devices you use every day.
- The Polycom Collaboration Services help your business succeed and get the most out of your investment through the benefits of collaboration.

Additional Command-Line API Resources

The following resources are available to help you use the command-line API.

Contacting Technical Support

Go to the Poly Online Support Center to contact technical support.

Requesting Feature Enhancements

Contact your Poly sales engineer to submit a feature request.

Video Test Numbers

See the Video Test Numbers page to test with various Poly systems worldwide.

Getting Started

The command-line API lets you externally configure and control a G7500 system. You can access the API with the following connections:

- SSH
- Telnet
- Serial

Enable SSH Access

Use SSH on port 22 if you want encrypted access to the G7500 system command-line API.

1. In the web interface, go to **Security > Access**.
2. Select the **Enable Legacy API Over SSH** check box if it's cleared.

Access the Command-Line API Over SSH

You can use your G7500 system local administrator credentials or external authentication to start an SSH session. For information on configuring system accounts, see the system *Administrator Guide* at the Poly Online Support Center.

1. Start an SSH session using the system IP address (for example, `ssh 10.xxx.xx.xx`).
2. When prompted, enter the system credentials.

Enable Telnet Access

Use port 24 or 23 to access the G7500 system command-line API using telnet.

1. In the web interface, go to **Security > Access**.
2. Select the **Enable Telnet Access** check box.
3. Choose an **API Port** for telnet connections: **24** (default) or **23**.

Access the Command-Line API Over Telnet

From a device connected to the same LAN as your G7500 system, you can access the command-line API using telnet (port 23 or 24 depending on how you've configured the system).

1. Start a telnet session using the system IP address and port number configured for the **API Port** setting (for example, `telnet 10.xxx.xx.xx 24`).
2. If prompted, log in with the system's local administrator password.

Enable Serial Port Access

To use the command-line API with an RS-232 interface, you must connect and configure the control system and G7500 system for serial communication.

1. In the system web interface, go to **General Settings > Serial Port** and set **RS-232 Mode** to **Control**.
2. Configure a **Baud Rate** (for example, **19200**).
3. Configure the same baud rate for your control system.
4. For the G7500 system **Login Mode** setting, choose one of the following:
   - **Admin password only**: (Default) Requires the administrator password (if you set one) when the control system connects.
   - **Username/Password**: Requires the user name and administrator password (if you set one) when the control system connects.
   - **None**: The system doesn't require a user name or password when the control system connects.

Access the Command-Line API Using a Serial Connection

Once you've configured your G7500 system and control system for serial communication, you can start an command-line API session.

1. Connect an RS-232 cable from your control system to the system's serial port.
2. From the control system, start a serial session using PuTTY or a similar utility.
3. If prompted, log in with the system's local administrator password.

Command-Line API Reference

The G7500 system command-line API is organized in alphabetical order.

Note: While every attempt is made to ensure that the expected results of executing API commands are accurate, Polycom cannot be responsible for system behaviors and control actions that are not explicitly documented.

About the Command-Line API

Use the following guidelines to help you understand the command-line API.

Definitions

The following definitions help you read the command-line API reference.

Terms	Description
Command Description	Brief statement about the purpose of the command.
Syntax	Format required to execute the command.
Parameter	A list of parameters defined for the command.
Description (parameter)	A description of each parameter that is defined for the command.
Feedback Examples	Examples of expected results when a command and variant are executed.
Limitations	Important notes about support for the command on the system.
Comments	Important notes about the command.

Syntax Conventions

The following conventions are used for the API command descriptions. Commands are case sensitive.

Convention	Meaning
`<param1|param2|param3>`	Multiple valid parameters are enclosed in angle brackets and separated by the pipe (“|”) character.
`[param]` `["param"]`	Optional parameters are enclosed in square brackets. Quotation marks indicate string input.
`{a..z}`	A range of possible alphanumeric values is enclosed in braces.
`"x"`	Quotation marks indicate string input. You don’t need to enclose the value in quotes unless it contains a space.

Although the command-line API parser may accept the minimum number of characters in a command that makes it unique, you should always use the full command string.

Command Availability

API command availability depends on the connected equipment, security settings, installed software version, among other factors. If a particular command isn’t supported, the command returns feedback such as, “`command is not available in current system configuration`”.

If a setting is configured by a provisioning service, the command may return, “`this setting is controlled by a provisioning service and cannot be changed. For more information about provisioned settings, refer to your provisioning service administrator.`”

Commands that are not listed are not supported by Polycom. Commands might change or be removed at any time. Polycom discourages integrators from using unpublished commands.

Command Response Syntax

When you send a command, the system returns responses using the syntax described in the following sections, where <CR> indicates a carriage return and <LF> indicates a line feed.

Unregistered for Notifications

When your system is not registered to receive notifications and you send an API command, a single API acknowledgement is returned.

For example:

`camera near 1<CR>` API command
returns
`camera near 1<CR><LF>` API acknowledgement

In the previous example, the command was sent with a carriage return <CR>.

The API expects a carriage return <CR> and the standard end of line characters carriage return/line feed <CR><LF>. All API responses end in carriage return/line feed <CR><LF>.

Registered for Notifications

Registering for notifications adds extra line responses. The number of additional lines depends on the specific registration. In the following example, the response shows an API acknowledgement and registration response returned:

`camera near 1 <CR>` API command
returns
`camera near 1<CR><LF>` API acknowledgement
`notification:vidsourcechange:near:1:Main:people<CR><LF>`
API registration response

When your system is registered for notifications, always use the API registration response for status.

Commands that Restart the System Without a Prompt

The following API commands restart your system without notification:

- `reboot now`
- `resetsettings`

Additional Tips

Refer to the following information when using the command-line API:

- The system does not provide flow control.
- If you lose an API connection (e.g., the system restarts), you must re-establish it.
- The API processes one command at a time.
- Polycom does not recommend sending multiple commands simultaneously without a pause or delay between each.
- For commands with a single action and response: A delay of 200 milliseconds between commands is usually sufficient. Examples of these commands include the commands for switching cameras (`camera near 1`), sending content (`vcbutton play`), and checking the status of audio mute (`mute near get`).
- For commands with a single action and more extensive response: The time it takes to receive the response (and in effect the time between commands) may be longer than 200 milliseconds. The response length, which can vary, determines the time required to receive the response. Examples include the commands for retrieving directory information (such as `addrbook all`) and system session information (such as `whoami`).
- When developing your application, always allow enough time for a command response before sending another command.
- Polycom does not recommend sending commands while a call is being established.
- The API provides feedback status in two ways: registrations or polling.
- Send registration and notification API commands only once. Registrations are written to flash memory and retained when the system restarts.
- Polycom recommends putting registrations in the initialization or startup of Crestron and AMX systems.

- Registrations are recommended over polling since they provide status updates without having to query for changes.
- Never poll for registrations.
- Registrations are specific to the port from which they are registered. If you register for notifications from com port 1, registration will not be sent to com port 2 or telnet port 24.

addrbook

Returns local directory (address book) entries.

Syntax

Commands for local directory

```
addrbook all
addrbook batch {0..59}
addrbook batch search "pattern" "count"
addrbook batch define "start_no" "stop_no"
addrbook letter {a..z}
addrbook range "start_no" "stop_no"
```

Commands for groups

```
addrbook names <all|video> [<range_start>] [<range_end>]
addrbook names <all|video> size
addrbook names search "search_pattern" <all|video>
                      [<range_start>] [<range_end>]
addrbook names search "search_pattern" <all|video size
addrbook group "group_name" [<range_start>] [<range_end>]
addrbook group "group_name" size
addrbook address "sys_name" ["sys_label"]
```

Parameter	Description
`all`	Returns all the entries in the local directory.
`batch`	Returns a batch of 10 local directory entries. Requires a batch number, which must be an integer in the range `{0..59}`.
`search`	Specifies a batch search.
`"pattern"`	Specifies a pattern to match for the batch search.
`"count"`	Specifies the number of entries to list that match the pattern.
`define`	Returns a batch of entries in the range defined by "`start_no`" to "`stop_no`."

`letter`	Returns entries beginning with the letter specified from the range `{a..z}`. Requires one or two alphanumeric characters. Valid characters are:
	`- _ / ; @ , . \`
	`0` through `9`
	`a` through `z`

`range`	Returns local directory entries numbered `"start_no"` through `"stop_no"`. Requires two integers.
`"start_no"`	Specifies the beginning of the range of entries to return.

Parameter	Description
"stop_no"	Specifies the end of the range of entries to return.

names	Returns a list of system names in the local address book. Also returns the system type: video, multicodec, phone, or multisite. A multicodec system will appear as a single row.
	The response is in the following format:
	`addrbook names {0..n}. name:"sys_name"`
	`sys_label:"sys_label"`
	`type: <video|multicodec|phone|group>`
	`...`
	`addrbook names <all|video|phone> done`

<all|video>	Specifies the type of entries to return. `video` returns entries that have video addresses. `all` returns entries with video numbers or phone numbers or both.

size	Returns the size of the result set that will be returned by the command. The size parameter can be used with the names and the names search commands.
	The response is in the following format:
	`addrbook names <all|video|phone> size {0..n}`
	`addrbook names search "search_pattern"`
	`<all|video|phone> size {0..n}`

range_start	For the names, names search, and group commands, specifies the beginning of the range of entries to return.
range_end	For the names, names search, and group commands, specifies the end of the range of entries to return. If a `range_start` is specified without a `range_end`, then the single `range_start` entry will be returned. If `range_end` is -1, all entries starting with `range_start` will be returned.

search	Returns a list local directory names that match the search criteria.
	The response is similar to the `names` command described above:
	`addrbook search {0..n}. name:"sys_name"`
	`sys_label:"sys_label"`
	`type: <video|multicodec|phone|group>`
	`...`
	`addrbook names search "search_pattern"`
	`<all|video|phone> done`

search_pattern	Specifies the string pattern for which to search. Wildcard characters are not supported. The search string is used to match the beginning of any of the attributes listed using descriptions for the `names` and `search` parameters. For example, the search string "Jo" would match any name that begins with Jo, such as John or Jones. The search is not case sensitive.

group	Returns a list of the names of all the sites included in a local directory group in this format:
	`addrbook group {0..n}. name:"site_sys_name"`
	`sys_label:"site_sys_label"`
	`...`
	`addrbook group "group_name" [range] done`
	`addrbook group size <num_entries>`

group_name	A local address book group name.

Parameter	Description
address	Obtains the address information for a specified entry. If the entry is an ITP system, the results will include the addresses for all codecs. If codecs support multiple protocols, the different addresses will be returned on separate lines. This command is not supported for multisite entries.
sys_name	The friendly name for an address book entry. It is the name of the person or the room. It is surrounded by quotes if it contains spaces.
sys_label	If a person/room has more than one system, the result set will include a row for each system. If those systems are of the same type the client will consider that entry to be a telepresence system with multiple codecs rather than separate systems. If the systems are of different types, then this sys_label attribute will be included to differentiate the systems.
type	The type of local address book entry. Possible values are: video, multicodec, phone, group
site_sys_name	The name of a site in a group. It is surrounded by quotes if it contains spaces
site_sys_label	The label associated with a site name in a local group. It is surrounded by quotes if it contains spaces.
codec:<1..4>	If the entry is a telepresence system, each codec will include a codec number attribute.
h323_spd	The preferred speed for an H.323 call to this entry. If no speed is associated with the entry, then the value of the configuration variable globaladdrmaxh323 is returned. The default is 384.
h323_num	H.323 address or alias.
h323_ext	H.323 extension or E.164 number.
sip_spd	The preferred speed for a SIP call to this entry. If no speed is associated with the entry, then this is the same as the h323_spd.
sip_num	IP address.

Feedback Examples

- addrbook all
  returns
  addrbook 0. "Polycom Demo 1" h323_spd:384 h323_num:192.168.1.101 h323_ext:7878
  addrbook 1. "Polycom Demo 2" sip_spd:384 sip_num:polycomg7500@polycom.com
  addrbook 2. "Polycom Demo 3" phone_num:1.512.5121212
  (and so on, until all entries in the local directory are listed, then:)
  addrbook all done
- addrbook batch 0
  returns
  addrbook 0. "Polycom Demo 1" h323_spd:384 h323_num:192.168.1.101 h323_ext:7878
  addrbook 1. "Polycom Demo 2" sip_spd:384 sip_num:polycomg7500@polycom.com
  addrbook 2. "Polycom Demo 3" phone_num:1.512.5121212
  (and so on, through the last entry in the batch of 10 directory entries, such as:)

```
addrbook 9. "Polycom Demo 20" h323_spd:384 h323_num:192.168.1.120
h323_ext:
addrbook batch 0 done
```

- `addrbook batch define 0 2`
  returns
  ```
  addrbook 0. "Polycom  Demo 1" h323_spd:384 h323_num:192.168.1.101
  h323_ext:7878
  addrbook 1. "Polycom  Demo 2" sip_spd:384 sip_num:polycom@polycom.com
  addrbook batch define 0 2 done
  ```
- `addrbook names all size`
  returns
  ```
  addrbook names all size 21
  ```
- `addrbook names all size 21`
  returns
  ```
  addrbook names all size 21
  addrbook names 0. name:"Eng RPX" sys_label:"" type:multicodec
  addrbook names 1. name:"Doe" sys_label:"" type:video
  addrbook names 2. name:"Gen Group" sys_label:"" type:group
  addrbook names 3. name:"John Doe" sys_label:"" type:video
  addrbook names 4. name:"John Doe" sys_label:"" type:video
  addrbook names 5. name:"Lab TPX" sys_label:"" type:video
  addrbook names 6. name:"Minuteman RPX" sys_label:"" type:multicodec
  addrbook names 7. name:"Monday Staff Mtg" sys_label:"" type:group
  addrbook names 8. name:"Polycom Austin Stereo" sys_label:"" type:video
  addrbook names 9. name:"Polycom Austin HD" sys_label:"" type:video
  addrbook names all 0 9 done
  ```
- `addrbook names all`
  returns
  ```
  addrbook names 0. name:"Eng RPX" sys_label:"HDX" type:multicodec
  addrbook names 1. name:"Doe" sys_label:"" type:video
  addrbook names 2. name:"Gen Group" sys_label:"" type:group
  addrbook names 3. name:"John Doe" sys_label:"" type:video
  addrbook names 4. name:"John Doe" sys_label:"" type:video
  addrbook names 5. name:"Lab TPX" sys_label:"" type:video
  addrbook names 6. name:"Minuteman RPX" sys_label:"" type: multicodec
  addrbook names 7. name:"Monday Staff Mtg" sys_label:"" type:group
  addrbook names 8. name:"Polycom Austin Stereo" sys_label:"" type:video
  addrbook names 9. name:"Polycom Austin HD" sys_label:"" type:video
  addrbook names 10. name:"Polycom Austin USA IP" sys_label:"" type:video
  addrbook names 11. name:"Polycom Japan" sys_label:"" type:video
  addrbook names 12. name:"Scott CMAD IP" sys_label:"" type:video
  addrbook names 13. name:"Scott Phone" sys_label:"" type:phone
  addrbook names 14. name:"Scott PVX" sys_label:"" type:video
  addrbook names 15. name:"Scott Quasar 19" sys_label:"" type:video
  addrbook names 16. name:"SQA " sys_label:"" type:video
  addrbook names 17. name:"John Doe" sys_label:"" type:video
  addrbook names 18. name:"Test System 1" sys_label:"" type:video
  addrbook names 19. name:"Test System 2A" sys_label:"" type:video
  addrbook names 20. name:"Test System 2B" sys_label:"" type:video
  addrbook names all done
  ```

- `addrbook names search "p" all`
  returns
  ```
  addrbook search 0. name:"Polycom Austin HD" sys_label:"" type:video
  addrbook search 1. name:"Polycom Austin Stereo" sys_label:"" type:video
  addrbook search 2. name:"Polycom Austin USA IP" sys_label:"" type:video
  addrbook search 3. name:"Polycom Japan" sys_label:"" type:video
  addrbook search 4. name:"Scott Phone" sys_label:"" type:phone
  addrbook search 5. name:"Scott Group Series" sys_label:"" type:video
  addrbook search search p all done
  ```
- `addrbook names search "p" all 0 2`
  returns
  ```
  addrbook search 0. name:"Polycom Austin HD" sys_label:"" type:video
  addrbook search 1. name:"Polycom Austin Stereo" sys_label:"" type:video
  addrbook search 2. name:"Polycom Austin USA IP" sys_label:"" type:video
  addrbook search search p all 0 2 done
  ```
- `addrbook group "Monday Staff Mtg"`
  returns
  ```
  addrbook group 0. name:"Eng RPX" sys_label:"HDX"
  addrbook group 1. name:"John Doe" sys_label:""
  addrbook group 2. name:"John Doe" sys_label:""
  addrbook group 3. name:"TPW" sys_label:"HDX"
  addrbook group "Monday Staff Mtg" done
  ```
- `addrbook address "John Doe"`
  return
  ```
  addrbook address 0. name:"John Doe" sys_label:"" codec:1
                      h323_spd:384 h323_num:172.25.137.101 h323_ext:
  addrbook address name:"John Doe" sys_label:"" done
  ```

Limitations

None

Comments

This command is deprecated. Polycom recommends using `localdir` instead.

advnetstats

Gets advanced network statistics for a call connection.

Syntax

```
advnetstats [{0..n}]
```

Parameter	Description
`{0..n}`	Specifies a connection in a call, where `n` is the maximum number of connections supported by the system. `0` is call #1, `1` is call #2, `2` is call #3, and so on. Select a number from this range to specify a remote site call for which you want to obtain advanced network statistics. Omit this parameter when retrieving statistics for a point-to-point call.

Feedback Examples

- `advnetstats 0`
  returns

```
call:0 tar:96 K rar:96 K tvr:224 K rvr:416 K
tvru:219 K rvru:154 K tvfr:29 rvfr:26 vfe:0
tapl:0 rapl:0 taj:6 ms raj:5 ms tvpl:0 rvpl:0
tvj:6 ms rvj:11 ms
dc:Disabled rsid:Sams RP700 ccaps:9
tcr:0 rcr:128 K tcru:0 rcru:128k
tcfr:0 rcfr:64 K tcpl:0 rcpl:0
```

  where:

```
tar = transmit audio rate
rar = receive audio rate
tvr = transmit video rate
rvr = receive video rate
tvru = transmit video rate used
rvru = receive video rate used
tvfr = transmit video frame rate
rvfr = receive video frame rate
vfe = video FEC errors
tapl = transmit audio packet loss (H.323 calls only)
tlsdp = transmit LSD protocol (H.320 calls only)
rapl = receive audio packet loss (H.323 calls only)
rlsdp = receive LSD protocol (H.320 calls only)
taj = transmit audio jitter (H.323 calls only)
tlsdr = transmit LSD rate (H.320 calls only)
raj = receive audio jitter (H.323 calls only)
rlsd = receive LSD rate (H.320 calls only)
tvpl = transmit video packet loss (H.323 calls only)
tmlpp = transmit MLP protocol (H.320 calls only)
rvpl = receive video packet loss (H.323 calls only)
rmlpp = receive MLP protocol (H.320 calls only)
tvj = transmit video jitter (H.323 calls only)
tmlpr = transmit MLP rate (H.320 calls only)
rvj = receive video jitter (H.323 calls only)
rmlpr = receive MLP rate (H.320 calls only)
```

```
dc = encryption information
rsid = remote system id
ccaps = content capability, where possible responses include “E”
(enterprise dual streams), “N” (none), and “P” (content over the people
stream)
tcr = transmit content rate
rcr = receive content rate
tcru = transmit content rate used
rcru = receive content rate used
tcfr = transmit content frame rate
rcfr = receive content frame rate
tcpl = transmit content packet loss
rcpl = receive content packet loss
```

Limitations

None

Comments

None

See Also

To return network statistics for a call, use the nearloop command.

all register

Registers for most commonly-used user registration events.

Syntax

```
all register
```

Additional Restrictions

None

Feedback Examples

- `all register`
  returns
  ```
  callstate registered
  camera registered
  linestate registered
  mute registered
  preset registered
  vcbutton registered
  volume registered
  ```

Comments

Registers changes to any of the following types of parameters:

- Current near- or far-site source
- State of privacy
- Current volume level
- Active camera presets
- Call status
- IP connection to codec
- System information

This command is useful when two control systems are used simultaneously, such as the web and API commands. The system maintains the registration changes through restarts.

To register for events not included in this feedback, refer to the specific registration command.

This is a one-time registration command that is retained in flash memory. Sending the command a second time results in the following feedback response:

- ```
  info: event/notification already active:callstate
  info: event/notification already active:camera

  info: event/notification already active:linestate
  info: event/notification already active:mute
  ```
  ```

```
info: event/notification already active:preset
info: event/notification already active:vcbutton
info: event/notification already active:volume
```

The `all register` command does not return local camera movements if the camera is moved using the remote control.

Use the notify command for camera notifications.

Limitations

None

all unregister

Simultaneously unregisters all registered user feedback so that the API no longer reports changes to the parameters.

Syntax

```
all unregister
```

Additional Restrictions

None

Feedback Examples

- `all unregister`
  returns
  ```
  callstate unregistered
  camera unregistered
  linestate unregistered
  mute unregistered
  preset unregistered
  vcbutton unregistered
  volume unregistered
  ```

Limitations

None

Comments

The following types of parameters are unregistered:

- Current near-site or far-site source
- State of privacy
- Current volume level
- Active camera presets
- Status of point-to-point or multipoint calls
- IP connection to codec
- System information

amxdd

Gets or sets the AMX Device Discovery beacon.

Syntax

```
amxdd get
amxdd <on|off>
```

Parameter	Description
`get`	Returns the current setting.
`on`	Turns on the AMX Device Discovery beacon.
`off`	Turns off the AMX Device Discovery beacon.

Feedback Examples

- `amxdd get`
  returns
  `amxdd off`
- `amxdd on`
  returns
  `amxdd on`

Limitations

None

Comments

The default setting for this signal is `off`.

answer

Answers incoming video calls.

Syntax

```
answer <video>
```

Parameter	Description
`video`	Answers incoming video calls when Auto Answer Point-to-Point Video or Auto Answer Multipoint Video is set to No.

Feedback Examples

- `answer video`
  returns
  `answer incoming video call failed`
- `answer video`
  returns
  `answer incoming video call passed`

Limitations

None

Comments

None

apiport

Gets or sets the command-line API telnet port.

Syntax

```
apiport get
apiport <23|24>
```

Parameter	Description
`get`	Returns the configured command-line API port.
`23`	Sets the command-line API telnet port to 23.
`24`	Sets the command-line API telnet port to 24. Default setting.

Feedback Examples

- `apiport get`
  returns
  `apiport 24`
- `apiport 23`
  returns
  `apiport 23`

Limitations

None

Comments

After sending the command to change the port, you must exit the current session and reconnect on the new port.

audio3p5inputfaronly

Gets or sets the preference for 3.5mm audio input from the system's 3.5mm audio port.

Syntax

```
audio3p5inputfaronly <get|enable|disable>
```

Parameter	Description
`get`	Returns the current settings.
`enable`	3.5 mm audio input is sent only to the far site.
`disable`	3.5 mm audio input is sent to both far and near sites.

Feedback Examples

- `audio3p5inputfaronly get`
  returns
  `audio3p5inputfaronly enable`
- `audio3p5inputfaronly disable`
  returns
  `audio3p5inputfaronly disable`

Limitations

None

Comments

None

audiotransmitlevel

Sets or gets the audio volume transmitted to the far site. Also register for notifications of audio transmit level changes.

Syntax

```
audiotransmitlevel <get|up|down|register|unregister>
audiotransmitlevel set {-6..18}
```

Parameter	Description
`get`	Returns the current setting.
`up`	Sets the volume 1 decibel higher than the current setting.
`down`	Sets the volume 1 decibel lower than the current setting.
`register`	Registers to receive notification when audio transmit level changes.
`unregister`	Unregisters to receive notification when audio transmit level changes.
`set`	Sets the volume to the specified dB level. Valid values are: `{-6..18}`.

Feedback Examples

- `audiotransmitlevel set 2`
  returns
  `audiotransmitlevel 2`
- `audiotransmitlevel get`
  returns
  `audiotransmitlevel 2`
- `audiotransmitlevel up`
  returns
  `audiotransmitlevel 3`
- `audiotransmitlevel down`
  returns
  `audiotransmitlevel 2`
- `audiotransmitlevel register`
  returns
  `audiotransmitlevel registered`
- `audiotransmitlevel unregister`
  returns
  `audiotransmitlevel unregistered`

Limitations

None

Comments

None

autoanswer

Sets how the system handles incoming point-to-point calls.

Syntax

```
autoanswer <get|yes|no|donotdisturb>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Incoming calls are connected automatically. This is the default setting.
`no`	Prompts the user to answer incoming calls.
`donotdisturb`	Notifies the user of incoming calls but does not connect. The far side receives a rejected call notification.

Feedback Examples

- `autoanswer yes`
  returns
  `autoanswer yes`
- `autoanswer no`
  returns
  `autoanswer no`
- `autoanswer get`
  returns
  `autoanswer no`
- `autoanswer donotdisturb`
  returns
  `autoanswer donotdisturb`

Limitations

None

Comments

If `autoanswer` is set to `no` or `donotdisturb`, you must rely on API session notifications to answer inbound calls.

calendardiscovery

Gets the Microsoft Exchange Server address based on the associated email address or registered SIP server address configured for the system.

Syntax

```
calendardiscovery get
calendardiscovery emaildomain
calendardiscovery sipdomain
```

Parameter	Description
`get`	Gets the Microsoft Exchange Server address that the system is using to register with the calendaring service.
`emaildomain`	Gets the Microsoft Exchange Server address based on an email address.
`sipdomain`	Gets the Microsoft Exchange Server address based on a SIP server address.

Feedback Examples

- `calendardiscovery sipdomain get`
  returns
  `calendardiscovery 192.168.44.168`
- `calendardiscovery emaildomain get`
  returns
  `calendardiscovery mail.exchangeserver.local.com`
- `calendardiscovery get`
  returns
  `calendardiscovery not available (if not configured or not found)`
- `calendardiscovery emaildomain get`
  returns
  `calendardiscovery not available (if not configured or not found)`
- `calendardiscovery get`
  returns
  `error: command needs more parameters to execute successfully`
- `calendardiscovery`
  returns
  `error: command needs more parameters to execute successfully`

Limitations

None

Comments

None

calendardomain

Gets or sets the domain used by the calendaring service to log in to the Microsoft Exchange Server.

Syntax

```
calendardomain get
calendardomain set "domain"
```

Parameter	Description
`get`	Returns the domain used by the calendaring service.
`set`	Sets the domain used by the calendaring service.
`"domain"`	Specifies the domain for the calendaring service when using the `set` command.

Feedback Examples

- `calendardomain get`
  returns
  `calendardomain smithfield`
- `calendardomain set fairview`
  returns
  `calendardomain fairview`

Limitations

None

Comments

None

calendarmeetings

Retrieves scheduled meetings within a provided time or for a meeting ID.

Syntax

```
calendarmeetings list "starttime" ["endtime"]
calendarmeetings info "meetingid"
```

Parameter	Description
`list`	Returns the meeting id or ids for meetings that start at or after the specified start time and end time.
`"starttime"`	The start time of meetings to be retrieved. The start time can be entered in one of the following formats: • YYYY-MM-DD:HH:MM • today:HH:MM • today • tomorrow:HH:MM • tomorrow The times are interpreted to be local times in the time zone the system was configured for.
`"endtime"`	The end time of meetings to be retrieved. This parameter can be given in the following format. • YYYY-MM-DD:HH:MM • today:HH:MM • today • tomorrow:HH:MM • tomorrow The times are interpreted to be local times in the time zone the system was configured for.
`info`	Retrieves meeting details for scheduled meetings when the system is registered with the calendaring service. Returns information such as the location, subject and organizer of the meeting.
`"meetingid"`	The ID of the meeting for which you want to find details.

Feedback Examples

- `calendarmeetings list tomorrow`
  returns
  ```
  calendarmeetings list begin
  meeting|AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMne2/ndgARgAAAADr9
  GlhsSjWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAADe7hJleQIOS7j2mzRJxk
  LKAAADI/F8BAAA|2010-03-30:08:30|2010-03-30:09:00|Discuss Budget
  meeting|AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMne2/ndgARgAAAADr9
  GlhsSjWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAADe7hJleQIOS7j2mzRJxk
  ```

```
LKAAAA/9PhAAAQ|2010-03-30:09:00|2010-03-30:09:30|Program Review
meeting|AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMne2/ndgARgAAAADr9
GlhsSjWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAABZ29fOU0S5Q6xzZ1lzDD
NnAABFQAQ3AAAQ|2010-03-30:10:00|2010-03-30:11:00|Customer Care Commitment
Meeting
calendarmeetings list end
```

- `calendarmeetings list 2010-03-30:08:00 2010-04-01:17:00`
  returns
  ```
  calendarmeetings list begin
  meeting|AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMne2/ndgARgAAAADr9
  GlhsSjWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAADe7hJleQIOS7j2mzRJxk
  LKAAADI/G8AAAQ|2010-03-30:08:30|2010-03-30:09:00|Bug Scrub
  meeting|AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMne2/ndgARgAAAADr9
  GlhsSjWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAABZ29fOU0S5Q6xzZ1lzDD
  NnAABFQARCAAAQ|2010-03-30:11:30|2010-03-30:12:30|groupseries/IP7000/Confe
  rence Coordination
  meeting|AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMne2/ndgARgAAAADr9
  GlhsSjWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAABZ29fOU0S5Q6xzZ1lzDD
  NnAABFQAQ3AAAQ|2010-04-01:16:30|2010-04-01:17:00|Customer Care Commitment
  Meeting
  calendarmeetings list end
  ```

- `calendarmeetings info AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMne2/ndgARgAAAADr9GlhsSjWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAADe7hJleQIOS7j2mzRJxkLKAAADI/G8AAAQ`
  returns
  ```
  calendarmeetings info start
  id|AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMne2/ndgARgAAAADr9GlhsS
  jWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAADe7hJleQIOS7j2mzRJxkLKAAA
  DI/G8AAAQ
  2010-03-30:08:30|2010-03-30:09:00|dialable|public
  organizer|Russell Bell
  location|Russell's Meeting Room - IP Video Number: 123456 (if registered
  to corp GK); 888-123-4567/978-123-4567 with passcode: #760900
  subject|Bug Scrub
  dialingnumber|video|733397@vsgwstdma01.r13.vsg.local2|sip
  dialingnumber|video|733397|h323
  dialingnumber|audio|48527
  meetingpassword|none
  attendee|Russell Bell
  attendee|Rebecca Sharp
  calendarmeetings info end
  ```

- `calendarmeetings info AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMn4AUcVgARgAAAADr9GlhsSjWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAADe7hJleQIOS7j2mzRJxkLKAAAA30GwAAAQ`
  returns
  ```
  calendarmeetings info start
  id|AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMn4AUcVgARgAAAADr9GlhsS
  jWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAADe7hJleQIOS7j2mzRJxkLKAAA
  A30GwAAAQ
  2010-04-01:10:30|2010-04-01:11:00|nondialable|private
  organizer|Rebecca Sharp
  ```

```
location|Red River conference room
subject|Escalations Review
attendee|Roslin Adam
attendee|Conference.Main
attendee|Claudia Nevarez
calendarmeetings info end
```

Limitations

None

Comments

If the meeting's end time is more than 31 days from the meeting's start time, the response is shortened to `starttime+31days`, and meetings that start in that time span are returned.

If an API client is logged in with user-level credentials and if the system is configured to hide private meeting information on the web interface, the API hides the information from the API client and shows the subject of the meeting as "Private Meeting"; for example:

```
calendarmeetings list begin
meeting|AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMn4AUcVgARgAAAADr9GlhsS
jWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAADe7hJleQIOS7j2mzRJxkLKAAAA30Gw
AAAQ|2009-09-25:08:30|2009-09-25:09:15|private meeting
calendarmeetings list end
```

If a system is configured to provide private meeting information on the web interface, the API provides the same information to the API client; for example:

```
calendarmeetings list begin
meeting|AAAZAGV4Y2H1C2VYMDFACJEZLNZZZY5SB2NHBDIARGAAAAAAKQKC8WW3CUWGCPM+AP66WQ
CASOLXUYMOMEKYBQJJ1Z0MBWASDQANHQAASOLXUYMOMEKYBQJJ1Z0MBWASDQASVGAA|2009-09-25:0
8:30|2009-09-25:09:15| Demo
calendarmeetings list end
```

If the API client is logged in with admin-level credentials, the API provides private meeting information to the API client, regardless of the configuration for displaying private meeting information; for example:

```
calendarmeetings list begin
meeting|AAAZAGV4Y2H1C2VYMDFACJEZLNZZZY5SB2NHBDIARGAAAAAAKQKC8WW3CUWGCPM+AP66WQ
CASOLXUYMOMEKYBQJJ1Z0MBWASDQANHQAASOLXUYMOMEKYBQJJ1Z0MBWASDQASVGAA|2009-09-25:0
8:30|2009-09-25:09:15|Release plan
meeting|AAAZAGV4Y2H1C2VYMDFACJEZLNZZZY5SB2NHBDIARGAAAAAAKQKC8WW3CUWGCPM+AP66WQ
CASOLXUYMOMEKYBQJJ1Z0MBWASDQANHQAASOLXUYMOMEKYBQJJ1Z0MBWASDQASVGAA|2009-09-23:1
1:00|2009-09-23:11:45|Product roadmap for 2010
calendarmeetings list end
```

The calendaring service must be registered with Microsoft Exchange Server for the `calendarmeetings` command to work successfully. If the calendar credentials are invalid, the server address is not valid, or the configured user credentials don't have access permissions to the resource mailbox calendar, the service will fail to register.

This command has multiline output.

The following characters in the meeting subject will not be displayed:

- | (vertical bar)
- CR (carriage return)
- LF (line feed)

See Also

To enable or disable the calendaring service, use the calendarregisterwithserver command.

To configure the Microsoft Exchange Server address that is used by this service, use the calendarserver command.

calendarpassword

Sets the password used by the calendaring service to log in to the Microsoft Exchange Server.

Syntax

```
calendarpassword "password"
```

Parameter	Description
"password"	The password used by the calendaring service to log in to the Microsoft Exchange Server.

Feedback Examples

- `calendarpassword Dsca1end@r`
  returns
  `calendarpassword Dsca1end@r`

Limitations

None

Comments

The password is case-sensitive and can contain a maximum of 15 characters. Use strong passwords that combine uppercase and lowercase letters, numbers, and symbols.

calendarplaytone

Gets or sets the reminder alert tone that plays with the meeting reminder when the system is registered with the calendaring service.

Syntax

```
calendarplaytone get
calendarplaytone <yes|no>
```

Parameter	Description
`get`	Gets the current setting for the alert tone.
`yes`	Enables the alert tone.
`no`	Disables the alert tone.

Feedback Examples

- `calendarplaytone get`
  returns
  `calendarplaytone yes`
- `calendarplaytone yes`
  returns
  `calendarplaytone yes`
- `calendarplaytone no`
  returns
  `calendarplaytone no`

Limitations

None

Comments

None

calendarprotocol

Gets or sets the protocol to use when connecting to the calendaring service.

Syntax

```
calendarprotocol <get|auto|tls>
```

Parameter	Description
`get`	Gets the current setting.
`auto`	Sets the connection protocol to automatic discovery.
`tls`	Sets the connection protocol to TLS.

Feedback Examples

- `calendarprotocol get`
  returns
  `calendarprotocol tls`
- `calendarprotocol auto`
  returns
  `calendarprotocol auto`

Limitations

None

Comments

None

calendarregisterwithserver

Enables or disables the calendaring service.

Syntax

```
calendarregisterwithserver get
calendarregisterwithserver <yes|no>
```

Parameter	Description
`get`	Returns the current server registration status.
`yes`	Enables the calendaring service.
`no`	Disables the calendaring service.

Feedback Examples

- `calendarregisterwithserver get`
  returns
  `calendarregisterwithserver no`
- `calendarregisterwithserver yes`
  returns
  `calendarregisterwithserver yes`
- `calendarregisterwithserver no`
  returns
  `calendarregisterwithserver no`

Limitations

None

Comments

To configure the Microsoft Exchange Server address used by the calendaring service, use the calendarserver command.

calendarremindertime

Gets or sets the meeting reminder time when the system is registered with the calendaring service.

Syntax

```
calendarremindertime <get|1|5|10|15|30|none>
```

Parameter	Description
`get`	Gets the current reminder time.
`1|5|10|15|30|none`	The number of minutes before a meeting starts that a meeting reminder is given. The default is 5.

Feedback Examples

- `calendarremindertime get`
  returns
  `calendarremindertime 5`
- `calendarremindertime 15`
  returns
  `calendarremindertime 15`
- `calendarremindertime none`
  returns
  `calendarremindertime none`

Limitations

None

Comments

None

See Also

Use the notify command to register for meeting reminders.

See also calendarplaytone command.

calendarresource

Gets or sets the resource (i.e., account) monitored for calendar events.

Syntax

```
calendarresource get
calendarresource "resource"
```

Parameter	Description
`get`	Returns the resource monitored for calendar events.
`"resource"`	The resource to monitor for calendaring events.

Feedback Examples

- `calendarresource get`
  returns
  `calendarresource radam@abcde.com`
- `calendarresource jmcnulty@abcde.com`
  returns
  `calendarresource jmcnulty@abcde.com`

Limitations

None

Comments

A resource can be a user or resource account. A resource account is assigned to a meeting room.

See Also

Use the calendarregisterwithserver command to enable or disable the calendaring service. See the calendarserver command to configure the Microsoft Exchange Server address used by the calendaring service.

calendarserver

Gets or sets the Microsoft Exchange Server used by the calendaring service.

Syntax

```
calendarserver get
calendarserver "server"
```

Parameter	Description
`get`	Gets the current Microsoft Exchange Server used by the calendaring service.
`"server"`	The IP address or DNS name of the Microsoft Exchange Server used by the calendaring service.

Feedback Examples

- `calendarserver get`
  returns
  `calendarserver 192.168.44.168`
- `calendarserver 192.168.23.221`
  returns
  `calendarserver 192.168.23.221`
- `calendarserver get`
  returns
  `calendarserver mail.exchangeserver.local.com`
- `calendarserver mail2.exchserver.local.com`
  returns
  `calendarserver mail2.exchserver.local.com`

Limitations

None

Comments

None

See Also

Use the calendarregisterwithserver command to enable or disable the calendaring service.

calendarshowpvtmeetings

Enables or disables the display of private meetings in the calendar when the system is registered with the calendaring service.

Syntax

```
calendarshowpvtmeetings get
calendarshowpvtmeetings <yes|no>
```

Parameter	Description
`get`	Gets the current setting for private meeting display.
`yes`	Enables the display of private meetings.
`no`	Blocks the display of private meetings.

Feedback Examples

- `calendarshowpvtmeetings get`
  returns
  `calendarshowpvtmeetings no`
- `calendarshowpvtmeetings yes`
  returns
  `calendarshowpvtmeetings yes`
- `calendarshowpvtmeetings no`
  returns
  `calendarshowpvtmeetings no`

Limitations

None

Comments

None

calendarstatus

Returns the status of the Microsoft Exchange Server connection.

Syntax

```
calendarstatus get
```

Parameter	Description
`get`	Returns the Microsoft Exchange Server connection status.

Feedback Examples

- `calendarstatus get`
  returns
  `calendarstatus established`
- `calendarstatus get`
  returns
  `calendarstatus unavailable`

Limitations

None

Comments

None

See Also

Use the calendarregisterwithserver command to enable or disable the calendaring service.

calendaruser

Gets or sets the user name the calendaring service uses to log in to the Microsoft Exchange Server.

Syntax

```
calendaruser get
calendaruser "username"
```

Parameter	Description
`get`	Returns the user name being used by the calendaring service.
`username`	The user name the calendaring service uses to log in to the Microsoft Exchange Server.

Feedback Examples

- `calendaruser get`
  returns
  `calendaruser jpolycom`

Limitations

None

Comments

None

See Also

See the calendarserver command to configure the Microsoft Exchange Server address used by this service.

callinfo

Returns information about the current call.

Syntax

```
callinfo all
callinfo callid "callid"
```

Parameter	Description
`all`	Returns information about each connection in the call.
`callid`	Returns information about the connection with the specified call ID.

Feedback Examples

- `callinfo callid 36`
  returns
  `callinfo:36:192.168.1.102:256:connected:muted:outgoing:videocall`
- `callinfo all`
  returns
  `system is not in a call`
  when no call is currently connected

Limitations

None

Comments

The `callid` information is returned using the following format:

```
callinfo:<callid>:<far site name>:<far site number>:<speed>:
<connection status>:<mute status>:<call direction>:<call type>
```

callstate

Sets or gets the call state notification for call state events.

Syntax

```
callstate <get|register|unregister>
```

Parameter	Description
`get`	Returns the current setting.
`register`	Registers the system to give notification of call activities.
`unregister`	Disables the register mode.

Feedback Examples

- `callstate register`
  returns
  `callstate registered`
- `callstate unregister`
  returns
  `callstate unregistered`
- `callstate get`
  returns
  `callstate unregistered`

After registering, the following callstate (cs:) data is returned when connecting an IP call:

```
cs: call[34] chan[0] dialstr[192.168.1.103] state[ALLOCATED]
cs: call[34] chan[0] dialstr[192.168.1.103] state[RINGING]
cs: call[34] chan[0] dialstr[192.168.1.103] state[COMPLETE]
active: call[34] speed [384]
```

After registering, the following response occurs when disconnecting an IP call:

```
cleared: call[34]
dialstr[IP:192.168.1.103 NAME:Polycom Demo]
ended: call[34]
```

Limitations

None

Comments

None

See Also

You can also use the notify command and the nonotify command for notifications.

camera

Sets or gets the near- or far-site camera settings.

Syntax

```
camera near {1..4}
camera far {1..4}
camera <near|far> move <left|right|up|down|zoom+|zoom-|stop>
camera <near|far> source
camera <near|far> stop
camera near <getposition|setposition “x” “y” “z”>
camera near tracking statistics
camera near tracking <get|on|off>
camerainvert near <get|on|off>
```

Parameter	Description
`get`	Returns the current setting.
`on`	Sets the camera to present an inverted (upside down) video image.
`off`	Sets the camera to present a normal (right-side up) video image.
`near`	Specifies that the command selects or controls the near camera.
`far`	Specifies that the command selects or controls the far camera.
`{1..4}`	Specifies a near or far camera as the main video source.
`move`	Changes the near or far camera’s direction or zoom. Valid directions are: left, right, up, down, zoom+, zoom-, and stop.
`left`	Starts moving the camera left.
`right`	Starts moving the camera right.
`up`	Starts moving the camera up.
`down`	Starts moving the camera down.
`zoom+`	Starts zooming in.
`zoom-`	Starts zooming out.
`stop`	Stops the movement of the near or far camera. Returns no feedback.
`source`	Returns the number of the near or far camera source currently selected.
`getposition`	Gets the pan, tilt, and zoom coordinates of the currently selected PTZ camera in the format of `pan tilt zoom`.

Parameter	Description
`setposition "x" "y" "z"`	Sets the pan (`x`), tilt (`y`), and zoom (`z`) coordinates of the selected PTZ camera. **Notes**: Different cameras might have different PTZ values. Some D30 cameras might not be able to reach the full range limit. For example, although the pan limit is 5000, the camera might only be able to reach a nearby value.
`tracking statistics`	Gets tracking statistics. Tracking statistics measure: • The amount of time tracking is turned off divided by the total call time in the most recent 100 calls lasting more than five minutes. • The amount of room and close-up view switches divided by the total call time in the most recent 100 calls lasting more than five minutes.
`tracking <get|on|off>`	Enables or disables the Polycom EagleEye Director II or EagleEye Producter camera tracking feature. `on` turns the tracking feature on, `off` turns the tracking feature off, and `get` returns the current tracking feature setting.
`camerainvert near`	Sets the video image of the EagleEye IV camera to upside down (`on`) or normal (`off`).

Feedback Examples

- `camera far 2`
  specifies camera 2 at the far-site and returns
  `camera far 2`
- `camera far move left`
  causes the far-site camera to start panning to the left and returns
  `event: camera far move left`
- `camera near move zoom+`
  causes the near-site camera to zoom in and returns
  `event: camera near move zoom+`
- `camera near tracking off`
  returns
  `camera near tracking off`
- `camera near tracking on`
  returns
  `camera near tracking on`
- `camera near setposition 100 100 219`
  returns
  `camera near setposition 100 100 219`
- `camera near getposition`
  returns
  `camera near getposition 100 99 218`
- `camerainvert near get`
  returns
  `camerainvert near off`

- `camerainvert near on`
  returns
  `camerainvert near on`
- `camerainvert near off`
  returns
  `camerainvert near off`

Limitations

None.

Comments

The `camera` commands function only when the system is in a wake state. If necessary, use the `wake` command prior to using the `camera` commands.

If the `camera near {1..4}` API command is used for an input configured as content, the command becomes a toggle. You must send the command once to send the content source and a second time to stop the content source.

After using a `camera` command to move a Polycom EagleEye Producer or Polycom EagleEye Director II camera, you must use the `camera <near|far> stop` command to update the camera position.

camera near tracking

Enables or disables camera tracking for a Polycom EagleEye Director II or EagleEye Producer camera.

The `camera near tracking get` command returns the following value: `GroupFrame`.

Syntax

```
camera near tracking <get|on|off>
cameratracking near calibrate <get|on|off>
cameratracking near framing <get|wide|medium|tight>
cameratracking near mode <get|off|group|speaker|groupwithtransition>
cameratracking near participant <get|on|off>
cameratracking near pip <get|on|off>
cameratracking near speed <get|slow|normal|fast>
cameratracking near wake <get|on|off>
```

Parameter	Description
`get`	Returns the current setting.
`on`	Enables camera near tracking.
`off`	Disables camera near tracking.
`calibrate`	Enables (`on`) or disables (`off`) automatic calibration for the EagleEye Producer camera.
`framing`	Sets camera tracking framing adjustments for a EagleEye Producer camera to one of the following: • `wide` - wide view of meeting participants • `medium` - default (normal) framing • `tight` - close-up view of meeting participants
`mode`	Set the tracking mode for the EagleEye Director II or EagleEye Producer camera. EagleEye Director II: • `group` - Frames the meeting participants. • `speaker` - Frames the active speaker. EagleEye Producer: • `groupwithtransition` - Enables the automatic locating and framing of participants in the room while displaying camera motion.
`participant`	Enables (`on`) or disables (`off`) the Participant feature for the EagleEye Producer camera.
`pip`	Enables (`on`) or disables (`off`) the self view setting of the EagleEye Director II camera.

Parameter	Description
`speed`	Sets the rate at which an EagleEye Director II or EagleEye Producer camera detects active speakers to `slow`, `normal`, or `fast`. Camera tracking must be enabled.
`wake`	Enables (`on`) or disables (`off`) the auto wake up feature for the EagleEye Producer camera. To use these commands, you must go to **Admin Settings > System Settings > Polycom Labs** in the web interface and enable the auto wake up feature for the EagleEye Producer camera.

Feedback Examples

- `camera near tracking get`
  returns
  `camera near tracking GroupFrame`
- `camera tracking get`
  returns
  `error: There is no tracking camera`
- `camera near tracking off`
  returns
  `camera near tracking off`
- `camera near tracking on`
  returns
  `camera near tracking on`
- `cameratracking near calibrate get`
  returns
  `cameratracking near calibrate on`
- `cameratracking near framing get`
  returns
  `cameratracking near framing medium`
- `ccameraautohanguptimer off`
  returns
  `cameraautohanguptimer off`
- `cameraautohanguptimer 30`
  returns
  `cameraautohanguptimer 30`
- `cameratracking near mode get`
  returns
  `cameratracking near mode group`
- `cameratracking near participant get`
  returns
  `cameratracking near participant on`
- `cameratracking near participant off`
  returns
  `cameratracking near participant off`

- `cameratracking near pip get`
  returns
  `cameratracking near pip on`

Limitations

None

Comments

None

configdisplay

Gets or sets the resolution and refresh rate for Monitor 1 or Monitor 2.

Syntax

```
configdisplay [<monitor1|monitor2>] get
configdisplay monitor1
<auto|50hz1920x1080p|60hz1920x1080p|25hz3840x2160p|30hz3840x2160p|50hz3840x2160p|60hz3
840x2160p>
configdisplay monitor2 <off|auto|50hz1920x1080p|60hz1920x1080p>
```

Parameter	Description
`get`	Returns the current setting.
`monitor1`	Specifies Monitor 1.
`monitor2`	Specifies Monitor 2.
`off`	Sets Monitor 2 to `off`.
`auto`	Sets the monitor signal type to auto detection.
`<refreshrateResolution>`	Sets the resolution and refresh rate. For example, entering `configdisplay monitor1 60hz3840x2160p` configures Monitor 1 to 3840x2160p, 60Hz.

Feedback Examples

- `configdisplay get`
  returns
  `configdisplay monitor1 1920x1080p 60Hz, monitor2 1920x1080p 60Hz`
- `configdisplay monitor1 30hz1920x1080p`
  returns
  `configdisplay monitor1 1920x1080p 30Hz`
- `configdisplay monitor2 get`
  returns
  `configdisplay monitor2 1920x1080p 60Hz`
- `configdisplay monitor2 off`
  returns
  `configdisplay monitor2 off`

Limitations

None

Comments

None

configparam

Gets or sets the video quality setting for the specified video input for motion or sharpness.

Syntax

```
configparam <"parameter"> get
configparam <"parameter"> set <"value">
```

Parameter	Description
`get`	Gets the video quality setting for the specified video input.
`set`	Sets the video quality setting for the specified video input.
`camera_video_quality <motion|sharpness>`	Sets the video quality setting for the specified video input for motion or for sharpness (for images without motion).

Feedback Examples

- `configparam camera_video_quality 1 set motion`
  returns
  `camera1_video_quality motion`
- `configparam camera_video_quality 1 get`
  returns
  `camera1_video_quality sharpness`

Limitations

None

Comments

None

configpresentation

Gets or sets the dual display settings, including self view and content.

Syntax

```
configpresentation [<self-view>|<content>] get
configpresentation self-view <corner|full-screen>
configpresentation content <single|dual>
```

Parameter	Description
get	Returns the current dual-display settings for `self-view` or `content`.
self-view	Specifies settings for self view. Precedes the `corner` or `full-screen` parameter.
corner	Displays self view in the bottom corner of a monitor.
full-screen	Displays full screen Self View.
content	Specifies where content displays on the connected monitor(s). Precedes the `single` or `dual` parameter.
single	Displays content on one monitor.
dual	Displays content on both monitors.

Feedback Examples

- `configpresentation self-view get`
  returns
  `configpresentation self-view Corner`
- `configpresentation self-view full-screen`
  returns
  `configpresentation self-view full-screen`
- `configpresentation content get`
  returns
  `configpresentation content dual`
- `configpresentation content single`
  returns
  `configpresentation content single`

Limitations

None

Comments

None

contentauto

Gets or sets the automatic bandwidth adjustment for people and content in point-to-point H.323 calls. Automatic adjustment maintains equal image quality in the two streams.

Syntax

```
contentauto <get|on|off>
```

Parameter	Description
`get`	Returns the current setting.
`on`	Enables automatic bandwidth adjustment for people and content.
`off`	Disables automatic bandwidth adjustment for people and content. The Quality Preference setting is used instead.

Feedback Examples

- `contentauto off`
  returns
  `contentauto off`
- `contentauto on`
  returns
  `contentauto on`
- `contentauto get`
  returns
  `contentauto on`

Limitations

None

Comments

None

daylightsavings

Gets or sets the daylight saving time setting. When you enable this setting, the system clock automatically changes for daylight saving time.

Syntax

```
daylightsavings <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables automatic adjustment for daylight savings time.
`no`	Disables automatic adjustment for daylight savings time.

Feedback Examples

- `daylightsavings no`
  returns
  `daylightsavings no`
- `daylightsavings yes`
  returns
  `daylightsavings yes`
- `daylightsavings get`
  returns
  `daylightsavings yes`

Limitations

None

Comments

None

defaultgateway

Gets or sets the default gateway.

Syntax

```
defaultgateway get
defaultgateway set "xxx.xxx.xxx.xxx"
```

Parameter	Description
`get`	Returns the default gateway IP address.
`set`	Sets the default gateway when followed by the `"xxx.xxx.xxx.xxx"` parameter.
`"xxx.xxx.xxx.xxx"`	IP address to use as the default gateway.

Feedback Examples

- `defaultgateway set 192.168.1.101`
  returns
  `defaultgateway 192.168.1.101`

Limitations

None

Comments

You can only change the `defaultgateway` setting if DHCP is turned off.

dhcp

Gets or sets DHCP options.

Syntax

```
dhcp <get|off|client>
```

Parameter	Description
`get`	Returns the selected DHCP option.
`off`	Disables DHCP.
`client`	Enables DHCP client. The system obtains an IP address from a server on your network.

Feedback Examples

- `dhcp off`
  returns
  `dhcp off`
- `dhcp client`
  returns
  `dhcp client`
- `dhcp get`
  returns
  `dhcp client`

Limitations

None

Comments

You must restart the system after making a change to a setting.

dial

Dials video or audio calls manually or from the directory.

Syntax

```
dial addressbook "addr book name"
dial auto "speed" "dialstr"
dial manual "speed" "dialstr1" ["dialstr2"] [h323|ip|sip]
dial phone <sip|h323|auto|> "dialstring"
```

Parameter	Description
`addressbook`	Dials a directory (address book) entry. Requires the name of the entry.
`"addr book name"`	The name of the directory (address book) entry. The name may be up to 25 characters. Use quotation marks around strings that contain spaces. For example: `"John Doe"`.
`auto`	Automatically dials a number. When used with "`speed`" and "`dialstr`, dials a video call number `dialstr1` at speed of type h323.
`"speed"`	Valid data rate for the network.
`"dialstr"`, `"dialstr1"`, `"dialstr2"`	IP directory number.
`manual`	Dials a video call number `dialstr1` at speed of type h323. Requires the parameters "`speed`" and "`dialstr1`". Use `dial manual "speed" "dialstr" "type"` when you do not want automatic call rollover or when the dialstring might not convey the intended transport. Use dial manual without specifying a call type
`h323|ip|sip`	Call type
`phone`	Dials an audio call. This option Is supported only when the **Enable Audio Add In** call feature is enabled.
`"dialstring"`	Numeric string specifying the phone number to dial. Enclose the string in quotation marks if it includes spaces. Example: `"512 555 1212"`

Feedback Examples

- `If registered for callstate notifications (`callstate register), the API returns

```
cs: call[44] chan[0] dialstr[5551212] state[ALLOCATED]
cs: call[44] chan[0] dialstr[5551212] state[RINGING]
cs: call[44] chan[0] dialstr[5551212] state[CONNECTED]
cs: call[44] chan[0] dialstr[5551212] state[CONNECTED]
cs: call[44] chan[0] dialstr[5551212] state[COMPLETE]
cs: call[44] chan[0] dialstr[5551212] state[COMPLETE]
active: call[44] speed[64]
```

- `dial addressbook "John Polycom"`
  returns
  `dialing addressbook "John Polycom"`
- `dial phone sip 1234`
  returns
  `dialing voice_sip`
- If SIP is not enabled `dial phone sip 1234`
  returns
  `info: IP line (SIP) not enabled.`
- If registered for callstate notifications (callstate register), the API returns
  ```
  cs: call[44] chan[0] dialstr[192.168.1.101] state[ALLOCATED]
  cs: call[44] chan[0] dialstr[192.168.1.101] state[RINGING]
  cs: call[44] chan[0] dialstr[192.168.1.101] state[COMPLETE]
  active: call[44] speed[384]
  ```

  Notes: The [BONDING] responses in IP calls are extraneous text that will be removed in a subsequent software version.

  Call ID (call [44]) is an example of the response. The Call ID number depends upon the call type.
- If registered for callstatus notifications (notify callstatus), the API returns,
  ```
  notification:callstatus:outgoing:45:null 1::opened::0:videocall
  notification:callstatus:outgoing:45: Polycom Austin:
  192.168.1.101:connecting:384:0:videocall
  notification:callstatus:outgoing:45: Polycom Austin:
  192.168.1.101:connected:384:0:videocall
  ```

  Note: The call ID number (45) is an example of the response. The Call ID number depends upon the call type.

Limitations

None

Comments

None

See Also

You can use `callstate register` or `notify callstatus` to obtain updated information on the status of a call. For example, when using `dial manual` to place a call, both registration commands will tell you when the call is connected. Refer to the callstate command and the notify command.

dial addressbook_entry

Dials a system using a unique identifier retrieved by the `globaldir` command.

Syntax

```
dial addressbook_entry "UID"
```

Parameter	Description
`"UID"`	Unique identifier associated with a site or group, for example: ldap#g#f82be96eea3bd644a1963dc7fdf45011 The complete UID must be specified

Feedback Examples

- `dial addressbook_entry ldap#g#35086aa0ecc9014facdcaa89bd34ccf6`
  returns
  `dialing addressbook_entry ldap#g#35086aa0ecc9014facdcaa89bd34ccf6Comments`

Limitations

None

Comments

The “UID” value must be retrieved by the globaldir command.

dns

Gets or sets the configuration for up to four DNS servers.

Syntax

```
dns get {1..4}
dns set {1..4} "xxx.xxx.xxx.xxx"
```

Parameter	Description
`get`	Returns the current IP address of the specified server. A server identification number `{1..4}` is required.
`{1..4}`	Specifies the server identification number.
`set`	Sets the IP address of the specified DNS server when followed by the `"xxx.xxx.xxx.xxx"` parameter. A server identification number `{1..4}` is required.
`"xxx.xxx.xxx.xxx"`	Specifies the IP address for the specified server.

Feedback Examples

- `dns set 1 192.168.1.205`
  returns
  `dns 1 192.168.1.205`

Limitations

None

Comments

After making a change, you must restart the system for the setting to take effect.

You cannot set these values if the system is in DHCP client mode.

dynamicbandwidth

Gets or sets the use of dynamic bandwidth allocation for Quality of Service.

Syntax

```
dynamicbandwidth <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables the dynamic bandwidth option.
`no`	Disables the dynamic bandwidth option.

Feedback Examples

- `dynamicbandwidth yes`
  returns
  `dynamicbandwidth yes`
- `dynamicbandwidth no`
  returns
  `dynamicbandwidth no`
- `dynamicbandwidth get`
  returns
  `dynamicbandwidth no`

Limitations

None

Comments

The system's dynamic bandwidth function automatically finds the optimum speed for a call. If you experience excessive packet loss during a call, the dynamic bandwidth function decrements the speed until there's no packet loss. This is supported in calls with endpoints that also support dynamic bandwidth.

e164ext

Gets or sets an H.323 (IP) extension (also known as an E.164 name).

Syntax

```
e164ext get
e164ext set "e.164name"
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the E.164 extension when followed by the `"e.164name"` parameter. To erase the current setting, omit `"e.164name"`.
`"e.164name"`	A valid E.164 extension (usually a four-digit number).

Feedback Examples

- `e164ext set 7878`
  returns
  `e164ext 7878`
- `e164ext get`
  returns
  `e164ext 7878`

Limitations

None

Comments

The extension number is associated with a specific LAN device.

echo

Returns a string that is sent to the system.

Syntax

```
echo <string>
```

Parameter	Description
`echo <string>`	Returns a string sent to the system.

Feedback Examples

- `echo "Are you there?"`
  returns
  `Are you there?`
- `echo KA`
  returns
  `KA`

Limitations

None

Comments

None

echocanceller

Gets and sets the configuration of line-input port echo cancellation that prevents users from hearing their voices loop back from the far site.

Syntax

```
echocanceller <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables the echo canceller option.
`no`	Disables the echo canceller option.

Feedback Examples

- `echocanceller get`
  returns
  `echocanceller no`

Limitations

None

Comments

None

echoreply

Gets or sets the system's ability to send an Echo Reply message in response to an Echo Request message sent to an IPv4 multicast/anycast address.

Syntax

```
echoreply <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables the echo reply option.
`no`	Disables the echo reply option.

Feedback Examples

- `echoreply get`
  returns
  `echoreply yes`
- `echoreply no`
  returns
  `echoreply no`

Limitations

None

Comments

The number of responses may be traffic-conditioned to limit the effect of a denial-of-service (DoS) attack.

You must restart the system after making a change to a setting.

enableacousticfence

Gets or sets the current setting for the Polycom® Acoustic Fence™ technology feature.

Syntax

```
enableacousticfence <get|on|off>
```

Parameter	Description
`get`	Returns the current setting.
`on`	Enables Polycom Acoustic Fence.
`off`	Disables Polycom Acoustic Fence.

Feedback Examples

- `enableacousticfence get`
  returns
  `enableacousticfence on`
- `enableacousticfence on`
  returns
  `enableacousticfence on`
- `enableacousticfence off`
  returns
  `enableacousticfence off`

Limitations

None

Comments

None

enableaudioadd

Enables or disables the Audio Add In feature, which allows one additional outbound, audio-only call from a G7500 system when the maximum number of calls allowed for a license is reached.

Syntax

```
enableaudioadd <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables the Audio Add-In features. This is the default setting.
`no`	Disables the Audio Add-In feature.

Feedback Examples

- `enableaudioadd get`
  returns
  `enableaudioadd yes`
- `enableaudioadd yes`
  returns
  `enableaudioadd yes`
- `enableaudioadd no`
  returns
  `enableaudioadd no`

Limitations

None

Comments

None

enablefirewalltraversal

Gets or sets the system's ability to traverse firewalls. This feature requires a session border controller that supports H.460.

Syntax

```
enablefirewalltraversal <get|on|off>
```

Parameter	Description
`get`	Returns the current setting.
`on`	Enables the firewall traversal feature.
`off`	Disables the firewall traversal feature.

Feedback Examples

- `enablefirewalltraversal on`
  returns
  `enablefirewalltraversal on`
- `enablefirewalltraversal off`
  returns
  `enablefirewalltraversal off`
- `enablefirewalltraversal get`
  returns
  `enablefirewalltraversal off`

Limitations

None

Comments

None

enablekeyboardnoisereduction

Gets or sets the keyboard noise reduction feature.

Syntax

```
enablekeyboardnoisereduction <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables keyboard noise reduction.
`no`	Disables keyboard noise reduction.

Feedback Examples

- `enablekeyboardnoisereduction yes`
  returns
  `enablekeyboardnoisereduction yes`
- `enablekeyboardnoisereduction no`
  returns
  `enablekeyboardnoisereduction no`
- `enablekeyboardnoisereduction get`
  returns
  `enablekeyboardnoisereduction no`

Limitations

None

Comments

None

enablelivemusicmode

Gets or sets the M-Mode feature.

Syntax

```
enablelivemusicmode <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables M-Mode.
`no`	Disables M-Mode.

Feedback Examples

- `enablelivemusicmode yes`
  returns
  `enablelivemusicmode yes`
- `enablelivemusicmode no`
  returns
  `enablelivemusicmode no`

Limitations

None

Comments

M-Mode was previously known as MusicMode. The feature functions the same way as before despite the name change.

enablepvec

Gets or sets the Polycom Video Error Concealment (PVEC) setting on the system.

Syntax

```
enablepvec <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables the PVEC option.
`no`	Disables the PVEC option.

Feedback Examples

- `enablepvec yes`
  returns
  `enablepvec yes`
- `enablepvec no`
  returns
  `enablepvec no`
- `enablepvec get`
  returns
  `enablepvec no`

Limitations

None

Comments

This option, **Enable Lost Packet Recovery** in the web interface, is enabled by default.

enablersvp

Gets or sets the Resource Reservation Protocol (RSVP) setting, which requests that routers reserve bandwidth along an IP connection path.

Syntax

```
enablersvp <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables the RSVP option.
`no`	Disables the RSVP option.

Feedback Examples

- `enablersvp yes`
  returns
  `enablersvp yes`
- `enablersvp no`
  returns
  `enablersvp no`
- `enablersvp get`
  returns
  `enablersvp no`

Limitations

None

Comments

This option is enabled by default.

enablesipka

Gets or sets the option to send SIP keep-alive messages.

Syntax

```
enablesipka <get|on|off>
```

Parameter	Description
`get`	Returns the current setting.
`on`	Enables SIP keep alive messages.
`no`	Disables SIP keep alive messages.

Feedback Examples

- `enablesipka get`
  returns
  `enablesipka off`
- `enablesipka on`
  returns
  `enablesipka on`

Limitations

None

Comments

None

enablesnmp

Gets or enables/disables SNMP.

Syntax

```
enablesnmp <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables SNMP.
`no`	Disables SNMP.

Feedback Examples

- `enablesnmp yes`
  returns
  `enablesnmp yes`
- `enablesnmp no`
  returns
  `enablesnmp no`
- `enablesnmp get`
  returns
  `enablesnmp no`

Limitations

None

Comments

None

encryption

Gets or sets the AES encryption mode for the system.

Syntax

```
encryption <get|yes|no|requiredvideocallsonly|requiredallcalls>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Use encryption when the far site is capable of encryption. **Note:** This parameter is **When Available** in the user interface.
`no`	Disables encryption. **Note:** This parameter is **Off** in the web interface.
`requiredvideocallsonly`	Enforces encryption on all video endpoints. Any video calls to or from systems that do not have encryption enabled are not connected. Audio-only calls are connected.
`requiredallcalls`	Enforces encryption on all endpoints. Any video or audio calls to or from systems that do not have encryption enabled are rejected and are not connected.

Feedback Examples

- `encryption yes`
  returns
  `encryption yes`
- `encryption no`
  returns
  `encryption no`
- `encryption get`
  returns
  `encryption no`
- `encryption requiredvideocallsonly`
  returns
  `encryption requiredvideocallsonly`
- `encryption requiredallcalls`
  returns
  `encryption requiredallcalls`

Limitations

None

Comments

You cannot execute the `encryption` command while a call is in progress. Using this command while the system is in a call returns the following message:

```
error: command has illegal parameters
```

exit

Ends the command-line API session.

Syntax

```
exit
```

Additional Restrictions

None

Feedback Examples

- `exit`
  returns
  `Connection to host lost.`

Limitations

None

Comments

For serial sessions, this command starts a new session.

exportdirectory

Exports a directory in XML format.

Syntax

```
exportdirectory
```

Additional Restrictions

None

Feedback Example

```
exportdirectory
```

returns

```
exportdirectory started
<?xml version="1.0" encoding="UTF-8" ?>
<addresses>
<entrytype type="entry"  name="dawn" filename="dawn" uniqueid="local:26">
<address filename="dawn
" langid="
" displayname="dawn
" name="dawn">
<h323 address="192.168.1.120"
 speed="0"/>
<sip address="192.168.1.120"
 speed="0"/>
<category category="CONTACTS"/>
</address>
</entrytype>
<entrytype type="entry"  name="dawn " filename="dawn " uniqueid="local:28">
<address filename="dawn
" langid="
" displayname="dawn
" name="dawn ">
<h323 address="192.168.1.120"
 speed="0"/>
<sip address="192.168.1.120"
 speed="0"/>
<category category="CONTACTS"/>
</address>
</entrytype>
<address filename="testGroup
" langid="
" displayname="testGroup
" name="testGroup ">
<multisitename meeting_name="testGroup " />
```

```
<multisitespeed meeting_speed="auto"/>
<multisitename0 site_name_0="dawn "/>
<mulitsitetype0 site_type_0="2" type_0="1000"/>
<mulitsiteprefcalltype0 pref_call_type_0="H323"/>
<multisiteuniqueid0 unique_id_0="local:28"/>
<multisitename1 site_name_1="dawn2 "/>
<mulitsitetype1 site_type_1="2" type_1="1000"/>
<mulitsiteprefcalltype1 pref_call_type_1="H323"/>
<multisiteuniqueid1 unique_id_1="local:30"/>
<multisitename2 site_name_2="dawn3 "/>
<mulitsitetype2 site_type_2="2" type_2="1000"/>
<mulitsiteprefcalltype2 pref_call_type_2="H323"/>
<multisiteuniqueid2 unique_id_2="local:29"/>
</address>
</entrytype>
<entrytype type="group"  name="testGroup1" filename="testGroup1"
uniqueid="local:38">
<address filename="testGroup1
" langid="
" displayname="testGroup1
" name="testGroup1">
<multisitename meeting_name="testGroup1" />
<multisitespeed meeting_speed="auto"/>
</address>
</entrytype>
</addresses>
</xml>
exportdirectory done
```

Limitations

None

Comments

`exportdirectory done` indicates that all directory data has been exported.

Do not use the `exportdirectory command` to interpret the data that is returned. Simply store and use the data as input to the `importdirectory` command or import directory utility in the web interface. The format of the exported directory data might change in future software releases and any application attempting to interpret the data could find its ability to do so compromised in later software releases.

Exporting a directory on one system model and importing the directory on another model is not supported. Attempts to export and import directory information between different systems might also fail. The message `importdirectory failed` indicates that the system was not able to import the information.

When importing directory data back into the system, use the data in its entirety (not edited in any form). There is information that is used by the system to determine what type (XML or CSV) of data is being imported.

See Also

See the importdirectory command.

exportprofile

Exports system and user profile information in CSV format. The output is available through a telnet or serial port connection.

Syntax

```
exportprofile
```

Additional Restrictions

None

Feedback Example

```
exportprofile
```

returns

```
exportprofile started
profileversion,0.2
system.info.eulafile,eula
system.info.hardwareversion,9
system.info.humanreadablemodel,RealPresence
system.info.humanreadableplatform,
system.info.humanreadableversion,Dev - 4.1.3-0
system.info.plcmstandardversion,Dev - 4.1.3-0
system.info.serialnumber,8213130FE433CV
audio.lineIO.lineinechocanceller,"False"
audio.volume.speakervolume,"46"
comm.Firewall.fixedportstcphigh,"3241"
comm.Firewall.fixedportsudphigh,"3301"
comm.NICs.H323Nic.h323extension,"177704997"
comm.NICs.H323Nic.h323name,"177704997"
comm.NICs.SipNic.bfcptransportprotocol,"Prefer_UDP"
comm.NICs.SipNic.thirdpartyinterop.ocs.sipuuid,"d503b976-c62f-5484-82c0-64a479
63      18d1"
comm.Qos.tos.tosaudio,"5"
comm.Qos.tos.tosfecc,"3"
comm.Qos.tos.tosoam,"0"
comm.Qos.tos.tosvideo,"4"
```

```
location.country,"United States"
location.language,"ENGLISHUS"
pm.monRoleAuto,"True"
pm.monitor[1].enable,"True"
softupdate.url,"http://builds.softupdate.com/~test/softupdate      /"
sourceman.camera[1].autowhitebalancegainb,"33"
sourceman.camera[1].autowhitebalancegainr,"37"
sourceman.camera[1].backlightcomp,"False"
sourceman.camera[1].brightness,"11"
sourceman.camera[1].contrast,"13"
sourceman.camera[1].name,"Main"
sourceman.camera[1].role,"People"
sourceman.camera[1].saturation,"6"
sourceman.camera[1].sharpness,"3"
sourceman.camera[1].videoquality,"Sharpness"
sourceman.camera[1].whitebalancemode,"atw"
video.monitor[1].Resolution,"1920x1080p 60Hz"
video.monitor[2].Resolution,"1920x1080p 60Hz"
exportprofile done
```

Comments

`exportprofile done` indicates that all profile data is exported.

When importing directory data back into the system, use the data in its entirety (not edited in any form). There is information that is used by the system to determine what type data (XML or CSV) is being imported.

See Also

See the importprofile command.

farcontrolnearcamera

Gets or sets far control of the near camera, which allows far sites to control the camera on your system.

Syntax

```
farcontrolnearcamera <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Allows the far site to control the near camera if the far site has this capability.
`no`	Disables far-site control of the near camera.

Feedback Examples

- `farcontrolnearcamera yes`
  returns
  `farcontrolnearcamera yes`
- `farcontrolnearcamera no`
  returns
  `farcontrolnearcamera no`
- `farcontrolnearcamera get`
  returns
  `farcontrolnearcamera no`

Limitations

None

Comments

None

farnametimedisplay

Enables or disables the name that is displayed on a far site monitor.

Syntax

```
farnametimedisplay <get|on|off>
```

Parameter	Description
`get`	Returns the current setting.
`on`	Enables the name to be displayed for defined amount of time.
`off`	Disables the name on a far site monitor.

Feedback Examples

- `farnametimedisplay get`
  returns
  `farnametimedisplay on`
- `farnametimedisplay on`
  returns
  `farnametimedisplay on`
- `farnametimedisplay off`
  returns
  `farnametimedisplay off`

Limitations

None

Comments

The name will time out in 10 seconds after displayed.

favorites

Returns entries in the Favorites list. Favorites can link to local or global directory entries.

Syntax

```
favorites names <all|video|phone> [range_start] [range_end]
favorites names <all|video|phone> size
favorites address "sys_name" ["sys_label"]
```

Parameter	Description

`names`	Returns names in the Favorites list and the associated system type: `video`, `multicodec`, `phone`, or `multisite` (a multicodec system displays as a single row).
	The response is in the following format:
	`favorites names`
	`name:"sys_name" sys_label:"sys_label" type: <video|multicodec|phone|multisite>`
	`favorites names done`

`<all|video|phone>`	`video` returns entries with video systems, `phone` returns entries with phone systems, and `all` returns entries with both types.
`size`	Specifies the size of the returned set based on the parameters you are including. For example, a response could look like this: `favorites names video size 9`
`range_start`	For the `names` parameter, it specifies the beginning of the range of entries to return.
`range_end`	For the `names` parameter, it specifies the end of the range of entries to return. If `range_start` is specified without a `range_end`, the single `range_start` entry is returned. If `range_end` is `-1`, all entries starting with `range_start` are returned.
`address`	Obtains the address information for a specified entry. If the entry is an ITP system, the results include the addresses for all its codecs. If the codecs support multiple protocols, the different addresses are returned on separate lines. This parameter is not supported for multisite entries.
`sys_name`	The user-friendly name for a Favorites entry (for example, the name of a person or room). It is surrounded by quotes if it contains spaces.
`sys_label`	If a person or room has more than one system, the returned set includes a row for each system. If those systems are of the same type, it is considered to be a single ITP system with multiple codecs rather than separate systems. If there are different types of systems, the `sys_label` parameter is included to differentiate the systems.
`type`	The type of Favorites entry (possible values are `video, multicodec, phone,` and `multisite`).
`codec: <1..4>`	If the entry is an ITP system, each codec has a unique value.
`h323_spd`	The preferred speed for an H.323 call to this entry.

Parameter	Description
h323_num	H.323 address or alias.
h323_ext	H.323 extension or E.164 number.
sip_spd	The preferred speed for a SIP call to this entry. If no speed is associated with the entry, it is the same as h323_spd.
sip_num	SIP address.
xmpp_addr	XMPP address, also known as the Jabber ID (JID).

Feedback Examples

- favorites names all
  returns
  ```
  favorites names 0. name:"Evergreen" sys_label:"" type:video
  favorites names 1. name:"Lab" sys_label:"groupseries" type:video
  favorites names 2. name:"Magnolia" sys_label:"" type:video
  favorites names 3. name:"Vineyard" sys_label:"" type:multicodec
  favorites names all done
  ```
- favorites names all 0 1
  returns
  ```
  favorites names 0. name:"Evergreen" sys_label:"" type:video
  favorites names 1. name:"Vineyard" sys_label:"" type:multicodec
  favorites names all 0 1 done
  ```

If an entry is an ITP system, the address parameter returns information about each of its codecs. A sys_label attribute is also returned to identify the endpoint types.

- favorites address "Vineyard" “”
  returns
  ```
  favorites address 0. name:"Vineyard" sys_label:"" codec:1 h323_spd:384
  h323_num: h323_ext:44042
  favorites address 1. name:"Vineyard" sys_label:"" codec:2 h323_spd:384
  h323_num: h323_ext:44043
  favorites address 2. name:"Vineyard" sys_label:"" codec:3 h323_spd:384
  h323_num: h323_ext:44044
  favorites address name:"Vineyard" sys_label:"" done
  ```

Limitations

None

Comments

You do not need to enclose a value in quotes unless it contains a space.

gatekeeperip

Gets or sets the IP address of the H.323 gatekeeper.

Syntax

```
gatekeeperip get
gatekeeperip set ["xxx.xxx.xxx.xxx"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the gatekeeper IP address when followed by the `"xxx.xxx.xxx.xxx"` parameter. To erase the current setting, omit `"xxx.xxx.xxx.xxx"`.
`"xxx.xxx.xxx.xxx"`	IP address of the gatekeeper.

Feedback Examples

- `gatekeeperip set 192.168.1.205`
  returns
  `gatekeeperip 192.168.1.205`
- `gatekeeperip get`
  returns
  `gatekeeperip 192.168.1.205`

Limitations

None

Comments

The `gatekeeperip get` command feedback may include the port number after the IP address.

gdsdirectory

Gets or sets options for the Polycom Global Directory Service (GDS).

Syntax

```
gdsdirectory <get|on|off|status>
```

Parameter	Description
`get`	Returns the current setting.
`On`	Enables GDS registration.
`Off`	Disables GDS registration.
`status`	Returns the current GDS registration status.

Feedback Examples

- `gdsdirectory get`
  returns
  `gdsdirectory off`
- `gdsdirectory on`
  returns
  `gdsdirectory on`
- `gdsdirectory status`
  returns
  `gdsdirectory online`

Limitations

None

Comments

The `gdsdirectory` command is supported only when H.323 is enabled.

gdspassword

Sets the password for Polycom GDS registration.

Syntax

```
gdspassword set <"password">
```

Parameter	Description
`set`	Sets the GDS registration password.
`"password"`	The GDS password when using the `set` command.

Feedback Examples

- `gdspassowrd set "polycomuser 01"`
  returns
  `gdspassword failed`
- `gdspassword set "polycomuser01"`
  returns
  `gdspassword accepted`

Limitations

None

Comments

The `gdspassword` command is supported only when H.323 is enabled.

gdsserverip

Gets or sets the GDS server IP address.

Syntax

```
gdsserverip <get|set> <"ipaddress">
```

Parameter	Description
`get`	Returns the current setting.
`set`	Specifies the IP address.
`"ipaddress"`	IP address to use with `set` command.

Feedback Examples

- `gdsserverip get`
  returns
  `gdsserverip 192.168.1.1`
- `gdsserverip set 192.168.1.1`
  returns
  `gdsserverip 192.168.1.1`

Limitations

None

Comments

The `gdsserverip` command is supported only when H.323 is enabled.

gendial

Generates DTMF dialing tones.

Syntax

```
gendial <{0..9}|#|*>
```

Parameter	Description
`{0..9}`	Generates the DTMF tone corresponding to telephone buttons 0-9.
`#`	Generates the DTMF tone corresponding to a telephone # button.
`*`	Generates the DTMF tone corresponding to a telephone * button.

Feedback Examples

- `gendial 2`
  returns
  `gendial 2`
  and causes the system to produce the DTMF tone corresponding to a telephone's 2 button

Limitations

None

Comments

None

generatetone

Turns the test tone on or off. The tone is used to check the monitor audio cable connections or to monitor the volume level.

Syntax

```
generatetone <on|off>
```

Parameter	Description
`on`	Turns on the test tone.
`off`	Turns off the test tone.

Feedback Examples

- `generatetone on`
  returns
  `generatetone on`
  and the system produces a test tone
- `generatetone off`
  returns
  `generatetone off`
  and the system stops producing a test tone

Limitations

None

Comments

None

getcallstate

Gets the state of the calls in the current conference.

Syntax

```
getcallstate
```

Additional Restrictions

None

Feedback Examples

- `getcallstate`
  returns
  ```
  cs: call[34] speed[384] dialstr[192.168.1.101] state[connected]
  cs: call[1] inactive
  cs: call[2] inactive
  ```

Limitations

None

Comments

None

See Also

To register the shell session to receive notifications about call state activities, see the callstate command.

getconfiguredipaddress

Retrieves the currently configured IPv4 address from the system.

Syntax

`getconfiguredipaddress`

Additional Restrictions

None

Feedback Examples

- `getconfiguredipaddress`
  returns
  `getconfiguredipaddress 1.2.3.4`

Limitations

None

Comments

`getconfiguredipaddress` returns the system's configured IPv4 address regardless of the status of the LAN connection. This differs from the `ipaddress get` command, which returns the system's IP address if it has an active LAN connection (if not, `0.0.0.0` returns).

The definition of "configured IPv4 address" depends on how the IPv4 address is configured:

- If the IP address is set manually, the configured IP address is returned regardless if the LAN connection is active.
- If the IP address is obtained automatically, the currently-assigned address is returned. `0.0.0.0` returns if there is no active connection.

globaldir

Retrieves global directory entries. Polycom recommends that you use this command for the Global Directory. This command supports all global directory types, including Polycom GDS and LDAP.

Syntax

```
globaldir "search_string"
globaldir "search_string" "size"
globaldir entry "UID"
globaldir range "start_no" "end_no"
globaldir "search_string" range "start_no" "end_no"
```

Multi-Tiered Directory Commands

```
globaldir grouplist
globaldir grouplist "UID"
globaldir grouplist "UID" "search_string"
globaldir grouplist "UID" range "start_no" "end_no"
globaldir grouplist "UID" "search_string" range "start_no" "end_no"
```

Parameter	Description
"search string"	The name or string to use for the search. If the string has a space, you must enclose it in quotations.
"size"	Specifies the maximum number of entries to return in the search.
entry	Retrieves information about a specific site.
grouplist	Retrieves the top tier of the group list when using a multitiered directory on Polycom® RealPresence® Resource Manager.
"UID"	Unique identifier associated with a site or group. For example: ldap#g#f82be96eea3bd644a1963dc7fdf45011 The complete UID must be specified.
range	Returns local directory entries numbered "start_no" through "stop_no". Requires two integers.
"start_no"	Specifies the beginning of the range of entries to return.
"stop_no"	Specifies the end of the range of entries to return.

Feedback Examples

LDAP

- `globaldir sd 5`

  returns

  ```
  globaldir 0. SD-Austin-01@polycom.com:
  ldap#g#840780b28ef4234f84f64298909aca07:site
  globaldir 1. SD-Austin-02@ polycom.com:
  ldap#g#8852f4c7cb6d9b4fab7e53e2730a5219:site
  globaldir 2. SD-Dallas-01@ polycom.com:
  ldap#g#83840767145bf04a9ce2b307af6d5688:site
  globaldir 3. SD-Dallas-02@ polycom.com:
  ldap#g#158aa86dd780ca4f8731fcfd627e05ad:site
  globaldir 4. SD-Houston-01@ polycom.com:
  ldap#g#e2859e0318bca145ba9b6f641e7f39d2:site
  globaldir 5. SD-Houston-02@ polycom.com:
  ldap#g#f82be96eea3bd644a1963dc7fdf45011:site
  globaldir sd 5 done
  ```

- `globaldir sd`

  returns

  ```
  globaldir 0. SD-Austin-01@polycom.com:
  ldap#g#840780b28ef4234f84f64298909aca07:site
  globaldir 1. SD-Austin-02@ polycom.com :
  ldap#g#8852f4c7cb6d9b4fab7e53e2730a5219:site
  through
  globaldir 401. SD-Wyoming-01@ polycom.com:
  ldap#g#3e98beb689622445af6f35bb0634ea02:site
  globaldir 402. SD-Wyoming-02@ polycom.com:
  ldap#g#81b735ce3111c445b85c0d0ddf3fd7a4:site
  globaldir sd done
  ```

GDS

- `globaldir gro 5`

  returns

  ```
  globaldir 0. Group Conf Room : gds#485:site
  globaldir 1. 1: gds#484:site
  globaldir 2.  2 : gds#466:site
  globaldir 3.  3 : gds#512:site
  globaldir 4.  Austin : gds#474:site
  globaldir 5.  Boston : gds#394:site
  ```

- `globaldir entry gds#485`

  returns

  ```
  globaldir 0. " Group Conf Room " h323_spd:1024 h323_num:10.223.17.147
  h323_ext: : site
  globaldir entry gds#485 done
  ```

RANGE

- `globaldir range 0 9`

  returns

  ```
  globaldir 0. AUSTIN LAB : ldap#g#2f83d8e0542dc74fac5c2f6e55035cff:group
  globaldir 1. Admin Admin : ldap#g#589feda2e097073b52134c7984ca6b44:site
  globaldir 2. Admin2 Admin2 : ldap#g#e6b660a112b25d4cb2067243e73da458:site
  globaldir 3. G7500 : ldap#g#0410894cfa213c418df5bd1226d46491:group
  globaldir 4. Group Series : ldap#g#d62644529aae1643ac7b418b1e404fe4:site
  ```

```
globaldir 5. HDX : ldap#g#011d8db58de14d48838549c5e0ec7465:group
globaldir 6. HDX8000 : ldap#g#38317b15022dc94f83650937c8aa0a48:group
globaldir 7. HDX9000 : ldap#g#5b97459113158744a3989d0bb40ce89e:group
globaldir 8. HDX_MISC : ldap#g#2331576d60cf9948a09860946f38a42b:group
globaldir 9. Sams 700 : ldap#g#35086aa0ecc9014facdcaa89bd34ccf6:site
globaldir range 0 9 done
```

- `globaldir gro range 0 9`
  returns
  ```
  globaldir 0. Group : ldap#g#35086aa0ecc9014facdcaa89bd34ccf6:site
  globaldir 1. Group 9006 : ldap#g#e64ffc28a13917488dec8ac97959c80f:site
  globaldir 2. Group GS300 : ldap#g#f7474445f7a8cc4d8221e7f452233446:site
  globaldir 3. Group GS700 : ldap#g#7922434fc77b6442bd74643f337f7a8e:site
  globaldir 4. Group HDX8006A : ldap#g#578b37ab9167d343853e4200145e119c:site
  globaldir 5. Group HDX8006B : ldap#g#2ce9b1cf64090e41a0b3e9b42a11edd5:site
  globaldir 6. Group HDX8006C : ldap#g#4275fd987e12e445bde9bcbb551dc7e8:site
  globaldir 7. Group HDX9004A : ldap#g#f3030565ec10bf4bbbfd1f77e1bdc483:site
  globaldir 8. Group HDX9004B : ldap#g#3e0b4c247225014682dbdebc5d6d935b:site
  globaldir 9. Group Saturn : ldap#g#5cb47f04e402d7478631ad45b5e6b493:site
  globaldir group range 0 9 done
  ```

MULTI-TIERED DIRECTORY

- `globaldir grouplist`
  returns
  ```
  globaldir 0. Admin Admin:ldap#g#589feda2e097073b52134c7984ca6b44:site
  globaldir 1. Admin2 Admin2:ldap#g#e6b660a112b25d4cb2067243e73da458:site
  globaldir 2. G7500:ldap#g#0410894cfa213c418df5bd1226d46491:group
  globaldir 3. Group Series:ldap#g#011d8db58de14d48838549c5e0ec7465:group
  globaldir 4. HDX_MISC:ldap#g#2331576d60cf9948a09860946f38a42b:group
  globaldir 5. Sams 9006:ldap#g#e64ffc28a13917488dec8ac97959c80f:site
  globaldir 6. Sams Saturn:ldap#g#5cb47f04e402d7478631ad45b5e6b493:site
  globaldir grouplist done
  ```
- `globaldir grouplist ldap#g#011d8db58de14d48838549c5e0ec7465`
  returns
  ```
  globaldir 0. HDX8000:ldap#g#38317b15022dc94f83650937c8aa0a48:group
  globaldir 1. HDX9000:ldap#g#5b97459113158744a3989d0bb40ce89e:group
  globaldir grouplist ldap#g#011d8db58de14d48838549c5e0ec7465 done
  ```
- `globaldir grouplist ldap#g#0410894cfa213c418df5bd1226d46491 boston`
  returns
  ```
  globaldir 0. Boston GS300:ldap#g#f7474445f7a8cc4d8221e7f452233446
  globaldir grouplist ldap#g#0410894cfa213c418df5bd1226d46491 boston done
  ```
- `globaldir grouplist range 0 6`
  returns
  ```
  globaldir 0. Admin Admin:ldap#g#589feda2e097073b52134c7984ca6b44:site
  globaldir 1. Admin2 Admin2:ldap#g#e6b660a112b25d4cb2067243e73da458:site
  globaldir 2. G7500:ldap#g#0410894cfa213c418df5bd1226d46491:group
  globaldir 3. Group Series:ldap#g#011d8db58de14d48838549c5e0ec7465:group
  globaldir 4. HDX_MISC:ldap#g#2331576d60cf9948a09860946f38a42b:group
  globaldir 5. Sams 9006:ldap#g#e64ffc28a13917488dec8ac97959c80f:site
  globaldir 6. Sams Saturn:ldap#g#5cb47f04e402d7478631ad45b5e6b493:site
  globaldir grouplist range 0 6 done
  ```

- `globaldir grouplist ldap#g#0410894cfa213c418df5bd1226d46491 range 0 1`
  returns
  ```
  globaldir 0. GS700:ldap#g#d62644529aae1643ac7b418b1e404fe4:group
  globaldir 1. Sams GS300:ldap#g#f7474445f7a8cc4d8221e7f452233446:site
  globaldir grouplist ldap#g#0410894cfa213c418df5bd1226d46491 range 0 1 done
  ```
- `globaldir grouplist ldap#g#e6b660a112b25d4cb2067243e73da458 austin range 0 9`
  returns
  ```
  globaldir 0. Austin 700 : ldap#g#35086aa0ecc9014facdcaa89bd34ccf6:site
  globaldir 1. Austin 9006 : ldap#g#e64ffc28a13917488dec8ac97959c80f:site
  globaldir 2. Austin GS300 : ldap#g#f7474445f7a8cc4d8221e7f452233446:site
  globaldir 3. Austin GS700 : ldap#g#7922434fc77b6442bd74643f337f7a8e:site
  globaldir 4. Austin HDX8006A :
  ldap#g#578b37ab9167d343853e4200145e119c:site
  globaldir 5. Austin HDX8006B :
  ldap#g#2ce9b1cf64090e41a0b3e9b42a11edd5:site
  globaldir 6. Austin HDX8006C :
  ldap#g#4275fd987e12e445bde9bcbb551dc7e8:site
  globaldir 7. Austin HDX9004A :
  ldap#g#f3030565ec10bf4bbbfd1f77e1bdc483:site
  globaldir 8. Austin HDX9004B :
  ldap#g#3e0b4c247225014682dbdebc5d6d935b:site
  globaldir 9. Austin Saturn : ldap#g#5cb47f04e402d7478631ad45b5e6b493:site
  globaldir grouplist ldap#g#e6b660a112b25d4cb2067243e73da458 austin range 0
  9 done
  ```

Limitations

None

Comments

Multitiered directory commands are supported only when using the RealPresence Resource Manager LDAP function.

Using multitiered directory commands on a system that does not support multitiered directory returns the following message: `error: command not supported in current configuration`.

h323authenticate enable

Enables or disables H.323 authentication.

Syntax

```
h323authenticate enable <get|true|false>
```

Parameter	Description
`get`	Returns the current setting.
`true`	Enables H.323 authentication.
`false`	Disables H.323 authentication.

Feedback Examples

- `h323authenticate enable get`
  returns
  `h323authenticate enable true`
- `h323authenticate enable true`
  returns
  `h323authenticate enable true`
- `h323authenticate enable false`
  returns
  `h323authenticate enable false`

Limitations

None

Comments

None

h323authenticate name

Sets the H.323 name to use to identify the system.

Syntax

```
h323authenticate name get
h323authenticate name "name"
```

Parameter	Description
`get`	Returns the current H.323 name.
`True`	The H.323 name to use to identify the system.

Feedback Examples

- `h323authenticate name get`
  returns
  `h323authenticate name Administrator`
- `h323authenticate name Administrator`
  returns
  `h323authenticate name Administrator`

Limitations

None

Comments

None

h323authenticate password

Sets the password for H.323 authentication.

Syntax

```
h323authenticate password set "password"
```

Parameter	Description
`"password"`	Password for H.323 authentication.

Feedback Examples

- `h323authenticate password set Polycom`
  returns
  `h323authenticate password accepted`

Limitations

None

Comments

None

h323name

Gets or sets the system's H.323 name.

Syntax

```
h323name get
h323name set ["H.323name"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the H.323 name when followed by the `"H.323name"` parameter. To erase this setting, omit the `"H.323name"` parameter.
`"H.323name"`	Character string specifying the H.323 name. Use quotation marks around strings that contain spaces. For example: `"Demo"`

Feedback Examples

- `h323name set My`
  returns
  `h323name my`
- `h323name set "Demo"`
  returns
  `h323name "Demo"`
- `h323name get`
  returns
  `h323name "Demo"`

Limitations

None

Comments

None

hangup

Hangs up the video call.

Syntax

```
hangup video ["callid"]
hangup all
```

Parameter	Description
`video`	Disconnects the current video call. If the "`callid`" parameter is omitted, the system disconnects all video far sites in the call.
`all`	Disconnects all video and audio sites in the call.

Feedback Examples

- `hangup video`
  returns
  `hanging up video`
- `hangup video 42`
  returns
  `hanging up video`
  and disconnects the specified site, leaving other sites connected
- If `callstate register` is used for notifications,
  `hangup video 42`
  returns
  ```
  hanging up video
  cleared: call[42]
  dialstring[IP:192.168.1.101 NAME:Demo]
  ended: call[42]
  ```
  and disconnects the specified site, leaving other sites connected

Limitations

None

Comments

After sending the `hangup` command, if registered for notification, the feedback response will notify that the call has ended. The feedback response can take up to 15 seconds.

hostname

Gets or sets the LAN host name, which is assigned to the system for TCP/IP configuration and can be used in place of an IP address when dialing IP calls.

Syntax

```
hostname get
hostname set ["hostname"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the system's LAN host name when followed by the `"hostname"` parameter.
`"hostname"`	Character string specifying the LAN host name of the system. The LAN host name follows these format rules: Starts with a letter (A-a to Z-z). It is not case sensitive. Ends with a letter (A-a to Z-z) or a number (0 to 9). May include letters, numbers, and a hyphen. May not be longer than 36 characters. Note: The LAN host name is initialized during the setup wizard sequence. The LAN host name is the same as the system name, if the system name conforms to the rules above. If the system name does not conform to these rules, the invalid characters are removed from the system name.

Feedback Examples

- `hostname set`
  returns
  `hostname ADMIN`
- `hostname set "My"`
  returns
  `hostname My`
- `hostname get`
  returns
  `hostname My`

Limitations

None

Comments

A LAN host name is required; it cannot be deleted or left blank.

After making a change, you must restart the system for the setting to take effect.

importdirectory

Imports local directory information in XML format.

Syntax

```
importdirectory
<import data line 1>
<import data line 2>
<import data line 3>
.
.
.
importcomplete
```

Additional Restrictions

None

Feedback Example

- `importdirectory`

  returns

```
<?xml version="1.0" encoding="UTF-8" ?>
<addresses>
<entrytype type="entry"  name="dawn" filename="dawn" uniqueid="local:26">
<address filename="dawn" langid="" displayname="dawn" name="dawn">
<h323 address="192.168.1.120"
speed="0"/>
<sip address="192.168.1.120"
 speed="0"/>
<category category="CONTACTS"/>
</address>
</entrytype>
<entrytype type="entry"  name="dawn " filename="dawn "
uniqueid="local:28">
<address filename="dawn
" langid="
" displayname="dawn
" name="dawn ">
<h323 address="192.168.1.120"
 speed="0"/>
<sip address="192.168.1.120"
 speed="0"/>
<category category="CONTACTS"/>
</address>
</entrytype>
<address filename="test
" langid="
" displayname="test
" name="test ">
<multisitename meeting_name="test " />
<multisitespeed meeting_speed="auto"/>
<multisitename0 site_name_0="dawn "/>
<mulitsitetype0 site_type_0="2" type_0="1000"/>
<mulitsiteprefcalltype0 pref_call_type_0="H323"/>
<multisiteuniqueid0 unique_id_0="local:28"/>
```

```
<multisitename1 site_name_1="dawn2 "/>
<mulitsitetype1 site_type_1="2" type_1="1000"/>
<mulitsiteprefcalltype1 pref_call_type_1="H323"/>
<multisiteuniqueid1 unique_id_1="local:30"/>
<multisitename2 site<?xml version="1.0" encoding="UTF-8" ?>
<addresses>
<entrytype type="entry"  name="dawn" filename="dawn" uniqueid="local:26">
<address filename="dawn
" langid="
" displayname="dawn
" name="dawn">
<h323 address="192.168.1.120"
 speed="0"/>
<sip address="192.168.1.120"
 speed="0"/>
<category category="CONTACTS"/>
</address>
</entrytype>
<entrytype type="entry"  name="dawn " filename="dawn "
uniqueid="local:28">
<address filename="dawn
" langid="
" displayname="dawn
" name="dawn ">
<h323 address="192.168.1.120"
 speed="0"/>
<sip address="192.168.1.120"
 speed="0"/>
<category category="CONTACTS"/>
</address>
</entrytype>
<address filename="test
" langid="
" displayname="test
" name="test ">
<multisitename meeting_name="test " />
<multisitespeed meeting_speed="auto"/>
<multisitename0 site_name_0="dawn "/>
```

```
<mulitsitetype0 site_type_0="2" type_0="1000"/>
<mulitsiteprefcalltype0 pref_call_type_0="H323"/>
<multisiteuniqueid0 unique_id_0="local:28"/>
<multisitename1 site_name_1="dawn2 "/>
<mulitsitetype1 site_type_1="2" type_1="1000"/>
<mulitsiteprefcalltype1 pref_call_type_1="H323"/>
<multisiteuniqueid1 unique_id_1="local:30"/>
<multisitename2 site_name_2="dawn3 "/>
<mulitsitetype2 site_type_2="2" type_2="1000"/>
<mulitsiteprefcalltype2 pref_call_type_2="H323"/>
<multisiteuniqueid2 unique_id_2="local:29"/>
</address>
</entrytype>
<entrytype type="group"  name="test1" filename="test1"
uniqueid="local:38">
<address filename="test1
" langid="
" displayname="test1
" name="test1">
<multisitename meeting_name="test1" />
<multisitespeed meeting_speed="auto"/>
</address>
</entrytype>
</addresses>_name_2="dawn3 "/>
<mulitsitetype2 site_type_2="2" type_2="1000"/>
<mulitsiteprefcalltype2 pref_call_type_2="H323"/>
<multisiteuniqueid2 unique_id_2="local:29"/>
</address>
</entrytype>
<entrytype type="group"  name="test1" filename="test1"
uniqueid="local:38">
<address filename="test1
" langid="
" displayname="test1
" name="test1">
<multisitename meeting_name="test1" />
<multisitespeed meeting_speed="auto"/>
</address>
```

```
</entrytype>
</addresses>
```

- `importcomplete`
  returns
  `import succeeded`

Limitations

None

Comments

A restart of the system is required after successfully importing directory information and occurs automatically after the import is complete.

When importing XML-formatted data, the imported data must be in the same format as was obtained from the system through the `exportdirectory` command or the export directory utility in the web interface. When importing data back into the system, use the data in its entirety (not edited in any form). The system may use the checksum utility to verify of integrity of the data when it is imported back into the system.

Duplicate entries are overwritten; other entries in the imported directory are added into the system's local directory.

All of the lines entered into the session after `importdirectory` is issued are interpreted as directory data.

You must include the `importcomplete` command as the last entry. Issuing the `importcomplete` command on its own line indicates that the directory import is complete.

If no data is received for 60 seconds during import, the import ends, and an `importdirectory timed out` error response is sent to the API session. All previous data entered is ignored.

Attempts to export and import directory information between different systems might fail. The message `import failed` indicates that the system was not able to import the information.

See Also

See the exportdirectory command.

importprofile

Imports system and user profile information in a CSV format.

Syntax

```
importprofile
<import data line 1>
<import data line 2>
<import data line 3>
. . .
importcomplete
```

Additional Restrictions

None

Feedback Example

- `importprofile`

  returns

```
import started
profileversion,0.2
system.info.eulafile,eula
system.info.hardwareversion,9
system.info.humanreadablemodel,RealPresence Group 500
system.info.humanreadableplatform,GROUPSERIES
system.info.humanreadableversion,Dev - 4.1.3-0
system.info.plcmstandardversion,Dev - 4.1.3-0
system.info.serialnumber,8213130FE433CV
audio.lineIO.lineinechocanceller,"False"
audio.volume.speakervolume,"46"
comm.Firewall.fixedportstcphigh,"3241"
comm.Firewall.fixedportsudphigh,"3301"
comm.NICs.H323Nic.h323extension,"177704997"
comm.NICs.H323Nic.h323name,"G7500 177704997"
comm.NICs.SipNic.bfcptransportprotocol,"Prefer_UDP"
comm.NICs.SipNic.thirdpartyinterop.ocs.sipuuid,"d503b976-c62f-5484-82c0-6
4a47963      18d1"
comm.Qos.tos.tosaudio,"5"
comm.Qos.tos.tosfecc,"3"
comm.Qos.tos.tosoam,"0"
comm.Qos.tos.tosvideo,"4"
location.country,"United States"
location.language,"ENGLISHUS"
pm.monRoleAuto,"True"
pm.monitor[1].enable,"True"
softupdate.url,"http://builds.softupdate.com/~test/softupdate      /"
sourceman.camera[1].autowhitebalancegainb,"33"
sourceman.camera[1].autowhitebalancegainr,"37"
sourceman.camera[1].backlightcomp,"False"
sourceman.camera[1].brightness,"11"
sourceman.camera[1].contrast,"13"
sourceman.camera[1].name,"Main"
sourceman.camera[1].role,"People"
```

```
sourceman.camera[1].saturation,"6"
sourceman.camera[1].sharpness,"3"
sourceman.camera[1].videoquality,"Sharpness"
sourceman.camera[1].whitebalancemode,"atw"
video.monitor[1].Resolution,"1920x1080p 60Hz"
video.monitor[2].Resolution,"1920x1080p 60Hz"

importcomplete

importprofile succeeded
```

Limitations

None

Comments

When importing profile data, the imported data must be in the same format as was obtained from the system using the `exportprofile` command. When importing profile data back into the system, use the data in its entirety (not edited in any form). The system may use the checksum utility to verify of integrity of the data when it is imported.

`importprofile done` indicates that all the profile data has been imported.

A restart of the system is required after successfully importing system and user profile information.

You must include the `importcomplete` command as the last entry. Issuing the `importcomplete` command on its own line indicates that the profile import is complete. If no data is received for 60 seconds during import, the import ends, and an `importprofile timed out` error response displays. All previous data entered is ignored.

The system might not allow certain parameters, such as passwords or software build information, to be updated during the import process (logs messages indicate if a parameter is ignored).

Exporting a profile on one system model and importing the profile on another model is not supported. Attempts to export and import profile information between different systems might also fail. The message `importprofile failed` indicates that the system was not able to import the information.

See Also

See the exportprofile command.

ipaddress

Gets or sets the LAN IP address (IPv4) of the system.

Syntax

```
ipaddress get
ipaddress set "xxx.xxx.xxx.xxx"
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the LAN IP address to the `"xxx.xxx.xxx.xxx"` parameter. This setting can only be changed when DHCP is off.
`"xxx.xxx.xxx.xxx"`	IP address of the system.

Feedback Examples

- `ipaddress set 192.168.1.101`
  returns
  `ipaddress 192.168.1.101`
- `ipaddress get`
  returns
  `ipaddress 192.168.1.101`

Limitations

None

Comments

Use this command when you need to assign a static IP address to your system.

You must restart the system for the setting to take effect.

lanport

Gets or sets the LAN port settings of the system.

Syntax

```
lanport <get|auto|10hdx|10fdx|100hdx|100fdx|1000hdx|1000fdx>
```

Parameter	Description
`get`	Returns the current setting.

`auto|10hdx|10fdx|`
`100hdx|100fdx|`
`1000hdx|1000fdx`
	Sets the LAN speed and duplex mode. This parameter is not allowed while in a call.
	`auto`: Automatically negotiates the LAN speed and duplex mode.
	`10hdx`: 10 Mbps, half duplex
	`10fdx`: 10 Mbps, full duplex
	`100hdx`: 100 Mbps, half duplex
	`100fdx`: 100 Mbps, full duplex
	`1000hdx`: 1000 Mbps, half duplex
	`1000fdx`: 1000 Mbps, full duplex

Feedback Examples

- `lanport auto`
  returns
  `lanport auto`
  `restart system for changes to take effect. restart now? <y,n>`
- `lanport get`
  returns
  `lanport auto`

Limitations

None

Comments

After making a change, you are prompted to restart the system.

ldapauthenticationtype

Gets or sets the authentication type required to authenticate with an LDAP server.

Syntax

```
ldapauthenticationtype get
ldapauthenticationtype set <anonymous|basic>
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the authentication type of an LDAP server. **Note:** This parameter does not change the setting on the server. Instead, this parameter changes how the Polycom system recognizes the server.
`anonymous`	Specifies "`anonymous`" as the authentication type of an LDAP server.
`basic`	Specifies "`basic`" as the authentication type of an LDAP server.
`ntlm`	Specifies "`ntlm`" as the authentication type of an LDAP server. This is the default setting.

Feedback Examples

- `ldapauthenticationtype get`
  returns
  `ldapauthenticationtype anonymous`
- `ldapauthenticationtype set basic`
  returns
  `ldapauthenticationtype basic`
- `ldapauthenticationtypeset ntlm`
  returns
  `ldapauthenticationtype ntlm`

Limitations

None

Comments

None

ldapbasedn

Gets or sets the base distinguished name (DN) of an LDAP server.

Syntax

```
ldapbasedn get
ldapbasedn set ["base dn"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the base DN of an LDAP server. To erase the current setting, omit the "`base dn`" parameter. **Note:** This parameter does not change the setting on the server. Instead, this parameter changes how the Polycom system recognizes the server.
`"base dn"`	Specifies the base DN of an LDAP server. Valid characters include: Unicode (ISO-10646) characters, including IA5/ASCII characters and extended characters such as é, Ø, and å. **Note:** To avoid LDAP registration issues, make sure the base DN is at least one level deeper than your domain. For example, set `ou=users,dc=example,dc=com` instead of `dc=example,dc=com`.

Feedback Examples

- `ldapbasedn get`
  returns
  `ldapbasedn dc=hardware,dc=domain,dc=Polycom,dc=com`
  where:
  `dc`=domain component
- `ldapbasedn set dc=software,dc=domain,dc=Polycom,dc=com`
  returns
  `ldapbasedn dc=software,dc=domain,dc=Polycom,dc=com`
  where:
  `dc`=domain component

Limitations

None

Comments

None

ldapbinddn

Gets or sets the bind DN for LDAP Simple Authentication.

Syntax

```
ldapbinddn get
ldapbinddn set ["bind dn"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the bind DN for LDAP Simple Authentication. To erase the current setting, omit the "`bind dn`" parameter. Note: This parameter does not change the setting on the server. Instead, this parameter changes how the Polycom system recognizes the server.
`"bind dn"`	Specifies the bind DN of an LDAP server. Valid characters include: Unicode (ISO-10646) characters, including IA5/ASCII characters and extended characters such as é, Ø, and å.

Feedback Examples

- `ldapbinddn get`
  returns
  `ldapbinddn cn=plcm admin1,ou=plcmsupport,ou=plcmhelp, dc=hardware,dc=domain,dc=polycom,dc=com`
  where:
  `cn=common name`
  `ou=organizational unit`
  `dc=domain component`
- `ldapbinddn set cn=plcm admin2,ou=plcmaccounts,ou=plcmservice, dc=hardware,dc=domain,dc=polycom,dc=com`
  returns
  `ldapbinddn cn=plcm admin2,ou=plcmaccounts,ou=plcmservice, dc=hardware,dc=domain,dc=polycom,dc=com`
  where:
  `cn=common name`
  `ou=organizational unit`
  `dc=domain component`

Limitations

None

Comments

None

ldapdirectory

Gets or sets the LDAP directory server setting.

Syntax

```
ldapdirectory <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables the LDAP directory server.
`no`	Disables the LDAP directory server. This is the default setting.

Feedback Examples

- `ldapdirectory get`
  returns
  `ldapdirectory yes`
- `ldapdirectory no`
  returns
  `ldapdirectory no`

Limitations

None

Comments

Each Polycom system supports a single global directory server at any given time. Therefore, enabling the LDAP directory server automatically disables any other global directory server, such as Polycom GDS, which is enabled.

If the Polycom GDS and another directory server are defined on the system, Polycom GDS becomes the default directory server after upgrading the system software.

ldapntlmdomain

Gets or sets the domain in which authentication takes place in the LDAP server.

Syntax

```
ldapntlmdomain get
ldapntlmdomain set ["domain"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the domain in which authentication takes place in the LDAP server. To erase the current setting, omit the "`domain`" parameter. **Note:** This parameter does not change the setting on the server. Instead, this parameter changes how the Polycom system recognizes the server.
`"domain"`	Specifies the domain in which authentication takes place in the LDAP server. Valid characters include: `0` through `9`, `a` through `z`, `A` through `Z`, hyphen `(-)`, and period (`.`) Note: The domain name cannot begin or end with a hyphen or a period.

Feedback Examples

- `ldapntlmdomain get`
  returns
  `ldapntlmdomain AUSTIN`
- `ldapntlmdomain set ANDOVER`
  returns
  `ldapntlmdomain ANDOVER`

Limitations

None

Comments

None

ldappassword

Sets the password for Simple or NT LAN Manager (NTLM) authentication of an LDAP server.

Syntax

```
ldappassword set ["password"]
```

Parameter	Description
`set`	Sets the password for Simple authentication of an LDAP server. To erase the current setting, omit the "`password`" parameter. Note: This parameter does not change the setting on the server. Instead, this parameter changes how the Polycom system recognizes the server.
`ntlm`	Specifies setting the password for NTLM authentication of an LDAP server.
`basic`	Specifies setting the password for Simple authentication of an LDAP server.
`"password"`	Specifies the password for Simple or NTLM authentication of an LDAP server. Valid characters include: Unicode (ISO-10646) characters, including IA5/ASCII characters and extended characters such as é, Ø, and å. **Note:** The server administrator may specify additional restrictions for password creation.

Feedback Examples

- `ldappassword set ntlm P!cmp@s5wd`
  returns
  `ldappassword NTLM P!cmp@s5wd`
- `ldappassword set basic P0!yc0mp@s5`
  returns
  `ldappassword BASIC P0!yc0mp@s5`

Limitations

None

Comments

None

ldapserveraddress

Gets or sets the LDAP server address.

Syntax

```
ldapserveraddress get
ldapserveraddress set ["address"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the IP address or the DNS name of an LDAP server. To erase the current setting, omit the "`address`" parameter. **Note:** This parameter does not change the setting on the server. Instead, this parameter changes how the Polycom system recognizes the server.

`"address"`	Specifies the IP address or the DNS name of an LDAP server.
	The DNS name requires alphanumeric characters. Valid characters include:
	`0` through `9`
	`a` through `z`
	`A` through `Z`
	-
	Note: The "-" character cannot be used as the first or last character in the DNS name.

Feedback Examples

- `ldapserveraddress get`
  returns
  `ldapserveraddress hardware.domain.polycom.com`
- `ldapserveraddress set software.domain.polycom.com`
  returns
  `ldapserveraddress software.domain.polycom.com`

Limitations

None

Comments

None

ldapserverport

Gets or sets the port number of an LDAP server.

Syntax

```
ldapserverport get
ldapserverport set ["port number"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the port number of an LDAP server. To erase the current setting, omit the "`port number`" parameter. **Note:** This parameter does not change the setting on the server. Instead, this parameter changes how the Polycom system recognizes the server.
`"port number"`	Specifies the port number of an LDAP server. The default setting is 389.

Feedback Examples

- `ldapserverport get`
  returns
  `ldapserverport 389`
- `ldapserverport set 636`
  returns
  `ldapserverport 636`

Limitations

None

Comments

None

ldapsslenabled

Gets or sets the Transport Layer Security (TLS) encryption state for LDAP operations.

Syntax

```
ldapsslenabled get
ldapsslenabled set [on|off]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the TLS encryption state for LDAP operations. **Note:** This parameter does not change the setting on the server. Instead, this parameter changes how the Polycom system recognizes the server.
`on`	Specifies "`on`" as the encryption state for LDAP operations. This is the default setting.
`off`	Specifies "`off`" as the encryption state for LDAP operations.

Feedback Examples

- `ldapsslenabled get`
  returns
  `ldapsslenabled off`
- `ldapsslenabled set on`
  returns
  `ldapsslenabled on`

Limitations

None

Comments

None

ldapusername

Gets or sets the user name for NTLM authentication of an LDAP server.

Syntax

```
ldapusername get
ldapusername set ["user name"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the user name for NTLM authentication of an LDAP server. To erase the current setting, omit the "`user name`" parameter. **Note:** This parameter does not change the setting on the server. Instead, this parameter changes how the Polycom system recognizes the server.
`"user name"`	Specifies the user name for NTLM authentication of an LDAP server. Valid characters include: Unicode (ISO-10646) characters, including IA5/ASCII characters and extended characters such as é, Ø, and å.

Feedback Examples

- `ldapusername get`
  returns
  `ldapusername jpolycom`
- `ldapusername set mpolycom`
  returns
  `ldapusername mpolycom`

Limitations

None

Comments

None

listen

Registers the API session to listen for the following events and statuses: incoming video calls, system sleep/awake state, and notifications when the registered state occurs.

Syntax

```
listen <video|sleep>
```

Parameter	Description
`video`	Instructs the session to listen for incoming video calls. When this event occurs, the message "`listen video ringing`" is received.
`sleep`	Instructs the session to listen for when the system goes into sleep mode. When this event occurs, the message "`listen going to sleep`" is received. When the system wakes up, the message "`listen waking up`" is received. Deprecated. Polycom recommends using `sleep register` instead of this command.

Feedback Examples

- `listen sleep`
  returns
  `listen sleep registered`
  to acknowledge that the session is now registered to listen for sleep mode
- `listen video`
  returns
  `listen video registered`
  to acknowledge that the session is now registered to listen for incoming video calls

Limitations

None

Comments

None

localdir

Retrieves local directory entries (Favorites).

Syntax

```
localdir <all>
localdir <search string>
localdir <search string> <size>
localdir entry <UID>
localdir range "start number" "end number"
localdir <search string> range "start number" "end number"
localdir grouplist
localdir grouplist <UID>
localdir grouplist <UID> <search string>
localdir grouplist range "start number" "end number"
```

Parameter	Description
*	Returns all site and group entries from the local directory in flat list form.
search string	The name or string to use for the search. If the string has a space you must enclose it in quotations.
size	Specifies the maximum number of entries to return in the search.
entry	Retrieves information about a specific site when using a site UID.
UID	Unique identifier associated with a site or group. The UID is the second part of the returned response that follows the colon (":"). You must use the complete UID.
grouplist	Displays entries in the specified group. Using this parameter alone retrieves the top group tier, including entries.
grouplist<UID>	Retrieves a list of sites and groups in the specified group.
grouplist<UID> <search string>	Retrieves directories that match the string inside of the specified group.
range	Returns directory entries in the range specified.
"start_no"	Specifies the beginning of the range of entries to return.
"stop_no"	Specifies the end of the range of entries to return.

Feedback Examples

- `localdir sd 5`
  returns
  `localdir 0. SD-Austin-01@polycom.com:`
  `local#840780b28ef4234f84f64298909aca07:site`
  `localdir 1. SD-Austin-02@polycom.com:`
  `local#8852f4c7cb6d9b4fab7e53e2730a5219:site`
  `localdir 2. SD-Dallas-01@polycom.com:`
  `local#83840767145bf04a9ce2b307af6d5688:site`
  `localdir 3. SD-Dallas-02@polycom.com:`
  `local#158aa86dd780ca4f8731fcfd627e05ad:site`
  `localdir 4. SD-Houston-01@polycom.com:`
  `local#e2859e0318bca145ba9b6f641e7f39d2:site`
  `localdir 5. SD-Houston-02@polycom.com:`
  `local#f82be96eea3bd644a1963dc7fdf45011:site`
  `localdir sd 5 done`
- `localdir entry ldap#g#8852f4c7cb6d9b4fab7e53e2730a5219`
  returns
  `localdir 0. "SD-Austin-02@polycom.com" sip_spd:Auto sip_num:sip:SEA18-09.106@vtc.austin.com:site`
  `localdir 1. "SD-Austin-02@polycom.com”h323_spd:Autoh323_num:`
  `h323_ext:12067406489:site`
  `localdir entry ldap#g#8852f4c7cb6d9b4fab7e53e2730a5219 done`
- `localdir grouplist`
  returns
  `localdir 0. Admin Admin:ldap#g#589feda2e097073b52134c7984ca6b44:site`
- `localdir 1. Admin2 Admin2:ldap#g#e6b660a112b25d4cb2067243e73da458:site`
- `localdir 2. G7500:ldap#g#0410894cfa213c418df5bd1226d46491:group`
- `localdir 3. Group Series:ldap#g#011d8db58de14d48838549c5e0ec7465:group`
- `localdir 4. HDX_MISC:ldap#g#2331576d60cf9948a09860946f38a42b:group`
- `localdir 5. Sams 9006:ldap#g#e64ffc28a13917488dec8ac97959c80f:site`
- `localdir 6. Sams Saturn:ldap#g#5cb47f04e402d7478631ad45b5e6b493:site`
- `localdir grouplist done`

Limitations

None

Comments

None

loglevel

Gets or sets the minimum log level of messages stored in the system's flash memory.

Syntax

```
loglevel get
loglevel set <debug|info|warning|error|critical>
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the debug level.
`debug`	Sets debug level to log all messages. The default.
`info`	Sets debug level to log all informational messages.
`warning`	Sets debug level to log all informational and warning messages.
`error`	Sets debug level to log all informational, warning, and error messages.
`critical`	Sets debug level to log all informational, warning, error, and critical messages.

Feedback Examples

- `loglevel get`
  returns
  `loglevel info`
- `loglevel set warning`
  returns
  `loglevel warning`
- `loglevel set error`
  returns
  `loglevel error`

Limitations

None

Comments

`warning` logs the fewest number of messages.

Polycom recommends leaving this setting at the default value of `debug`.

maxtimeincall

Gets or sets the maximum number of minutes allowed for call length.

Syntax

```
maxtimeincall get
maxtimeincall set [{0..2880}]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the maximum time for calls when followed by a parameter from `{0..2880}`. To erase the current setting, omit the time parameter or set it to `0`. The call will then stay up indefinitely.
`{0..2880}`	Maximum call time in minutes. Must be an integer in the range `{0..2880}`. The value in minutes will be rounded up to hours in the system, the valid hour values are 1_hour, 2_hours to 12_hours, 24_hours and 48_hours.

Feedback Examples

- `maxtimeincall set`
  returns
  `maxtimeincall <empty>`
- `maxtimeincall set 180`
  returns
  `maxtimeincall 180`
- `maxtimeincall get`
  returns
  `maxtimeincall 180`

Limitations

None

Comments

When the time has expired in a call, a message asks you if you want to hang up or stay in the call. If you do not answer within one minute, the call automatically disconnects.

monitor1screensaveroutput

Gets the current setting or sets whether to send black video or "No Signal" to Monitor 1 when its screen saver starts.

Syntax

```
monitor1screensaveroutput <get|black|no_signal>
```

Parameter	Description
`get`	Returns the current setting.
`black`	Sends black video to Monitor 1 when the system goes to sleep and the screen saver activates.
`no_signal`	Sends no signal to Monitor 1 when the system goes to sleep and the screen saver activates.

Feedback Examples

- `monitor1screensaveroutput black`
  returns
  `monitor1screensaveroutput black`
- `monitor1screensaveroutput no_signal`
  returns
  `monitor1screensaveroutput no_signal`
- `monitor1screensaveroutput get`
  returns
  `monitor1screensaveroutput no_signal`

Limitations

None

Comments

Setting Monitor 1 automatically sets Monitor 2 to the same setting.

See Also

See the monitor2screensaveroutput command.

monitor2screensaveroutput

Gets the current setting or sets whether to send black video or "No Signal" to Monitor 2 when its screen saver starts.

Syntax

```
monitor2screensaveroutput <get|black|no_signal>
```

Parameter	Description
`get`	Returns the current setting.
`black`	Sends black video to Monitor 2 when the system goes to sleep and the screen saver activates.
`no_signal`	Sends no signal to Monitor 2 when the system goes to sleep and the screen saver activates.

Feedback Examples

- `monitor2screensaveroutput black`
  returns
  `monitor2screensaveroutput black`
- `monitor2screensaveroutput no_signal`
  returns
  `monitor2screensaveroutput no_signal`
- `monitor2screensaveroutput get`
  returns
  `monitor2screensaveroutput no_signal`

Limitations

None

Comments

Setting Monitor 2 automatically sets Monitor 1 to the same setting.

See Also

See the monitor1screensaveroutput command.

mute

Gets or sets near- or far-site mute settings.

Syntax

```
mute <register|unregister>
mute near <get|on|off|toggle>
mute far get
```

Parameter	Description
`register`	Registers to receive notification when the mute mode changes.
`unregister`	Disables register mode.
`near`	Sets the command for the near site. Requires `on`, `off`, `toggle`, or `get`.
`get`	Returns the current setting for the near or far site.
`on`	Mutes the near site (`mute near on`).
`off`	Unmutes the near site (`mute near off`).
`toggle`	If mute near mode is `mute near on`, this switches to `mute near off`, and vice versa.
`far`	Returns the mute state of the far site system. Requires the parameter `get`.

Feedback Examples

- `mute register`
  returns
  `mute registered`
- `mute near on`
  returns
  `mute near on`
- `mute far get`
  returns
  `mute far off`

Limitations

None

Comments

In register mode, the system sends notification to the API session when the far or near site is muted or unmuted.

muteautoanswer

Gets or sets if the audio is muted for auto-answered calls. When this is on, your microphone is muted to prevent the far site from immediately hearing the near site.

Syntax

```
muteautoanswer <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Enables Mute Auto Answer Calls mode. The microphone will be muted when the system receives a call while in Auto Answer mode.
`no`	Disables Mute Auto Answer Calls mode. The microphone is not muted when the system automatically answers calls.

Feedback Examples

- `muteautoanswer yes`
  returns
  `muteautoanswer yes`
- `muteautoanswer no`
  returns
  `muteautoanswer no`
- `muteautoanswer get`
  returns
  `muteautoanswer no`

Limitations

None

Comments

None

natconfig

Gets or sets the NAT configuration.

Syntax

```
natconfig <get|auto|manual|off>
```

Parameter	Description
`get`	Returns the current setting.
`auto`	Specifies that the system is behind a NAT and that the system will automatically discover the public (WAN) address.
`manual`	Specifies that the system is behind a NAT. Requires you to assign the WAN address using the `wanipaddress` command.
`off`	Disables the option when the system is not behind a NAT.

Feedback Examples

- `natconfig auto`
  returns
  `natconfig auto`
- `natconfig manual`
  returns
  `natconfig manual`
- `natconfig off`
  returns
  `natconfig off`
- `natconfig get`
  returns
  `natconfig off`

Limitations

None

Comments

None

nath323compatible

Gets or sets whether the NAT is H.323 compatible.

Syntax

```
nath323compatible <get|yes|no>
```

Parameter	Description
`get`	Returns the current setting.
`yes`	Specifies that NAT is capable of translating H.323 traffic.
`no`	Specifies that NAT is not capable of translating H.323 traffic.

Feedback Examples

- `nath323compatible yes`
  returns
  `nath323compatible yes`
- `nath323compatible no`
  returns
  `nath323compatible no`
- `nath323compatible get`
  returns
  `nath323compatible no`

Limitations

None

Comments

None

nearloop

Activates or deactivates the Near End Loop test.

Syntax

```
nearloop <on|off>
```

Parameter	Description
`on`	Activates the Near End Loop, a complete internal test of the system.
`off`	Deactivates the Near End Loop.

Feedback Examples

- `nearloop on`
  returns
  `nearloop on`
- `nearloop off`
  returns
  `nearloop off`

Limitations

None

Comments

When Near End Loop is on, you can test the encoder/decoder on the system. This test is not available when you are in a call.

netstats

Returns network statistics for each call connection.

Syntax

```
netstats [{0..n}]
```

Parameter	Description
`{0..n}`	Call in a multipoint call, where `n` is the maximum number of calls supported by the system. `0` is the first site connected. If no call is specified, `netstats` returns information about the near site.

Feedback Examples

- `netstats 0`
  returns
  ```
  call:0 txrate:128 K rxrate:128 K pktloss:0 %pktloss:0.0%
  tvp:H.263 rvp:H.263 tvf:CIF rvf:CIF tap:G.722.1 rap:G.722.
  1 tcp:H.323 rcp:H.323 tcp:- rcp:-- tcf:-- rcf:H.239
  ```
  where:
  ```
  txrate = transmit clock rate
  rxrate = receive clock rate
  pktloss = number of packet loss/errors
  %pktloss = percentage of packet loss/errors
  tvp = transmit video protocol
  rvp = receive video protocol
  tvf = transmit video format
  rvf = receive video format
  tap = transmit audio protocol
  rap = receive audio protocol
  tcp = transmit comm protocol
  rcp = receive comm protocol
  tcp = transmit content protocol
  rcp = receive content protocol
  tcf = transmit content format
  rcf = receive content format
  ```

Limitations

None

Comments

Both `pktloss` and `%pktloss` report only numbers related to packet loss on the transmit. These numbers are not affected by packet loss on the Real-time Transport Protocol (RTP) that is received.

The number listed for `%pktloss` is not cumulative and is calculated every five seconds. The number listed for `pktloss` is calculated every 5 seconds and is cumulative.

nonotify

Unregisters the API client to receive status notifications.

Syntax

```
nonotify <callstatus|linestatus|mutestatus|screenchanges>
nonotify <sysstatus|sysalerts|vidsourcechanges>
```

Parameter	Description
`calendarmeetings`	Stops the system from receiving meeting reminders.
`callstatus`	Stops the system from receiving changes in call status, such as a connection or disconnection.
`linestatus`	Stops the system from receiving line status notifications.
`mutestatus`	Stops the system from receiving changes in audio mute status.
`screenchanges`	Stops the system from receiving notification when a user interface screen is displayed.
`sysstatus`	Stops the system from receiving system status notifications.
`sysalerts`	Stops the system from receiving system alerts.
`vidsourcechanges`	Stops the system from receiving notification of camera source changes.

Feedback Examples

- `nonotify callstatus`
  returns
  `nonotify callstatus success`
- If entered again,
  `nonotify callstatus`
  returns
  `info: event/notification not active:callstatus`
- `nonotify calendarmeetings`
  returns
  `nonotify calendarmeetings success`

Limitations

None

Comments

None

See Also

See the related notify command.

notify

Lists the types of notifications being received or registers to receive status notifications.

Syntax

```
notify
notify <callstatus|linestatus|mutestatus|screenchanges>
notify <sysstatus|sysalerts|vidsourcechanges>
notify calendarmeetings
```

Parameter	Description
`notify`	Lists the notification types that are being received in the following format: `registered for <num> notifications[:notification type>...]`
`calendarmeetings`	Registers the API client to receive meeting reminders.
`callstatus`	Registers the system to receive changes in call status, such as a connection or disconnection in the following format: `notification:callstatus:<call direction>:<call id>:<far site name>:<far site number>:<connection status>:<call speed>:<status-specific cause code from call engine>:<calltype>`
`linestatus`	Registers the system to receive line status notifications as they occur in the following format: `notification:linestatus:<direction>:<call id>:<line id>:<channel id>:<connection status>`
`mutestatus`	Registers the system to receive changes in audio mute status in the following format: `notification:mutestatus:<near or far>:<call id>:<site name>:<site number>:<mute status>`
`screenchanges`	Registers the system to receive notification when a user interface screen is displayed in the following format: `notification:screenchange:<screen name>:<screen def name>`
`sysstatus`	Registers the system to receive system status notifications in the following format: `notification:sysstatus:<sys parameter name>:<value1>[:<value2>...]`
`sysalerts`	Registers the system to receive system alerts in the following format: `notification:sysalert:<alert name>:<value1>[:<value2>...]`
`vidsourcechanges`	Registers the system to receive notification of camera source changes in the following format: `notification:vidsourcechange:<near or far>:<camera index>:<camera name>:<people or content>`

Feedback Examples

- `notify mutestatus`
  returns
  `notify mutestatus success`
  acknowledging that the session is registered to receive mutestatus notifications
- `notify callstatus`
  returns
  `notify callstatus success`
  acknowledging that the session is registered to receive callstatus notifications
- If entered again,
  `notify callstatus`
  returns
  `info: event/notification already active:callstatus`
- `notify`
  returns
  `registered for 2 notifications:callstatus:mutestatus`
- `notify calendarmeetings`
  returns
  `notify calendarmeetings success`

The following are examples of notifications that may be returned after registering to receive them.

- `notification:callstatus:outgoing:34:Polycom G7500 Demo:192.168.1.101:connected:384:0:videocall`
- `notification:mutestatus:near:near:near:near:muted`
- `notification:screenchange:systemsetup:systemsetup_a`
- `notification:vidsourcechange:near:1:Main:people`
- `notification:linestatus:outgoing:32:0:0:disconnected`
- `notification:vidsourcechange:near:none:none:content`
- `notification: calendarmeetings: AAAaAEFsZXguTWFjRG9uYWxkQHBvbHljb20uY29tAVEACIjMne2/ndgARgAAAADr9GlhsSjWEZBcAAKzMphJBwA4wicbtr3UEZArAKAk09LtAAACZpKWAADe7hJleQIOS7j2mzRJxkLKAAADI/G8AAAQ:Product Planning:10`

Limitations

None

Comments

The `notify callstatus` command registers the current API session for call status notifications. The API client receives call status notifications as a call progresses.

Registration for status notifications is session-specific. For example, registering for alerts in a Telnet session does not return alerts in a simultaneous RS-232 session with the same system.

Duplicate registrations produce another success response. The notify setting remains in effect, even if you restart the system or update the software with system settings saved.

See Also

See also the nonotify command and the callinfo command.

ntpmode

Sets the Network Time Protocol (NTP) server mode, which determines how the system connects to the time server to obtain time settings.

Syntax

```
ntpmode <get|auto|off|manual>
```

Parameter	Description
`get`	Returns the current time server mode.
`auto`	Sets the connection to the time server as automatic.
`off`	Turns off the connection to the time server.
`manual`	Sets the connection to the time server as manual. You can then use the `ntpserver` command to manually set the NTP server address.

Feedback Examples

- `ntpmode get`
  returns
  `ntpmode manual`
- `ntpmode auto`
  returns
  `ntpmode auto`
- `ntpmode off`
  returns
  `ntpmode off`
- `ntpmode manual`
  returns
  `ntpmode manual`

Limitations

None

Comments

None

ntpsecondaryserver

Sets the NTP server to use for time settings when the primary time server does not respond.

Syntax

```
ntpsecondaryserver get
ntpsecondaryserver set <"xxx.xxx.xxx.xxx"|server name">
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the IP address of the NTP server using the specified IP address or DNS name.

Feedback Examples

- `ntpsecondaryserver get`
  returns
  `ntpsecondaryserver 172.26.44.22`
- `ntpsecondaryserver set`
  returns
  `ntpsecondaryserver ""`
- `ntpsecondaryserver set 172.26.44.22`
  returns
  `ntpsecondaryserver 172.26.44.22`

Limitations

None

Comments

You must first set the `ntpmode` command to manual before using the `ntpsecondaryserver` command.

ntpserver

Sets the NTP server to use for time settings when the time server is set to manual.

Syntax

```
ntpserver get
ntpserver set <"xxx.xxx.xxx.xxx"|server name">
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the IP address of the NTP server using the specified IP address or DNS name.

Feedback Examples

- `ntpserver get`
  returns
  `ntpserver 192.168.1.205`
- `ntpserver set`
  returns
  `ntpserver <empty>`
- `ntpserver set 192.168.1.205`
  returns
  `ntpserver 192.168.1.205`

Limitations

None

Comments

You must first set the `ntpmode` command to manual before using the `ntpserver` command.

oobcomplete

Completes the onscreen setup instructions and restarts the system.

Syntax

`oobcomplete`

Feedback Examples

- `oobcomplete`
  returns
  `oobcomplete`

Limitations

None

Comments

The `oobcomplete` command is processed only when the system is in setup mode.

To execute `oobcomplete` successfully, the system name must be configured.

powerdown

Turns the system off. The `powerdown` command does not prompt the user to confirm and turns off the system with no other feedback returned. After the system turns off, it cannot be restarted remotely; it must be restarted manually.

Syntax

`powerdown`

Parameter	Description
`powerdown`	Turns the system off.

Feedback Examples

- `powerdown`
  returns
  `powerdown`

Limitations

None

Comments

None

preset

Sets the presets or goes (moves) to the presets for the near or far camera. Also registers or unregisters the API session to notify when the user sets or goes to presets.

Syntax

```
preset <register|unregister>
preset register get
preset far <go|set> <{0..15}>
preset near <go|set> <{0..99}>
```

Parameter	Description
`register`	Registers the system to give notification when the user or far site sets or goes to a preset. Returns the current preset registration state when followed by the `get` parameter.
`unregister`	Disables register mode.
`far`	Specifies the far camera. Requires a `set` or `go` parameter and a preset identifier.
`go`	Moves the camera to a camera preset. Requires a `"preset"` parameter.
`set`	Sets a camera preset. Requires a `"preset"` parameter.
`{0..15}`, `{0..99}`	Camera preset identifier. Must be an integer in the range `{0..15}` for a far-site camera or `{0..99}` for a near-site camera.
`near`	Specifies the near camera. Requires a `set` or `go` parameter and a preset identifier.

Feedback Examples

- `preset register`
  returns
  `preset registered`
- `preset near go 1`
  returns
  `preset near go 1`
  and moves the near-site camera to the preset 1 position
- `preset near set 2`
  returns
  `preset near set 2`
  and saves the current location/position of the near-site camera as preset 2

Comments

Up to 100 preset camera positions can be set.

provisionserveraddress

Gets or sets the IP address for the provisioning server.

Syntax

```
provisionserveraddress <get|set> <”Server Address”>
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the IP address of the provisioning server.
`“Server Address”`	Specifies the IP address to use when using the `set` command.

Feedback Examples

- `provisionserveraddress get`
  returns
  `provisionserveraddress 10.223.15.152`
- `provisionserveraddress set 192.168.1.1`
  returns
  `provisionserveraddress 192.168.1.1`

Limitations

None

Comments

None

provisionserverdomain

Gets or sets the domain name of the provisioning server.

Syntax

```
provisionserverdomain <get|set|"domain name">
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the domain name of the provisioning server.
`"Server Address"`	Specifies the domain name for the provisioning server address when using the `set` command.

Feedback Examples

- `provisionserverdomain get`
  returns
  `provisionserverdomain Polycom`
- `provisionserverdomain set corporate1`
  returns
  `provisionserverdomain corporate1`

Limitations

None

Comments

None

provisionserverenable

Gets or sets the current setting for the provisioning server.

Syntax

```
provisionserverenable <get|true|false>
```

Parameter	Description
`get`	Returns the current setting.
`true`	Enables the provisioning server.
`false`	Disables the provisioning server.

Feedback Examples

- `provisionserverenable get`
  returns
  `provisionserverenable false`
- `provisionserverenable true`
  returns
  `provisionserverenable true`
- `provisionserverenable false`
  returns
  `provisionserverenable false`

Limitations

None

Comments

None

provisionserverpassword

Sets the password for the provisioning server.

Syntax

```
provisionserverpassword set <”password”>
```

Parameter	Description
`set`	Sets the password for the provisioning server.
`“password”`	Specifies the password for the provisioning server when using the `set` command.

Feedback Examples

- `provisionserverpassword set “Polycom01”`
  returns
  `provisionserverpassword accepted`
- `provisionserverpassword set Pcom 01`
  returns
  `error: command has illegal parameters.`
- `provisionserverpassword set “Pcom 01”`
  returns
  `provisionserverpassword accepted`

Limitations

None

Comments

None

provisionserverstatus

Gets the current status of the provisioning server.

Syntax

```
provisionserverstatus <get>
```

Parameter	Description
`get`	Returns the current status of the provisioning server.

Feedback Examples

- `provisionserverstatus get`
  returns
  `provisionserverstatus registered`
- `provisionserverstatus get`
  returns
  `provisionserverstatus unregistered`

Limitations

None

Comments

None

provisionservertype

Gets or sets the provisioning server type.

Syntax

```
provisionservertype <get|rprm>
```

Parameter	Description
`get`	Returns the current setting.
`rpm`	Sets the provisioning server type to RealPresence Resource Manager.

Feedback Examples

- `provisionservertype get`
  returns
  `provisionservertype rprm`
- `provisionservertype rprm`
  returns
  `provisionservertype rprm`

Limitations

None

Comments

None

provisionserverupdate

Updates the connection to the provisioning server.

Syntax

```
provisionserverupdate
```

Additional Restrictions

None

Feedback Examples

- `provisionserverupdate`
  returns
  `provisionserverupdate success`
- `provisionserverupdate`
  returns
  `provisionserverupdate failed`
- `provisionserverupdate`
  returns
  `provisioning is already in progress`

Limitations

None

Comments

None

provisionserveruser

Gets or sets the username assigned to the provisioning server account.

Syntax

```
provisionserveruser <get|set> <"Username">
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the username for the provisioning server.
`"User Name"`	Specifies the username for the provisioning server when using the `set` command.

Feedback Examples

- `provisionserveruser get`
  returns
  `provisionserveruser "John Smith"`
- `provisionserveruser set "Harry Thomas"`
  returns
  `provisionserveruser "Harry Thomas"`

Limitations

None

Comments

None

reboot

Restarts the system.

Syntax

```
reboot [now]
```

Parameter	Description
now	Restarts the system without prompting you.

Feedback Examples

```
reboot now
```

Limitations

None

Comments

None

recentcalls

Returns a list of recent calls.

Syntax

```
recentcalls
```

Additional Restrictions

None

Feedback Examples

- `recentcalls`
  returns

  ```
  “Polycom Demo” 30/Sep/2015 14:39:56 Out
  192.168.1.101 30/Sep/2015 14:39:56 Out
  192.168.1.102 30/Sep/2015 14:40:35 Out
  192.168.1.103 30/Sep/2015 20:27:33 Out
  "John Polycom" 30/Sep/2015 02:13:23 In
  192.168.1.104 30/Sep/2015 02:20:08 In
  192.168.1.105 30/Sep/2015 02:21:40 In
  192.168.1.106 30/Sep/2015 05:53:04 In
  "Mary Polycom" 30/Sep/2015 07:00:19 In
  ```

Limitations

None

Comments

The number of items returned depends on the value entered for the **Maximum Number to Display** option in the web interface.

resetsettings

Resets your system to default settings. Nothing is retained during the reset if you do not include at least one of the following parameters.

Syntax

resetsettings <keepcertificates|keeplocaldirectory|keepcdr|keeplogs>

Parameter	Description
`keepcertificates`	Resets your system settings but keeps installed PKI certificates.
`keeplocaldirectory`	Resets your system settings but keeps local directory entries.
`keepcdr`	Resets your system settings but keeps the call detail report (CDR).
`keeplogs`	Resets your system settings but keeps system logs.

Feedback Examples

- `resetsettings`
  returns
  `resetsettings, are you sure? <y,n>`
- `resetsettings keepcertificates`
  returns
  `resetsettings, are you sure? <y,n>`

Limitations

None

Comments

None

rs232 baud

Gets or sets the baud rate for the first RS-232 port.

Syntax

```
rs232 baud <get|9600|19200|38400|57600|115200>
```

Parameter	Description
`get`	Returns the current baud rate setting.
`9600|19200|38400|57600|115200`	Sets the RS-232 port to this baud rate.

Feedback Examples

- `rs232 baud 9600`
  returns
  `rs232 baud 9600`
- `rs232 baud get`
  returns
  `rs232 baud 9600`

Limitations

None

Comments

None

rs232 mode

Gets or sets the operational mode of the system's serial port.

Syntax

```
rs232 mode <get|off|control>
```

Parameter	Description
`get`	Returns the current mode setting.
`off`	Sets the operational mode of the RS-232 port to off.
`control`	Sets the RS-232 port to Control mode.

Feedback Examples

- `rs232 mode control`
  returns
  `rs232 mode control`

Limitations

None

Comments

None

rs232login

Gets or sets the serial port login requirements.

Syntax

```
rs232login <get|off|pwonly|pwuser>
```

Parameter	Description
`get`	Returns the current setting.
`off`	Disables RS232 login requirements.
`pwonly`	Sets the serial port login requirement to use only the admin password.
`pwuser`	Sets the serial port login requirement to use both admin and user passwords.

Feedback Examples

- `rs232login get`
  returns
  `rs232login off`
- `rs232login pwonly`
  returns
  `rs232login pwonly`

Limitations

None

Comments

None

serialnum

Returns the serial number of the system.

Syntax

```
serialnum
```

Additional Restrictions

None

Feedback Examples

- `serialnum`
  returns
  `serialnum 82065205E72E1`

Limitations

None

Comments

None

session

Names or finds an active API session.

Syntax

```
session name “session-name”
session find “session-name”
```

Parameter	Description
`name`	Names the current API session.
`find`	Finds an active API session for this system.
`session-name`	Unique string that identifies the session.

Feedback Examples

- `session name sessionone`
  returns
  `session name sessionone success`
- `If entered again,`
  `session name sessionone`
  returns
  `info: the supplied session name is already in use`
  `session name sessionone failed`
- `session find sessionone`

  returns
  `info: session sessionone attached`
- `session find sessiontwo`

  returns
  `info: session sessiontwo not connected`

Limitations

None

Comments

None

setpassword

Sets the system's local administrator account password.

Syntax

```
setpassword admin room "currentacctpasswd" "newacctpasswd"
```

Parameter	Description
admin	Specifies the system's local administrator account.
room	Needed to change the password.
"currentacctpasswd"	The current password.
"newacctpasswd"	The new password.

Feedback Examples

- `setpassword admin room 123 456`
  returns
  `password changed`
- `setpassword admin room '' 456`
  returns
  `password changed`
- `setpassword admin room 123 ''`
  returns

  `password changed`

Limitations

None

Comments

If the account doesn't have password, enter a pair of single quotes ('') to denote an empty password.

sleep

Gets or sets options for system sleep mode.

Syntax

```
sleep
sleep <register|unregister>
sleep mute <get|on|off>
```

Parameter	Description
`get`	Returns the current setting for the `sleep mute` command.
`on`	Mutes the system microphone while the system is in sleep mode.
`off`	Unmutes the microphone while the system is in sleep mode.
`mute`	Mutes the system microphone while the system is in sleep mode.
`sleep`	Puts the system in sleep mode if not followed by other parameters.
`register`	Registers the system for sleep or wake events.
`unregister`	Unregisters the system for sleep or wake events.

Feedback Examples

- `sleep`
  returns
  `sleep`
- `sleep register`
  returns
  `sleep registered`
- If entered again,
  `sleep register`
  returns
  `info: event/notification already active:sleep`
- `sleep unregister`
  returns
  `sleep unregistered`
- If entered again,
  `sleep unregister`
  returns
  `info: event/notification not active:sleep`
- `sleep mute get`
  returns
  `sleep mute off`
- `sleep mute on`
  returns
  `sleep mute on`

Limitations

None

Comments

None

sleeptime

Gets or sets the time before the system goes to sleep.

Syntax

```
sleeptime <get|0|1|3|15|30|60|120|240|480>
```

Parameter	Description
`get`	Returns the current setting.
`off|1|3|15|30|45|60|120|240|480`	Sets the number of minutes from last user interaction to entering sleep mode. The default value is `3`. A value of `0` indicates that the system will never go to sleep.

Feedback Examples

- `sleeptime 30`
  returns
  `sleeptime 30`

Limitations

None

Comments

None

snmpadmin

Gets or sets your SNMP support contact name.

Syntax

```
snmpadmin get
snmpadmin set ["admin name"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the administrator name when followed by the `"admin name"` parameter. To erase the current setting, omit `"admin name"`.
`"admin name"`	SNMP administrator contact name. Character string. Enclose the character string in quotation marks if it includes spaces. Example: `"John Admin"`

Feedback Examples

- `snmpadmin get`
  returns
  `snmpadmin "John Admin"`
- `snmpadmin set "John Admin"`
  returns
  `snmpadmin "John Admin"`
- `snmpadmin set`
  returns
  `error: command needs more parameters to execute successfully`

Limitations

None

Comments

After making a change, you must restart the system for the setting to take effect.

snmpcommunity

Gets or sets the SNMP community string.

Syntax

```
snmpcommunity get
snmpcommunity set ["community name"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the SNMP community name when followed by the `"community name"` parameter. To erase the current setting, omit the parameter.
`"community name"`	SNMP community name. Character string. Enclose the character string in quotation marks if it includes spaces.

Feedback Examples

- `snmpcommunity set`
  returns
  `snmpcommunity <empty>`
- `snmpcommunity set Public`
  returns
  `snmpcommunity Public`
- `snmpcommunity get`
  returns
  `snmpcommunity Public`

Limitations

None

Comments

After making a change, you must restart the system for the setting to take effect.

snmpconsoleip

Gets or sets the SNMP console IP address.

Syntax

```
snmpconsoleip get
snmpconsoleip set ["xxx.xxx.xxx.xxx"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the SNMP console application IP address when followed by the "`xxx.xxx.xxx.xxx`" parameter. To erase the current setting, omit the parameter.
"`xxx.xxx.xxx.xxx`"	IP address of the console.

Feedback Examples

- `snmpconsoleip set`
  returns
  `snmpconsoleip <empty>`
- `snmpconsoleip set 192.168.1.111`
  returns
  `snmpconsoleip 192.168.1.111`
- `snmpconsoleip get`
  returns
  `snmpconsoleip 192.168.1.111`

Limitations

None

Comments

After making a change, you must restart the system for the setting to take effect.

snmplocation

Gets or sets the location of the SNMP system.

Syntax

```
snmplocation get
snmplocation ["location name"]
```

Parameter	Description
`get`	Returns the current setting.
`"location name"`	SNMP system location. Enclose the location name in quotation marks if it includes spaces. To erase the current setting, omit the parameter.

Feedback Examples

- `snmplocation`
  returns
  `snmplocation <empty>`
- `snmplocation set "Polycom1 in United States"`
  returns
  `snmplocation "Polycom1 in United States"`
- `snmplocation get`
  returns
  `snmplocation "Polycom1 in United States"`

Limitations

None

Comments

You must restart the system after making a change to the SNMP setting.

snmpnotification

Enables or disables SNMP notifications for the Polycom MIB, which can be downloaded from the SNMP page in the system web interface.

Syntax

```
snmpnotification <get|true|false>
```

Parameter	Description
`get`	Returns the current setting for SNMP notifications.
`true`	Enables SNMP notifications.
`false`	Disables SNMP notifications.

Feedback Examples

- `snmpnotification get`
  returns
  `snmpnotification true`
- `snmpnotification true`
  returns
  `snmpnotification true`
- `snmpnotification false`
  returns
  `snmpnotification false`

Limitations

None

Comments

None

snmpsystemdescription

Gets or sets the SNMP system description.

Syntax

```
snmpsystemdescription get
snmpsystemdescription set ["system description"]
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the SNMP system description when followed by the `"system description"` parameter. To erase the current setting, omit the parameter.
`"system description"`	SNMP system description.

Feedback Examples

- `snmpsystemdescription set`
  returns
  `snmpsystemdescription <empty>`
- `snmpsystemdescription set "videoconferencing system"`
  returns
  `snmpsystemdescription "videoconferencing system"`
- `snmpsystemdescription get`
  returns
  `snmpsystemdescription "videoconferencing system"`

Limitations

None

Comments

After making a change, you must restart the system for the setting to take effect.

snmptrapversion

Gets or sets the SNMP trap version.

Syntax

```
snmptrapversion get
snmptrapversion set <v1|v2c|v3>
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the SNMP trap protocol that the system uses.
`v1|v2c|v3`	SNMP trap version 1, 2c, or 3.

Feedback Examples

- `snmptrapversion get`
  returns
  `snmptrapversion v2c`
- `snmptrapversion set v3`
  returns
  `snmptrapversion v3`

Limitations

None

Comments

After making a change, you must restart the system for the setting to take effect.

sshenable

Enables command-line API access over SSH.

Syntax

```
sshenable <true|false>
```

Parameter	Description
true	Enables command-line API access over SSH.
false	Disables command-line API access over SSH

Feedback Examples

- `sshenable true`
  returns
  `sshenable true`
- `sshenable false`
  returns
  `sshenable false`

Limitations

None

Comments

None

status

Returns the current status of devices and primary system services.

Syntax

```
status
```

Parameter	Description
status	Returns the current status of system settings.

Feedback Examples

- `status`
  returns
  ```
  inacall offline
  autoanswerp2p online
  remotecontrol online
  microphones online
  globaldirectory offline
  ipnetwork online
  gatekeeper online
  sipserver online
  calendar online
  logthreshold offline
  provisioning online
  wifi offline
  status end
  ```

Limitations

None

Comments

None

subnetmask

Gets or sets the subnet mask of the system.

Syntax

```
subnetmask get
subnetmask set ["xxx.xxx.xxx.xxx"]
```

Parameter	Description
`get`	Returns the current subnet mask.
`set`	Sets the subnet mask of the system when followed by the `"xxx.xxx.xxx.xxx"` parameter. To erase the current setting, omit `"xxx.xxx.xxx.xxx"`. This parameter is not allowed while in a call.
`"xxx.xxx.xxx.xxx"`	Subnet mask of the system.

Feedback Examples

- `subnetmask set 255.255.255.0`
  returns
  `subnetmask 255.255.255.0`
- `subnetmask get`
  returns
  `subnetmask 255.255.255.0`

Limitations

None

Comments

After making a change, you must restart the system for the setting to take effect.

systemname

Gets or sets the name of the system.

Syntax

```
systemname get
systemname set "system name"
```

Parameter	Description
`get`	Returns the current setting.
`set`	Sets the system name to `"system name"`.
`"system name"`	Character string specifying the system name. Enclose the string in quotation marks if it includes spaces. Example: `"Polycom G7500 Demo"`

Feedback Examples

- `systemname set "Demo"`
  returns
  `systemname "Demo"`
- `systemname set get`
  returns
  `systemname "Demo"`

Limitations

None

Comments

The first character must be a numeric (a digit) or an alphabetic (a letter) character including foreign language characters. The name can be any combination of alphanumeric characters and may be up to 30 characters in length. The system name cannot be blank.

systemsetting 323gatewayenable

Gets the current setting or enables H.323 calling through a gateway.

Syntax

```
systemsetting 323gatewayenable <True|False>
systemsetting get 323gatewayenable
```

Parameter	Description
`get`	Returns the current setting.
`True`	Enables H.323 gateway calls.
False	Disables H.323 gateway calls.

Feedback Examples

- `systemsetting 323gatewayenable True`
  returns
  `systemsetting 323gatewayenable True`
- `systemsetting get 323gatewayenable`
  returns
  `systemsetting 323gatewayenable True`

Limitations

None

Comments

None

systemsetting bfcptransportprotocol

Gets the current setting or indicates the Binary Floor Control Protocol (BFCP) connection and provides an option to set the connection preference to UDP or TCP.

Syntax

```
systemsetting bfcptransportprotocol <Prefer_UDP|Prefer_TCP|UDP_Only|TCP_Only>
systemsetting get bfcptransportprotocol
```

Parameter	Description
`get`	Returns the current setting.
`Prefer_TCP`	Specifies TCP as the BFCP connection preference.
`Prefer_UDP`	Specifies UDP as the BFCP connection preference.
`UDP_Only`	Specifies UDP as the BFCP transport protocol.
`TCP_Only`	Specifies TCP as the BFCP transport protocol.

Feedback Examples

- `systemsetting get bfcptransportprotocol`
  returns
  `systemsetting bfcptransportprotocol Prefer_UDP`
- `systemsetting bfcptransportprotocol Prefer_TCP`
  returns
  `systemsetting bfcptransportprotocol Prefer_TCP`
- `systemsetting get bfcptransportprotocol`
  returns
  `systemsetting bfcptransportprotocol Prefer_TCP`
- `systemsetting bfcptransportprotocol UDP_Only`
  returns
  `systemsetting bfcptransportprotocol UDP_Only`

Limitations

None

Comments

The BFCP Transport Protocol in which your system is operating determines which protocol is required.

systemsetting dialingmethod

Gets or sets the preferred method for dialing various call types.

Syntax

```
systemsetting dialingmethod <Auto|Manual>
systemsetting get dialingmethod
```

Parameter	Description
`get`	Returns the current setting.
`Auto`	Sets the dialing mode to Auto. Calls use the configured dialing order.
`Manual`	Sets the dialing mode to Manual. The system prompts the user to select the call type from a list when placing a call.

Feedback Examples

- `systemsetting dialingmethod Auto`
  returns
  `systemsetting dialingmethod Auto`
- `systemsetting get dialingmethod`
  returns
  `systemsetting dialingmethod Auto`

Limitations

None

Comments

None

systemsetting displayiconsincall

Gets or specifies whether to display icons on the Home Screen during a call.

Syntax

```
systemsetting displayiconsincall <True|False>
systemsetting get displayiconsincall
```

Parameter	Description
`get`	Returns the current setting.
`True`	Specifies to display the icons on the info bar while in a call.
`False`	Specifies to not display the icons on the info bar while in a call.

Feedback Examples

- `systemsetting displayiconsincall True`
  returns
  `systemsetting displayiconsincall True`
- `systemsetting get displayiconsincall`
  returns
  `systemsetting displayiconsincall True`

Limitations

None

Comments

None

systemsetting iph323enable

Gets the current setting or specifies whether H.323 calls are allowed.

Syntax

```
systemsetting iph323enable <True|False>
systemsetting get iph323enable
```

Parameter	Description
`get`	Returns the current setting.
`True`	Enables H.323 call capability.
`False`	Disables H.323 call capability.

Feedback Examples

- `systemsetting iph323enable True` returns
  `systemsetting iph323enable True`
- `systemsetting get iph323enable` returns
  `systemsetting iph323enable True`

Limitations

None

Comments

None

systemsetting lineinlevel

Gets the current setting or returns the volume level for 3.5 mm stereo audio input.

Syntax

```
systemsetting lineinlevel {0..10}
systemsetting get lineinlevel
```

Parameter	Description
`get`	Returns the current setting.
`0..10`	Sets the volume level for input 1. Valid range is `0` to `10`.

Feedback Examples

- `systemsetting lineinlevel 5`
  returns
  `systemsetting lineinlevel 5`
- `systemsetting get lineinlevel`
  returns
  `systemsetting lineinlevel 5`

Limitations

None

Comments

None

systemsetting lineoutmode

Gets the current setting or specifies whether the volume for a device connected to the 3.5 mm line stereo audio output port is variable or fixed.

Syntax

```
systemsetting lineoutmode <fixed|variable>
systemsetting get lineoutmode
```

Parameter	Description
`get`	Returns the current setting.
`fixed`	Sets the volume to the audio level specified in the interface.
`variable`	Allows users to set the volume with the remote control.

Feedback Examples

- `systemsetting lineoutmode fixed`
  returns
  `systemsetting lineoutmode fixed`
- `systemsetting get lineoutmode`
  returns
  `systemsetting lineoutmode fixed`

Limitations

None

Comments

None

systemsetting maxrxbandwidth

Gets the sets the maximum receive line speed between 64 kbps and 6144 kbps.

Syntax

```
systemsetting maxrxbandwidth [speed]
systemsetting get maxrxbandwidth
```

Parameter	Description
`get`	Returns the current setting.
`speed`	Sets the maximum speed for receiving calls.

Feedback Examples

- `systemsetting maxrxbandwidth 1920`
  returns
  `systemsetting maxrxbandwidth 1920`
- `systemsetting get maxrxbandwidth`
  returns
  `systemsetting maxrxbandwidth 1920`

Limitations

None

Comments

None

systemsetting maxtxbandwidth

Gets or sets the maximum transmit line speed between 64 kbps and 6144 kbps.

Syntax

```
systemsetting maxtxbandwidth [speed]
systemsetting get maxtxbandwidth
```

Parameter	Description
`get`	Returns the current setting.
`speed`	Sets the maximum speed for placing calls.

Feedback Examples

- `systemsetting maxtxbandwidth 1920`
  returns
  `systemsetting maxtxbandwidth 1920`
- `systemsetting get maxtxbandwidth`
  returns
  `systemsetting maxtxbandwidth 1920`

Limitations

None

Comments

None

systemsetting mediainlevel

Gets or specifies the volume level for the content 3.5 mm stereo audio input.

Syntax

```
systemsetting mediainlevel <0..10>
systemsetting get mediainlevel
```

Parameter	Description
`get`	Returns the current setting.
`0..10`	Sets the volume level of the media input to the specified value.

Feedback Examples

- `systemsetting mediainlevel 5`
  returns
  `systemsetting mediainlevel 5`
- `systemsetting get mediainlevel`
  returns
  `systemsetting mediainlevel 5`

Limitations

None

Comments

None

systemsetting model

Returns the model of the system.

Syntax

```
systemsetting get model
```

Parameter	Description
`get`	Returns the current setting.

Feedback Examples

- `systemsetting get model`
  returns
  `systemsetting model “RealPresence ”`

Limitations

None

Comments

None

systemsetting selfview

Gets or sets the Self View.

Syntax

```
systemsetting selfview <on|off|auto>
systemsetting get selfview
```

Parameter	Description
`get`	Returns the current setting.
`on`	Enables Self View.
`off`	Disables Self View.
`auto`	Sets Self View to auto mode.

Feedback Examples

- `systemsetting selfview on` returns
  `systemsetting selfview on`
- `systemsetting selfview off` returns
  `systemsetting selfview off`
- `systemsetting get selfview` returns
  `systemsetting selfview on`

Limitations

None

Comments

None

systemsetting sipaccountname

Gets or sets the SIP user account name.

Syntax

```
systemsetting sipaccountname <"sipuser">
systemsetting get sipaccountname
```

Parameter	Description
`get`	Returns the current setting.
`"sipuser"`	Specifies the user account name.

Feedback Examples

- `systemsetting sipaccountname polycom_user`
  returns
  `systemsetting sipaccountname polycom_user`
- `systemsetting get sipaccountname`
  returns
  `systemsetting sipaccountname polycom_user`

Limitations

None

Comments

None

systemsetting sipdebug

Gets or sets the state of SIP debug tracing in the system log.

Syntax

```
systemsetting sipdebug <True|False>
systemsetting get sipdebug
```

Parameter	Description
`get`	Returns the current setting.
`True`	Enables SIP debug tracing in the system log.
`False`	Disables SIP debug tracing in the system log.

Feedback Examples

- `systemsetting sipdebug True`
  returns
  `systemsetting sipdebug True`
- `systemsetting get sipdebug`
  returns
  `systemsetting sipdebug True`

Limitations

None

Comments

None

systemsetting sipenable

Enables or disables SIP calling.

Syntax

```
systemsetting sipenable <True|False>
systemsetting get sipenable
```

Parameter	Description
`get`	Returns the current setting.
`True`	Enables SIP calling.
`False`	Disables SIP calling.

Feedback Examples

- `systemsetting sipenable True`
  returns
  `systemsetting sipenable True`
- `systemsetting get sipenable`
  returns
  `systemsetting sipenable True`

Limitations

None

Comments

None

systemsetting sipforcereuse

Enables or disables the SIP force reuse function, which forces the proxy server to reuse the existing SIP connection for requests in the reverse direction by using the SIP port as the source port.

Syntax

```
systemsetting get sipforcereuse
systemsetting sipforcereuse <True|False>
```

Parameter	Description
`get`	Returns the current setting.
`True`	Enables the SIP force reuse function.
`False`	Disables the SIP force reuse function.

Feedback Examples

- `systemsetting get sipforcereuse`
  returns
  `systemsetting sipforcereuse True`
- `systemsetting sipforcereuse True`
  returns
  `systemsetting sipforcereuse True`
- `systemsetting sipforcereuse False`
  returns
  `systemsetting sipforcereude False`

Limitations

None

Comments

None

systemsetting sippassword

Sets the SIP server password.

Syntax

```
systemsetting sippassword <"password">
```

Parameter	Description
"password"	Password used to register with SIP server.

Feedback Examples

- `systemsetting sippassword secret`
  returns
  `systemsetting sippassword secret`

Limitations

None

Comments

None

systemsetting sipproxyserver

Gets or sets the address of the SIP proxy server.

Syntax

```
systemsetting sipproxyserver <address>
systemsetting get sipproxyserver
```

Parameter	Description
`get`	Returns the current setting.
`"address"`	Address of the proxy server. Can be an IP address or fully qualified domain name (FQDN).

Feedback Examples

- `systemsetting sipproxyserver pserver.abc.com`
  returns
  `systemsetting sipproxyserver pserver.abc.com`
- `systemsetting get sipproxyserver`
  returns
  `systemsetting sipproxyserver pserver.abc.com`

Limitations

None

Comments

None

systemsetting sipregistrarserver

Gets or sets the address of the SIP registrar server.

Syntax

```
systemsetting sipregistrarserver <address>
systemsetting get sipregistrarserver
```

Parameter	Description
`get`	Returns the current setting.
`"address"`	Address of the registrar server. Can be an IP address or fully qualified domain name (FQDN).

Feedback Examples

- `systemsetting sipregistrarserver pserver.abc.com`
  returns
  `systemsetting sipregistrarserver pserver.abc.com`
- `s`systemsetting get sipregistrarserver
  returns
  `systemsetting sipregistrarserver pserver.abc.com`

Limitations

None

Comments

None

systemsetting siptransportprotocol

Gets or sets the protocol the system uses for SIP signaling.

Syntax

```
systemsetting siptransportprotocol <Auto|TLS|TCP|UDP>
systemsetting <get> siptransportprotocol
```

Parameter	Description
`get`	Returns the current setting.
`Auto`	Sets the SIP transport protocol to automatic negotiation.
`TLS`	Sets TLS as the SIP transport protocol. TLS provides a secure transport.
`TCP`	Sets TCP as the SIP transport protocol. TCP provides a reliable transport.
`UDP`	Sets UDP as the SIP transport protocol. UDP provides a best-effort transport.

Feedback Examples

- `systemsetting get siptransportprotocol`
  returns
  `systemsetting siptransportprotocol Auto`
- `systemsetting siptransportprotocol TLS`
  returns
  `systemsetting siptransportprotocol TLS`
- `systemsetting siptransportprotocol TCP`
  returns
  `systemsetting siptransportprotocol TCP`
- `systemsetting siptransportprotocol UDP`
  returns
  `systemsetting siptransportprotocol UDP`

Limitations

None

Comments

None

systemsetting sipusername

Gets or sets the system's SIP name.

Syntax

```
systemsetting sipusername ["name"]
systemsetting get sipusername
```

Parameter	Description
`get`	Returns the current setting.
`"name"`	Specifies the SIP URI for SIP registration.

Feedback Examples

- `systemsetting sipusername Polycom`
  returns
  `systemsetting sipusername Polycom`
- `systemsetting get sipusername`
  returns
  `systemsetting sipusername Polycom`

Limitations

None

Comments

None

systemsetting stereoenable

Gets the current setting or specifies whether Polycom® StereoSurround™ technology is used for all calls.

Syntax

```
systemsetting stereoenable <True|False>
systemsetting get stereoenable
```

Parameter	Description
`get`	Returns the current setting.
`True`	Enables Polycom StereoSurround.
`False`	Disables Polycom StereoSurround.

Feedback Examples

- `systemsetting stereoenable True`
  returns
  `systemsetting sstereoenable True`
- `systemsetting get stereoenable`
  returns
  `systemsetting stereoenable True`

Limitations

None

Comments

None

systemsetting telnetenabled

Gets or sets the telnet ports.

Syntax

```
systemsetting telnetenabled <True|False>
systemsetting get telnetenabled
```

Parameter	Description
`get`	Returns the current setting.
`True`	Enables ports 23 and 24.
`False`	Disables ports 23 and 24.

Feedback Examples

- `systemsetting get telnetenabled`
  returns
  `systemsetting telnetenabled True`
- `systemsetting telnetenabled True`
  returns
  `systemsetting telnetenabled True`
- `systemsetting telnetenabled`
  returns
  `error: command needs more parameters to execute successfully`

Limitations

None

Comments

After making a change, you must restart the system for the setting to take effect.

systemsetting transcodingenabled

Gets or specifies whether the system allows each far-site system to connect using the best possible call rate and audio/video algorithm.

Syntax

```
systemsetting transcodingenabled <True|False>
systemsetting get transcodingenabled
```

Parameter	Description
`get`	Returns the current setting.
`True`	Enables transcoding.
`False`	Disables transcoding.

Feedback Examples

- `systemsetting transcodingenabled True`
  returns
  `systemsetting transcodingenabled True`
- `systemsetting get transcodingenabled`
  returns
  `systemsetting transcodingenabled True`

Limitations

None

Comments

None

systemsetting webenabled

Gets or specifies whether to allow remote access to the system using the web interface.

Syntax

```
systemsetting webenabled <True|False>
systemsetting get webenabled
```

Parameter	Description
`get`	Returns the current setting.
`True`	Enables remote access from the web interface.
`False`	Disables remote access from the web interface.

Feedback Examples

- `systemsetting webenabled True`
  returns
  `systemsetting webenabled True`
- `systemsetting get webenabled`
  returns
  `systemsetting webenabled True`

Limitations

None

Comments

You must restart the system for changes to take effect.

systemsetting whitebalancemode

Gets or sets the white balance mode for a connected Polycom camera.

Syntax

```
systemsetting whitebalancemode
<Auto|Manual|2300K|2856K|3200K|3450K|3680K|4160K|4230K|4640K|5120K|5200K|5600K|6504K>
systemsetting get whitebalancemode
```

Parameter	Description
`get`	Returns the current setting.

`Auto|Manual|3200K|3680K|4160K|4640K|5120K|5600K`	`Auto` - Automatic white balance.
	`Manual` - Fixed white balance (measured in Kelvin).
	`2300K`
	`2856K`
	`3200K`
	`3450K`
	`3680K`
	`4160K`
	`4230K`
	`4640K`
	`5120K`
	`5200K`
	`5600K`
	`6504K`

Feedback Examples

- `systemsetting whitebalancemode Auto`
  returns
  `systemsetting whitebalancemode Auto`
- `systemsetting get whitebalancemode`
  returns
  `systemsetting whitebalancemode Auto`

Limitations

None

Comments

None

uptime

Returns the total time the system has been running since the last system start.

Syntax

```
uptime get
```

Parameter	Description
`get`	Returns the current setting.

Feedback Example

- `uptime get`
  returns
  `1 Hour, 10 Minutes`

Limitations

None

Comments

None

usegatekeeper

Gets or sets whether the system can use an H.323 gatekeeper.

Syntax

```
usegatekeeper <get|off|specify|auto>
```

Parameter	Description
`get`	Returns the current setting.
`off`	Select this option if no gatekeeper is required.
`specify`	Specifies a gatekeeper. If this option is selected, you must enter the gatekeeper IP address or name using `gatekeeperip`.
`auto`	Sets the system to automatically find an available gatekeeper.

Feedback Examples

- `usegatekeeper off`
  returns
  `usegatekeeper off`
- `usegatekeeper specify`
  returns
  `usegatekeeper specify`
- `usegatekeeper auto`
  returns
  `usegatekeeper auto`
- `usegatekeeper get`
  returns
  `usegatekeeper auto`

Limitations

None

Comments

None

vcbutton

Controls a content video source. It can also register or unregister the API session to receive notification of content events.

Syntax

```
vcbutton play {1..6}
vcbutton <get|stop|register|unregister>
vcbutton map <get|{1..6}>
vcbutton source get
```

Parameter	Description
`get`	Returns the current setting (`play` or `stop`).
`play`	Starts sending the content from the specified content video source. If no content video source is specified, starts sending content from the default content video source. Starts content from any content video source without the need to change source mapping and without needing to stop the currently playing content video source. Fails and does not stop the current content video source if the specified content video source is not valid. Stops the current content video source if the specified content video source is valid but is currently unavailable.
`{1..6}`	Specifies a content video source. `5` is not supported.
`stop`	Stops sending content from the content video source that is currently playing.
`register`	Registers the API session to receive notifications about content events.
`unregister`	Unregisters the API session to receive notifications about content events.
`map get`	Gets the content video source currently specified for control.
`map {1..6}`	Specifies the content video source to control. Note: This parameter is only necessary if no video source was specified when using the vcbutton play command. `5` is not supported.
`source get`	Gets the content video source that is currently playing.

Feedback Examples

If not registered for notifications:

- `vcbutton play 4`
  returns
  ```
  vcbutton play 4
  vcbutton play succeeded
  camera near 4
  ```

If registered for notifications:

- `vcbutton play 4`
  returns
  ```
  Control event: vcbutton play
  ```

```
Control event: vcbutton source 4
Control event: vcbutton play
vcbutton play 4
vcbutton play succeeded
camera near 4
```

- `vcbutton play 3`
  returns
  `vcbutton play failed`
- `vcbutton play`
  returns
  ```
  Control event: vcbutton play
  vcbutton play succeeded
  ```
- `vcbutton play`
  returns
  `vcbutton play failed`
- `vcbutton play 2`
  returns
  ```
  error: input 2 is not a content source
  vcbutton play failed
  ```
- `vcbutton play 7`
  returns
  ```
  error: invalid value! (valid ranges 2..6)
  vcbutton play failed
  ```
- `vcbutton register`
  returns
  `vcbutton registered`
- `vcbutton stop`
  returns
  ```
  Control event: vcbutton stop
  Camera near none
  vcbutton stop
  vcbutton stop succeeded
  ```
- `vcbutton get`
  returns
  ```
  vcbutton stop
  vcbutton get succeeded
  ```
- `vcbutton source get`
  returns
  ```
  vcbutton source get 1
  vcbutton source get succeeded
  ```
- `vcbutton source get`
  returns
  ```
  vcbutton source get none
  vcbutton source get succeeded
  ```

Limitations

None

Comments

`vcbutton map` defaults to input 6.

`vcbutton map` is only required if you do not specify the input number when sending `vcbutton play`.

version

Returns the current system's version information.

Syntax

```
version
```

Additional Restrictions

None

Feedback Examples

- `version`
  returns
  `version 2.0`

Limitations

None

Comments

None

videocallorder

Gets or sets the video call order of the specified protocol to the specified slot.

Syntax

```
videocallorder <h323|sip> <1|2|3|4>
```

Parameter	Description
`h323`	Specifies IP protocol.
`sip`	Specifies SIP protocol.
`1|2|3|4`	Sets the order in which the specified protocol is attempted when a video call is placed.

Feedback Examples

- `videocallorder h323 1`
  returns
  `videocallorder h323 1`
- `videocallorder sip 2`
  returns
  `videocallorder sip 2`

Limitations

None

Comments

None

videomute

Gets or sets the transmission of local video to the far site.

Syntax

```
videomute near <get|on|off>
```

Parameter	Description
`get`	Returns the current setting.
`near`	Specifies local video.
`on`	Enables information.
`off`	Returns the current setting.

Feedback Examples

- `videomute near get`
  returns
  `vidoemute near off`
- `videomute near on`
  returns
  `videomute near on`
- `videomute near off`
  returns
  `videomute near off`

Limitations

None

Comments

None

volume

Gets or sets the call audio volume (not sound effects) on the system or registration for volume changes. Changes the call audio volume (not sound effects) on the system.

Syntax

```
volume <register|unregister>
volume <get|up|down|set {0..50}>
volume range
```

Parameter	Description
`register`	Registers to receive notification when the volume changes.
`unregister`	Disables register mode.
`get`	Returns the current volume level.
`up`	Increases the audio volume by 1.
`down`	Decreases the audio volume by 1.
`set`	Sets the volume to a specified level. Requires a volume setting from `{0..50}`.
`range`	Returns the valid volume range available to the user.

Feedback Examples

- `volume register`
  returns
  `volume registered`
- If entered again,
  `volume register`
  returns
  `info: event/notification already active:volume`
- `volume set 23`
  returns
  `volume 23`
- `volume up`
  returns
  `volume 24`
- `volume get`
  returns
  `volume 24`

Limitations

None

Comments

Changes the call audio volume (not sound effects) on the system.

The `button` command also allows you to control the system volume. Note that it does not return feedback about the current volume level.

wake

Wakes the system from sleep mode.

Syntax

```
wake
```

Additional Restrictions

None

Feedback Examples

- `wake`
  returns
  `wake`
  and wakes the system from sleep mode

Limitations

None

Comments

None

See Also

To put the system in sleep mode, use the sleep command.

wanipaddress

Gets or sets the WAN IP address.

Syntax

```
wanipaddress get
wanipaddress set ["xxx.xxx.xxx.xxx"]
```

Parameter	Description
`get`	Returns the WAN IP address.
`set`	Sets the WAN IP address when followed by the `"xxx.xxx.xxx.xxx"` parameter. To erase the current setting, omit the `"xxx.xxx.xxx.xxx"` parameter.
`"xxx.xxx.xxx.xxx"`	WAN IP address.

Feedback Examples

- `wanipaddress set 192.168.1.101`
  returns
  `wanipaddress 192.168.1.101`
- `wanipaddress get`
  returns
  `wanipaddress 192.168.1.101`

Limitations

None

Comments

The **NAT Configuration** setting must be set to **Auto** or **Manual** to set this command.

whoami

Displays the same initial banner information when an API session starts.

Syntax

```
whoami
```

Additional Restrictions

None

Feedback Examples

- `whoami`
  returns

```
Hi, my name is: Demo
Here is what I know about myself:
Model:
Serial Number: 82065205E72E1
Software Version: 1.0
Build Information: root on domain.polycom.com
Contact Number: <empty>
Time In Last Call: 01:43:50
Total Time In Calls: 3 days, 08:17:17
Total Calls: 819
SNTP Time Service: auto insync ntp1.polycom.com
Local Time is: Wed, 30 Nov 2008 10:41:46
Network Interface: NONE
IP Video Number: 192.168.1.101
MP Enabled: AB1C-2D34-5EF6-7890-GHI1
H323 Enabled: True
HTTP Enabled: True
SNMP Enabled: True
```

Limitations

None

Comments

The response can vary depending on your system configuration.