API Commands Introduction for Yealink Video Conferencing Systems

Nov. 2019

V1.08

Table of Contents

Introduction

About This Guide

The features and syntax of central control API commands are covered in this guide. This guide provides central control module introduction for readers to quickly understand the related functions, and helps developers to complete follow-up programming work to realize controlling Yealink video conferencing systems via LAN connection and Serial connection.

Changes for Release 1, Guide Version 1.06

The following parameters are new for this edition:

- callinfo all
- callinfo callid "called"
- getcallid
- sysstatus get

Changes for Release 1, Guide Version 1.07

The following parameters are new for this edition:

- layout near get
- layout near get list
- layout near set layoutmode idlist
- camera near get_id_list {"id:int"}
- camera near get_id_detial "id:int" "status:int" "name:string" "default:int" "ip:string" "mac:string"
- camera near set_active_status "id:int"
- camera near move_id "id:int" "direct:x"|stop
- camera near zoom_id "id:x" "direct:int"|stop

Changes for Release 1, Guide Version 1.08

The following parameters are new for this edition:

- layout near camera_layout_get
- layout near camera_layout_set layoutmode idlist

Overview

Supported Environment

- The API commands described in this guide apply to Yealink video conferencing systems 2.0 or later.
- The API commands can be sent to Yealink video conferencing systems over network or serial port, to realize controlling the Yealink video conferencing systems.
- In order to make it easier for readers to understand the API syntax, the detailed API syntax is introduced in the following chapters. For more information, refer to About the API Commands on page 3.

Note The API commands are case sensitive and use half-width input.

Using the API with a LAN Connection

When using LAN connection method, you can send API commands to Yealink video conferencing system through TCP protocol. The control system needs to know the IP address and port of the Yealink video conferencing system.

Configuration parameters needed:

- Control TCP port should be set to 6024
- The IP address of Yealink video conferencing system

Using the API with a Serial Connection

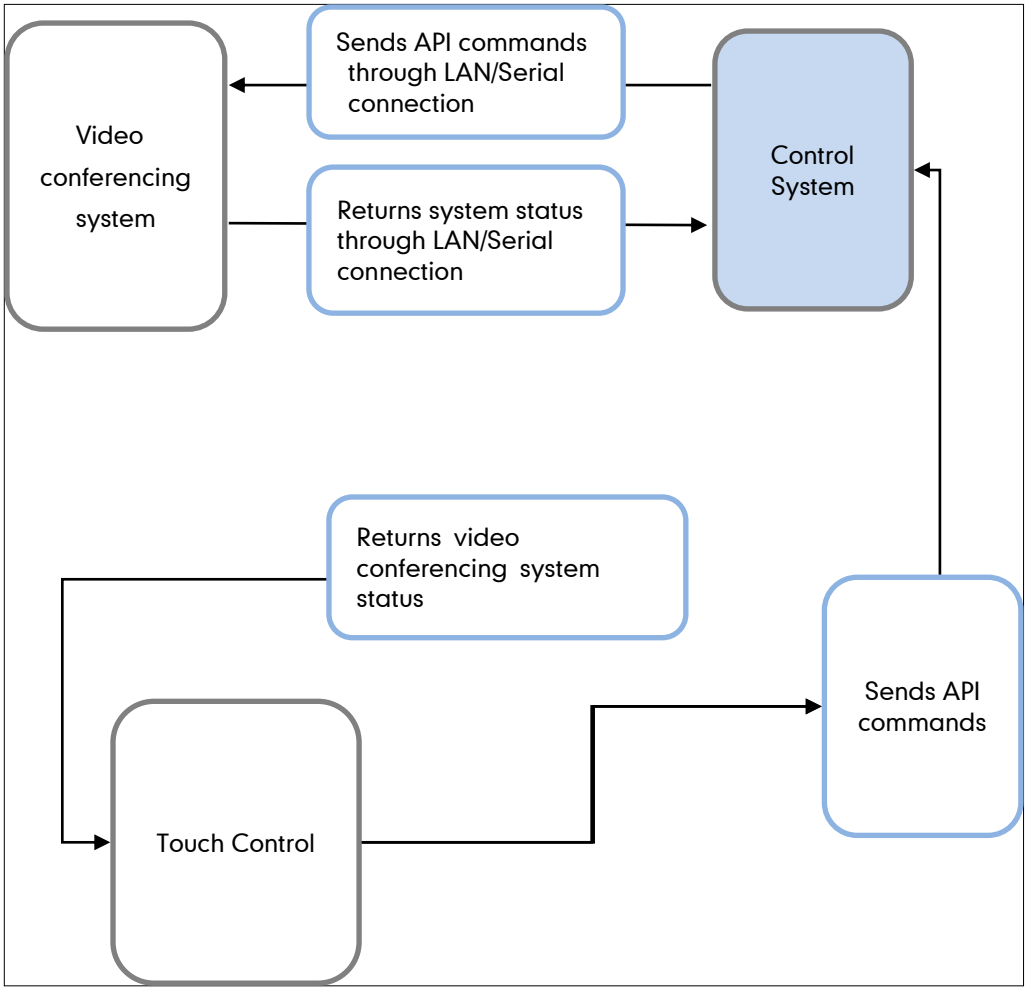
You can choose serial connection method to control Yealink video conferencing system. The USB port on the Yealink video conferencing system can be connected to the serial port on the control system through a serial cable.

Configuration parameters needed:

- Baudrate: 115200
- Databits: 8
- Stopbits: 1
- Parity: N (None)

Note The parameter configuration should be consistent with the central control host, the above is only for reference.
When using the computer's serial port debugging assistant, you can set the same parameters as above for testing. In the actual connection with the central control, in the security control of the VCS 2.0 endpoint should be modified based on the relevant parameters of the actual central control host.

Diagram of Central Control System



About the API Commands

Syntax Conventions

The following conventions are used for the API command. All of the commands are case sensitive.

Convention	Meaning
<param1|param2|param3>	Multiple valid parameters are enclosed in angle brackets and separated by the pipe ("|") character. **Example**: button <0|1|2|3|4|5|6|7|8|9|*|#> **Returning Example:** Send: button 0\r\n return button 0\r\n
"x"	Quotation marks indicate strings to be supplied by the user. **Example**: dial auto "dialstring" **Returning Example** Send: dial auto "1235"\r\n return dial auto "1235"\r\n
{1..n}	A range of possible numeric values is enclosed in braces.

Note

The API commands listed in this guide are in alphabetical order. To make it easier for users to manage and organize these commands, new commands will be added to the table based on this rule.

This rule applies to reading and sorting API commands. It is not a rule for sending API commands. The format of sending API commands is detailed in each API command.

Format Description of Data Packet

Format of Sending API Command

\r\n identifier represents the end of the API command. Every API command must end with \r\n.

Format of Returning API Command

\r\n identifier represents the end of the API command. Every returned API command ends with \r\n.

Basic Command format

1. Command (without parameter)
2. Command+ space +param1 (with a parameter)
3. Command+ space + param1 + space + param2 + space +... + space + paramN (with multiple parameters)
4. Command + space + "string" (command with a string, the string is enclosed in quotes)
5. Command+ space + param1 + space + "string"

Additional Tips

- Parameter list is optional. The command may or may not have parameters.
- Commands may or may not have string. The string should be enclosed in quotes.
- Command and parameter should be separated by a space.
- Multiple parameters should be separated by a space.
- String should be enclosed in quotes.

Command Response Syntax

Successful Response

- Returning control commands (the control system sends control commands to video conferencing system)

  Command param\r\n

- Returning query commands (the control system sends query commands to video conferencing system)

  Command data\r\n

- Returning feedback commands (the video conferencing system provides feedback to the control system)

  Command data\r\n

Note Param represents parameter supported by command. Data represents the data user wants to query.

Invalid Response

- Command is correct but the parameter is wrong:

  error: command has illegal parameters\r\n

- Command is wrong:

  error: command not found\r\n

Detailed Description of API Commands

The API commands supported by Yealink VCS V2.0 are described below. Readers can use TCP tools or serial tools to test these API commands in the environment mentioned above.

The availability of API commands depends on the current firmware of the video conferencing system.

The commands listed below are in alphabetical order. To make it easier for users to manage and organize these commands, new commands will be added to the table based on this rule.

answer

1. answer <yes|no>

answer	**answer <yes|no>**	
Parameter introduction (control command type)	param	Valid values: - **yes**: answers calls. - **no**: refuses calls.
Sending format	answer <yes|no>\r\n	
Returning format	answer <yes|no>\r\n	
Example	Send: answer yes\r\n return answer yes\r\n	

addrbook

1. addrbook all

addrbook	**addrbook all**	
Parameter introduction (control command type)	all	Obtains all contacts' information.
Sending format	addrbook all\r\n	
Returning format	addrbook numid type "name" "numberlist"\r\n **Note:** - **numid**: contact id - **type**: contact type	

	local: local contact **conf**: conference contact - **"name"**: contact's name strings - **"numberlist"**: contact's number strings（maybe multiple numbers）
Example	Send: addrbook all return addrbook 1 local "xiaoming" "123456" "231456" \r\n addrbook 2 local "zhangpeng" "123456" "65412" "98745"\r\n addrbook 3 local "pengp" "63251"\r\n addrbook 4 conf "groud1_conf" "11111" "22222" "5565655" "96363524"\r\n addrbook 5 conf "group2_conf" "222241" "65412" "654253"\r\n addrbook all all done!

2. addrbook <local|conf> get all

addrbook	**addrbook <local|conf> get all**	
Parameter introduction (control command type)	local	Configures local contacts information.
	conf	Configures conference contact information.
	get	Obtains the selected contacts' information.
	all	Obtains all local or conference contacts' information.
Sending format	addrbook <local|conf> get all\r\n	
Returning format	addrbook numid type "name" "numberlist"\r\n **Note:** - **numid**: contact id - **type**: contact type **local**: local contact **conf**: conference contact - **"name"**: contact's name strings - **"numberlist"**: contact's number strings（maybe multiple numbers）	
Example	Send: addrbook local get all return addrbook 1 local "xiaoming" "123456" "231456" \r\n addrbook 2 local "zhangpeng" "123456" "65412" "98745"\r\n	

	addrbook 4 local "zzddff" "123456" "231456" \r\n addrbook 5 local "zmndhdn" "123456" "65412" "98745"\r\n addrbook local get all all done!

3. addrbook <local|conf> get {1…n}

addrbook	**addrbook <local|conf> get {1…n}**	
Parameter introduction (control command type)	local	Configures local contacts information.
	conf	Configures conference contact information.
	get	Obtains the selected contacts' information.
	{1…n}	n should be a positive integer. It represents the number of contacts.
Sending format	addrbook <local|conf> get {1…n}\r\n	
Returning format	addrbook numid type "name" "numberlist"\r\n **Note:** - **numid**: contact id - **type**: contact type **local**: local contact **conf**: conference contact - **"name"**: contact's name strings - **"numberlist"**: contact's number strings（maybe multiple numbers）	
Example	Send: addrbook local get 2 return addrbook 1 local "xiaoming" "123456" "231456" \r\n addrbook 2 local "zhangpeng" "123456" "65412" "98745"\r\n addrbook local get 2 all done! send: addrbook conf get 3 return: addrbook 1 conf "groud1_conf" "11111" "22222" "5565655" "96363524"\r\n addrbook 2 conf "group2_conf" "222241" "65412" "654253"\r\n addrbook 3 conf "group3_conf" "123654" "32165" "52314"\r\n addrbook conf get 3 all done!	

4. addrbook search "searchstring"

addrbook	**addrbook search "searchstring"**

Parameter introduction (control command type)	search	Searches contacts' information.
	"searchstring"	Configures the name string to be searched.
Sending format	addrbook search "searchstring"\r\n	
Returning format	addrbook numid type "name" "numberlist"\r\n **Note:** - **numid**: contact id - **type**: contact type **local**: local contact **conf**: conference contact - **"name"**: contact's name strings - **"numberlist"**: contact's number strings（maybe multiple numbers）	
Example	Send: addrbook search "xiaoh" return addrbook 1 local "xiaoh" "123456" "231456" \r\n addrbook 2 conf "xiaoh" "123456" "65412" "98745"\r\n	

button

1. button power

button	button power
Parameter introduction (control command type)	Sends the power button signal to the video conferencing system.
Sending format	button power\r\n
Returning format	button power\r\n
Example	Send: button power\r\n return button power\r\n

2. button F1

button	**button F1**
Parameter introduction (control command type)	For VC110/VC120/VC400: sends the red button signal to the video conferencing system. For VC500/VC800/VC880: sends the recording button signal to the video conferencing system.
Sending format	button F1\r\n
Returning format	button F1\r\n
Example	Send: button F1\r\n return button F1\r\n

3. button F2

button	**button F2**
Parameter introduction (control command type)	For VC110/VC120/VC400: sends the yellow button signal to the video conferencing system. For VC500/VC800/VC880: sends the layout button signal to the video conferencing system.
Sending format	button F2\r\n
Returning format	button F2\r\n
Example	Send: button F2\r\n return button F2\r\n

4. button F3

button	**button F3**
Parameter introduction (control command type)	For VC110/VC120/VC400: sends the blue button signal to the video conferencing system. For VC500/VC800/VC880: sends the custom button (Presentation, Input, Screenshot or Mute Speaker) signal to the video conferencing system.
Sending format	button F3\r\n
Returning format	button F3\r\n
Example	Send: button F3\r\n return button F3\r\n

5. button volume+

button	button volume+
Parameter introduction (control command type)	Sends the volume + button signal to the video conferencing system.
Sending format	button volume+\r\n
Returning format	button volume+\r\n
Example	Send: button volume+\r\n return button volume+\r\n

6. button volume-

button	button volume-
Parameter introduction (control command type)	Sends the volume button signal to the video conferencing system.
Sending format	button volume-\r\n
Returning format	button volume-\r\n
Example	Send: button volume-\r\n return button volume-\r\n

7. button zoom+

button	button zoom+
Parameter introduction (control command type)	Sends the zoom+ button signal to the video conferencing system.
Sending format	button zoom+\r\n
Returning format	button zoom+\r\n
Example	Send: button zoom+\r\n return button zoom+\r\n

8. button zoom-

button	button zoom-
Parameter introduction (control command type)	Sends the zoom - button signal to the video conferencing system.
Sending format	button zoom-\r\n
Returning format	button zoom-\r\n
Example	Send: button zoom-\r\n

	return button zoom-\r\n

9. button up

button	**button up**
Parameter introduction (control command type)	Sends the up arrow button signal to the video conferencing system.
Sending format	button up\r\n
Returning format	button up\r\n
Example	Send: button up\r\n return button up\r\n

10. button down

button	**button down**
Parameter introduction (control command type)	Sends the down arrow button signal to the video conferencing system.
Sending format	button down\r\n
Returning format	button down\r\n
Example	Send: button down\r\n return button down\r\n

11. button right

button	**button right**
Parameter introduction (control command type)	Sends the right arrow button signal to the video conferencing system.
Sending format	button right\r\n
Returning format	button right\r\n
Example	Send: button right\r\n return button right\r\n

12. button left

button	**button left**
Parameter introduction (control command type)	Sends the left arrow button signal to the video conferencing system.
Sending format	button left\r\n

button	button left
Returning format	button left\r\n
Example	Send: button left\r\n return button left\r\n

13. button select

button	button select
Parameter introduction (control command type)	Sends the OK button signal to the video conferencing system.
Sending format	button select\r\n
Returning format	button select\r\n
Example	Send: button select\r\n return button select\r\n

14. button mute

button	button mute
Parameter introduction (control command type)	Sends the mute button signal to the video conferencing system.
Sending format	button mute\r\n
Returning format	button mute\r\n
Example	Send: button mute\r\n return button mute\r\n

15. button home

button	button home
Parameter introduction (control command type)	Sends the home button signal to the video conferencing system.
Sending format	button home\r\n
Returning format	button home\r\n
Example	Send: button home\r\n return button home\r\n

16. button show

button	button show
Parameter introduction (control command type)	Sends the video source button signal to the video conferencing system. **Note**: this parameter is only applicable to VC110/VC120/VC400.
Sending format	button show\r\n
Returning format	button show\r\n
Example	Send: button show\r\n return button show\r\n

17. button back

button	button back
Parameter introduction (control command type)	Sends the return button signal to the video conferencing system. **Note**: this parameter is only applicable to VC500/VC800/VC880.
Sending format	button back\r\n
Returning format	button back\r\n
Example	Send: Button back\r\n return button back\r\n

18. button call

button	button call
Parameter introduction (control command type)	Sends the off-hook button signal to the video conferencing system.
Sending format	button call\r\n
Returning format	button call\r\n
Example	Send: button call\r\n return button call\r\n

19. button delete

button	button delete
Parameter introduction	Sends the delete button signal to the video conferencing

(control command type)	system.
Sending format	button delete\r\n
Returning format	button delete\r\n
Example	Send: button delete\r\n return button delete\r\n

20. button hangup

button	**button hangup**
Parameter introduction (control command type)	Sends the on-hook button signal to the video conferencing system.
Sending format	button hangup\r\n
Returning format	button hangup\r\n
Example	Send: button hangup\r\n return button hangup\r\n

21. button <1|2|3|4|5|6|7|8|9|0|*|#>

button	**button <1|2|3|4|5|6|7|8|9|0|*|#>**
Parameter introduction (control command type)	Sends corresponding numeric button or # or * button signal to the video conferencing system.
Difference of models:	• For VC110/VC120/VC400: the* button can also be used to record video, # button can also be used to capture screenshot. • For VC500/VC800/VC880: # or * button does not provide special behaviors.
Sending format	button <1|2|3|4|5|6|7|8|9|0|*|#>\r\n
Returning format	button <1|2|3|4|5|6|7|8|9|0|*|#>\r\n
Example	Send: button 1\r\n return button 1\r\n

22. button recordstart

button	**button recordstart**
Parameter introduction (control command type)	Sends video recording button signal to the video conferencing system to start recording.
Sending format	button recordstart\r\n

Returning format	button recordstart\r\n
Example	Send: button recordstart\r\n return button recordstart\r\n

23. button recordstop

button	**button recordstop**
Parameter introduction (control command type)	Sends video recording button signal to the video conferencing system to stop recording.
Sending format	button recordstop\r\n
Returning format	button recordstop\r\n
Example	Send: button recordstop\r\n return button recordstop\r\n

24. button screenshot

button	**button screenshot**
Parameter introduction (control command type)	Sends snapshot button signal to the video conferencing system.
Sending format	button screenshot\r\n
Returning format	button screenshot\r\n
Example	Send: button screenshot\r\n return button screenshot\r\n

camera

1. camera near move <left|right|up|down|zoom+|zoom-|stop>

camera	**camera near move <left|right|up|down|zoom+|zoom-|stop>**	
Parameter introduction (control command type)	near	Controls the near-site camera.
	move	Changes the near-site camera's direction or zoom.
	<left|right|up|down|zoom+|zoom-|stop>	**left**: Starts moving the camera left.
		right: Starts moving the camera right.
		down:Starts moving the camera down.
		up: Starts moving the camera up.

		zoom+: Starts zooming in.
		zoom-: Starts zooming out.
		stop: Stops moving the near-site camera.
Sending format	camera near move <left|right|up|down|zoom+|zoom-|stop>\r\n	
Returning format	camera near move <left|right|up|down|zoom+|zoom-|stop>\r\n	
Example	Send: camera near move left\r\n return camera near move left\r\n	

2. camera near <getposition|setposition "x" "y" "z">

camera	**camera near <getposition|setposition "x" "y" "z">**	
Parameter introduction (control command type)	near	Controls the near-site camera.
	getposition	Obtains the pan(x), tilt(y), and zoom(z) coordinates of the currently selected PTZ camera in the format of pan tilt zoom.
	setposition	Configures the pan(x), tilt(y), and zoom(z) coordinates of the currently selected PTZ camera in the format of pan tilt zoom.
	"x"	0 <= pan <= 1920
	"y"	0 <= tilt <= 1080
	"z"	0 <= zoom <= 100
Sending format	camera near <getposition|setposition "x" "y" "z">\r\n	
Returning format	camera near <getposition|setposition "x" "y" "z">\r\n	
Example	Send: camera near get_id_list\r\n return camera near get_id_list "id:0" "id:1" "id:2" "id:3" "id:4"\r\n	

3. camera near get_id_list {"id:int"}

camera	**camera near get_id_list {"id:int"}** **Note**: This parameter only applies to video conferencing system that connects multiple cameras (VC800/VC880).	
Parameter introduction (control command type)	near	Controls the near-site camera.
	get_id_list	Obtains the cameras list.
	{"id:int"}	Indicates the camera Id. **Note:** This value depends on the cameras you connect to the video conferencing system (up to 9).

Sending format	camera near get_id_list\r\n
Returning format	
Example	Send: camera near get_id_list\r\n return camera near get_id_list "id:0" "id:1" "id:2" "id:3" "id:4"\r\n

4. camera near get_id_detial "id:int" "status:int" "name:string" "default:int" "ip:string" "mac:string"

camera	**camera near get_id_detial "id:int" "status:int" "name:string" "default:int" "ip:string" "mac:string"** **Note**: This parameter only applies to video conferencing system that connects multiple cameras (VC800/VC880).	
Camera Parameter	id	Indicates the camera Id.
	status	Indicates the camera status. • unkown = (1<<0) • disconnect = (1<<1), • connect = (1<<2) • connect_disable =(1<<3) • connect_available = (1<<4) • connect_active=(1<<5) **Note:** separate multiple camera statuses by pipes. **For example:** status = connect|active|available
	name	The camera name.
	default	Indicates whether the camera is using the default name.
	ip	The IP address of the camera
	mac	The MAC address of the camera
Parameter introduction (control command type)	near	Controls the near-site camera.
	get_id_detial	Obtains the camera detail.
	{"id:int"}	Indicates the camera Id. **Note:** This value depends on the cameras you connect to the video conferencing system (up to 9).
Sending format	camera near get_id_detial "id:int"\r\n	
Returning format	camera near get_id_detial "id:int" "status:int" "name:string" "default:int" "ip:string" "mac:string"	
Example	Send: camera near get_id_detial "id:int"\r\n return camera near get_id_detial "id:0" "status:4" "name:camera 1"	

	"default:1" "ip:192.168.0.122" "mac:00-15-65-20-36-14"\r\n

5. camera near set_active_status "id:int"

camera	**camera near set_active_status "id:int"** **Note**: This parameter only applies to the video conferencing system that connects multiple cameras (VC800/VC880).	
Parameter introduction (control command type)	near	Controls the near-site camera.
	set_active_status	Activates the desired camera. **Note**: you can configure the camera parameters only when it is activated.
	{"id:int"}	Indicates the camera Id. **Note:** This value depends on the cameras you connect to the video conferencing system (up to 9).
Sending format	camera near set_active_status "id:int"\r\n	
Returning format	camera near set_active_status "id:int"\r\n	
Example	Send: camera near set_active_status "id:1"\r\n return camera near set_active_status "id:1"\r\n	

6. camera near move_id "id:int" "direct:x"|stop

camera	**camera near move_id "id:int" "direct:x"|stop** **Note**: This parameter only applies to video conferencing system that connects multiple cameras (VC800/VC880).	
Parameter introduction (control command type)	near	Controls the near-site camera.
	move_id	Moves the desired camera.
	{"id:int"}	Indicates the camera Id. **Note:** This value depends on the cameras you connect to the video conferencing system (up to 9).
	"direct:x"|stop	• Direct: move the camera • 2: move down • 4:move up • 6:move right • 8:move up • Stop: stop moving
Sending format	camera near move_id "id:int" "direct:int"|stop\r\n	
Returning format	camera near move_id "id:int" "direct:int"|stop	
Example	Send:	

	camera near move_id "id:1" "direct:2"\r\n return camera near move_id "id:1" "direct:2"\r\n Send: camera near move_id "id:1" stop\r\n return camera near move_id "id:1" stop\r\n

7. camera near zoom_id "id:x" "direct:int"|stop

camera	**camera near zoom_id "id:int" "direct:int"|stop** **Note**: This parameter only applies to video conferencing system that connects multiple cameras (VC800/VC880).	
Parameter introduction (control command type)	near	Controls the near-site camera.
	zoom_id	Zoom the camera in or out.
	{"id:int"}	Indicates the camera Id. **Note:** This value depends on the cameras you connect to the video conferencing system (up to 9).
	"direct:x"|stop	• Direct: move the camera • 0:zoom out • 1:zoom in • Stop: stop zooming
Sending format	camera near zoom_id "id:int" "direct:int"|stop\r\n	
Returning format	camera near zoom_id "id:int" "direct:int"|stop	
Example	Send: camera near zoom_id "id:1" "direct:0"\r\n return camera near zoom_id "id:1" "direct:0"\r\n Send: camera near zoom_id "id:1" stop\r\n return camera near zoom_id "id:1" stop\r\n	

callinfo

1. callinfo all

callinfo	**callinfo all**	
Parameter introduction (control command type)	all	Returns information about each connection in the call.

Sending format	callinfo all\r\n
Returning format	**For audio call:** callinfo audio "callid:63214" "RemoteStr:10.10.37.21@10.10.37.21" "direction:incomging" "protocol:sip" "devInfo:YealiInk VC110 50.20.251.31 38/2""TotalBwRecv:48" "TotalBwSend:48" "AcodecRecv:G.722.1C" "AcodecSend:G.722.1C" "ABwRecv:48" "ABwSend:48′ "ASrRecv:32" "ASrSend:32" "AJrRecv:6" "AJrSend:6" "AtplRecv:0" "AtplSend:0" "AtplpRecv:0" "AtplpSend:0"\r\n **For video call with received content:** callinfo video "callid:63214" "RemoteStr:10.10.37.21@10.10.37.21" "direction:outgoing" "protocol:sip" "devInfo:YealiInk VC110 50.20.251.31 38/2""TotalBwRecv:1664" "TotalBwSend:2096" "VResRecv:192*1080" "VResSend:1920*1080" "VCodecRecv:H.264" "VCodecSend:H.264" "VBwRecv:1547" "VBwSend:2036" "VFrRecv:25:" "VFrSend:29" "VJrRecv:17" "VJrSend:16" "VtplRecv:0" "VtplSend:0" "VtplpRecv:0" "VtplpSend:0" "AcodecRecv:G.722.1C" "AcodecSend:G.722.1C" "ABwRecv:48" "ABwSend:48′ "ASrRecv:32" "ASrSend:32" "AJrRecv:6" "AJrSend:6" "AtplRecv:0" "AtplSend:0" "AtplpRecv:0" "AtplpSend:0" "SResRecv:0" "SResSend:0" "SCodecRecv:0" "SCodecSend:0" "SBwRecv:0" "SBwSend:0" "SFrRecv:0" "SFrSend:0"\r\n **Parameter explanation:** 1. **TotalBwRecv**: receive total bandwidths 2. **TotalBwSend**: transmit total bandwidths 3. **VResRecv:** receive video resolution 4. **VResSend:** transmit video resolution 5. **VCodecRecv:** receive video Codec type 6. **VCodecSend:** transmit video Codec type 7. **VBwRecv:** receive video bandwidths 8. **VBwSend:** transmit video bandwidths 9. **VFrRecv:** receive video frame rate 10. **VFrSend:**transmit video frame rate 11. **VJrRecv:** receive video Jitter (ms) 12. **VJrSend:** transmit video Jitter (ms) 13. **VtplRecv:** receive video packet loss 14. **VtplSend:** transmit video packet loss 15. **VtplpRecv:** receive video packet loss(%) 16. **VtplpSend:** transmit video packet loss(%)

	17. **AcodecRecv:** receive audio Codec type 18. **AcodecSend:** transmit audio Codec type 19. **ABwRecv:** receive audio bandwidths 20. **ABwSend:** transmit audio bandwidths 21. **ASrRecv:** receive audio sample rate(k) 22. **ASrSend:** transmit audio sample rate(k) 23. **AJrRecv:** receive audio Jitter (ms) 24. **AJrSend:** transmit audio Jitter (ms) 25. **AtplRecv:** receive audio packet loss 26. **AtplSend:** transmit audio packet loss 27. **AtplpRecv:** receive audio packet loss (%) 28. **AtplpSend:** transmit audio packet loss (%) 29. **SResRecv:** receive content resolution 30. **SResSend:** transmit content resolution 31. **SCodecRecv:** receive content Codec type 32. **SCodecSend:** transmit content Codec type 33. **SBwRecv:** receive content bandwidths 34. **SBwSend:** transmit content bandwidths 35. **SFrRecv:** receive content frame rate 36. **SFrSend:** transmit content frame rate **Units of bandwidth: kb/s** **Units of frame rate: fps**
Example	When the video conferencing system is during a video call and an audio call, the control system can get all call statistics. Send: callinfo all\r\n Return callinfo audio "callid:63214" "RemoteStr:10.10.37.21@10.10.37.21" "direction:incomging" "protocol:sip" "devInfo:YealiInk VC110 50.20.251.31 38/2""TotalBwRecv:48" "TotalBwSend:48" "AcodecRecv:G.722.1C" "AcodecSend:G.722.1C" "ABwRecv:48" "ABwSend:48′ "ASrRecv:32" "ASrSend:32" "AJrRecv:6" "AJrSend:6" "AtplRecv:0" "AtplSend:0" "AtplpRecv:0" "AtplpSend:0"\r\n callinfo video "callid:63214" "RemoteStr:10.10.37.21" "direction:outgoing" "protocol:sip" "devInfo:YealiInk VC110 50.20.251.31 38/2""TotalBwRecv:1664" "TotalBwSend:2096" "VResRecv:192*1080" "VResSend:1920*1080" "VCodecRecv:H.264" "VCodecSend:H.264" "VBwRecv:1547" "VBwSend:2036" "VFrRecv:25:" "VFrSend:29" "VJrRecv:17" "VJrSend:16" "VtplRecv:0" "VtplSend:0"

	"VtplpRecv:0" "VtplpSend:0" "AcodecRecv:G.722.1C" "AcodecSend:G.722.1C" "ABwRecv:48" "ABwSend:48′ "ASrRecv:32" "ASrSend:32" "AJrRecv:6" "AJrSend:6" "AtplRecv:0" "AtplSend:0" "AtplpRecv:0" "AtplpSend:0" "SResRecv:0" "SResSend:0" "SCodecRecv:0" "SCodecSend:0" "SBwRecv:0" "SBwSend:0" "SFrRecv:0" "SFrSend:0"\r\n callinfo all all done\r\n

2. callinfo callid "callid"

callinfo	**callinfo callid "callid"**	
Parameter introduction (control command type)	callid	Returns information about the connection with the specified call ID.
Sending format	callinfo callid "63214"\r\n	
Returning format	**For audio call:** callinfo audio "callid:63214" "RemoteStr:10.10.37.21@10.10.37.21" "direction:incomging" "protocol:sip" "devInfo:YealiInk VC110 50.20.251.31 38/2""TotalBwRecv:48" "TotalBwSend:48" "AcodecRecv:G.722.1C" "AcodecSend:G.722.1C" "ABwRecv:48" "ABwSend:48′ "ASrRecv:32" "ASrSend:32" "AJrRecv:6" "AJrSend:6" "AtplRecv:0" "AtplSend:0" "AtplpRecv:0" "AtplpSend:0"\r\n **For video call without received content:** callinfo video "callid:63214" "RemoteStr:10.10.37.21@10.10.37.21" "direction:outgoing" "protocol:sip" "devInfo:YealiInk VC110 50.20.251.31 38/2""TotalBwRecv:1664" "TotalBwSend:2096" "VResRecv:192*1080" "VResSend:1920*1080" "VCodecRecv:H.264" "VCodecSend:H.264" "VBwRecv:1547" "VBwSend:2036" "VFrRecv:25:" "VFrSend:29" "VJrRecv:17" "VJrSend:16" "VtplRecv:0" "VtplSend:0" "VtplpRecv:0" "VtplpSend:0" "AcodecRecv:G.722.1C" "AcodecSend:G.722.1C" "ABwRecv:48" "ABwSend:48′ "ASrRecv:32" "ASrSend:32" "AJrRecv:6" "AJrSend:6" "AtplRecv:0" "AtplSend:0" "AtplpRecv:0" "AtplpSend:0" "SResRecv:0" "SResSend:0" "SCodecRecv:0" "SCodecSend:0" "SBwRecv:0" "SBwSend:0" "SFrRecv:0" "SFrSend:0"\r\n **For video call with received content (local system is the receiver):** callinfo video "callid:63214" "RemoteStr:10.10.37.21@10.10.37.21" "direction:outgoing" "protocol:sip" "devInfo:YealiInk VC110 50.20.251.31 38/2""TotalBwRecv:1664" "TotalBwSend:2096" "VResRecv:192*1080" "VResSend:1920*1080" "VCodecRecv:H.264" "VCodecSend:H.264" "VBwRecv:1547" "VBwSend:2036" "VFrRecv:25:" "VFrSend:29" "VJrRecv:17" "VJrSend:16" "VtplRecv:0" "VtplSend:0" "VtplpRecv:0" "VtplpSend:0" "AcodecRecv:G.722.1C"	

	"AcodecSend:G.722.1C" "ABwRecv:48" "ABwSend:48′ "ASrRecv:32" "ASrSend:32" "AJrRecv:6" "AJrSend:6" "AtplRecv:0" "AtplSend:0" "AtplpRecv:0" "AtplpSend:0" "SResRecv:1920*1080" "SResSend:0" "SCodecRecv:H.264" "SCodecSend:0" "SBwRecv:1974" "SBwSend:0" "SFrRecv:30" "SFrSend:0"\r\n **For video call with shared content (local system is the sender):** callinfo video "callid:63214" "RemoteStr:10.10.37.21@10.10.37.21" "direction:outgoing" "protocol:sip" "devInfo:Yealilnk VC110 50.20.251.31 38/2""TotalBwRecv:1664" "TotalBwSend:2096" "VResRecv:192*1080" "VResSend:1920*1080" "VCodecRecv:H.264" "VCodecSend:H.264" "VBwRecv:1547" "VBwSend:2036" "VFrRecv:25:" "VFrSend:29" "VJrRecv:17" "VJrSend:16" "VtplRecv:0" "VtplSend:0" "VtplpRecv:0" "VtplpSend:0" "AcodecRecv:G.722.1C" "AcodecSend:G.722.1C" "ABwRecv:48" "ABwSend:48′ "ASrRecv:32" "ASrSend:32" "AJrRecv:6" "AJrSend:6" "AtplRecv:0" "AtplSend:0" "AtplpRecv:0" "AtplpSend:0" "SResRecv:0" "SResSend:1920*1080" "SCodecRecv:" "SCodecSend:H.264" "SBwRecv:" "SBwSend:1974" "SFrRecv:0" "SFrSend:30"\r\n **Parameter explanation:** 1. **TotalBwRecv**: receive total bandwidths 2. **TotalBwSend**: transmit total bandwidths 3. **VResRecv:** receive video resolution 4. **VResSend:** transmit video resolution 5. **VCodecRecv:** receive video Codec type 6. **VCodecSend:** transmit video Codec type 7. **VBwRecv:** receive video bandwidths 8. **VBwSend:** transmit video bandwidths 9. **VFrRecv:** receive video frame rate 10. **VFrSend:**transmit video frame rate 11. **VJrRecv:** receive video Jitter (ms) 12. **VJrSend:** transmit video Jitter (ms) 13. **VtplRecv:** receive video packet loss 14. **VtplSend:** transmit video packet loss 15. **VtplpRecv:** receive video packet loss(%) 16. **VtplpSend:** transmit video packet loss(%) 17. **AcodecRecv:** receive audio Codec type 18. **AcodecSend:** transmit audio Codec type 19. **ABwRecv:** receive audio bandwidths

	20. **ABwSend:** transmit audio bandwidths 21. **ASrRecv:** receive audio sample rate(k) 22. **ASrSend:** transmit audio sample rate(k) 23. **AJrRecv:** receive audio Jitter (ms) 24. **AJrSend:** transmit audio Jitter (ms) 25. **AtplRecv:** receive audio packet loss 26. **AtplSend:** transmit audio packet loss 27. **AtplpRecv:** receive audio packet loss (%) 28. **AtplpSend:** transmit audio packet loss (%) 29. **SResRecv:** receive content resolution 30. **SResSend:** transmit content resolution 31. **SCodecRecv:** receive content Codec type 32. **SCodecSend:** transmit content Codec type 33. **SBwRecv:** receive content bandwidths 34. **SBwSend:** transmit content bandwidths 35. **SFrRecv:** receive content frame rate 36. **SFrSend:** transmit content frame rate **Units of bandwidth: kb/s** **Units of frame rate: fps**

dial

1. dial auto "dialstring"

dial	**dial auto "dialstring"** (dial one contact) or **dial auto "dialnumber1" "dialnumber2" "dialnumber3"** (dial the conference contacts) **Note**: This is a dial command. When you dial a contact, the video conferencing system will call a number. When you dial a conference contact, the video conferencing system will dial multiple numbers. In other words, when sending one number, the video conferencing system will call a local contact, when sending multiple numbers, the video conferencing system will call the conference contact.	
Parameter introduction (control command type)	auto	Allows the user to dial a number using default call type and call protocol.
	"dialstring"	Valid phone numbers string.

Sending format	dial auto "dialstring"\r\n or dial auto "dialnumber" "dialnumber" "dialnumber"\r\n
Returning format	dial auto "dialstring"\r\n or dial auto "dialnumber" "dialnumber" "dialnumber"\r\n
Example	When dialing a local contact: Send: dial auto "7001"\r\n return dial auto "7001"\r\n When dialing a conference contact: Send: dial auto "7001" "7002" "7003"\r\n return dial auto "7001" "7002" "7003"\r\n

2. dial manual <video|audio|auto> <auto|sip|h323> "speed" "dialstring"

Dial	**dial manual <video|audio|auto> <auto|sip|h323> "speed" "dialstring"** **Note**: It is only applicable to dialing a contact.	
Parameter introduction (control command type)	manual	Allows the user to dial a number using call type and call protocol set manually.
	"speed"	Specifies the bandwidth. Default value is "Auto".
	<video|audio|auto>	Specifies the desired call type for placing calls. Valid values: - **Auto**: the video conferencing system automatically uses the default call type. Default value is Video. - **Video**: the video conferencing system dials a video call. - **Audio**: the video conferencing system dials an audio call.
	<auto|sip|h323>	Specifies the desired call protocol for placing calls. Valid values: - **Auto**: the video conferencing system automatically uses the default call protocol. Default value is H.323. - **SIP**: the video conferencing system

		uses the SIP protocol for placing calls. - **H.323**: the video conferencing system uses H.323 protocol for placing calls.
	"dialstring"	Valid phone numbers string.
Sending format	dial manual <video|audio|auto> <auto|sip|h323> "speed" "dialstring"\r\n	
Returning format	dial manual <video|audio|auto> <auto|sip|h323> "speed" "dialstring"\r\n	
Example	Send: dial manual video sip "auto" "70001"\r\n return dial manual video sip "auto" "70001"\r\n **Specific bandwidth:** Send: dial manual video sip "1024" "70001"\r\n return dial manual video sip "1024" "70001"\r\n **Note**: enclose the variable values (number and bandwidth) in quotation marks.	

donotdisturb

1. donotdisturb global <get|on|off>

donotdisturb	**donotdisturb global <get|on|off>**	
Parameter introduction (control command type)	global	Configures DND for the video conferencing system.
	params	Valid values: - **get**: obtains the DND status. - **on**: enables the DND feature. - **off**: disables the DND feature.
Sending format from the control system	donotdisturb global <get|on|off>\r\n	
Returning format received by the control system	donotdisturb global get <on|off>\r\n or donotdisturb global <on|off>\r\n	
The video conferencing system provides active feedback to the control system	donotdisturb global get <on|off>\r\n **Note:** Once the DND status changes, the video conferencing system will provide active feedback to the control system. **Format:**	

	• The video conferencing system provides active feedback to the control system: donotdisturb global get on\r\n • The control system receives the command: donotdisturb global get on\r\n • The video conferencing system provides active feedback to the control system: donotdisturb global get off\r\n • The control system receives the command: donotdisturb global get off\r\n
Example	Send: donotdisturb global get\r\n return donotdisturb global get on\r\n send: donotdisturb global get\r\n return donotdisturb global get off\r\n send: donotdisturb global on\r\n return donotdisturb global on\r\n

2. donotdisturb talk <get|on|off>

donotdisturb	**donotdisturb talk <get|on|off>**	
Parameter introduction (control command type)	talk	Configures DND during a call.
	params	Valid values: - **get**: obtains the DND status during a call. - **on**: enables the DND feature during a call. - **off**: disables the DND feature during a call.
Sending format	donotdisturb talk <get|on|off>\r\n	
Returning format	donotdisturb talk get <on|off>\r\n or donotdisturb talk <on|off>\r\n	
The video conferencing system provides active feedback to the control system	donotdisturb talk get <on|off>\r\n **Note:** Once the DND status in a call changes, the video conferencing system will provide active feedback to the control system. **Format:**	

donotdisturb	donotdisturb talk <get|on|off>
	• The video conferencing system provides active feedback to the control system: donotdisturb talk get on\r\n • The control system receives the command: donotdisturb talk get on\r\n • The video conferencing system provides active feedback to the control system: donotdisturb talk get off\r\n • The control system receives the command: donotdisturb talk get off\r\n
Example	Send: donotdisturb talk get\r\n return donotdisturb talk get on\r\n send: donotdisturb talk get\r\n return donotdisturb talk get off\r\n send: donotdisturb talk on\r\n return donotdisturb talk on\r\n

gendial

1. gendial <0|1|2|3|4|5|6|7|8|9|*|#>

gendial	gendial <0|1|2|3|4|5|6|7|8|9|*|#>
Parameter introduction (control command type)	Generates DTMF dialing tones.
Sending format	gendial <0|1|2|3|4|5|6|7|8|9|#|*>\r\n
Returning format	gendial <0|1|2|3|4|5|6|7|8|9|#|*>\r\n
Example	Send: gendial 0\r\n return gendial 0\r\n

getcallid

1. getcallid

getcallid	getcallid
Parameter introduction (control command type)	Returns call ID information about each connection in the call.
Sending format	getcallid\r\n
Returning format	getcallid "callid: 654123" "remotestr: 10.3.3.2"\r\n
Example	Send: getcallid\r\n return getcallid "callid: 654121" "remotestr: 10.3.3.1"\r\n getcallid "callid: 654122" "remotestr: 10.3.3.2"\r\n getcallid "callid: 654123" "remotestr: 10.3.3.3"\r\n getcallid "callid: 654124" "remotestr: 10.3.3.4"\r\n getcallid all done\r\n

history

1. history all

history	history all	
Parameter introduction (control command type)	all	Lists all call history.
Sending format	history all\r\n	
Returning format	history numid type "name" "date" "duration" "numberlist"\r\n **Note:** - **numid**: call history id - **type**: call history type **placed**: placed calls **received**: received calls **misscalled**: missed calls - **"name"**: name strings of the call history - **"date"** : date string of the call history - **"duration"**: duration of the call history - **"numberlist"**: number strings of the call history	
Example	Send: history all	

	return history 1 placed "xiaom" "2015-06-01" "00:00:43" "123456" "321654" "222222333"\r\n history 2 placed "mengsde" "2015-06-01" "00:01:43" "16532" "328888" "565622333"\r\n history 3 received "mengsde" "2015-06-01" "00:11:43" "3616532" "865328888" "96365622333"\r\n history 4 misscalled "mengsde" "2015-07-01" "00:00:00" "3363633" "756288" "363333"\r\n history all all done!

2. history <placed|received|misscalled> get all

history	**history <placed|received|misscalled> get all**	
Parameter introduction (control command type)	placed	Configures the placed calls. **Note**: Placed call record that has only one number represents a call to a contact. Placed call record that has multiple numbers represents a call to a conference contact. Conference contact has "placed calls" only. It does not support other call history type.
	received	Configures the received calls.
	misscalled	Configures the missed calls.
	get	Obtains the call history.
	all	Obtains all call history in selected history type.
Sending format	history <placed|received|misscalled> get all\r\n	
Returning format	history numid type "name" "date" "duration" "numberlist"\r\n **Note:** - **numid**: call history id - **type**: call history type **placed**: placed calls **received**: received calls **misscalled**: missed calls - **"name"**: name strings of the call history - **"date"** : date string of the call history - **"duration"**: duration of the call history - **"numberlist"**: number strings of the call history	

history	history <placed|received|misscalled> get all
Example	Send: history placed get all\r\n return history 1 placed "xiaom" "2015-06-01" "00:00:43" "123456" "321654" "222222333"\r\n history 2 placed "mengsde" "2015-06-01" "00:01:43" "16532" "328888" "565622333"\r\n history placed get all all done!

3. history <placed|received|misscalled> get {1…n}

history	history <placed|received|misscalled> get {1…n}	
Parameter introduction (control command type)	placed	Configures the placed calls.
	received	Configures the received calls.
	misscalled	Configures the missed calls.
	get	Obtains the call history.
	{1…n}	n should be a positive integer. It represents the number of call history.
Sending format	history <placed|received|misscalled> get {1…n}\r\n	
Returning format	history numid type "name" "date" "duration" "numberlist"\r\n **Note:** - **numid**: call history id - **type**: call history type **placed**: placed calls **received**: received calls **misscalled**: missed calls - **"name"**: name strings of the call history - **"date"** : date string of the call history - **"duration"**: duration of the call history - **"numberlist"**: number strings of the call history	
Example	Send: history placed get 2\r\n return history 1 placed "xiaom" "2015-06-01" "00:00:43" "123456" "321654" "222222333"\r\n history 2 placed "mengsde" "2015-06-01" "00:01:43" "16532" "328888" "565622333"\r\n	

	history placed get 2 all done!

inputsource

1. inputsource camera

inputsource	**inputsource camera**	
Parameter introduction (control command type)	camera	Specifies the camera to be the video source.
Sending format	inputsource camera\r\n	
Returning format	inputsource camera\r\n	
Example	Send: inputsource camera\r\n return inputsource camera\r\n	

2. inputsource pc

inputsource	**inputsource pc**	
Parameter introduction (control command type)	pc	Specifies the PC to be the video source.
Sending format	inputsource pc\r\n	
Returning format	inputsource pc\r\n	
Example	Send: inputsource pc\r\n return inputsource pc\r\n	

3. inputsource share

inputsource	**inputsource share**	
Parameter introduction (control command type)	share	Specifies the PC+camera to be the video source. **Note:** PC+camera can be selected during a call only.
Sending format	inputsource share\r\n	
Returning format	inputsource share\r\n	
Example	Send: inputsource share\r\n return inputsource share\r\n	

incoming

1. incoming "num:string" "name:string"

incoming	incoming "num:string" "name:string"
Parameter introduction (feedback command type)	The video conferencing system will provide active feedback to the control system when receiving an incoming call. Valid values: - Incoming phone number - Caller name
Sending format from the video conferencing system	incoming "num:string" "name:string"\r\n
Format received by control system	incoming "num:string" "name:string"\r\n
Example	Send: incoming "num:700051" "name:xiaopeng"\r\n return incoming "num:700051" "name:xiaopeng"\r\n

layout

1. layout near get

layout	layout near get	
Parameter introduction	near	Local video layout
	get	Obtain local video layout
Sending format from the video conferencing system	layout near get \r\n	
Difference of models	This API command is not applicable to VC110/VC120/VC400.	
Format received by control system	layout near get layoutmode idlist\r\n **Note**: the supported layout modes are: - **equal**: every participant is given equal prominence in equal-sized panes. - **surround**: the assigned participant is given prominence in the largest pane regardless of who is currently speaking. Other participants are displayed in a strip beside the assigned speaker. - **Fullscreen**: only the selected speaker is seen in a large pane.	

layout	layout near get
	- **share**: only the PC content is seen in a large pane. - **pip**: During two-way video calls, one participant is shown in full screen, while the other participant is shown in the PIP (Picture-in-Picture). This video layout is only applicable to VC500/VC800/VC880. - **empspk**: during multi-way video calls, the active speaker is given prominence in the largest pane. Other participants are displayed in a strip beside the active speaker. - **Idlist:** indicates the call ID, or pc | share |camera
Example	Send: layout near get\r\n return layout near get equal "23211" \r\n

2. layout near get list

layout	layout near get list	
Parameter introduction	near	Local video layout
	get	Obtain local video layout
	list	Obtain local video layout list
Sending format from the video conferencing system	layout near get list\r\n	
Difference of models	This API command is not applicable to VC110/VC120/VC400.	
Format received by control system	layout near get list layoutmode idlist\r\n **Note**: the supported layout modes are: - **equal**: every participant is given equal prominence in equal-sized panes. - **surround**: the assigned participant is given prominence in the largest pane regardless of who is currently speaking. Other participants are displayed in a strip beside the assigned speaker. - **Fullscreen**: only the selected speaker is seen in a large pane. - **share**: only the PC content is seen in a large pane. - **pip**: During two-way video calls, one participant is shown in full screen, while the other participant is shown in the PIP (Picture-in-Picture). This video layout is only applicable to	

layout	layout near get list
	VC500/VC800/VC880. - **empspk**: during multi-way video calls, the active speaker is given prominence in the largest pane. Other participants are displayed in a strip beside the active speaker. - **Idlist:** indicates the call ID, or pc | share |camera
Example	Send: layout near get list\r\n return layout near get list equal pip fullscreen \r\n

3. layout near set layoutmode idlist

layout	layout near set layoutmode idlist	
Parameter introduction	near	Local video layout
	set	Set local video layout
	layoutmode	Supported layout modes (as described above).
	idlist	Indicates the call ID, or pc | share |camera
Difference of models	This API command is not applicable to VC110/VC120/VC400.	
Sending format from the video conferencing system	layout near set equal callid\r\n	
Format received by control system	layout near|far set equal callid\r\n	
Example	Send: layout near set equal "123"\r\n return layout near set equal "123"\r\n	

4. layout near camera_layout_get

layout	layout near camera_layout_get	
Parameter introduction	near	Local video layout
	camera_layout_get	Obtain the current multi-camera layout
Difference of	This API command is not applicable to VC110/VC120/VC400.	

layout	layout near camera_layout_get
models	
Sending format from the video conferencing system	layout near camera_layout_get \r\n
Format received by control system	layout near get layoutmode idlist\r\n **Note**: the supported layout modes are: - **equal**: every participant is given equal prominence in equal-sized panes. - **surround**: the assigned participant is given prominence in the largest pane regardless of who is currently speaking. Other participants are displayed in a strip beside the assigned speaker. - **Fullscreen**: only the selected speaker is seen in a large pane. **Idlist:** indicates the corresponding camera ID in the current mode (It is obtained by the parameter "camera near get_id_list").
Example	Send layout near camera_layout_get \r\n return layout near camera_layout_get fullscreen "2" \r\n

5. layout near camera_layout_set layoutmode idlist

layout	layout near camera_layout_set layoutmode idlist	
Parameter introduction	near	Local video layout
	camera_layout_set	Set multi-camera layout
	layoutmode	Supported layout modes (as described above).
	idlist	Indicates the corresponding camera ID in the current mode (It is obtained by the parameter "camera near get_id_list")
Difference of models	This API command is not applicable to VC110/VC120/VC400.	
Sending format from the video conferencing system	layout near camera_layout_set fullscreen "1"\r\n	
Format received by control	layout near camera_layout_set fullscreen "1" \r\n	

layout	layout near camera_layout_set layoutmode idlist
system	
Example	Send: layout near camera_layout_set fullscreen "1" \r\n return layout near camera_layout_set fullscreen "1" \r\n

Mute

1. mute near <get|on|off|toggle>

mute	mute near <get|on|off|toggle>	
Parameter introduction (control command type)	near	Mutes local video conferencing system.
	<get|on|off|toggle>	Valid values: **get**: obtains the mute status **on**: enables the mute feature **off**: disables the mute feature **toggle**: toggles between mute and unmute status.
Sending format from the control system	mute near <get|on|off|toggle>\r\n	
Returning format received by the control system	mute near get <on|off>\r\n or mute near <on|off|toggle>\r\n	
The video conferencing system provides active feedback to the control system	mute near get <on|off>\r\n **Note:** Once the mute status changes, the video conferencing system will provide active feedback to the control system. **Format:** • The video conferencing system provides active feedback to the control system: mute near get on\r\n • The control system receives the command: mute near get on\r\n • The video conferencing system provides active feedback to the control system: mute near get off\r\n • The control system receives the command:	

	mute near get off\r\n
Example	Send: mute near get return mute near get on\r\n send: mute near get return mute near get off\r\n send: mute near on return mute near on\r\n

preset

1. preset near <go|set> <0|1|2|3|4|5|6|7|8|9>

Preset	**preset near <go|set> <0|1|2|3|4|5|6|7|8|9>**	
Parameter introduction (control command type)	near	Configures the presets for the near-site camera.
	go	Moves the camera to a camera preset.
	set	Configures a camera preset.
	<0|1|2|3|4|5|6|7|8|9>	Camera preset identifier.
Sending format	preset near <go|set> <0|1|2|3|4|5|6|7|8|9>\r\n	
Returning format	preset near <go|set> <0|1|2|3|4|5|6|7|8|9>\r\n	
Example	Send: preset near go 0\r\n return preset near go 0\r\n	

storage

1. storage get

storage	**storage get**	
Checking parameter, but the system will provide active feedback too	get	Obtains the USB status.
Sending format	Checking format	storage get\r\n
Returning format	The video conferencing system provides active feedback to	

storage	storage get
	the control system or the control system receives the command: storage get available\r\n or storage get unavailable\r\n **Note:** Once USB status changes, the video conferencing system will provide active feedback to the control system. If LAN mode and serial port mode are configured, both of them are able to receive the feedback.
Example	Send: storage get\r\n return storage get available\r\n storage get\r\n return storage get unavailable\r\n

sysstatus

1. sysstatus get

sysstatus	Sysstatus get
Parameter introduction	Obtains status notifications. **Note:** The video conferencing system may in multiple states. The control system can query the video conferencing system's status. The video conferencing system can also provide feedback to the control system.
Sending format	sysstatus get\r\n
Video conferencing system status	**1.** **sleeping** (the video conferencing system is sleeping) **2.** **Idle** (the video conferencing system is idle) **3.** **outgoing** (the video conferencing system is placing a call) **4.** **ringing** (the video conferencing system receives an incoming calls) **5.** **talking** (the video conferencing system establishes a call) **6.** **finished** (the video conferencing system finishes a call) **7.** **talking max** (the video conferencing system has reached maximum sessions)

Example	**sleeping (the video conferencing system is sleeping）** **The video conferencing system provides active feedback to the control system:** sysstatus get sleeping\r\n **Sending format from the control system** Send sysstatus get\r\n return sysstatus get sleeping\r\n sysstatus get all done! **Idle (the video conferencing system is idle）** **The video conferencing system provides active feedback to the control system:** sysstatus get idle\r\n **Sending format from the control system** Send sysstatus get\r\n return sysstatus get idle\r\n sysstatus get all done! **talking max (the video conferencing system has reached maximum sessions)** **The video conferencing system provides active feedback to the control system:** sysstatus get talking max\r\n **Sending format from the control system** Send sysstatus get\r\n Return sysstatus get talking max\r\n sysstatus get all done! **outgoing (the video conferencing system is placing a call)** **The video conferencing system provides active feedback to the control system:** sysstatus get outgoing "diastr:9865412" "callid: 653214"

	"calltype: video"\r\n **Sending format from the control system** sysstatus get\r\n Return sysstatus get outgoing "diastr:9865412" "callid: 653214" "calltype: video"\r\n sysstatus get all done! **ringing (the video conferencing system receives an incoming calls)** **The video conferencing system provides active feedback to the control system:** sysstatus get ringing "dialstr:9865412" "callid:653214" "calltype:autdio"\r\n **Sending format from the control system** Send: sysstatus get\r\n return sysstatus get ringing "dialstr:9865412" "callid:653214" "calltype:autdio"\r\n sysstatus get all done! **talking (the video conferencing system establishes a call)** **The video conferencing system provides active feedback to the control system:** systatus get talking "dialstr:9865412" "callid:654321" "calltype:audio" "protocol:sip" "direction: incoming/outgoing"\r\n **Sending format from the control system** Send systatus get\r\n return sysstatus get talking "dialstr:9865412" "callid:654321" "calltype:audio" "protocol:sip" "direction: incoming/outgoing"\r\n sysstatus get all done! **finished (the video conferencing system finishes a call)** **The video conferencing system provides active feedback to the control system:**

	sysstatus get finished "dialstr: 98653214" "callid:632156"\r\n Finished status is a real-time status, it does not last for a certain period, so you can obtain finished status from the feedback only. You cannot query it from your control system.
Note	**1.** You may get multiple states from the **Sysstatus get** command. For example, the video conferencing system receives an incoming call during a call: Send syssstatus get\r\n Return sysstatus get talking "dialstr:9865412" "callid:654321" "calltype:audio" "protocol: sip" "direction: incoming/outgoing"\r\n sysstatus get ringing "dialstr:9865412" "callid:653214" "calltype:autdio"\r\n sysstatus get all done! **2.** The video conferencing system does not keep noticing its status during a call, it only notice its status every time a call is established. And for H.323 calls, the video conferencing system will notice its status again when switching an audio call to a video call.

volume

1. volume <get|up|down|set {0..10}>

volume	**volume <get|up|down|set {0..10}>**	
Parameter introduction	get	Obtains the audio volume on the video conferencing system.
	up	Increases the audio volume.
	down	Decreases the audio volume.
	set	Configures the volume to a specified level. Requires a volume setting from 0-10.
Sending format	volume <get|up|down|set>\r\n	
Returning format	volume <get|up|down|set>\r\n	
Example	Send: volume get\r\n return	

	volume get 10\r\n send: volume up return: volume up\r\n send: volume set 10 return volume set 10\r\n

version

version	version
Sending format	version\r\n **Note:** The control system can query the video conferencing system's version information. The video conferencing system can also provide version feedback to the control system when a connection between them is just established. If it is the first time LAN mode and serial port mode are configured, both of them are able to receive the version feedback. **Format:** ● The video conferencing system provides active feedback to the control system: version: "model:Yealink VC400" "firmware:30.20.254.12" "hardware:10.8.10.1.0.0.2" "productId:201601081434" "1.00"\r\n ● The control system receives the command: version: "model:Yealink VC400" "firmware:30.20.254.12" "hardware:10.8.10.1.0.0.2" "productId:201601081434" "1.00"\r\n
Returning format	version: "model: string1" " firmware: string2" "hardware: string3" "productId: string4" "cc_version: string5"\r\n
Example	send: version return version "model: Yealink VC400" "firmware:30.20.254.12" "hardware:10.8.10.1.0.0.2" "productId:201601081434" "cc_version:1.00"\r\n