Software version TC6.2
JULY 2013

Application Programmer Interface (API)
Reference Guide

Cisco TelePresence SX20 Codec

What's in this guide?

The top menu bar and the entries in the table of contents are all hyperlinks, just click on them to go to the topic.

We recommend you visit our web site regularly for updated versions of the user documentation.

Go to: ▸ http://www.cisco.com/go/quickset-docs

Table of Contents

Chapter 1

Introduction

About this guide

This guide introduces you to the Application Programmer Interface (API) for the TC software in general, and serve as a reference guide for the command line commands for the following product:

· Cisco TelePresence SX20 Codec

User documentation

The user documentation for the Cisco TelePresence systems, running the **TC software**, have several guides suitable to various user groups.

· Video conference room primer
· Video conference room acoustics guidelines
· Installation guides for the TelePresence systems
· Software release notes for the TC software
· Getting started guide for the TelePresence systems
· User guide for the TelePresence systems (Touch controller)
· User guide for the TelePresence systems (Remote Control)
· Quick reference guides for the TelePresence systems
· Administrator guides for the TelePresence systems
· Camera user guide for the PrecisionHD cameras
· API reference guide for the SX20 Codec
· Regulatory compliance and safety information guides
· Legal and license information for the TC software

To download the user documentation go to:
▸ http://www.cisco.com/go/quickset-docs

What's new in this version

This section provides an overview of the new and changed advanced settings and new features in the TC6.2 software version.

Software release notes

For a complete overview of new features and changes, we recommend reading the Software Release Notes (TC6). Go to:

► http://www.cisco.com/en/US/products/ps11424/prod_release_notes_list.html

Software download

For software download go to:
► http://www.cisco.com/cisco/software/navigator.html

New features and improvements

Cisco Unified Communication Manager (CUCM) redundancy support

The TelePresence endpoints have CUCM redundancy support with failover and fallback. The TelePresence endpoint will automatically register to the next CUCM in the cluster if the connection to the active CUCM is lost.

If the TelePresence endpoint loses its connection to the primary CUCM, the call is preserved, but no additional features like content sharing, putting participants on hold etc. will be available. The next call will be handled by the secondary CUCM and all call features are available again. When the primary CUCM is back online, the endpoint will automatically register with the primary CUCM.

More details can be found in the *Administering TC endpoints on CUCM* guide for TC6.2.

Improved provisioning from CUCM

Several TelePresence endpoint parameters are added to CUCM and can be provisioned to the endpoint.

More details can be found in the *Administering TC endpoints on CUCM* guide for TC6.2.

Support for CUCM provisioning of the endpoint password when using the encrypted security profile

TelePresence endpoints can be fully and securely provisioned from CUCM. When the TelePresence endpoint is set up with an encrypted security profile in CUCM, the endpoint will read the *admin* password from the CUCM. The password can not be empty, and the user name must be *admin*.

More details can be found in the *Administering TC endpoints on CUCM* guide for TC6.2.

Support for auto discovery of the CUCM

The TelePresence endpoints have support for auto discovery of the CUCM through DHCP Option 150. If the network provides the CUCM address through DHCP Option 150, the endpoint will automatically register to the CUCM after a factory reset.

More details can be found in the *Administering TC endpoints on CUCM* guide for TC6.2.

Support for IPv4, IPv6 and dual stack

The video system is able to operate on IPv4, IPv6 or both at the same time (dual stack). This is configured using the *xConfiguration Network IPStack* setting. Which IP version to use for calls is configured using the *xConfiguration Conference CallProtocolIPStack* setting.

G.729 audio codec support in H.323 calls

Support for the G.729AB audio codec is added in H.323 calls in order to provide better IP phone interoperability. G.729 has been supported in SIP calls as from software version TC6.1.

Lync 2013 support for point-to-point calls

You can set up video calls between TelePresence systems and Lync 2013 users. HD quality at full frame rate (720p30) is supported in both directions. This feature is disabled by default. It must be enabled using the *xConfiguration Conference LyncCompatibility Mode* setting. It also requires version X8 of the VCS (Video Communication Server).

You can see the Lync 2013 users in the TelePresence system's directory (phone book), provided that TMS (TelePresence Management System) version 13.x or later is used.

Interactive Connectivity Establishment (ICE) support

TelePresence systems can use the ICE protocol (RFC 5245) to find the best media path through the network. ICE separates signalling and media, and is a NAT traversal solution that finds the shortest route for audio and video.

This feature is disabled by default. It must be enabled using the *xConfiguration SIP Profile Ice* settings. Both endpoints in the call must support ICE, and a valid TURN (Traversal Using Relay NAT) server must be defined.

ICE is not supported when registered to the current version of CUCM.

Check the TC6 software release notes for further details.

Active control: Providing end users conference control with TelePresence Server

Active Control gives more control and improved experience for the end user during conferences hosted on a Cisco TelePresence Server. Using your Cisco Touch controller, you can interact with the server to control the local experience of the meeting. You will see the participant list with indicators that show who is the active speaker and who is sharing content, you can change your local video layout, disconnect participants, etc.

This feature is disabled by default. It must be enabled using the *xConfiguration Experimental Conference ActiveControl Mode* setting.

Active control requires a Cisco TelePresence Server 3.1 or later, and the server must be behind a Conductor (version XC2.2 or later), i.e. not registered directly to VCS or CUCM. CUCM must be version 9.1.2 or later, and VCS must be version X7 or later. The feature is only supported for SIP calls. H.323 interworking scenarios are not supported.

Check the TC6 software release notes for further details.

Extended Binary Floor Control Protocol (BFCP) port range

When using SIP as a call protocol, UDP ports 5070 to 5077 are used for the BFCP connection (presentation channel). If a firewall is blocking ports in this range, the presentation channel will not be established and the presentation will be sent in the main video channel.

In earlier software releases, only port 5070 was used. The range is extended to support ICE.

Overriding manual positioning of the PrecisionHD 1080p12x camera

If the camera position is adjusted by hand, the camera will automatically return to the position it had before. This feature is disabled by default. It must be enabled using the *xConfiguration Camera Camera[n] MotorMoveDetection* setting.

Web interface enhancements

- A *System recovery* page is introduced. The page provides access to the following recovery alternatives: software image swapping and factory reset. Easy access to configuration and log file back up is also provided.
- A VU meter (volume meter) is available on the *Peripherals* page. The VU meter shows the signal level for the microphone(s).

xConfiguration changes

New configurations

xConfiguration Cameras Camera [1..1] MotorMoveDetection: <Off/On>

xConfiguration Conference CallProtocolIPStack: <IPv4/IPv6>

xConfiguration Conference LyncCompatibility: <Off/On>

xConfiguration Network IPv4 Assignment: <Static/DHCP>
    Replacing xConfiguration Network Assignment

xConfiguration SIP Profile Ice Mode: <Off/On>

xConfiguration SIP Profile Ice DefaultCandidate: <Host/Rflx/Relay>

xConfiguration SIP Profile Turn Server: <S: 0, 255>

xConfiguration SIP Profile Turn UserName: <S: 0, 128>

xConfiguration SIP Profile Turn Password: <S: 0, 128>

xConfiguration Video Input Source 3 Name: <S: 0, 50>

xConfiguration Video Input Source 3 Type: <other/camera/PC/DVD/document_camera>

xConfiguration Video Input Source 3 Connector: <USB>

xConfiguration Video Input Source 3 PresentationSelection: <Manual/Automatic/Hidden>

xConfiguration Video Input Source 3 CameraControl Mode: <Off/On>

xConfiguration Video Input Source 3 CameraControl CameraId: <1>

xConfiguration Video Input Source 3 OptimalDefinition Profile: <Normal/Medium/High>

xConfiguration Video Input Source 3 OptimalDefinition Threshold60fps: <512_288/768_448/1024_576/1280_720/1920_1080/Never>

xConfiguration Video Input Source 3 Quality: <Motion/Sharpness>

xConfiguration Video OSD WallPaperSelection: <Off/On>

xConfiguration Video OSD LanguageSelection: <Off/On>

Configurations that are removed

xConfiguration Network Assignment

    Replaced by xConfiguration Network IPv4 Assignment

Configurations that are modified

xConfiguration Network IPStack

    OLD: <IPv4/IPv6>

    NEW: <Dual/IPv4/IPv6>

xConfiguration Video MainVideoSource

    OLD: <1/2>

    NEW: <1/2/3>

xCommand changes

Commands that are modified

xCommand Message FarendMessage

    OLD: CallId was required

    NEW: CallId is optional

xCommand Phonebook Search

    ADDED: ContactType: <Any/Folder/Contact>

xStatus changes

New status

xStatus Conference Site Hold

xStatus Conference Site Preserved

xStatus ICE Configured

xStatus ICE Defaultcandidate

xStatus ICE Turn IP

xStatus ICE Turn Hostname

xStatus ICE Turn Username

xStatus ICE Call Result

xStatus ICE Call Local Candidate

xStatus ICE Call Local IP

xStatus ICE Call Remote Candidate

xStatus ICE Call Remote IP

xStatus SIP Profile Turn ServerAddress

xStatus SIP Profile 1 DirectoryURI Primary URI

xStatus SIP Profile 1 DirectoryURI Alias URI

xStatus Video Layout Mode

xStatus Video Input Source [1..n] MediaChannelId

xStatus Video Input Source 3 Resolution Height

xStatus Video Input Source 3 Resolution Width

xStatus Video Input Source 3 Resolution RefreshRate

xStatus Video Input Source 3 Resolution FormatType

xStatus Video Input Source 3 Resolution FormatStatus

xStatus Video Input USB 3 Connected

xStatus Video Input USB 3 SignalState

Status that is modified

xStatus Call Status

    OLD: <Idle/Dialling/Ringing/Connecting/Connected/ Disconnecting/OnHold>

    NEW: <Idle/Dialling/Ringing/Connecting/Connected/ Disconnecting/OnHold/EarlyMedia/Preserved/ RemotePreserved>

xStatus H320 Gateway Status

    OLD: <OK/OKWithWarning/Error/Inactive>

    NEW: <OK/OKWithWarning/Error/Inactive/Warning>

xStatus SystemUnit Diagnostics Message Type

    OLD: <ValidAdminPassword/CamerasDetected/ H320GatewayStatus/ISDNLinkCompatibility/ SIPProfileRegistration/SIPListenPortAndOutboundMode/ TLSVerifyRequiredCerts/DefaultCallProtocolRegistered/ NetSpeedAutoNegotiated/HasValidReleaseKey/ EthernetDuplexMatches/IPv4Assignment/ IPv6Assignment>

    NEW: <ValidAdminPassword/CamerasDetected/ H320GatewayStatus/ISDNLinkCompatibility/ SIPProfileRegistration/SIPListenPortAndOutboundMode/ TLSVerifyRequiredCerts/DefaultCallProtocolRegistered/ NetSpeedAutoNegotiated/HasValidReleaseKey/ EthernetDuplexMatches/IPv4Assignment/ IPv6Assignment/SIPProfileType/ CallProtocolIPStackPlatformCompatability/ CameraStatus/CameraPairing/CameraSoftwareVersion/ SelectedVideoInputSourceConnected/OSDVideoOutput/ VideoFromInternalCamera/H323GatekeeperStatus>

xStatus MediaChannels Call IncomingVideoChannel Video Protocol

    OLD: <Off/Raw/H261/H263/H263p/H263pp/H264/MP4V/ JPEG>

    NEW: <Off/Raw/H261/H263/H263p/H263pp/H264/MP4V/ JPEG/HEVC>

xStatus MediaChannels Call OutgoingVideoChannel Video Protocol

    OLD: <Off/Raw/H261/H263/H263p/H263pp/H264/MP4V/ JPEG>

    NEW: <Off/Raw/H261/H263/H263p/H263pp/H264/MP4V/ JPEG/HEVC>

Chapter 2

About the API

API fundamentals

This chapter contains a top-level view of the mechanisms supported by the codec API. You can use the API to manage all parts of the TelePresence system.

Here you can read about how to access the API, how to use the command line and what the different parts of the API can be used for. Information on how to use the feedback functions that are available for the codec is included in this chapter.

The API consists of four major groups:

· Commands
· Configurations
· Status
· Events

These four groups are hierarchically organized, which makes it easier to find related functionality. You can find the complete lists of all commands, configurations and statuses in the following chapters.

Connecting to the API

There are several ways to access the codec API. Regardless of which method you choose, the structure of the API stays the same. Choose the connection method that suites your application best. Before you start, please read this section about the different methods, as some of those may require additional configuration changes before being enabled.

The following commands can be set from the System configuration menu in the web interface or on the on-screen-diplay, or from the command line interface. All of the examples are for the command line interface.

Password

Initially, no password is set for the default admin user. We strongly recommend that you set a password for this user, and any other users possessing an ADMIN role, to restrict access to system configuration. The password can be changed by issuing the following command:

```
xCommand SystemUnit AdminPassword Set Password:
<password>
```

RS-232/serial connection

This is one of the most used connections to our products for integrators, as it can be used without an IP-address, DNS or a network.

Codec configurations

There are two different configurations that can be changed for serial connections; the baud-rate and whether login is required or not. A reboot of the codec is required for these settings to take effect.

```
xConfiguration SerialPort BaudRate:
<9600/19200/38400/57600/115200>

xConfiguration SerialPort LoginRequired: <Off/On>
```

Connecting

For the SX20 codec you need a USB to RS-232 adapter to connect to the USB port.

Note:

· For security reasons the password prompting is turned on by default, but it can be turned off if preferred.
· The default baud rate is set to 38400. The reason for this is that the codec can give very much feedback. When using 9600 baud it may cause your integration to appear sluggish. We therefore recommend keeping the connection at this speed or higher.
· During the initial boot sequence, the codec uses a baud rate of 38400 regardless of the application baud rate you have set.

Telnet

Telnet can be viewed as the equivalent of the serial protocol in the TCP/IP world. Telnet is disabled by default. Before connecting to the codec with Telnet you have to enable it.

Codec configuration

To enable Telnet service, configure the following setting on the codec. Changing this setting does not require a reboot of the device, but it may take some time to take effect.

```
xConfiguration NetworkServices Telnet Mode: On
```

SSH

SSH is a secure TCP/IP connection and it is enabled by default on the codec. It can be disabled. You need a SSH client, such as PuTTY, to access the API over SSH.

HTTP/HTTPS

As HTTP/HTTPS are connectionless protocols, there is no persistent connection. There are several ways of communicating with the API over HTTP.

Codec configuration

In order to enable or disable the HTTP and HTTPS services, configure the following settings on the codec. Changing these settings requires a reboot of the device.

```
xConfiguration NetworkServices HTTP Mode: <Off/On>
xConfiguration NetworkServices HTTPS Mode:
<Off/On>
```

Connecting

You can inspect the API by entering the IP-address or host name of the codec in your favorite web browser. In the web interface, you can find the API documents under the menu section *System Configuration > API > XML API.* The HTTP POST and GET methods are used to execute commands and get feedback from the codec. This is described in ▶"Using HTTP" on page 20 in this document.

API output

The xPreferences is used to set preferences for the RS-232, Telnet and SSH sessions.

The output modes

- Terminal: Line based output for use with line based control systems
- XML: XML output for use with control systems that understand XML.

The default output mode is terminal. To change this you have to define your preferences for each session individually. Examples in this guide are in terminal mode.

To set ouput mode to XML, issue the command:

```
xPreferences outputmode xml
```

To revert to terminal mode, issue the command:

```
xPreferences outputmode terminal
```

Example: Layout command in terminal mode

```
xCommand Video Layout AssignCall CallId: 2 LayoutId: 1
```

Example: Layout command in XML mode

```
<Command>
<Video>
<Layout>
<AssignCall command="True">
<CallId>2</CallId>
<LayoutId>1</LayoutId>
</AssignCall>
</Layout>
</Video>
</Command>
```

Using the command line

Help

To get a list of all supported root commands you can type ? or help after connecting to the TelePresence System using RS-232, Telnet or SSH.

Bye

Typing the bye command closes the command line interface.

```
?
- User Commands -

help            xconfiguration     xfeedback          xgetxml          xhistory
xstatus         xevent             xpreferences       xcommand         systemtools
bye             log                echo               whoami


OK
```

API commands

xConfiguration

Configurations are system settings, such as system name and network settings. These are persistent across boots. Refer to ▶"Configurations" on page 14.

xCommand

Commands instruct the codec to execute actions, such as to dial a number or to search the phone book. Refer to ▶"Commands" on page 13.

xStatus

A status contains the current state of the codec, such as connected calls, the status of the gatekeeper registration, connected inputs and output sources. Refer to ▶"Status" on page 15.

xFeedback

The Feedback commands are used to specify what parts of the configuration and status hierarchies to monitor. Feedback is only issued on the Telnet or SSH session for which it is specified. If you are connecting to the codec with multiple sessions, you have to define feedback individually for each session. Refer to ▶"Feedback mechanism" on page 18.

xPreferences

The xPreferences command is used to set preferences for the RS-232, Telnet and SSH sessions. Refer to ▶"API output" on page 10.

Echo <on/off>

If echo is set to On the key inputs are displayed when entering text in a command line interface.

If echo is set to Off no user input is displayed when entering text in a command line interface.

xEvent

the xEvent command returns information on which events are available for feedback. Refer to ▶"Events" on page 16.

xHistory

The xHistory command returns the call log.

xGetxml

The xGetxml request returns an XML document based on the location parameter attached to the request. The elements (or a complete document) matching the expression will be returned. Refer to ▶"Feedback mechanism" on page 18.

Other commands

Systemtools

The systemtools commands are used for administrative control of the codec and are only available from the command line interface. Systemtools are not a part of the programming API. Refer to ▶"The SystemTools commands" on page 182.

Log

The log command is used to enable advanced logs. It is only used for debugging the system.

Command line shortcuts

If your client supports it, there are some timesaving shortcuts you can use:

- Tab-completion to complete the commands and arguments.
- Arrow up and arrow down keys to navigate your command history.
- <CTRL-a>: Jump to the beginning of the line.
- <CTRL-e>: Jump to the end of the line.
- <CTRL-r>: Incremental command history search.
- <CTRL-w>: Erase the current line.

Searching

You can use // to search for elements anywhere in the status or configuration hierarchy (Example 1).

You can also combine multiple //'s (Example 2).

WARNING: The search shortcuts work well for inspecting the API, but should not be used in applications. We recommend that you always use the complete paths to avoid command ambiguity when upgrading to newer firmware releases.

Value types and formats

The system supports the following value types (Example 3):

- Integer values: <x..y>
  Defines the valid range for an integer input. x = min value, y = max value.

- Literal values: <X/Y/../Z>
  Defines the possible values for a given configuration.

- String values: <S: x, y>
  Defines that the valid input for this configuration is a string with minimum length of x and maximum length of y characters. Strings can have rules that further specify their format and length.

Input values that contain spaces need to be quoted

Any values for configurations and commands containing spaces must be enclosed in quotes. Quotes are not necessary for values without spaces.

Examples:

Correct: xCommand dial number: "my number contains spaces"

Correct: xCommand dial number: 12345

Incorrect: xCommand dial number: my number contains spaces

Case sensitivity

All commands are case-insensitive. All of the following commands will work.

```
XCOMMAND DIAL NUMBER: foo@bar.org

xcommand dial number: foo@bar.org

xCommand Dial Number: foo@bar.org
```

Example 1: List all configurations that include a word that starts with DVI:

```
xConfiguration //dvi
*c xConfiguration Video Input DVI 2 RGBQuantizationRange: Full
*c xConfiguration Video Input DVI 2 Type: AutoDetect
** end
```

Example 2: Get the resolution width of all connected sources for both inputs and outputs:

```
xStatus //vid//res//wid
*s Video Input Source 1 Resolution Width: 0
*s Video Input Source 2 Resolution Width: 0
*s Video Output HDMI 1 Resolution Width: 1280
*s Video Output HDMI 2 Resolution Width: 1680
** end
```

Example 3: List the value types and formats

```
xConfiguration ??
*? xConfiguration Audio Volume: <0..100>
*? xConfiguration Audio SoundsAndAlerts RingVolume: <0..100>
*? xConfiguration Audio SoundsAndAlerts RingTone: <Marbles/IceCrystals/Polaris/Alert/
Discreet/Fantasy/Jazz/Nordic/Echo/Rhythmic>.
.
.
.
*? xConfiguration Video OSD InputMethod Cyrillic: <Off/On>
*? xConfiguration Video OSD LoginRequired: <Off/On>

OK
```

Commands

Commands instruct the codec to execute actions, such as to dial a number or to search the phone book. All commands start with the prefix xCommand followed by a command path. Writing xCommand ? on the command line will list all the top level commands.

To view the complete list of commands and their parameters, write xCommand ?? on the command line.

Command arguments are key-value pairs. The (r) behind the argument name indicates that the argument is required for the command.

When issuing a xCommand, the command consists of one argument and one required parameter:

Example: xCommand Dial Number: 123

- xCommand is the command prefix. The command to be executed is Dial.
- The example contains one argument, Number: 123. Number is the key and 123 is its value. The key/value pair is separated with ':'.

```
xCommand ?


- User Commands -

Audio                    Camera                   FarEndControl            Preset

Bookings                 Conference               HttpFeedback             Provisioning

Boot                     Dial                     Key                      Security

Call                     DTMFSend                 Message                  Standby

CallLog                  Experimental             Phonebook                SystemUnit

CamCtrlPip               FacilityService          Presentation             Video


OK
```

```
xCommand ??


xCommand Call DisconnectAll
xCommand Call Accept
     CallId: <0..65534>
     CallType: <Audio/Video>
.
.
.
xCommand Phonebook Folder Delete
     FolderId(r): <S: 0, 255>
xCommand Phonebook Search
     PhonebookId: <S: 0, 255>
     PhonebookType: <Corporate/Local>
     SearchString: <S: 0, 255>
     SearchField: <Name/Number>
     Offset: <0..65534>
     FolderId: <S: 0, 255>
     Limit: <0..65534>
     Recursive: <False/True>


OK
```

Configurations

Configurations are system settings that are persistent across boots. Like commands, also configurations are structured in a hierarchy.

Writing xConfiguration ? on the command line lists all the top level configurations.

Writing xConfiguration ?? lists all configurations and their value spaces.

Writing xConfiguration lists all configurations and their current values. To list out only parts of the configuration, write xConfiguration followed by one or more parts of the configuration paths.

Example: Set the H323 Alias ID

Write in:

```
xConfiguration H323 Profile 1 H323Alias ID:
"changed@company.com"
```

Example: Get the H323 Alias ID

Write in:

```
xConfiguration H323 Profile 1 H323Alias ID
```

Result:

```
*c xConfiguration H323 Profile 1 H323Alias ID:
"changed@company.com"
**end
```

```
xConfiguration ?


- User Configurations -

Audio                    H323                   RTP                 SystemUnit

Cameras                  Network                Security            Time

Conference               NetworkServices        SerialPort          UserInterface

Experimental             Phonebook              SIP                 Video

FacilityService          Provisioning           Standby


OK
```

```
xConfiguration ??


*? xConfiguration Audio Volume: <0..100>

*? xConfiguration Audio SoundsAndAlerts RingVolume: <0..100>

*? xConfiguration Audio SoundsAndAlerts RingTone: <Marbles/IceCrystals/Polaris/Alert/Discreet/Fantasy/Jazz/
 Nordic/Echo/Rhythmic>

*? xConfiguration Audio SoundsAndAlerts KeyTones Mode: <Off/On>

*? xConfiguration Audio Microphones Mute Enabled: <True/InCallOnly>

*? xConfiguration Audio Input HDMI [1..1] Mode: <Off/On>

.

.

.

*? xConfiguration Video OSD AutoSelectPresentationSource: <Off/On>

*? xConfiguration Video OSD TodaysBookings: <Off/On>

*? xConfiguration Video OSD MyContactsExpanded: <Off/On>

*? xConfiguration Video OSD Output: <1/2>

*? xConfiguration Video OSD InputMethod InputLanguage: <Latin/Cyrillic>

*? xConfiguration Video OSD InputMethod Cyrillic: <Off/On>

*? xConfiguration Video OSD LoginRequired: <Off/On>


OK
```

Status

A status contains the current state of the codec, such as connected calls, the status of the gatekeeper registration, connected inputs and output sources.

Writing xStatus ? on the command line lists all top level statuses.

Writing xStatus lists all statuses and their current values.

To list out only some of the statuses, write xstatus followed by the relevant part of the status path.

```
xStatus ?
– Status –
Audio                    H323                    Security
Call[...]                HttpFeedback[1..4]      SIP
Camera                   MediaChannels           Standby
Conference               Network                 SystemUnit
Diagnostics              NetworkServices         Time
Experimental             Preset[1..15]           Video
H320                     Provisioning
OK
```

Addressing status information with xStatus

To read the status from the system type the root command (xStatus) followed by a path (address expression):

```
xStatus <address expression>
```

Example 1: Query all ongoing Call information:

```
xStatus Call
*s Call 3 Status: Connected
*s Call 3 Direction: Outgoing
*s Call 3 Protocol: "sip"
*s Call 3 RemoteNumber: "firstname.lastname@
company.com"
*s Call 3 CallbackNumber: "sip:firstname.
lastname@company.com"
*s Call 3 DisplayName: "firstname.lastname@
company.com"
*s Call 3 TransmitCallRate: 768
*s Call 3 ReceiveCallRate: 768
*s Call 3 FacilityServiceId: 0
*s Call 3 Encryption Type: "None"
*s Call 3 PlacedOnHold: False
*s Call 3 Duration: 9
** end
 OK
```

Example 2: Query the protocol for a call:

```
xStatus Call Protocol
*s Call 3 Protocol: "sip"
OK
```

Events

Event returns information about the events that are available for feedback. This overview presents examples of some the events that are available on the API.

To get an overview of the supported events type ?, ?? or help after xEvent:

- xEvent ? Lists the top level events
- xEvent ?? List all of the available events
- xEvent help Lists the top level events

The result for events depends on the state of the codec.

Example 1: Outgoing Call Indication

Outgoing Call Indication is an event reported when an outgoing call is about to be dialled. Returns the CallId the call has been assigned.

```
*e OutgoingCallIndication CallId: x

** end
```

Example 2: Call Disconnect

Call Disconnect is an event reported when a call is disconnected. Returns the CallId of the disconnected call and reason for the call's disconnection.

```
*e CallDisconnect CallId: x CauseValue: 0
CauseString: "" CauseType: LocalDisconnect
OrigCallDirection: "outgoing"

** end
```

Example 3: Call Successful

Call Succesful is an event reported when a call is connected successfully, that is when all channels are up and established.

```
*e CallSuccessful CallId: 132 Protocol: "h223"
Direction: "outgoing" CallRate: 768 RemoteURI:
"h223:integratorHQ@company.com" EncryptionIn:
"Off" EncryptionOut: "Off"

** end
```

Example 4: FECC Action request

FECC Action request is an event reported when far end is sending FECC commands.

```
*e FeccActionInd Id: 132 Req: 1 Pan: 1 PanRight:
1 Tilt: 0 TiltUp: 0 Zoom: 0 ZoomIn: 0 Focus: 0
FocusIn: 0 Timeout: 300 VideoSrc: 0 m: 0

** end
```

Call history

The xHistory command returns the call logs.

Example with xHistory CallLogs

```
xhistory
*h xHistory CallLogs Call 1 CallId: 13
*h xHistory CallLogs Call 1 Protocol: "h323"
*h xHistory CallLogs Call 1 Direction: Outgoing
*h xHistory CallLogs Call 1 CallType: Video
*h xHistory CallLogs Call 1 RemoteNumber: "h323:firstname.lastname.office@company.com"
*h xHistory CallLogs Call 1 CallbackNumber: "h323:firstname.lastname.office@company.com"
*h xHistory CallLogs Call 1 DisplayName: "firstname.lastname@company.com"
*h xHistory CallLogs Call 1 CallRate: 768
*h xHistory CallLogs Call 1 DisconnectCause: ""
*h xHistory CallLogs Call 1 DisconnectCauseType: LocalDisconnect
*h xHistory CallLogs Call 1 DisconnectCauseCode: 16
*h xHistory CallLogs Call 1 DisconnectCauseOrigin: Q850
*h xHistory CallLogs Call 1 StartTime: "2012/03/01 11:04:14"
*h xHistory CallLogs Call 1 Duration: 184
*h xHistory CallLogs Call 1 Encryption: ""Aes-128"
*h xHistory CallLogs Call 1 BookingId: ""
...
*h xHistory CallLogs Recent 6 CounterMissed: 2
*h xHistory CallLogs Recent 6 Counter: 2
...
*h xHistory CallLogs Outgoing 29 Counter: 1
...
*h xHistory CallLogs Received 40 Counter: 6
...
*h xHistory CallLogs Missed 50 Counter: 2
*h xHistory CallLogs Missed 50 NewCounter: 0
** end
```

Feedback mechanism

To build solutions that can reliably keep the state between the application and the codec synchronized, you need to set up a notification system to report the changes in the state of the codec.

The API supports notifications on the following:

· Configuration changes

· Status changes

· Event notifications

These notifications will not be sent unless the user has explicitly told the codec to do so. The user is required to subscribe to all the feedback the application needs. This is done by registering feedback expressions. The way of registering feedback expressions varies according to the connection method used.

When using HTTP, the method of handling feedback differs slightly from what is presented in this section. Refer to ▶"Feedback from codec over HTTP" on page 21.

> WARNING: A codec may give very much feedback, especially when calls are connected and disconnected. Therefore, you should only subscribe to the feedback that you need.
>
> Never register for all status feedback by issuing xFeedback register /Status. This may give the control application too much data to handle, which may lead to sluggish or unpredictable behavior.

Feedback expressions

The expression used when registering for feedback is a variant of the XPath language. This language describes a way to select nodes from an XML document. TC software contains three main feedback documents:

Document	API command	Path
Status	xStatus	/Status
Configuration	xConfiguration	/Configuration
Event	xEvent	/Event

The syntax for feedback registering is: xFeedback register <path>

Never register for all status feedback by issuing xFeedback register /Status.

It is safe to register for all configuration changes using xFeedback register /Configuration, as configuration changes will most likely not occur that often.

By going through some examples, we can see how this information can be used to build feedback expressions. A good way to verify the expressions is to point your browser to http://<ip-address>/getxml?location=path or to execute xgetxml <path> from the terminal, and check that the output matches the nodes you want feedback on.

Example 1: Microphones Mute status.

Terminal query

```
xStatus Audio Microphones Mute
*s Audio Microphones Mute: Off
** end
```

Equivalent feedback expression

```
xFeedback register /Status/Audio/Microphones/Mute
```

Example 2: All video input connectors.

Terminal query

```
xConfiguration Video Input Source Connector
*c xConfiguration Video Input Source 1 Connector: HDMI
*c xConfiguration Video Input Source 2 Connector: DVI
** end
```

Equivalent feedback expression

```
xFeedback register /Configuration/Video/Input/Source/Connector
```

Example 3: Video input connector for source 2.

Terminal query

```
xConfiguration Video Input Source 2 Connector
*c xConfiguration Video Input Source 2 Connector: DVI
** end
```

Equivalent feedback expression

```
xFeedback register /Configuration/Video/Input/Source[@item='2']/Connector
```

Terminal connections

Managing feedback subscriptions

To register, list and deregister feedback expressions you use the command xFeedback and its corresponding sub commands.

The registered expressions are only valid for the currently active connection. If you open two Telnet sessions and register to get feedback in one session, you do not receive feedback in the other serssion. This also means that if you disconnect from a session, you have to re-register all expressions after reconnecting.

You can register up to 38 expressions.

Feedback output

The feedback output is exactly the same as you get when querying the system using the xConfiguration and xStatus commands. E.g., if you issue the command `xStatus Standby Active` on the command line the result is:

```
*s Standby Active: On
** end
```

If you have registered for feedback on status changes the feedback you get when the system goes to standby-mode will be exactly the same:

```
*s Standby Active: On
** end
```

This means that when you are programming against the device you only need to handle one format.

Example: Managing feedback subscriptions

A: Register feedback expressions.

Write in: `xFeedback register /Status/Audio`
Result: `** end`
      `OK`

Write in: `xFeedback register /Event/CallDisconnect`
Result: `** end`
      `OK`

Write in: `xFeedback register /Configuration/Video/MainVideoSource`
Result: `** end`
      `OK`

B: List out currently registered expressions.

Write in: `xFeedback list`
Result: `/Configuration/Video/MainVideoSource`
      `/Event/CallDisconnect`
      `/Status/Audio`
      `** end`
      `OK`

C: Deregister feedback expressions.

Write in: `xFeedback deregister /Event/CallDisconnect`
Result: `** end`
      `OK`

Write in: `xFeedback deregister /Status/Audio`
Result: `** end`
      `OK`

D: List the new feedback expressions.

Write in: `xFeedback list`
Result: `/Configuration/Video/MainVideoSource`
      `** end`
      `OK`

Using HTTP

The codec supports sending commands and configurations over HTTP and HTTPS. It is also possible to retrieve configurations and statuses this way. This interface exposes the same API as the command line, but in XML format.

URL cheat sheet

The following table contains the main URLs used when accessing the API over HTTP.

Method	URL	Description
GET	http://<ip-address>/status.xml	Complete status document
GET	http://<ip-address>/configuration.xml	Complete configuration document
GET	http://<ip-address>/command.xml	Complete command document
GET	http://<ip-address>/valuespace.xml	Complete valuespace document
GET	http://<ip-address>/getxml?location=<path>	Retrieve document based on a path
POST	http://<ip-address>/putxml	Configurations and commands in HTTP body
GET, POST	http://<ip-address>/formputxml?xmldoc=<xml>	Configurations and commands www-urlencoded

Getting status and configurations

Example 1: Get all status entries on the codec.

```
http://<ip-address>/getxml?location=/Status
```

Example 2: Get just the audio statuses of the codec.

```
http://<ip-address>/getxml?location=/Status/Audio
```

Example 3: Get all configurations of the codec.

```
http://<ip-address>/getxml?location=/Configuration
```

Example 4: Get all video configurations of the codec.

```
http://<ip-address>/getxml?location=/
Configuration/Video
```

Sending commands and configurations

Using HTTP GET

It is possible to use HTTP GET when sending commands or configurations to the codec. This makes it easy to test commands using your browser.

Example 1: Setting the camera position.

```
http://<ip-address>/formputxml?xmldoc=
<Command><Camera><PositionSet command="True">
<CameraId>1</CameraId><Pan>200</Pan>
<Tilt>200</Tilt></PositionSet></Camera></Command>
```

Example 2: Changing the system name.

```
http://<ip-address>/formputxml?xmldoc=
<Configuration><SystemUnit>
<Name>newName</Name></SystemUnit></Configuration>
```

Example 3: Changing multiple configurations in one go.

```
http://<ip-address>/formputxml?xmldoc=
<Configuration><Audio><Volume>80</Volume>
</Audio><Video><OSD><TodaysBookings>On
</TodaysBookings></OSD></video><Configuration>
```

Using HTTP POST

When sending configurations and commands to the codec, it is important that the HTTP header Content-Type is set to text/xml, i.e. Content-Type: text/xml. The body of the POST should contain the XML content.

Example 1: Changing the system name.

Request

```
POST /putxml HTTP/1.1
Content-Type: text/xml
Connection: close
<Configuration>
    <SystemUnit>
        <Name>newName</Name>
    </SystemUnit>
</Configuration>
```

Example 2: Setting the camera position.

Request

```
POST /putxml HTTP/1.1
Content-Type: text/xml
Connection: close
<Command>
    <Camera>
        <PositionSet command="True">
            <CameraId>1</CameraId>
            <Pan>200</Pan>
            <Tilt>200</Tilt>
        </PositionSet>
    </Camera>
</Command>
```

Response

```
HTTP/1.1 200 OK
Date: <date>
Server: WSGIServer/0.1 Python/2.5.4
Cache-Control: no-cache
Content-Type: text/xml
Content-Length: 91
Connection: close
<?xml version="1.0"?>
<Command>
<CameraPositionSetResult item="1" status="OK"/>
</Command>
```

Feedback from codec over HTTP

To get notifications from the codec, you need to register HTTP feedback expressions. The codec will then use HTTP POST to send feedback messages to the supplied IP-address. This means that you have to have a HTTP server running for your application to receive updates from the codec.

Registering for feedback

The command for registering is xCommand HttpFeedback Register. The syntax for this command and its arguments are described in this section.

HttpFeedback Register syntax:

```
xCommand HttpFeedback Register
    FeedbackSlot: <1..4>
    ServerUrl(r): <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
    Expression: <S: 1, 255>
```

HttpFeedback Register arguments:

FeedbackSlot: The codec can register up to 4 slots of servers requesting HTTP feedback. Set the registering to one of them.

Note: Avoid using FeedbackSlot 3 in an environment where Cisco TelePresence Management Suite (TMS) is used as TMS uses this feedbackslot to register its expressions.

ServerUrl: The URL that you want the codec to post the HTTP feedback messages to.

Expression 1-15: Register the expressions you want to receive feedback on. Refer to ▶"Feedback mechanism" on page 18.

Example: Registering feedback on configuration changes, disconnect events and call status changes.

```
<Command>
  <HttpFeedback>
    <Register command="True">
      <FeedbackSlot>1</FeedbackSlot>
      <ServerUrl>http://127.0.0.1/
        myhttppostscripturl</ServerUrl>
      <Expression item="1">/Configuration</
        Expression>
      <Expression item="2">/Event/
        CallDisconnect</Expression>
      <Expression item="3">/Status/Call</
        Expression>
    </Register>
  </HttpFeedback>
</Command>
```

Feedback output

When the codec notifies the registered HTTP server about changes, the body contains the same XML as when polling. There is however one small difference. The root-node contains an Identification node with children that specify the codec from which the notification originated. This means that you can handle multiple codecs with a single HTTP server URI.

Example: Audio volume changed.

```
<Configuration xmlns="http://www.company.com/XML/
CUIL/2.0">
  <Identification>
    <SystemName>integrator</SystemName>
    <MACAddress>00:00:de:ad:be:ef</MACAddress>
    <IPAddress>192.168.1.100</IPAddress>
    <ProductType>Cisco Codec</ProductType>
    <ProductID>Cisco Codec C90</ProductID>
    <SWVersion>TC6.0.0.199465</SWVersion>
    <HWBoard>101401-5 [08]</HWBoard>
    <SerialNumber>PH0000000</SerialNumber>
  </Identification>
  <Audio item="1">
    <Volume item="1">60</Volume>
  </Audio>
</Configuration>
```

Translating from terminal mode to XML

Translating commands

The XML commands maintain the same structure as the terminal commands, but they use a parent-child relationship to describe the hierarchy. You can see this structure in the examples below.

Example 1: Setting up a call.

Terminal

```
xCommand Dial Number: "12345" Protocol: H323
```

XML

```
<Command>
  <Dial command="True">
    <Number>12345</Number>
    <Protocol>H323</Protocol>
  </Dial>
</Command>
```

Example 2: Assigning video layout to a call.

Terminal

```
xCommand Video Layout AssignCall CallId: 2
LayoutId: 1
```

XML

```
<Command>
 <Video>
   <Layout>
   <AssignCall command="True">
   <CallId>2</CallId>
   <LayoutId>1</LayoutId>
   </AssignCall>
  </Layout>
 </Video>
</Command>
```

Translating configurations

Translating from xConfiguration to XML is similar to commands, but with the addition of a special attribute item="NN" for specifying the index in arrays.

Example: Configuring connector for input source 2.

Terminal

```
xConfiguration Video Input Source 2
Connector: HDMI
```

XML

```
<Configuration>
  <Video>
    <Input>
      <Source item="2">
        <Connector>HDMI</Connector>
      </Source>
    </Input>
  </Video>
</Configuration>
```

Dos and don'ts

Here are some issues you should consider when programming the Cisco C- and SX-series API.

AVOID remote control emulation

The use of xCommand Key Press and xCommand Key Click commands is highly discouraged. The commands are still available in the API, but we recommend the use of direct commands, as this ensures backwards compatibility in your integrations. Program against the codec, not the on-screen-display.

DO use complete commands

You should always use complete commands when programming, i.e. always use xConfiguration Video instead of xconf vid. The shortcuts can be used for searches in the API on the command line, but not for programming. The reason for this is that you might end up with ambiguous code when additional commands are added to the API.

DO NOT subscribe to unnecessary feedback

Subscribing to too much feedback may congest the control application. Although the amount of feedback may seem fine in the current version, the amount of feedback may grow in future releases.

DO present one screen to the end user

Avoid making the user look at two menus, one on the control panel and one on the video screen. Do not make the control panel a substitute for the remote control. The on-screen-display in is using the exact same API as you have access to through the command interface.

DO NOT use the experimental section in production

Under the listing of commands, status and configurations, you can find subsections that start with Experimental.

- xCommand Experimental
- xStatus Experimental
- xConfiguration Experimental

These sections give access to features that we are still working on and have yet to release for official use. This part of the API will NOT stay backwards compatible and the structure WILL CHANGE.

Do not use the experimental section in production - in future versions of the firmware these sections may be hidden or removed.

Chapter 3

xConfiguration commands

Description of the xConfiguration commands

In this chapter, you can find a complete list of the xConfiguration commands. The examples show either the default value or an example of a value.

We recommend you visit our web site regularly for updated versions of the manual.
Go to: ▸ http://www.cisco.com/go/quickset-docs

Audio configuration

xConfiguration Audio Input HDMI [1] Mode

If you are connecting a Cisco camera with an integrated microphone to the codec's HDMI 1 input, you can enable/disable audio input on the port; if you are using any other camera, the audio input will be disabled regardless of this setting.

If you are going to use the integrated microphone, this setting must be On (default value).

Whenever you are connecting a Cisco Table Microphone 20 to any of the codec's external microphone inputs, the integrated microphone will be disabled regardless of this setting.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable audio on the HDMI input.

On: Enable audio on the HDMI input.

Example: `xConfiguration Audio Input HDMI 1 Mode: On`

xConfiguration Audio Microphones Mute Enabled

Determine whether audio-mute is allowed or not. The default value is True.

Requires user role: ADMIN

Value space: <True/InCallOnly>

True: Muting of audio is always available.

InCallOnly: Muting of audio is only available when the device is in a call. When Idle it is not possible to mute the microphone. This is useful when an external telephone service/audio system is connected via the codec and is to be available when the codec is not in a call. When set to InCallOnly this will prevent the audio-system from being muted by mistake.

Example: `xConfiguration Audio Microphones Mute Enabled: True`

xConfiguration Audio SoundsAndAlerts KeyTones Mode

The system can be configured to make a keyboard click sound effect (key tone) when pressing a key on the remote control, or when typing text or numbers on a Touch controller.

Requires user role: USER

Value space: <Off/On>

Off: No key tones will be played when you type.

On: You will hear a key tone when you press a key or type text.

Example: `xConfiguration Audio SoundsAndAlerts KeyTones Mode: Off`

xConfiguration Audio SoundsAndAlerts RingTone

Select the ring tone for incoming calls.

Requires user role: USER

Value space: <Marbles/IceCrystals/Polaris/Alert/Discreet/Fantasy/Jazz/Nordic/Echo/Rhythmic>

Range: Select a tone from the list of ring tones.

Example: `xConfiguration Audio SoundsAndAlerts RingTone: Jazz`

xConfiguration Audio SoundsAndAlerts RingVolume

Sets the ring volume for an incoming call.

Requires user role: USER

Value space: <0..100>

Range: The value goes in steps of 5 from 0 to 100 (from -34.5 dB to 15 dB). Volume 0 = Off.

Example: `xConfiguration Audio SoundsAndAlerts RingVolume: 50`

xConfiguration Audio Volume

Adjust the speaker volume.

Requires user role: USER

Value space: <0..100>

Range: The value must be between 0 and 100. The values from 1 to 100 correspond to the range from -34.5 dB to 15 dB (0.5 dB steps). The value 0 means that the audio is switched off.

Example: `xConfiguration Audio Volume: 70`

Cameras configuration

xConfiguration Cameras PowerLine Frequency

Applies to cameras supporting PowerLine frequency anti-flickering, i.e. PrecisionHD 1080p cameras.

Requires user role: ADMIN

Value space: <50Hz/60Hz>

50Hz: Set to 50 Hz.

60Hz: Set to 60 Hz.

Example: `xConfiguration Cameras PowerLine Frequency: 50Hz`

xConfiguration Cameras Camera [1..1] Backlight

This configuration turns backlight compensation on or off. Backlight compensation is useful when there is much light behind the persons in the room. Without compensation the persons will easily appear very dark to the far end.

Requires user role: ADMIN

Value space: <Off/On>

Off: Turn off the camera backlight compensation.

On: Turn on the camera backlight compensation.

Example: `xConfiguration Cameras Camera 1 Backlight: Off`

xConfiguration Cameras Camera [1..1] Brightness Mode

Set the camera brightness mode.

Requires user role: ADMIN

Value space: <Auto/Manual>

Auto: The camera brightness is automatically set by the system.

Manual: Enable manual control of the camera brightness. The brightness level is set using the Cameras Camera Brightness Level setting.

Example: `xConfiguration Cameras Camera 1 Brightness Mode: Auto`

xConfiguration Cameras Camera [1..1] Brightness Level

Set the brightness level. NOTE: Requires the Camera Brightness Mode to be set to Manual.

Requires user role: ADMIN

Value space: <1..31>

Range: Select a value from 1 to 31.

Example: `xConfiguration Cameras Camera 1 Brightness Level: 1`

xConfiguration Cameras Camera [1..1] Flip

With Flip mode (vertical flip) you can flip the image upside down.

Requires user role: ADMIN

Value space: <Auto/Off/On>

Auto: When the camera is placed upside down the image is automatically flipped upside down. This setting will only take effect for a camera that automatically detects which way it is mounted.

Off: Display the video on screen the normal way.

On: When enabled the video on screen is flipped. This setting is used when a camera is mounted upside down, but cannot automatically detect which way it is mounted.

Example: `xConfiguration Cameras Camera 1 Flip: Off`

xConfiguration Cameras Camera [1..1] Focus Mode

Set the camera focus mode.

Requires user role: ADMIN

Value space: <Auto/Manual>

Auto: The camera will auto focus once a call is connected, as well as after moving the camera (pan, tilt, zoom). The system will use auto focus only for a few seconds to set the right focus; then auto focus is turned off to prevent continuous focus adjustments of the camera.

Manual: Turn the autofocus off and adjust the camera focus manually.

Example: `xConfiguration Cameras Camera 1 Focus Mode: Auto`

xConfiguration Cameras Camera [1..1] Gamma Mode

Applies to cameras which support gamma mode. The Gamma Mode setting enables for gamma corrections. Gamma describes the nonlinear relationship between image pixels and monitor brightness. The Cisco TelePresence PrecisionHD 720p camera supports gamma mode. The PrecisionHD 1080p camera does not support gamma mode.

Requires user role: ADMIN

Value space: <Auto/Manual>

Auto: Auto is the default and the recommended setting.

Manual: In severe light conditions, you may switch mode to manual and specify explicitly which gamma table to use by setting the Gamma Level.

Example: `xConfiguration Cameras Camera 1 Gamma Mode: Auto`

xConfiguration Cameras Camera [1..1] Gamma Level

By setting the Gamma Level you can select which gamma correction table to use. This setting may be useful in difficult lighting conditions, where changes to the brightness setting does not provide satisfactory results. NOTE: Requires the Gamma Mode to be set to Manual.

Requires user role: ADMIN

Value space: <0..7>

Range: Select a value from 0 to 7.

Example: `xConfiguration Cameras Camera 1 Gamma Level: 0`

xConfiguration Cameras Camera [1..1] IrSensor

The IR sensor LED is located in the front of the camera and flickers when the IR sensor is activated from the remote control. Both the Codec C Series and PrecisionHD camera have IR sensors, and only one of them needs to be enabled at the time.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable the IR sensor on the camera.

On: Enable the IR sensor on the camera.

Example: `xConfiguration Cameras Camera 1 IrSensor: On`

xConfiguration Cameras Camera [1..1] Mirror

With Mirror mode (horizontal flip) you can mirror the image on screen.

Requires user role: ADMIN

Value space: <Auto/Off/On>

Auto: When the camera is placed upside down the image is automatically mirrored. Use this setting with cameras that can be mounted upside down, and that can auto detect that the camera is mounted upside down.

Off: See the self view in normal mode, that is the experience of self view is as seeing yourself as other people see you.

On: See the self view in mirror mode, that is the self view is reversed and the experience of self view is as seeing yourself in a mirror.

Example: `xConfiguration Cameras Camera 1 Mirror: Off`

xConfiguration Cameras Camera [1..1] MotorMoveDetection

This setting applies only when using a Cisco TelePresence PrecisionHD 1080p12x camera.

If adjusting the camera position by hand you can configure whether the camera should keep its new position or return to the preset or position it had before.

Requires user role: ADMIN

Value space: <Off/On>

Off: When the camera position is adjusted manually the camera will keep this position until adjusted again. WARNING: If moving the camera by hand, the camera will not register the new pan and tilt values since there is no position feedback. This will result in wrong pan and tilt values when recalling the camera presets subsequently.

On: When the camera position is adjusted manually, or the camera detects that the motors have moved, it will first re-initialize (i.e. go to default position) then return to the preset/position it had before the camera was adjusted.

Example: `xConfiguration Cameras Camera 1 MotorMoveDetection: Off`

xConfiguration Cameras Camera [1..1] Whitebalance Mode

Set the camera whitebalance mode.

Requires user role: ADMIN

Value space: <Auto/Manual>

Auto: The camera will continuously adjust the whitebalance depending on the camera view.

Manual: Enables manual control of the camera whitebalance. The whitebalance level is set using the Cameras Camera Whitebalance Level setting.

Example: `xConfiguration Cameras Camera 1 Whitebalance Mode: Auto`

xConfiguration Cameras Camera [1..1] Whitebalance Level

Set the whitebalance level. NOTE: Requires the Camera Whitebalance Mode to be set to manual.

Requires user role: ADMIN

Value space: <1..16>
Range: Select a value from 1 to 16.

Example: `xConfiguration Cameras Camera 1 Whitebalance Level: 1`

xConfiguration Cameras Camera [1..1] DHCP

Applies to cameras which support DHCP (for example the Cisco TelePresence PrecisionHD 1080p 12X camera). The camera must be connected to a LAN. When set, the command enables support for SW upgrade of daisy chained cameras. It will enable the camera's DHCP function and force start of MAC and IP address retrieval. Remember to reset the DHCP when the camera is no longer connected to a LAN.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable DHCP in the camera. NOTE: This setting should be applied when the camera is not connected to a LAN.

On: Enable DHCP in the camera. The camera is automatically re-booted. After re-boot the DHCP is started and the IP address will be retrieved. Run the commnand "xStatus Camera" for result.

Example: `xConfiguration Cameras Camera 1 DHCP: Off`

Conference configuration

xConfiguration Conference [1..1] CallProtocolIPStack

Select if the call protocol (SIP, H323) should use the IPv4 or IPv6 address when the system has IP connectivity on both protocol versions at the same time, e.g. when the network interface (Network IPStack) is set to Dual.

Requires user role: ADMIN

Value space: <IPv4/IPv6>

IPv4: When set to IPv4, the call protocol (SIP, H323) will use IPv4.
IPv6: When set to IPv6, the call protocol (SIP, H323) will use IPv6.

Example: `xConfiguration Conference 1 CallProtocolIPStack: IPv4`

xConfiguration Conference [1..1] LyncCompatibility

The TelePresence endpoints, running TC6.2 software, or later are Lync 2013 ready.

NOTE: Requires version X8 of the VCS (Video Communication Server). You can make calls to Lync 2013 from your TelePresence endpoint and vice versa and the call will have support for audio and video.

NOTE: To make Lync 2013 contacts available in the phonebook on the endpoint the TMS (TelePresence Management System) version 13.x, or later, must be configured.

Requires user role: ADMIN

Value space: <Off/On>

Off: When set to Off, Lync 2013 compatibility on the endpoint is disabled. This is the default value.
On: When set to Off, Lync 2013 compatibility on the endpoint is enabled.

Example: `xConfiguration Conference 1 LyncCompatibility: Off`

xConfiguration Conference [1..1] AutoAnswer Mode

Set the auto answer mode.

Requires user role: ADMIN

Value space: <Off/On>

Off: An incoming call must be answered manually by pressing the OK key or the green Call key on the remote control, or by tapping the Accept key on the Touch controller.

On: Enable auto answer to let the system automatically answer all incoming calls.

Example: `xConfiguration Conference 1 AutoAnswer Mode: Off`

xConfiguration Conference [1..1] AutoAnswer Mute

Determine if the microphone shall be muted when an incoming call is automatically answered.
NOTE: Requires that AutoAnswer Mode is switched on.

Requires user role: ADMIN

Value space: <Off/On>

Off: The incoming call will not be muted.

On: The incoming call will be muted when automatically answered.

Example: `xConfiguration Conference 1 AutoAnswer Mute: Off`

xConfiguration Conference [1..1] AutoAnswer Delay

Define how long (in seconds) an incoming call has to wait before it is answered automatically by the
system. NOTE: Requires that AutoAnswer Mode is switched on.

Requires user role: ADMIN

Value space: <0..50>

Range: Select a value from 0 to 50 seconds.

Example: `xConfiguration Conference 1 AutoAnswer Delay: 0`

xConfiguration Conference [1..1] MicUnmuteOnDisconnect Mode

Determine if the microphones shall be unmuted automatically when all calls are disconnected. In a
meeting room or other shared resources this may be done to prepare the system for the next user.

Requires user role: ADMIN

Value space: <Off/On>

Off: If muted during a call, let the microphones remain muted after the call is disconnected.

On: Unmute the microphones after the call is disconnected.

Example: `xConfiguration Conference 1 MicUnmuteOnDisconnect Mode: On`

xConfiguration Conference [1..1] DoNotDisturb Mode

Determine if there should be an alert on incoming calls.

Requires user role: USER

Value space: <Off/On/Timed>

Off: The incoming calls will come through as normal.

On: All incoming calls will be rejected and they will be registered as missed calls. The calling side
will receive a busy signal. A message telling that Do Not Disturb is switched on will display on the
Touch controller or main display. The calls received while in Do Not Disturb mode will be shown as
missed calls.

Timed: Select this option only if using the API to switch Do Not Disturb mode on and off
(xCommand Conference DoNotDisturb Activate and xCommand Conference DoNotDisturb
Deactivate).

Example: `xConfiguration Conference 1 DoNotDisturb Mode: Off`

xConfiguration Conference [1..1] DoNotDisturb DefaultTimeout

This setting determines the default duration of a Do Not Disturb session, i.e. the period when
incoming calls are rejected and registered as missed calls. The session can be terminated earlier by
using the user interface (remote control or Touch controller) or the Conference DoNotDisturb Mode
setting. The default value is 60 minutes.

Requires user role: ADMIN

Value space: <0..1440>

Range: Select the number of minutes (between 0 and 1440, i.e. 24 hours) before the Do Not
Disturb session times out automatically.

Example: `xConfiguration Conference 1 DoNotDisturb DefaultTimeOut: 60`

xConfiguration Conference [1..1] FarEndControl Mode

Lets you decide if the remote side (far end) should be allowed to select your video sources and
control your local camera (pan, tilt, zoom).

Requires user role: ADMIN

Value space: <Off/On>

Off: The far end is not allowed to select your video sources or to control your local camera (pan,
tilt, zoom).

On: Allows the far end to be able to select your video sources and control your local camera (pan,
tilt, zoom). You will still be able to control your camera and select your video sources as normal.

Example: `xConfiguration Conference 1 FarEndControl Mode: On`

xConfiguration Conference [1..1] FarEndControl SignalCapability

Set the far end control (H.224) signal capability mode.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable the far end control signal capability.

On: Enable the far end control signal capability.

Example: `xConfiguration Conference 1 FarEndControl SignalCapability: On`

xConfiguration Conference [1..1] Encryption Mode

Set the conference encryption mode. A padlock with the text "Encryption On" or "Encryption Off" displays on screen for a few seconds when the conference starts.

Requires user role: ADMIN

Value space: <Off/On/BestEffort>

Off: The system will not use encryption.

On: The system will only allow calls that are encrypted.

BestEffort: The system will use encryption whenever possible.

 > In Point to point calls: If the far end system supports encryption (AES-128), the call will be encrypted. If not, the call will proceed without encryption.

 > In MultiSite calls: In order to have encrypted MultiSite conferences, all sites must support encryption. If not, the conference will be unencrypted.

Example: `xConfiguration Conference 1 Encryption Mode: BestEffort`

xConfiguration Conference [1..1] DefaultCall Protocol

Set the Default Call Protocol to be used when placing calls from the system.

Requires user role: ADMIN

Value space: <H323/Sip/H320>

H323: H323 ensures that calls are set up as H.323 calls.

Sip: Sip ensures that calls are set up as SIP calls.

H320: H320 ensures that calls are set up as H.320 calls (only applicable if connected to a Cisco TelePresence ISDN Link gateway).

Example: `xConfiguration Conference 1 DefaultCall Protocol: H323`

xConfiguration Conference [1..1] DefaultCall Rate

Set the Default Call Rate to be used when placing calls from the system.

Requires user role: ADMIN

Value space: <64..6000>

Range: Select a value between 64 and 6000 kbps.

Example: `xConfiguration Conference 1 DefaultCall Rate: 768`

xConfiguration Conference [1..1] MaxTransmitCallRate

Specify the maximum transmit bit rate to be used when placing or receiving calls. Note that this is the maximum bit rate for each individual call; use the Conference MaxTotalTransmitCallRate setting to set the aggregated maximum for all simultaneous active calls.

Requires user role: ADMIN

Value space: <64..6000>

Range: Select a value between 64 and 6000 kbps.

Example: `xConfiguration Conference 1 MaxTransmitCallRate: 6000`

xConfiguration Conference [1..1] MaxReceiveCallRate

Specify the maximum receive bit rate to be used when placing or receiving calls. Note that this is the maximum bit rate for each individual call; use the Conference MaxTotalReceiveCallRate setting to set the aggregated maximum for all simultaneous active calls.

Requires user role: ADMIN

Value space: <64..6000>

Range: Select a value between 64 and 6000 kbps.

Example: `xConfiguration Conference 1 MaxReceiveCallRate: 6000`

34

xConfiguration Conference [1..1] MaxTotalTransmitCallRate

This configuration applies when using a video system's built-in MultiSite feature (optional) to host a multipoint video conference.

Specify the maximum overall transmit bit rate allowed. The bit rate will be divided fairly among all active calls at any time. This means that the individual calls will be up-speeded or down-speeded as appropriate when someone leaves or enters a multipoint conference, or when a call is put on hold (suspended) or resumed.

The maximum transmit bit rate for each individual call is defined in the Conference MaxTransmitCallRate setting.

Requires user role: ADMIN

Value space: <64..10000>

Range: Select a value between 64 and 10000.

Example: `xConfiguration Conference 1 MaxTotalTransmitCallRate: 9000`

xConfiguration Conference [1..1] MaxTotalReceiveCallRate

This configuration applies when using a video system's built-in MultiSite feature (optional) to host a multipoint video conference.

Specify the maximum overall receive bit rate allowed. The bit rate will be divided fairly among all active calls at any time. This means that the individual calls will be up-speeded or down-speeded as appropriate when someone leaves or enters a multipoint conference, or when a call is put on hold (suspended) or resumed.

The maximum receive bit rate for each individual call is defined in the Conference MaxReceiveCallRate setting.

Requires user role: ADMIN

Value space: <64..10000>

Range: Select a value between 64 and 10000.

Example: `xConfiguration Conference 1 MaxTotalReceiveCallRate: 9000`

xConfiguration Conference [1..1] VideoBandwidth Mode

Set the conference video bandwidth mode.

Requires user role: ADMIN

Value space: <Dynamic/Static>

Dynamic: The available transmit bandwidth for the video channels are distributed among the currently active channels. If there is no presentation, the main video channels will use the bandwidth of the presentation channel.

Static: The available transmit bandwidth is assigned to each video channel, even if it is not active.

Example: `xConfiguration Conference 1 VideoBandwidth Mode: Dynamic`

xConfiguration Conference [1..1] VideoBandwidth MainChannel Weight

The available transmit video bandwidth is distributed on the main channel and presentation channel according to "MainChannel Weight" and "PresentationChannel Weight". If the main channel weight is 2 and the presentation channel weight is 1, then the main channel will use twice as much bandwidth as the presentation channel.

Requires user role: ADMIN

Value space: <1..10>

Range: 1 to 10.

Example: `xConfiguration Conference 1 VideoBandwidth MainChannel Weight: 5`

xConfiguration Conference [1..1] VideoBandwidth PresentationChannel Weight

The available transmit video bandwidth is distributed on the main channel and presentation channel according to "MainChannel Weight" and "PresentationChannel Weight". If the main channel weight is 2 and the presentation channel weight is 1, then the main channel will use twice as much bandwidth as the presentation channel.

Requires user role: ADMIN

Value space: <1..10>

Range: 1 to 10.

Example: `xConfiguration Conference 1 VideoBandwidth PresentationChannel Weight: 5`

xConfiguration Conference [1..1] PacketLossResilience Mode

Set the packetloss resilience mode. This configuration will only take effect for calls initiated after the configuration is set.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable the packetloss resilience.

On: Enable the packetloss resilience.

Example: `xConfiguration Conference 1 PacketLossResilience Mode: On`

xConfiguration Conference [1..1] Presentation Policy

Control how the presentation service is to be performed.

Requires user role: ADMIN

Value space: <LocalRemote/LocalOnly>

LocalRemote: The presentation will be shown locally and sent to remote side.

LocalOnly: The presentation will only be shown locally.

Example: `xConfiguration Conference 1 Presentation Policy: LocalRemote`

xConfiguration Conference [1..1] Presentation RelayQuality

This configuration applies to video systems that are using the built-in MultiSite feature (optional) to host a multipoint video conference. When a remote user shares a presentation, the video system (codec) will transcode the presentation and send it to the other participants in the multipoint conference. The RelayQuality setting specifies whether to give priority to high frame rate or to high resolution for the presentation source.

Requires user role: ADMIN

Value space: <Motion/Sharpness>

Motion: Gives the highest possible frame rate. Used when there is a need for higher frame rates, typically when there is a lot of motion in the picture.

Sharpness: Gives the highest possible resolution. Used when you want the highest quality of detailed images and graphics.

Example: xConfiguration Conference 1 Presentation RelayQuality: Sharpness

xConfiguration Conference [1..1] Presentation OnPlacedOnHold

Define whether or not to continue sharing a presentation after the remote site has put you on hold.

Requires user role: ADMIN

Value space: <Stop/NoAction>

Stop: The video system stops the presentation sharing when the remote site puts you on hold. The presentation will not continue when the call is resumed.

NoAction: The video system will not stop the presentation sharing when put on hold. The presentation will not be shared while you are on hold, but it will continue automatically when the call is resumed.

Example: xConfiguration Conference 1 Presentation OnPlacedOnHold: NoAction

xConfiguration Conference [1..1] Multipoint Mode

Define how the video system handles multiparty video conferences.

If registered to a Cisco TelePresence Video Communication Server (VCS), the video system can either use its own built-in MultiSite feature, or it can rely on the MultiWay network solution. MultiWay requires that the video network includes a multipoint control unit (MCU).

If registered to a Cisco Unified Communications Manager (CUCM) version 8.6.2 or newer, the video system can use either the CUCM conference bridge, or the video system's own built-in MultiSite feature. Which one to use is set-up by CUCM.

Both Multiway and the CUCM conference bridge allows you to set up conferences with many participants. The MultiSite feature allows up to four participants (yourself included) plus one additional audio call.

Note that the MultiSite feature is optional and may not be available on all video systems.

Requires user role: ADMIN

Value space: <Auto/Off/MulitSite/MultiWay/CUCMMediaResourceGroupList>

Auto: If a MultiWay address is specified in the NetworkServices MultiWay Address setting, MultiWay takes priority over MultiSite. If neither MultiWay nor MultiSite is available, the Multipoint Mode is set to Off automatically.

Off: Multiparty conferences are not allowed.

MultiSite: Multiparty conferences are set up using MultiSite. If MultiSite is chosen when the MultiSite feature is not available, the Multipoint Mode will be set to Off.

MultiWay: Multiparty conferences are set up using MultiWay. The Multipoint Mode will be set to Off automatically if the MultiWay service is unavailable, for example if a server address is not specified in the NetworkServices MultiWay Address setting.

CUCMMediaResourceGroupList: Multiparty conferences (ad hoc conferences) are hosted by the CUCM configured conference bridge. This setting is provisioned by CUCM in a CUCM environment and should never be set manually by the user.

Example: xConfiguration Conference 1 Multipoint Mode: Auto

xConfiguration Conference [1..1] IncomingMultisiteCall Mode

Select whether or not to allow incoming calls when already in a call/conference.

Requires user role: ADMIN

Value space: <Allow/Deny>

Allow: You will be notified when someone calls you while you are already in a call. You can accept the incoming call or not. The ongoing call may be put on hold while answering the incoming call; or you may merge the calls (requires MultiSite or MultiWay support).

Deny: An incoming call will be rejected if you are already in a call. You will not be notified about the incoming call. However, the call will appear as a missed call in the call history list.

Example: xConfiguration Conference 1 IncomingMultisiteCall Mode: Allow

FacilityService configuration

xConfiguration FacilityService Service [1..5] Type

Up to five different facility services can be supported simultaneously. With this setting you can select what kind of services they are.

A facility service is not available unless both the FacilityService Service Name and the FacilityService Service Number settings are properly set.

Only FacilityService Service 1 with Type Helpdesk is available on the Touch controller. Facility services are not available when using the remote control and on-screen menu.

Requires user role: ADMIN

Value space: <Other/Concierge/Helpdesk/Emergency/Security/Catering/Transportation>

Other: Select this option for services not covered by the other options.

Concierge: Select this option for concierge services.

Helpdesk: Select this option for helpdesk services.

Emergency: Select this option for emergency services.

Security: Select this option for security services.

Catering: Select this option for catering services.

Transportation: Select this option for transportation services.

Example: `xConfiguration FacilityService Service 1 Type: Helpdesk`

xConfiguration FacilityService Service [1..5] Name

Set the name of each facility service. Up to five different facility services are supported.

A facility service is not available unless both the FacilityService Service Name and the FacilityService Service Number settings are properly set.

Only FacilityService Service 1 is available on the Touch controller, and its Name is used on the facility service call button. Facility services are not available when using the remote control and on-screen menu.

Requires user role: ADMIN

Value space: <S: 0, 255>

Format: String with a maximum of 255 characters.

Example: `xConfiguration FacilityService Service 1 Name: ""`

xConfiguration FacilityService Service [1..5] Number

Set the number for each facility service. Up to five different facility services are supported.

A facility service is not available unless both the FacilityService Service Name and the FacilityService Service Number settings are properly set.

Only FacilityService Service 1 is available on the Touch controller. Facility services are not available when using the remote control and on-screen menu.

Requires user role: ADMIN

Value space: <S: 0, 255>

Format: String with a maximum of 255 characters.

Example: `xConfiguration FacilityService Service 1 Number: ""`

xConfiguration FacilityService Service [1..5] CallType

Set the call type for each facility service. Up to five different facility services are supported.

A facility service is not available unless both the FacilityService Service Name and the FacilityService Service Number settings are properly set.

Only FacilityService Service 1 is available on the Touch controller. Facility services are not available when using the remote control and on-screen menu.

Requires user role: ADMIN

Value space: <Video/Audio>

Video: Select this option for video calls.

Audio: Select this option for audio calls.

Example: `xConfiguration FacilityService Service 1 CallType: Video`

H323 configuration

xConfiguration H323 NAT Mode

The firewall traversal technology creates a secure path through the firewall barrier, and enables proper exchange of audio/video data when connected to an external video conferencing system (when the IP traffic goes through a NAT router). NOTE: NAT does not work in conjunction with gatekeepers.

Requires user role: ADMIN

Value space: <Auto/Off/On>

Auto: The system will determine if the H323 NAT Address or the real IP address should be used in signaling. This makes it possible to place calls to endpoints on the LAN as well as endpoints on the WAN. If the H323 NAT Address is wrong or not set, the real IP address will be used.

Off: The system will signal the real IP address.

On: The system will signal the configured H323 NAT Address instead of its real IP address in Q.931 and H.245. The NAT Server Address will be shown in the startup-menu as: "My IP Address: 10.0.2.1". If the H323 NAT Address is wrong or not set, H.323 calls cannot be set up.

Example: `xConfiguration H323 NAT Mode: Off`

xConfiguration H323 NAT Address

Enter the external/global IP address to the router with NAT support. Packets sent to the router will then be routed to the system. Note that NAT cannot be used when registered to a gatekeeper.

In the router, the following ports must be routed to the system's IP address:

 * Port 1720
 * Port 5555-6555
 * Port 2326-2487

Requires user role: ADMIN

Value space: <S: 0, 64>

Format: A valid IPv4 address or IPv6 address.

Example: `xConfiguration H323 NAT Address: ""`

xConfiguration H323 Profile [1..1] Authentication Mode

Set the authenticatin mode for the H.323 profile.

Requires user role: ADMIN

Value space: <Off/On>

Off: If the H.323 Gatekeeper Authentication Mode is set to Off the system will not try to authenticate itself to a H.323 Gatekeeper, but will still try a normal registration.

On: If the H.323 Gatekeeper Authentication Mode is set to On and a H.323 Gatekeeper indicates that it requires authentication, the system will try to authenticate itself to the gatekeeper. NOTE: Requires the Authentication LoginName and Authentication Password to be defined on both the codec and the Gatekeeper.

Example: `xConfiguration H323 Profile 1 Authentication Mode: Off`

xConfiguration H323 Profile [1..1] Authentication LoginName

The system sends the Authentication Login Name and the Authentication Password to a H.323 Gatekeeper for authentication. The authentication is a one way authentication from the codec to the H.323 Gatekeeper, i.e. the system is authenticated to the gatekeeper. If the H.323 Gatekeeper indicates that no authentication is required, the system will still try to register. NOTE: Requires the H.323 Gatekeeper Authentication Mode to be enabled.

Requires user role: ADMIN

Value space: <S: 0, 50>

Format: String with a maximum of 50 characters.

Example: `xConfiguration H323 Profile 1 Authentication LoginName: ""`

xConfiguration H323 Profile [1..1] Authentication Password

The system sends the Authentication Login Name and the Authentication Password to a H.323 Gatekeeper for authentication. The authentication is a one way authentication from the codec to the H.323 Gatekeeper, i.e. the system is authenticated to the gatekeeper. If the H.323 Gatekeeper indicates that no authentication is required, the system will still try to register. NOTE: Requires the H.323 Gatekeeper Authentication Mode to be enabled.

Requires user role: ADMIN

Value space: <S: 0, 50>

Format: String with a maximum of 50 characters.

Example: `xConfiguration H323 Profile 1 Authentication Password: ""`

xConfiguration H323 Profile [1..1] CallSetup Mode

The H.323 Call Setup Mode defines whether to use a Gatekeeper or Direct calling when establishing H323 calls.

NOTE: Direct H.323 calls can be made even though the H.323 Call Setup Mode is set to Gatekeeper.

Requires user role: ADMIN

Value space: <Direct/Gatekeeper>

Direct: An IP address must be used when dialing in order to make the H323 call.

Gatekeeper: The system will use a Gatekeeper to make a H.323 call. When selecting this option the H323 Profile Gatekeeper Address and H323 Profile Gatekeeper Discovery settings must also be configured.

Example: `xConfiguration H323 Profile 1 CallSetup Mode: Gatekeeper`

xConfiguration H323 Profile [1..1] Gatekeeper Discovery

Determine how the system shall register to a H.323 Gatekeeper.

Requires user role: ADMIN

Value space: <Manual/Auto>

Manual: The system will use a specific Gatekeeper identified by the Gatekeeper's IP address.

Auto: The system will automatically try to register to any available Gatekeeper. If a Gatekeeper responds to the request sent from the codec within 30 seconds this specific Gatekeeper will be used. This requires that the Gatekeeper is in auto discovery mode as well. If no Gatekeeper responds, the system will not use a Gatekeeper for making H.323 calls and hence an IP address must be specified manually.

Example: `xConfiguration H323 Profile 1 Gatekeeper Discovery: Manual`

xConfiguration H323 Profile [1..1] Gatekeeper Address

Enter the IP address of the Gatekeeper. NOTE: Requires the H.323 Call Setup Mode to be set to Gatekeeper and the Gatekeeper Discovery to be set to Manual.

Requires user role: ADMIN

Value space: <S: 0, 255>

Format: A valid IPv4 address, IPv6 address or DNS name.

Example: `xConfiguration H323 Profile 1 Gatekeeper Address: "192.0.2.0"`

xConfiguration H323 Profile [1..1] H323Alias E164

The H.323 Alias E.164 defines the address of the system, according to the numbering plan implemented in the H.323 Gatekeeper. The E.164 alias is equivalent to a telephone number, sometimes combined with access codes.

Requires user role: ADMIN

Value space: <S: 0, 30>

Format: Compact string with a maximum of 30 characters. Valid characters are 0-9, * and #.

Example: `xConfiguration H323 Profile 1 H323Alias E164: "90550092"`

xConfiguration H323 Profile [1..1] H323Alias ID

Lets you specify the H.323 Alias ID which is used to address the system on a H.323 Gatekeeper and will be displayed in the call lists. Example: "firstname.lastname@company.com", "My H.323 Alias ID"

Requires user role: ADMIN

Value space: <S: 0, 49>

Format: String with a maximum of 49 characters.

Example: `xConfiguration H323 Profile 1 H323Alias ID: "firstname.lastname@company.com"`

xConfiguration H323 Profile [1..1] PortAllocation

The H.323 Port Allocation setting affects the H.245 port numbers used for H.323 call signalling.

Requires user role: ADMIN

Value space: <Dynamic/Static>

Dynamic: The system will allocate which ports to use when opening a TCP connection. The reason for doing this is to avoid using the same ports for subsequent calls, as some firewalls consider this as a sign of attack. When Dynamic is selected, the H.323 ports used are from 11000 to 20999. Once 20999 is reached they restart again at 11000. For RTP and RTCP media data, the system is using UDP ports in the range 2326 to 2487. Each media channel is using two adjacent ports, ie 2330 and 2331 for RTP and RTCP respectively. The ports are automatically selected by the system within the given range. Firewall administrators should not try to deduce which ports are used when, as the allocation schema within the mentioned range may change without any further notice.

Static: When set to Static the ports are given within a static predefined range [5555-6555].

Example: `xConfiguration H323 Profile 1 PortAllocation: Dynamic`

Network configuration

xConfiguration Network [1..1] IPStack

Select if the system should use IPv4, IPv6 or Dual on the network interface. Dual means that the system will be able to operate on both IP versions at the same time. This means that the system can have both an IPv4 and an IPv6 address at the same time.

NOTE: Restart the system after changing this setting.

Requires user role: ADMIN

Value space: <Dual/IPv4/IPv6>

Dual: When set to Dual, the network interface can operate on both IP versions at the same time, and can have both an IPv4 and an IPv6 address at the same time.

IPv4: When set to IPv4, the system will use IPv4 on the network interface.

IPv6: When set to IPv6, the system will use IPv6 on the network interface.

Example: `xConfiguration Network 1 IPStack: IPv4`

xConfiguration Network [1..1] IPv4 Assignment

Define how the system will obtain its IPv4 address, subnet mask and gateway address. This setting only applies to systems on IPv4 networks.

Requires user role: ADMIN

Value space: <Static/DHCP>

Static: The addresses must be configured manually using the Network IPv4 Address, Network IPv4 Gateway and Network IPv4 SubnetMask settings (static addresses).

DHCP: The system addresses are automatically assigned by the DHCP server.

Example: `xConfiguration Network 1 IPv4 Assignment: DHCP`

xConfiguration Network [1..1] IPv4 Address

Enter the static IPv4 network address for the system. This setting is only applicable when Network Assignment is set to Static.

Requires user role: ADMIN

Value space: <S: 0, 64>

Format: A valid IPv4 address.

Example: `xConfiguration Network 1 IPv4 Address: "192.0.2.2"`

xConfiguration Network [1..1] IPv4 Gateway

Define the IPv4 network gateway. This setting is only applicable when the Network Assignment is set to Static.

Requires user role: ADMIN

Value space: <S: 0, 64>

Format: A valid IPv4 address.

Example: `xConfiguration Network 1 IPv4 Gateway: "192.0.2.1"`

xConfiguration Network [1..1] IPv4 SubnetMask

Define the IPv4 network subnet mask. This setting is only applicable when the Network Assignment is set to Static.

Requires user role: ADMIN

Value space: <S: 0, 64>

Format: The valid IPv4 address format.

Example: `xConfiguration Network 1 IPv4 SubnetMask: "255.255.255.0"`

xConfiguration Network [1..1] IPv6 Assignment

Define how the system will obtain its IPv6 address and the default gateway address. This setting only applies to systems on IPv6 networks.

Requires user role: ADMIN

Value space: <Static/DHCPv6/Autoconf>

Static: The codec and gateway IP addresses must be configured manually using the Network IPv6 Address and Network IPv6 Gateway settings. The options, for example NTP and DNS server addresses, must either be set manually or obtained from a DHCPv6 server. The Network IPv6 DHCPOptions setting determines which method to use.

DHCPv6: All IPv6 addresses, including options, will be obtained from a DHCPv6 server. See RFC 3315 for a detailed description. The Network IPv6 DHCPOptions setting will be ignored.

Autoconf: Enable IPv6 stateless autoconfiguration of the IPv6 network interface. See RFC 4862 for a detailed description. The options, for example NTP and DNS server addresses, must either be set manually or obtained from a DHCPv6 server. The Network IPv6 DHCPOptions setting determines which method to use.

Example: `xConfiguration Network 1 IPv6 Assignment: Autoconf`

xConfiguration Network [1..1] IPv6 Address

Enter the static IPv6 network address for the system. This setting is only applicable when the Network IPv6 Assignment is set to Static.

Requires user role: ADMIN

Value space: <S: 0, 64>
Format: A valid IPv6 address.

Example: `xConfiguration Network 1 IPv6 Address: "2001:0DB8:0000:0000:0000:0000:0000:000 2"`

xConfiguration Network [1..1] IPv6 Gateway

Define the IPv6 network gateway address. This setting is only applicable when the Network IPv6 Assignment is set to Static.

Requires user role: ADMIN

Value space: <S: 0, 64>
Format: A valid IPv6 address.

Example: `xConfiguration Network 1 IPv6 Gateway: "2001:0DB8:0000:0000:0000:0000:0000:000 1"`

xConfiguration Network [1..1] IPv6 DHCPOptions

Retrieve a set of DHCP options, for example NTP and DNS server addresses, from a DHCPv6 server.

Requires user role: ADMIN

Value space: <Off/On>
Off: Disable the retrieval of DHCP options from a DHCPv6 server.
On: Enable the retrieval of a selected set of DHCP options from a DHCPv6 server.

Example: `xConfiguration Network 1 IPv6 DHCPOptions: On`

xConfiguration Network [1..1] DHCP RequestTFTPServerAddress

This setting is used only for video systems that are registered to a Cisco Unified Communications Manager (CUCM).

The setting determines whether the endpoint should ask the DHCP server for DHCP option 150, so that it can discover the address of the TFTP server (provisioning server) automatically.

If this setting is Off or the DHCP server does not support option 150, the TFTP server address must be set manually using the Provisioning ExternalManager Address setting.

Note: If the Network VLAN Voice Mode setting is Auto and the Cisco Discovery Protocol (CDP) assigns an ID to the voice VLAN, then a request for option 150 will always be sent. That is, the Network DHCP RequestTFTFServerAddress setting will be ignored.

Requires user role: ADMIN

Value space: <Off/On>
Off: The video system will not send a request for DHCP option 150 and the address of the TFTP server must be set manually. See the note above for any exception to this rule.
On: The video system will send a request for option 150 to the DHCP server so that it can automatically discover the address of the TFTP server.

Example: `xConfiguration Network 1 DHCP RequestTFTPServerAddress: On`

xConfiguration Network [1..1] DNS Domain Name

DNS Domain Name is the default domain name suffix which is added to unqualified names.

Example: If the DNS Domain Name is "company.com" and the name to lookup is "MyVideoSystem", this will result in the DNS lookup "MyVideoSystem.company.com".

Requires user role: ADMIN

Value space: <S: 0, 64>
Format: String with a maximum of 64 characters.

Example: `xConfiguration Network 1 DNS Domain Name: ""`

xConfiguration Network [1..1] DNS Server [1..3] Address

Define the network addresses for DNS servers. Up to 3 addresses may be specified. If the network addresses are unknown, contact your administrator or Internet Service Provider.

Requires user role: ADMIN

Value space: <S: 0, 64>
Format: A valid IPv4 address or IPv6 address.

Example: `xConfiguration Network 1 DNS Server 1 Address: ""`

xConfiguration Network [1..1] QoS Mode

The QoS (Quality of Service) is a method which handles the priority of audio, video and data in the network. The QoS settings must be supported by the infrastructure. Diffserv (Differentiated Services) is a computer networking architecture that specifies a simple, scalable and coarse-grained mechanism for classifying, managing network traffic and providing QoS priorities on modern IP networks.

Requires user role: ADMIN

Value space: <Off/Diffserv>

Off: No QoS method is used.

Diffserv: When you set the QoS Mode to Diffserv, the Network QoS Diffserv Audio, Network QoS Diffserv Video, Network QoS Diffserv Data, Network QoS Diffserv Signalling, Network QoS Diffserv ICMPv6 and Network QoS Diffserv NTP settings are used to prioritize packets.

Example: xConfiguration Network 1 QoS Mode: Diffserv

xConfiguration Network [1..1] QoS Diffserv Audio

Note: This setting will only take effect if Network QoS Mode is set to Diffserv.

Define which priority Audio packets should have in the IP network.

The priority for the packets ranges from 0 to 63 - the higher the number, the higher the priority. The recommended class for Audio is CS4, which equals the decimal value 32. If in doubt, contact your network administrator.

The priority set here might be overridden when packets are leaving the network controlled by the local network administrator.

Requires user role: ADMIN

Value space: <0..63>

Range: Select a value between 0 to 63 - the higher the number, the higher the priority. The default value is 0 (best effort).

Example: xConfiguration Network 1 QoS Diffserv Audio: 0

xConfiguration Network [1..1] QoS Diffserv Video

Note: This setting will only take effect if Network QoS Mode is set to Diffserv.

Define which priority Video packets should have in the IP network. The packets on the presentation channel (shared content) are also in the Video packet category.

The priority for the packets ranges from 0 to 63 - the higher the number, the higher the priority. The recommended class for Video is CS4, which equals the decimal value 32. If in doubt, contact your network administrator.

The priority set here might be overridden when packets are leaving the network controlled by the local network administrator.

Requires user role: ADMIN

Value space: <0..63>

Range: Select a value between 0 to 63 - the higher the number, the higher the priority. The default value is 0 (best effort).

Example: xConfiguration Network 1 QoS Diffserv Video: 0

xConfiguration Network [1..1] QoS Diffserv Data

Note: This setting will only take effect if Network QoS Mode is set to Diffserv.

Define which priority Data packets should have in the IP network.

The priority for the packets ranges from 0 to 63 - the higher the number, the higher the priority. The recommended value for Data is 0, which means best effort. If in doubt, contact your network administrator.

The priority set here might be overridden when packets are leaving the network controlled by the local network administrator.

Requires user role: ADMIN

Value space: <0..63>

Range: Select a value between 0 to 63 - the higher the number, the higher the priority. The default value is 0 (best effort).

Example: xConfiguration Network 1 QoS Diffserv Data: 0

xConfiguration Network [1..1] QoS Diffserv Signalling

Note: This setting will only take effect if Network QoS Mode is set to Diffserv.

Define which priority Signalling packets that are deemed critical (time-sensitive) for the real-time operation should have in the IP network.

The priority for the packets ranges from 0 to 63 - the higher the number, the higher the priority. The recommended class for Signalling is CS3, which equals the decimal value 24. If in doubt, contact your network administrator.

The priority set here might be overridden when packets are leaving the network controlled by the local network administrator.

Requires user role: ADMIN

Value space: <0..63>

Range: Select a value between 0 to 63 - the higher the number, the higher the priority. The default value is 0 (best effort).

Example: `xConfiguration Network 1 QoS Diffserv Signalling: 0`

xConfiguration Network [1..1] QoS Diffserv ICMPv6

Note: This setting will only take effect if Network QoS Mode is set to Diffserv.

Define which priority ICMPv6 packets should have in the IP network.

The priority for the packets ranges from 0 to 63 - the higher the number, the higher the priority. The recommended value for ICMPv6 is 0, which means best effort. If in doubt, contact your network administrator.

The priority set here might be overridden when packets are leaving the network controlled by the local network administrator.

Requires user role: ADMIN

Value space: <0..63>

Range: Select a value between 0 to 63 - the higher the number, the higher the priority. The default value is 0 (best effort).

Example: `xConfiguration Network 1 QoS Diffserv ICMPv6: 0`

xConfiguration Network [1..1] QoS Diffserv NTP

Note: This setting will only take effect if Network QoS Mode is set to Diffserv.

Define which priority NTP packets should have in the IP network.

The priority for the packets ranges from 0 to 63 - the higher the number, the higher the priority. The recommended value for NTP is 0, which means best effort. If in doubt, contact your network administrator.

The priority set here might be overridden when packets are leaving the network controlled by the local network administrator.

Requires user role: ADMIN

Value space: <0..63>

Range: Select a value between 0 to 63 - the higher the number, the higher the priority. The default value is 0 (best effort).

Example: `xConfiguration Network 1 QoS Diffserv NTP: 0`

xConfiguration Network [1..1] IEEE8021X Mode

The system can be connected to an IEEE 802.1X LAN network, with a port-based network access control that is used to provide authenticated network access for Ethernet networks.

Requires user role: ADMIN

Value space: <Off/On>

Off: The 802.1X authentication is disabled (default).

On: The 802.1X authentication is enabled.

Example: `xConfiguration Network 1 IEEE8021X Mode: Off`

xConfiguration Network [1..1] IEEE8021X TlsVerify

Verification of the server-side certificate of an IEEE802.1x connection against the certificates in the local CA-list when TLS is used. The CA-list must be uploaded to the video system. This can be done from the web interface.

This setting takes effect only when Network [1..1] IEEE8021X Eap Tls is enabled (On).

Requires user role: ADMIN

Value space: <Off/On>

Off: When set to Off, TLS connections are allowed without verifying the server-side X.509 certificate against the local CA-list. This should typically be selected if no CA-list has been uploaded to the codec.

On: When set to On, the server-side X.509 certificate will be validated against the local CA-list for all TLS connections. Only servers with a valid certificate will be allowed.

Example: `xConfiguration xConfiguration Network 1 IEEE8021X TlsVerify: Off`

xConfiguration Network [1..1] IEEE8021X UseClientCertificate

Authentication using a private key/certificate pair during an IEEE802.1x connection. The authentication X.509 certificate must be uploaded to the video system. This can be done from the web interface.

Requires user role: ADMIN

Value space: <Off/On>

Off: When set to Off client-side authentication is not used (only server-side).

On: When set to On the client (video system) will perform a mutual authentication TLS handshake with the server.

Example: `xConfiguration Network 1 IEEE8021X UseClientCertificate: Off`

xConfiguration Network [1..1] IEEE8021X Identity

The 802.1X Identity is the user name needed for 802.1X authentication.

Requires user role: ADMIN

Value space: <S: 0, 64>

Format: String with a maximum of 64 characters.

Example: `xConfiguration Network 1 IEEE8021X Identity: ""`

xConfiguration Network [1..1] IEEE8021X Password

The 802.1X Password is the password needed for 802.1X authentication.

Requires user role: ADMIN

Value space: <S: 0, 32>

Format: String with a maximum of 32 characters.

Example: `xConfiguration Network 1 IEEE8021X Password: ""`

xConfiguration Network [1..1] IEEE8021X AnonymousIdentity

The 802.1X Anonymous ID string is to be used as unencrypted identity with EAP (Extensible Authentication Protocol) types that support different tunneled identity, like EAP-PEAP and EAP-TTLS. If set, the anonymous ID will be used for the initial (unencrypted) EAP Identity Request.

Requires user role: ADMIN

Value space: <S: 0, 64>

Format: String with a maximum of 64 characters.

Example: `xConfiguration Network 1 IEEE8021X AnonymousIdentity: ""`

xConfiguration Network [1..1] IEEE8021X Eap Md5

Set the Md5 (Message-Digest Algorithm 5) mode. This is a Challenge Handshake Authentication Protocol that relies on a shared secret. Md5 is a Weak security.

Requires user role: ADMIN

Value space: <Off/On>

Off: The EAP-MD5 protocol is disabled.

On: The EAP-MD5 protocol is enabled (default).

Example: `xConfiguration Network 1 IEEE8021X Eap Md5: On`

xConfiguration Network [1..1] IEEE8021X Eap Ttls

Set the TTLS (Tunneled Transport Layer Security) mode. Authenticates LAN clients without the need for client certificates. Developed by Funk Software and Certicom. Usually supported by Agere Systems, Proxim and Avaya.

Requires user role: ADMIN

Value space: <Off/On>

Off: The EAP-TTLS protocol is disabled.

On: The EAP-TTLS protocol is enabled (default).

Example: `xConfiguration Network 1 IEEE8021X Eap Ttls: On`

xConfiguration Network [1..1] IEEE8021X Eap Tls

Enable or disable the use of EAP-TLS (Transport Layer Security) for IEEE802.1x connections. The EAP-TLS protocol, defined in RFC 5216, is considered one of the most secure EAP standards. LAN clients are authenticated using client certificates.

Requires user role: ADMIN

Value space: <Off/On>

Off: The EAP-TLS protocol is disabled.

On: The EAP-TLS protocol is enabled (default).

Example: `xConfiguration Network 1 IEEE8021X Eap Tls: On`

xConfiguration Network [1..1] IEEE8021X Eap Peap

Set the Peap (Protected Extensible Authentication Protocol) mode. Authenticates LAN clients without the need for client certificates. Developed by Microsoft, Cisco and RSA Security.

Requires user role: ADMIN

Value space: <Off/On>
Off: The EAP-PEAP protocol is disabled.
On: The EAP-PEAP protocol is enabled (default).

Example: `xConfiguration Network 1 IEEE8021X Eap Peap: On`

xConfiguration Network [1..1] MTU

Set the Ethernet MTU (Maximum Transmission Unit).

Requires user role: ADMIN

Value space: <576..1500>
Range: Select a value from 576 to 1500 bytes.

Example: `xConfiguration Network 1 MTU: 1500`

xConfiguration Network [1..1] Speed

Set the Ethernet link speed.
NOTE: If running older software versions than TC6.0, restart the system for any change to this setting to take effect.

Requires user role: ADMIN

Value space: <Auto/10half/10full/100half/100full/1000full>
Auto: Autonegotiate link speed.
10half: Force link to 10 Mbps half-duplex.
10full: Force link to 10 Mbps full-duplex.
100half: Force link to 100 Mbps half-duplex.
100full: Force link to 100 Mbps full-duplex.
1000full: Force link to 1 Gbps full-duplex.

Example: `xConfiguration Network 1 Speed: Auto`

xConfiguration Network [1..1] TrafficControl Mode

Set the network traffic control mode to decide how to control the video packets transmission speed.

Requires user role: ADMIN

Value space: <Off/On>
Off: Transmit video packets at link speed.
On: Transmit video packets at maximum 20 Mbps. Can be used to smooth out bursts in the outgoing network traffic.

Example: `xConfiguration Network 1 TrafficControl: On`

xConfiguration Network [1..1] RemoteAccess Allow

Filter IP addresses for access to ssh/telnet/HTTP/HTTPS.

Requires user role: ADMIN

Value space: <S: 0, 255>
Format: String with a maximum of 255 characters, comma separated IP adresses or IP range.

Example: `xConfiguration Network 1 RemoteAccess Allow: "192.168.1.231, 192.168.1.182"`

xConfiguration Network [1..1] VLAN Voice Mode

Set the VLAN voice mode. The VLAN Voice Mode will be set to Auto automatically if you choose Cisco UCM (Cisco Unified Communications Manager) as provisioning infrastructure via the Provisioning Wizard on the Touch controller.

Requires user role: ADMIN

Value space: <Auto/Manual/Off>
Auto: The Cisco Discovery Protocol (CDP), if available, assigns an id to the voice VLAN. If CDP is not available, VLAN is not enabled.
Manual: The VLAN ID is set manually using the Network VLAN Voice VlanId setting. If CDP is available, the manually set value will be overruled by the value assigned by CDP.
Off: VLAN is not enabled.

Example: `xConfiguration Network 1 VLAN Voice Mode: Off`

xConfiguration Network [1..1] VLAN Voice VlanId

Set the VLAN voice ID. This setting will only take effect if VLAN Voice Mode is set to Manual.

Requires user role: ADMIN

Value space: <1..4094>
Range: Select a value from 1 to 4094.

Example: `xConfiguration Network 1 VLAN Voice VlanId: 1`

NetworkServices configuration

xConfiguration NetworkServices XMLAPI Mode

Enable or disable the video system's XML API. For security reasons this may be disabled. Disabling the XML API will limit the remote manageability with for example TMS, which no longer will be able to connect to the video system.

Requires user role: ADMIN

Value space: <Off/On>

Off: The XML API is disabled.

On: The XML API is enabled (default).

Example: `xConfiguration NetworkServices XMLAPI Mode: On`

xConfiguration NetworkServices MultiWay Address

The MultiWay address must be equal to the Conference Factory Alias, as configured on the Video Communication Server. The Multiway™ conferencing enables video endpoint users to introduce a 3rd party into an existing call.

Multiway™ can be used in the following situations:

1) When you want to add someone else in to your existing call.

2) When you are called by a 3rd party while already in a call and you want to include that person in the call.

Requirements: Video Communication Server (VCS) version X5 (or later) and Codian MCU version 3.1 (or later). Video systems invited to join the Multiway™ conference must support the H.323 routeToMC facility message if in an H.323 call, or SIP REFER message if in a SIP call.

Requires user role: ADMIN

Value space: <S: 0, 255>

Format: String with a maximum of 255 characters (a valid dial URI).

Example: `xConfiguration NetworkServices MultiWay Address: "h323:multiway@company.com"`

xConfiguration NetworkServices MultiWay Protocol

Determine the protocol to be used for MultiWay calls.

Requires user role: ADMIN

Value space: <Auto/H323/Sip>

Auto: The system will select the protocol for MultiWay calls.

H323: The H323 protocol will be used for MultiWay calls.

Sip: The SIP protocol will be used for MultiWay calls.

Example: `xConfiguration NetworkServices MultiWay Protocol: Auto`

xConfiguration NetworkServices H323 Mode

Determine whether the system should be able to place and receive H.323 calls or not.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable the possibility to place and receive H.323 calls.

On: Enable the possibility to place and receive H.323 calls (default).

Example: `xConfiguration NetworkServices H323 Mode: On`

xConfiguration NetworkServices HTTP Mode

Set the HTTP mode to enable/disable access to the system through a web browser. The web interface is used for system management, call management such as call transfer, diagnostics and software uploads.

NOTE: Restart the system for any change to this setting to take effect.

Requires user role: ADMIN

Value space: <Off/On>

Off: The HTTP protocol is disabled.

On: The HTTP protocol is enabled.

Example: `xConfiguration NetworkServices HTTP Mode: On`

xConfiguration NetworkServices HTTPS Mode

HTTPS is a web protocol that encrypts and decrypts user page requests as well as the pages that are returned by the web server.

NOTE: Restart the system for any change to this setting to take effect.

Requires user role: ADMIN

Value space: <Off/On>

Off: The HTTPS protocol is disabled.

On: The HTTPS protocol is enabled.

Example: `xConfiguration NetworkServices HTTPS Mode: On`

xConfiguration NetworkServices HTTPS VerifyServerCertificate

When the video system connects to an external HTTPS server (like a phone book server or an external manager), this server will present a certificate to the video system to identify itself.

Requires user role: ADMIN

Value space: <Off/On>

Off: Do not verify server certificates.

On: Requires the system to verify that the server certificate is signed by a trusted Certificate Authority (CA). This requires that a list of trusted CAs are uploaded to the system in advance.

Example: `xConfiguration NetworkServices HTTPS VerifyServerCertificate: Off`

xConfiguration NetworkServices HTTPS VerifyClientCertificate

When the video system connects to a HTTPS client (like a web browser), the client can be asked to present a certificate to the video system to identify itself.

Requires user role: ADMIN

Value space: <Off/On>

Off: Do not verify client certificates.

On: Requires the client to present a certificate that is signed by a trusted Certificate Authority (CA). This requires that a list of trusted CAs are uploaded to the system in advance.

Example: `xConfiguration NetworkServices HTTPS VerifyClientCertificate: Off`

xConfiguration NetworkServices HTTPS OCSP Mode

Define the support for OCSP (Online Certificate Status Protocol) responder services. The OCSP feature allows users to enable OCSP instead of certificate revocation lists (CRLs) to check the certificate status.

For any outgoing HTTPS connection, the OCSP responder is queried of the status. If the corresponding certificate has been revoked, then the HTTPS connection will not be used.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable OCSP support.

On: Enable OCSP support.

Example: `xConfiguration NetworkServices HTTPS OCSP Mode: Off`

xConfiguration NetworkServices HTTPS OCSP URL

Specify the URL of the OCSP responder (server) that will be used to check the certificate status.

Requires user role: ADMIN

Value space: <S: 0, 255>

Format: String with a maximum of 255 characters.

Example: `xConfiguration NetworkServices HTTPS OCSP URL: "http://ocspserver.company.com:81"`

xConfiguration NetworkServices NTP Mode

The Network Time Protocol (NTP) is used to synchronize the time of the system to a reference time server. The time server will subsequently be queried every 24th hour for time updates. The time will be displayed on the top of the screen. The system will use the time to timestamp messages transmitted to Gatekeepers or Border Controllers requiring H.235 authentication. The system will use the time to timestamp messages transmitted to Gatekeepers or Border Controllers that requires H.235 authentication. It is also used for timestamping Placed Calls, Missed Calls and Received Calls.

Requires user role: ADMIN

Value space: <Auto/Off/Manual>

Auto: The system will use the NTP server, by which address is supplied from the DHCP server in the network. If no DHCP server is used, or the DHCP server does not provide the system with a NTP server address, the system will use the static defined NTP server address specified by the user.

Off: The system will not use an NTP server.

Manual: The system will always use the static defined NTP server address specified by the user.

Example: `xConfiguration NetworkServices NTP Mode: Manual`

xConfiguration NetworkServices NTP Address

Enter the NTP Address to define the network time protocol server address. This address will be used if NTP Mode is set to Manual, or if set to Auto and no address is supplied by a DHCP server.

Requires user role: ADMIN

Value space: <S: 0, 64>

Format: A valid IPv4 address, IPv6 address or DNS name.

Example: `xConfiguration NetworkServices NTP Address: "1.ntp.tandberg.com"`

xConfiguration NetworkServices SIP Mode

Determine whether the system should be able to place and receive SIP calls or not.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable the possibility to place and receive SIP calls.

On: Enable the possibility to place and receive SIP calls (default).

Example: `xConfiguration NetworkServices SIP Mode: On`

xConfiguration NetworkServices SNMP Mode

SNMP (Simple Network Management Protocol) is used in network management systems to monitor network-attached devices (routers, servers, switches, projectors, etc) for conditions that warrant administrative attention. SNMP exposes management data in the form of variables on the managed systems, which describe the system configuration. These variables can then be queried (set to ReadOnly) and sometimes set (set to ReadWrite) by managing applications.

Requires user role: ADMIN

Value space: <Off/ReadOnly/ReadWrite>

Off: Disable the SNMP network service.

ReadOnly: Enable the SNMP network service for queries only.

ReadWrite: Enable the SNMP network service for both queries and commands.

Example: `xConfiguration NetworkServices SNMP Mode: ReadWrite`

xConfiguration NetworkServices SNMP Host [1..3] Address

Enter the address of up to three SNMP Managers.

The system's SNMP Agent (in the codec) responds to requests from SNMP Managers (a PC program etc.), for example about system location and system contact. SNMP traps are not supported.

Requires user role: ADMIN

Value space: <S: 0, 64>

Format: A valid IPv4 address, IPv6 address or DNS name.

Example: `xConfiguration NetworkServices SNMP Host 1 Address: ""`

xConfiguration NetworkServices SNMP CommunityName

Enter the name of the Network Services SNMP Community. SNMP Community names are used to authenticate SNMP requests. SNMP requests must have a password (case sensitive) in order to receive a response from the SNMP Agent in the codec. The default password is "public". If you have the Cisco TelePresence Management Suite (TMS) you must make sure the same SNMP Community is configured there too. NOTE: The SNMP Community password is case sensitive.

Requires user role: ADMIN

Value space: <S: 0, 50>

Format: String with a maximum of 50 characters.

Example: `xConfiguration NetworkServices SNMP CommunityName: "public"`

xConfiguration NetworkServices SNMP SystemContact

Enter the name of the Network Services SNMP System Contact.

Requires user role: ADMIN

Value space: <S: 0, 50>

Format: String with a maximum of 50 characters.

Example: `xConfiguration NetworkServices SNMP SystemContact: ""`

xConfiguration NetworkServices SNMP SystemLocation

Enter the name of the Network Services SNMP System Location.

Requires user role: ADMIN

Value space: <S: 0, 50>

Format: String with a maximum of 50 characters.

Example: `xConfiguration NetworkServices SNMP SystemLocation: ""`

xConfiguration NetworkServices SSH Mode

SSH (or Secure Shell) protocol can provide secure encrypted communication between the codec and your local computer.

Requires user role: ADMIN

Value space: <Off/On>

Off: The SSH protocol is disabled.

On: The SSH protocol is enabled.

Example: `xConfiguration NetworkServices SSH Mode: On`

xConfiguration NetworkServices SSH AllowPublicKey

Secure Shell (SSH) public key authentication can be used to access the codec.

Requires user role: ADMIN

Value space: <Off/On>

Off: The SSH public key is not allowed.

On: The SSH public key is allowed.

Example: `xConfiguration NetworkServices SSH AllowPublicKey: On`

xConfiguration NetworkServices Telnet Mode

Telnet is a network protocol used on the Internet or Local Area Network (LAN) connections.

Requires user role: ADMIN

Value space: <Off/On>

Off: The Telnet protocol is disabled. This is the factory setting.

On: The Telnet protocol is enabled.

Example: `xConfiguration NetworkServices Telnet Mode: Off`

xConfiguration NetworkServices CTMS Mode

This setting determines whether or not to allow multiparty conferences controlled by a Cisco TelePresence Multipoint Switch (CTMS).

Video systems running software TC5.0 or later are able to initiate or join non-encrypted multiparty conferences controlled by CTMS version 1.8 or later. Encrypted conferences are supported as from software versions TC6.0 and CTMS 1.9.1. Encryption is addressed in the NetworkServices CTMS Encryption setting.

Requires user role: ADMIN

Value space: <Off/On>

Off: Mulitparty conferencing via CTMS is prohibited.

On: Mulitparty conferencing via CTMS is allowed.

Example: `xConfiguration NetworkServices CTMS Mode: On`

xConfiguration NetworkServices CTMS Encryption

This setting indicates whether or not the video system supports encryption when participating in a multiparty meeting controlled by a Cisco TelePresence Multipoint Switch (CTMS).

CTMS allows three security settings for meetings: non-secure (not encrypted), best effort (encrypted if all participants support encryption, otherwise not encrypted) and secure (always encrypted).

Requires user role: ADMIN

Value space: <Off/BestEffort>

Off: The video system does not allow encryption and therefore cannot participate in a secure CTMS meeting (encrypted). When participating in a best effort CTMS meeting, the meeting will be downgraded to non-secure (not encrypted).

BestEffort: The video system can negotiate encryption parameters with CTMS and participate in a secure CTMS meeting (encrypted). Do not use this value if the CTMS version is older than 1.9.1.

Example: `xConfiguration NetworkServices CTMS Encryption: Off`

49

Phonebook configuration

xConfiguration Phonebook Server [1..1] ID

Enter a name for the external phone book.

Requires user role: ADMIN

Value space: <S: 0, 64>
Format: String with a maximum of 64 characters.

Example: xConfiguration Phonebook Server 1 ID: ""

xConfiguration Phonebook Server [1..1] Type

Select the phonebook server type.

Requires user role: ADMIN

Value space: <VCS/TMS/Callway/CUCM>
VCS: Select VCS if the phonebook is located on the Cisco TelePresence Video Communication Server.
TMS: Select TMS if the phonebook is located on the Cisco TelePresence Management Suite server.
Callway: Select Callway if the phonebook is to be provided by the WebEx TelePresence subscription service (formerly called CallWay). Contact your WebEx TelePresence provider for more information.
CUCM: Select CUCM if the phonebook is located on the Cisco Unified Communications Manager.

Example: xConfiguration Phonebook Server 1 Type: TMS

xConfiguration Phonebook Server [1..1] URL

Enter the address (URL) to the external phone book server.

Requires user role: ADMIN

Value space: <S: 0, 255>
Format: String with a maximum of 255 characters.

Example: xConfiguration Phonebook Server 1 URL: "http://tms.company.com/tms/public/external/phonebook/phonebookservice.asmx"

Provisioning configuration

xConfiguration Provisioning Connectivity

This setting controls how the device discovers whether it should request an internal or external configuration from the provisioning server.

Requires user role: ADMIN

Value space: <Internal/External/Auto>
Internal: Request internal configuration.
External: Request external configuration.
Auto: Automatically discover using NAPTR queries whether internal or external configurations should be requested. If the NAPTR responses have the "e" flag, external configurations will be requested. Otherwise internal configurations will be requested.

Example: xConfiguration Provisioning Connectivity: Auto

xConfiguration Provisioning Mode

It is possible to configure a video system using a provisioning system (external manager). This allows video conferencing network administrators to manage many video systems simultaneously.
With this setting you choose which type of provisioning system to use. Provisioning can also be switched off. Contact your provisioning system provider/representative for more information.

Requires user role: ADMIN

Value space: <Off/TMS/VCS/CallWay/CUCM/Auto>
Off: The video system will not be configured by a provisioning system.
TMS: The video system will be configured using TMS (Cisco TelePresence Management System).
VCS: Not applicable in this version.
Callway: The video system will be configured using the WebEx TelePresence subscription service (formerly called Callway).
CUCM: The video system will be configured using CUCM (Cisco Unified Communications Manager).
Auto: The provisioning server will automatically be selected by the video system.

Example: xConfiguration Provisioning Mode: TMS

xConfiguration Provisioning LoginName

This is the user name part of the credentials used to authenticate the video system with the provisioning server. This setting must be used when required by the provisioning server. If Provisioning Mode is Callway (WebEx TelePresence), enter the video number.

Requires user role: ADMIN

Value space: <S: 0, 80>
Format: String with a maximum of 80 characters.

Example: `xConfiguration Provisioning LoginName: ""`

xConfiguration Provisioning Password

This is the password part of the credentials used to authenticate the video system with the provisioning server. This setting must be used when required by the provisioning server. If Provisioning Mode is Callway (WebEx TelePresence), enter the activation code.

Requires user role: ADMIN

Value space: <S: 0, 64>
Format: String with a maximum of 64 characters.

Example: `xConfiguration Provisioning Password: ""`

xConfiguration Provisioning HttpMethod

Select the HTTP method to be used for the provisioning.

Requires user role: ADMIN

Value space: <GET/POST>
GET: Select GET when the provisioning server supports GET.
POST: Select POST when the provisioning server supports POST.

Example: `xConfiguration Provisioning HttpMethod: POST`

xConfiguration Provisioning ExternalManager Address

Enter the IP Address or DNS name of the external manager / provisioning system.

If an External Manager Address (and Path) is configured, the system will send a message to this address when starting up. When receiving this message the external manager / provisioning system can return configurations/commands to the unit as a result.

When using CUCM or TMS provisioning, the DHCP server can be set up to provide the external manager address automatically (DHCP Option 242 for TMS, and DHCP Option 150 for CUCM). An address set in the Provisioning ExternalManager Address setting will override the address provided by DHCP.

Requires user role: ADMIN

Value space: <S: 0, 64>
Format: A valid IPv4 address, IPv6 address or DNS name.

Example: `xConfiguration Provisioning ExternalManager Address: ""`

xConfiguration Provisioning ExternalManager Protocol

Determine whether to use secure management or not.

Requires user role: ADMIN

Value space: <HTTP/HTTPS>
HTTP: Set to HTTP to disable secure management. Requires HTTP to be enabled in the NetworkServices HTTP Mode setting.
HTTPS: Set to HTTPS to enable secure management. Requires HTTPS to be enabled in the NetworkServices HTTPS Mode setting.

Example: `xConfiguration Provisioning ExternalManager Protocol: HTTP`

xConfiguration Provisioning ExternalManager Path

Set the Path to the external manager / provisioning system. This setting is required when several management services reside on the same server, i.e. share the same External Manager address.

Requires user role: ADMIN

Value space: <S: 0, 255>
Format: String with a maximum of 255 characters.

Example: `xConfiguration Provisioning ExternalManager Path: "tms/public/external/management/SystemManagementService.asmx"`

xConfiguration Provisioning ExternalManager Domain

Enter the SIP domain for the VCS provisioning server.

Requires user role: ADMIN

Value space: <S: 0, 64>
Format: String with a maximum of 64 characters.

Example: xConfiguration Provisioning ExternalManager Domain: "any.domain.com"

RTP configuration

xConfiguration RTP Ports Range Start

Specify the first port in the range of RTP ports. Also see the H323 Profile [1..1] PortAllocation setting.
NOTE: Restart the system for any change to this setting to take effect.

Requires user role: ADMIN

Value space: <1024..65502>
Range: Select a value from 1024 to 65502.

Example: xConfiguration RTP Ports Range Start: 2326

xConfiguration RTP Ports Range Stop

Specify the last RTP port in the range. Also see the H323 Profile [1..1] PortAllocation setting.
NOTE: Restart the system for any change to this setting to take effect.

Requires user role: ADMIN

Value space: <1056..65535>
Range: Select a value from 1056 to 65535.

Example: xConfiguration RTP Ports Range Stop: 2486

Security configuration

xConfiguration Security Audit Logging Mode

Determine where to record or transmit the audit logs. When using the External or ExternalSecure modes, you also must enter the address and port number for the audit server in the Security Audit Server Address and Security Audit Server Port settings.

NOTE: Restart the system for any change to this setting to take effect.

Requires user role: AUDIT

Value space: <Off/Internal/External/ExternalSecure>

Off: No audit logging is performed.

Internal: The system records the audit logs to internal logs, and rotates logs when they are full.

External: The system sends the audit logs to an external audit server (syslog server). The audit server must support TCP.

ExternalSecure: The system sends encrypted audit logs to an external audit server (syslog server) that is verified by a certificate in the Audit CA list. The Audit CA list file must be uploaded to the codec using the web interface. The common_name parameter of a certificate in the CA list must match the IP address of the audit server.

Example: xConfiguration Security Audit Logging Mode: Off

xConfiguration Security Audit Server Address

Enter the IP address of the audit server. Only valid IPv4 or IPv6 address formats are accepted. Host names are not supported. This setting is only relevant when Security Audit Logging Mode is set to External or ExternalSecure.

NOTE: Restart the system for any change to this setting to take effect.

Requires user role: AUDIT

Value space: <S: 0, 64>

Format: A valid IPv4 address or IPv6 address.

Example: xConfiguration Security Audit Server Address: ""

xConfiguration Security Audit Server Port

Enter the port of the audit server that the system shall send its audit logs to. The default port is 514. This setting is only relevant when Security Audit Logging Mode is set to External or ExternalSecure.

NOTE: Restart the system for any change to this setting to take effect.

Requires user role: AUDIT

Value space: <0..65535>

Range: Select a value from 0 to 65535.

Example: xConfiguration Security Audit Server Port: 514

xConfiguration Security Audit OnError Action

Determine what happens when the connection to the audit server is lost. This setting is only relevant when Security Audit Logging Mode is set to ExternalSecure.

NOTE: Restart the system for any change to this setting to take effect.

Requires user role: AUDIT

Value space: <Halt/Ignore>

Halt: If a halt condition is detected the system is rebooted and only the auditor is allowed to operate the unit until the halt condition has passed. When the halt condition has passed the audit logs are re-spooled to the audit server. Halt conditions are: A network breach (no physical link), no audit server running (or wrong audit server address or port), TLS authentication failed (if in use), local backup (re-spooling) log full.

Ignore: The system will continue its normal operation, and rotate internal logs when full. When the connection is restored it will again send its audit logs to the audit server.

Example: xConfiguration Security Audit OnError Action: Ignore

xConfiguration Security Session ShowLastLogon

When logging in to the system using SSH or Telnet you will see the UserId, time and date of the last session that did a successful login.

Requires user role: ADMIN

Value space: <Off/On>

On: Show information about the last session.

Off: Do not show information about the last session.

Example: xConfiguration Security Session ShowLastLogon: Off

xConfiguration Security Session InactivityTimeout

Determine how long the system will accept inactivity from the user before he is automatically logged out.

Requires user role: ADMIN

Value space: <0..10000>

Range: Select a value between 1 and 10000 seconds; or select 0 when inactivity should not enforce automatic logout.

Example: xConfiguration Security Session InactivityTimeout: 0

SerialPort configuration

xConfiguration SerialPort Mode

Enable/disable the serial port (connection via USB and RS-232 adapter).

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable the serial port.

On: Enable the serial port.

Example: xConfiguration SerialPort Mode: On

xConfiguration SerialPort BaudRate

Specify the baud rate (data transmission rate, bits per second) for the serial port. The default value is 38400.

Other connection parameters for the serial port are: Data bits: 8; Parity: None; Stop bits: 1; Flow control: None.

Requires user role: ADMIN

Value space: <9600/19200/38400/57600/115200>

Range: Select a baud rate from the baud rates listed (bps).

Example: xConfiguration SerialPort BaudRate: 38400

xConfiguration SerialPort LoginRequired

Determine if login shall be required when connecting to the serial port.

Requires user role: ADMIN

Value space: <Off/On>

Off: The user can access the codec via the serial port without any login.

On: Login is required when connecting to the codec via the serial port.

Example: xConfiguration SerialPort LoginRequired: On

SIP configuration

xConfiguration SIP Profile [1..1] Ice

ICE (Interactive Connectivity Establishment, RFC 5245) is a NAT traversal solution that the endpoints can use to discover the optimized media path. Thus the shortest route for audio and video is always secured between the endpoints.

NOTE: ICE is not supported when registered to CUCM (Cisco Unified Communication Manager).

Requires user role: ADMIN

Value space: <Off/On>

Off: Set to Off to disable ICE.

On: Set to On, to enable ICE.

Example: xConfiguration SIP Profile 1 Ice: Off

xConfiguration SIP Profile [1..1] IceDefaultCandidate

This is the default IP address that the endpoint will receive media on until ICE has reached a conclusion about which media route to use (up to the first 5 seconds of a call).

Requires user role: ADMIN

Value space: <Off/On>

Host: The endpoint will receive media on its own IP address.

Rflx: The endpoint will receive media on its public IP address as seen by the TURN server.

Relay: The endpoint will receive media on the IP address and port allocated on the TURN server, and is used as a fallback until ICE has concluded.

Example: xConfiguration SIP Profile 1 IceDefaultCandidate: Host

xConfiguration SIP Profile [1..1] Turn Server

This is the address of the TURN (Traversal Using Relay NAT) server that the endpoints will use. It is used as a media relay fallback and it is also used to discover the endpoint's own public IP address.

Requires user role: ADMIN

Value space: <S: 0, 255>

Format: The preferred format is DNS SRV record (e.g. _turn._udp.<domain>), or it can be a valid IPv4 or IPv6 address.

Example: xConfiguration SIP Profile 1 Turn Server: "_turn._udp.example.com"

xConfiguration SIP Profile [1..1] Turn UserName

The user name needed for accessing the TURN server.

Requires user role: ADMIN

Value space: <S: 0, 128>
Format: String with a maximum of 128 characters.

Example: xConfiguration SIP Profile 1 Turn UserName: ""

xConfiguration SIP Profile [1..1] Turn Password

The password needed for accessing the TURN server.

Requires user role: ADMIN

Value space: <S: 0, 128>
Format: String with a maximum of 128 characters.

Example: xConfiguration SIP Profile 1 Turn Password: ""

xConfiguration SIP Profile [1..1] URI

The SIP URI (Uniform Resource Identifier) is the address that is used to identify the video system. The URI is registered and used by the SIP services to route inbound calls to the system. The SIP URI syntax is defined in RFC 3261.

Requires user role: ADMIN

Value space: <S: 0, 255>
Format: String with maximum 255 characters and compliant with the SIP URI syntax.

Example: xConfiguration SIP Profile 1 URI: "sip:firstname.lastname@company.com"

xConfiguration SIP Profile [1..1] DisplayName

When configured the incoming call will report the DisplayName instead of the SIP URI.

Requires user role: ADMIN

Value space: <S: 0, 255>
Format: String with a maximum of 255 characters.

Example: xConfiguration SIP Profile 1 DisplayName: ""

xConfiguration SIP Profile [1..1] Authentication [1..1] LoginName

This is the user name part of the credentials used to authenticate towards the SIP proxy.

Requires user role: ADMIN

Value space: <S: 0, 128>
Format: String with a maximum of 128 characters.

Example: xConfiguration SIP Profile 1 Authentication 1 LoginName: ""

xConfiguration SIP Profile [1..1] Authentication [1..1] Password

This is the password part of the credentials used to authenticate towards the SIP proxy.

Requires user role: ADMIN

Value space: <S: 0, 128>
Format: String with a maximum of 128 characters.

Example: xConfiguration SIP Profile 1 Authentication 1 Password: ""

xConfiguration SIP Profile [1..1] DefaultTransport

Select the transport protocol to be used over the LAN.

Requires user role: ADMIN

Value space: <TCP/UDP/Tls/Auto>
TCP: The system will always use TCP as the default transport method.
UDP: The system will always use UDP as the default transport method.
Tls: The system will always use TLS as the default transport method. For TLS connections a SIP CA-list can be uploaded to the video system. If no such CA-list is available on the system then anonymous Diffie Hellman will be used.
Auto: The system will try to connect using transport protocols in the following order: TLS, TCP, UDP.

Example: xConfiguration SIP Profile 1 DefaultTransport: Auto

xConfiguration SIP Profile [1..1] TlsVerify

For TLS connections a SIP CA-list can be uploaded to the video system. This can be done from the web interface.

Requires user role: ADMIN

Value space: <Off/On>

Off: Set to Off to allow TLS connections without verifying them. The TLS connections are allowed to be set up without verifying the x.509 certificate received from the server against the local CA-list. This should typically be selected if no SIP CA-list has been uploaded.

On: Set to On to verify TLS connections. Only TLS connections to servers, whose x.509 certificate is validated against the CA-list, will be allowed.

Example: `xConfiguration SIP Profile 1 TlsVerify: Off`

xConfiguration SIP Profile [1..1] Outbound

Turn on or off the client initiated connections mechanism for firewall traversal, connection reuse and redundancy. The current version supports RFC 5626.

Requires user role: ADMIN

Value space: <Off/On>

Off: Connect to the single proxy configured first in Proxy Address list.

On: Set up multiple outbound connections to servers in the Proxy Address list.

Example: `xConfiguration SIP Profile 1 Outbound: Off`

xConfiguration SIP Profile [1..1] Proxy [1..4] Address

The Proxy Address is the manually configured address for the outbound proxy. It is possible to use a fully qualified domain name, or an IP address. The default port is 5060 for TCP and UDP but another one can be provided. If SIP Profile Outbound is enabled, multiple proxies can be addressed.

Requires user role: ADMIN

Value space: <S: 0, 255>

Format: A valid IPv4 address, IPv6 address or DNS name.

Example: `xConfiguration SIP Profile 1 Proxy 1 Address: ""`

xConfiguration SIP Profile [1..1] Proxy [1..4] Discovery

Select if the SIP Proxy address is to be obtained manually or by using Dynamic Host Configuration Protocol (DHCP).

Requires user role: ADMIN

Value space: <Auto/Manual>

Auto: When Auto is selected, the SIP Proxy address is obtained using Dynamic Host Configuration Protocol (DHCP).

Manual: When Manual is selected, the manually configured SIP Proxy address will be used.

Example: `xConfiguration SIP Profile 1 Proxy 1 Discovery: Manual`

xConfiguration SIP Profile [1..1] Type

Enables SIP extensions and special behaviour for a vendor or provider.

Requires user role: ADMIN

Value space: <Standard/Alcatel/Avaya/Cisco/Microsoft/Nortel>

Standard: To be used when registering to standard SIP Proxy (tested with Cisco TelePresence VCS and Broadsoft)

Alcatel: To be used when registering to Alcatel-Lucent OmniPCX Enterprise. NOTE: This mode is not fully supported.

Avaya: To be used when registering to Avaya Communication Manager. NOTE: This mode is not fully supported.

Cisco: To be used when registering to Cisco Unified Communication Manager.

Microsoft: To be used when registering to Microsoft LCS or OCS. NOTE: This mode is not fully supported.

Nortel: To be used when registering to Nortel MCS 5100 or MCS 5200 PBX. NOTE: This mode is not fully supported.

Example: `xConfiguration SIP Profile 1 Type: Standard`

xConfiguration SIP Profile [1..1] Mailbox

When registered to a Cisco Unified Communications Manager (CUCM) you may be offered the option of having a private voice mailbox. Enter the number (address) of the mailbox in this setting, or leave the string empty if you do not have a voice mailbox.

Requires user role: ADMIN

Value space: <S: 0, 255>>

Format: String with a maximum of 255 characters.

Example: `xConfiguration SIP Profile 1 Mailbox: "12345678"`

xConfiguration SIP Profile [1..1] Line

When registered to a Cisco Unified Communications Manager (CUCM) the endpoint may be part of a shared line. This means that several devices share the same directory number. The different devices sharing the same number receive status from the other appearances on the line as defined in RFC 4235.

Note that shared lines are set up by CUCM, not by the endpoint. Therefore do not change this setting manually; CUCM pushes this information to the endpoint when required.

Requires user role: ADMIN

Value space: <Private/Shared>

Shared: The system is part of a shared line and is therefore sharing its directory number with other devices.

Private: This system is not part of a shared line (default).

Example: xConfiguration SIP Profile 1 Line: Private

xConfiguration SIP ListenPort

Turn on or off the listening for incoming connections on the SIP TCP/UDP ports. If turned off, the endpoint will only be reachable through the SIP registrar (CUCM or VCS). It is recommended to leave this setting at its default value.

Requires user role: ADMIN

Value space: <On/Off>

On: Listening for incoming connections on the SIP TCP/UDP ports is turned on.

Off: Listening for incoming connections on the SIP TCP/UDP ports is turned off.

Example: xConfiguration SIP ListenPort: On

Standby configuration

xConfiguration Standby Control

Determine whether the system should go into standby mode or not.

Requires user role: ADMIN

Value space: <Off/On>

Off: The system will not enter standby mode.

On: Enter standby mode when the Standby Delay has timed out. NOTE: Requires the Standby Delay to be set to an appropriate value.

Example: xConfiguration Standby Control: On

xConfiguration Standby Delay

Define how long (in minutes) the system shall be in idle mode before it goes into standby mode. NOTE: Requires the Standby Control to be enabled.

Requires user role: ADMIN

Value space: <1..480>

Range: Select a value from 1 to 480 minutes.

Example: xConfiguration Standby Delay: 10

xConfiguration Standby BootAction

Define the camera position after a restart of the codec.

Requires user role: ADMIN

Value space: <None/Preset1/Preset2/Preset3/Preset4/Preset5/Preset6/Preset7/Preset8/ Preset9/Preset10/Preset11/Preset12/Preset13/Preset14/Preset15/RestoreCameraPosition/ DefaultCameraPosition>

None: No action.

Preset1 to Preset15: After a reboot the camera position will be set to the position defined by the selected preset.

RestoreCameraPosition: After a reboot the camera position will be set to the position it had before the last boot.

DefaultCameraPosition: After a reboot the camera position will be set to the factory default position.

Example: xConfiguration Standby BootAction: DefaultCameraPosition

xConfiguration Standby StandbyAction

Define the camera position when going into standby mode.

Requires user role: ADMIN

Value space: <None/PrivacyPosition>
None: No action.
PrivacyPosition: Turns the camera to a sideways position for privacy.

Example: xConfiguration Standby StandbyAction: PrivacyPosition

xConfiguration Standby WakeupAction

Define the camera position when leaving standby mode.

Requires user role: ADMIN

Value space: <None/Preset1/Preset2/Preset3/Preset4/Preset5/Preset6/Preset7/Preset8/
Preset9/Preset10/Preset11/Preset12/Preset13/Preset14/Preset15/RestoreCameraPosition/
DefaultCameraPosition>
None: No action.
Preset1 to Preset15: When leaving standby the camera position will be set to the position defined
by the selected preset.
RestoreCameraPosition: When leaving standby the camera position will be set to the position it
had before entering standby.
DefaultCameraPosition: When leaving standby the camera position will be set to the factory default
position.

Example: xConfiguration Standby WakeupAction: RestoreCameraPosition

SystemUnit configuration

xConfiguration SystemUnit Name

Enter a System Name to define a name of the system unit. If the H.323 Alias ID is configured on the
system then this ID will be used instead of the system name. The system name will be displayed:
 1) When the codec is acting as an SNMP Agent.
 2) Towards a DHCP server.

Requires user role: ADMIN

Value space: <S: 0, 50>
Format: String with a maximum of 50 characters.

Example: xConfiguration SystemUnit Name: "Meeting Room"

xConfiguration SystemUnit MenuLanguage

Select the language to be used in the menus on screen or on the Touch controller.

Requires user role: USER

Value space: <English/ChineseSimplified/ChineseTraditional/Czech/Danish/Dutch/Finnish/French/
German/Hungarian/Italian/Japanese/Korean/Norwegian/Polish/PortugueseBrazilian/Russian/
Spanish/SpanishLatin/Swedish/Turkish>

Example: xConfiguration SystemUnit MenuLanguage: English

58

xConfiguration SystemUnit ContactInfo Type

Choose which type of contact information to show in the status field in the upper left corner of the main display and Touch controller. The information can also be read with the command xStatus SystemUnit ContactInfo.

Requires user role: ADMIN

Value space: <Auto/None/IPv4/IPv6/H323Id/E164Alias/H320Number/SipUri/SystemName/DisplayName>

Auto: Show the address which another system can dial to reach this system. The address depends on the default call protocol and system registration.

None: Do not show any contact information in the status field.

IPv4: Show the IPv4 address as contact information.

IPv6: Show the IPv6 address as contact information.

H323Id: Show the H.323 ID as contact information (see the H323 Profile [1..1] H323Alias ID setting).

E164Alias: Show the H.323 E164 Alias as contact information (see the H323 Profile [1..1] H323Alias E164 setting).

H320Number: Show the H.320 number as contact information (only applicable if connected to a Cisco TelePresence ISDN Link gateway).

SipUri: Show the SIP URI as contact information (see the SIP Profile [1..1] URI setting).

SystemName: Show the system name as contact information (see the SystemUnit Name setting).

DisplayName: Show the display name as contact information (see the SIP Profile [1..1] DisplayName setting).

Example: xConfiguration SystemUnit ContactInfo Type: Auto

xConfiguration SystemUnit CallLogging Mode

Set the call logging mode for calls that are received or placed by the system. The call logs may then be viewed via the web interface or using the xHistory command.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable logging.

On: Enable logging.

Example: xConfiguration SystemUnit CallLogging Mode: On

xConfiguration SystemUnit IrSensor

Both the Codec C Series and PrecisionHD camera have IR sensors, and only one of them needs to be enabled at the time. The IR sensor LED is located on the front of the codec and the camera and flickers when an IR signal is received from the remote control.

Requires user role: ADMIN

Value space: <Auto/Off/On>

Auto: The system will automatically disable the IR sensor on the codec if the IR sensor at camera is enabled. Otherwise, the IR sensor on the codec will be enabled.

Off: Disable the IR sensor on the codec.

On: Enable the IR sensor on the codec.

Example: xConfiguration SystemUnit IrSensor: Auto

Time configuration

xConfiguration Time Zone

Set the time zone where the system is located, using Windows time zone description format.

Requires user role: USER

Value space: <GMT-12:00 (International Date Line West)/GMT-11:00 (Midway Island, Samoa)/GMT-10:00 (Hawaii)/GMT-09:00 (Alaska)/GMT-08:00 (Pacific Time (US & Canada); Tijuana)/GMT-07:00 (Arizona)/GMT-07:00 (Mountain Time (US & Canada))/GMT-07:00 (Chihuahua, La Paz, Mazatlan)/GMT-06:00 (Central America)/GMT-06:00 (Saskatchewan)/GMT-06:00 (Guadalajara, Mexico City, Monterrey)/GMT-06:00 (Central Time (US & Canada))/GMT-05:00 (Indiana (East))/GMT-05:00 (Bogota, Lima, Quito)/GMT-05:00 (Eastern Time (US & Canada))/GMT-04:30 (Caracas)/GMT-04:00 (La Paz)/GMT-04:00 (Santiago)/GMT-04:00 (Atlantic Time (Canada))/GMT-03:30 (Newfoundland)/GMT-03:00 (Buenos Aires, Georgetown)/GMT-03:00 (Greenland)/GMT-03:00 (Brasilia)/GMT-02:00 (Mid-Atlantic)/GMT-01:00 (Cape Verde Is.)/GMT-01:00 (Azores)/GMT (Casablanca, Monrovia)/GMT (Coordinated Universal Time)/GMT (Greenwich Mean Time : Dublin, Edinburgh, Lisbon, London)/GMT+01:00 (West Central Africa)/GMT+01:00 (Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna)/GMT+01:00 (Brussels, Copenhagen, Madrid, Paris)/GMT+01:00 (Sarajevo, Skopje, Warsaw, Zagreb)/GMT+01:00 (Belgrade, Bratislava, Budapest, Ljubljana, Prague)/GMT+02:00 (Harare, Pretoria)/GMT+02:00 (Jerusalem)/GMT+02:00 (Athens, Istanbul, Minsk)/GMT+02:00 (Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius)/GMT+02:00 (Cairo)/GMT+02:00 (Bucharest)/GMT+03:00 (Nairobi)/GMT+03:00 (Kuwait, Riyadh)/GMT+03:00 (Moscow, St. Petersburg, Volgograd)/GMT+03:00 (Baghdad)/GMT+03:30 (Tehran)/GMT+04:00 (Abu Dhabi, Muscat)/GMT+04:00 (Baku, Tbilisi, Yerevan)/GMT+04:30 (Kabul)/GMT+05:00 (Islamabad, Karachi, Tashkent)/GMT+05:00 (Ekaterinburg)/GMT+05:30 (Chennai, Kolkata, Mumbai, New Delhi)/GMT+05:45 (Kathmandu)/GMT+06:00 (Sri Jayawardenepura)/GMT+06:00 (Astana, Dhaka)/GMT+06:00 (Almaty, Novosibirsk)/GMT+06:30 (Rangoon)/GMT+07:00 (Bangkok, Hanoi, Jakarta)/GMT+07:00 (Krasnoyarsk)/GMT+08:00 (Perth)/GMT+08:00 (Taipei)/GMT+08:00 (Kuala Lumpur, Singapore)/GMT+08:00 (Beijing, Chongqing, Hong Kong, Urumqi)/GMT+08:00 (Irkutsk, Ulaan Bataar)/GMT+09:00 (Osaka, Sapporo, Tokyo)/GMT+09:00 (Seoul)/GMT+09:00 (Yakutsk)/GMT+09:30 (Darwin)/GMT+09:30 (Adelaide)/GMT+10:00 (Guam, Port Moresby)/GMT+10:00 (Brisbane)/GMT+10:00 (Vladivostok)/GMT+10:00 (Hobart)/GMT+10:00 (Canberra, Melbourne, Sydney)/GMT+11:00 (Magadan, Solomon Is., New Caledonia)/GMT+12:00 (Fiji, Kamchatka, Marshall Is.)/GMT+12:00 (Auckland, Wellington)/GMT+13:00 (Nuku alofa)>

Range: Select a time zone from the list time zones. If using a command line interface; watch up for typos.

Example: xConfiguration Time Zone: "GMT (Greenwich Mean Time : Dublin, Edinburgh, Lisbon, London)"

xConfiguration Time TimeFormat

Set the time format.

Requires user role: USER

Value space: <24H/12H>

24H: Set the time format to 24 hours.
12H: Set the time format to 12 hours (AM/PM).

Example: xConfiguration Time TimeFormat: 24H

xConfiguration Time DateFormat

Set the date format.

Requires user role: USER

Value space: <DD_MM_YY/MM_DD_YY/YY_MM_DD>

DD_MM_YY: The date January 30th 2010 will be displayed: 30.01.10
MM_DD_YY: The date January 30th 2010 will be displayed: 01.30.10
YY_MM_DD: The date January 30th 2010 will be displayed: 10.01.30

Example: xConfiguration Time DateFormat: DD _ MM _ YY

UserInterface configuration

Video configuration

xConfiguration UserInterface TouchPanel DefaultPanel

Select whether to display the list of contacts, the list of scheduled meetings, or a dial pad on the Touch controller as default.

Requires user role: USER

Value space: <ContactList/MeetingList/Dialpad>

ContactList: The contact list (favorites, directory and history) will appear as default on the Touch controller.

MeetingList: The list of scheduled meetings will appear as default on the Touch controller.

Dialpad: A dial pad will appear as default on the Touch controller.

Example: `xConfiguration UserInterface TouchPanel DefaultPanel: ContactList`

xConfiguration Video Input Source [1..3] Name

Enter a name for the video input source.

Requires user role: ADMIN

Value space: <S: 0, 50>

Format: String with a maximum of 50 characters.

Example: `xConfiguration Video Input Source 1 Name: ""`

xConfiguration Video Input Source [1] Connector

Select which video input connector to be active on video input source 1.

Requires user role: ADMIN

Value space: <HDMI>

HDMI: Select HDMI when you want to use the HDMI as input source 1.

Example: `xConfiguration Video Input Source 1 Connector: HDMI`

xConfiguration Video Input Source [2] Connector

Select which video input connector to be active on video input source 2.

Requires user role: ADMIN

Value space: <DVI>

DVI: Select DVI-I when you want to use the DVI-I 2 as input source 2.

Example: `xConfiguration Video Input Source 2 Connector: DVI`

xConfiguration Video Input Source [3] Connector

Select which video input connector to be active on video input source 3.

Requires user role: ADMIN

Value space: <USB>

USB: Select USB when you want to use the USB as input source 3.

Example: `xConfiguration Video Input Source 3 Connector: USB`

xConfiguration Video Input Source [1..3] Type

Set which type of input source is connected to the video input.

Requires user role: ADMIN

Value space: <other/camera/PC/DVD/document_camera>

Other: Select Other when some other type of equipment is connected to the selected video input.

Camera: Select Camera when you have a camera connected to the selected video input.

PC: Select PC when you have a PC connected to the selected video input.

DVD: Select DVD when you have a DVD player connected to the selected video input.

Document_Camera: Select Document_Camera when you have a document camera connected to the selected video input.

Example: `xConfiguration Video Input Source 1 Type: PC`

xConfiguration Video Input Source [1..3] PresentationSelection

In general, any input source can be used as a presentation source; normally, the main camera (self view) will not be used as a presentation source.

This setting is used to define whether to display the presentation source on the local video system's display automatically or not. To share the presentation with the far end always requires additional action (tap Start Presenting on the Touch controller or the Presentation key on the remote control).

The default values for all input sources are Manual.

Requires user role: ADMIN

Value space: <Manual/Automatic/Hidden>

Manual: The content on the input source will not be presented on the local video system's display before you select it. Use either the remote control or the Touch controller to choose which input source to present.

Automatic: Any content on the input source will be presented on the local video system's display automatically. If there is active content on more than one input source (which is set to Automatic) the most recent one will be used.

Hidden: The input source is not expected to be used as a presentation source.

Example: `xConfiguration Video Input Source 1 PresentationSelection: Manual`

xConfiguration Video Input Source [1..3] CameraControl Mode

Select whether or not to enable camera control for the selected video input source when the video input is active.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable camera control.

On: Enable camera control.

Example: `xConfiguration Video Input Source 1 CameraControl Mode: On`

xConfiguration Video Input Source [1..3] CameraControl CameraId

Indicates the ID of the camera. This value is fixed in this product.

Value space: <1>

Range: Indicates the ID of the camera.

xConfiguration Video Input Source [1..3] OptimalDefinition Profile

The Video Input Source Quality setting must be set to Motion for the optimal definition settings to take any effect.

The optimal definition profile should reflect the lighting conditions in your room and the quality of the video input (camera); the better the lighting conditions and video input, the higher the profile. Then, in good lighting conditions, the video encoder will provide better quality (higher resolution or frame rate) for a given call rate.

Generally, we recommend using the Normal or Medium profiles. However, when the lighting conditions are good, the High profile can be set in order to increase the resolution for a given call rate.

Some typical resolutions used for different optimal definition profiles, call rates and transmit frame rates are shown in the table below. It is assumed that dual video is not used. The resolution must be supported by both the calling and called systems.

Use the Video Input Source OptimalDefinition Threshold60fps setting to decide when to use the 60 fps frame rate.

Typical resolutions used for different optimal definition profiles, call rates and frame rates								
Frame rate	Optimal Definition Profile	Call rate						
		256 kbps	768 kbps	1152 kbps	1472 kbps	2560 kbps	4 Mbps	6 Mbps
30 fps	Normal	512×288	1024×576	1280×720	1280×720	1920×1080	1920×1080	1920×1080
	Medium	640×360	1280×720	1280×720	1280×720	1920×1080	1920×1080	1920×1080
	High	768×448	1280×720	1280×720	1920×1080	1920×1080	1920×1080	1920×1080
60 fps	Normal	256×144	512×288	768×448	1024×576	1280×720	1280×720	1920×1080
	Medium	256×144	768×448	1024×576	1024×576	1280×720	1920×1080	1920×1080
	High	512×288	1024×576	1280×720	1280×720	1920×1080	1920×1080	1920×1080

Requires user role: ADMIN

Value space: <Normal/Medium/High>

Normal: Use this profile for a normally to poorly lit environment. Resolutions will be set rather conservative.

Medium: Requires good and stable lighting conditions and a good quality video input. For some call rates this leads to higher resolution.

High: Requires nearly optimal video conferencing lighting conditions and a good quality video input in order to achieve a good overall experience. Rather high resolutions will be used.

Example: xConfiguration Video Input Source 1 OptimalDefinition Profile: Medium

xConfiguration Video Input Source [1..3] OptimalDefinition Threshold60fps

For each video input, this setting tells the system the lowest resolution where it should transmit 60fps. So for all resolutions lower than this, the maximum transmitted framerate would be 30fps, while above this resolution 60fps would also be possible, if the available bandwidth is adequate.

Requires user role: ADMIN

Value space: <512_288/768_448/1024_576/1280_720/1920_1080/Never>

512_288: Set the threshold to 512x288.

768_448: Set the threshold to 768x448.

1024_576: Set the threshold to 1024x576.

1280_720: Set the threshold to 1280x720.

1920_1080: Set the threshold to 1920x1080.

Never: Do not set a threshold for transmitting 60fps.

Example: xConfiguration Video Input Source 1 OptimalDefinition Threshold60fps: 1280 _ 720

xConfiguration Video Input Source [1..3] Quality

When encoding and transmitting video there will be a trade-off between high resolution and high framerate. For some video sources it is more important to transmit high framerate than high resolution and vice versa. The Quality setting specifies whether to give priority to high frame rate or to high resolution for a given source.

Requires user role: ADMIN

Value space: <Motion/Sharpness>

Motion: Gives the highest possible framerate. Used when there is a need for higher frame rates, typically when a large number of participants are present or when there is a lot of motion in the picture.

Sharpness: Gives the highest possible resolution. Used when you want the highest quality of detailed images and graphics.

Example: xConfiguration Video Input Source 1 Quality: Motion

xConfiguration Video MainVideoSource

Define which video input source shall be used as the main video source. The video input source is configured with the "Video Input Source [1..n] Connector" setting.

Requires user role: USER

Value space: <1/2/3>

Range: Select the source to be used as the main video source.

Example: xConfiguration Video MainVideoSource: 1

xConfiguration Video DefaultPresentationSource

Define which video input source shall be used as the default presentation source when you press the Presentation key on the remote control. If using a Touch controller this setting has no effect. The Video Input Source n Connector setting defines which input connector to use for input source n.

Requires user role: USER

Value space: <1/2>

Range: Select the video source to be used as the presentation source.

Example: `xConfiguration Video DefaultPresentationSource: 2`

xConfiguration Video Input HDMI [1..1] RGBQuantizationRange

All devices with HDMI inputs should follow the rules for RGB video quantization range defined in CEA-861. Unfortunately some devices do not follow the standard and this configuration may be used to override the settings to get a perfect image with any source.

Requires user role: ADMIN

Value space: <Auto/Full/Limited>

Auto: RGB quantization range is automatically selected based on the RGB Quantization Range bits (Q0, Q1) in the AVI infoframe. If no AVI infoframe is available, RGB quantization range is selected based on video format according to CEA-861-E.

Full: Full quantization range. The R, G, B quantization range includes all code values (0 - 255). This is defined in CEA-861-E.

Limited: Limited Quantization Range. R, G, B quantization range that excludes some code values at the extremes (16 - 235). This is defined in CEA-861-E.

Example: `xConfiguration Video Input 1 HDMI 1 RGBQuantizationRange: Auto`

xConfiguration Video Input DVI [2] RGBQuantizationRange

All devices with DVI inputs should follow the rules for RGB video quantization range defined in CEA-861. Unfortunately some devices do not follow the standard and this configuration may be used to override the settings to get a perfect image with any source. The default value is set to Full because most DVI sources expects full quantization range.

Requires user role: ADMIN

Value space: <Auto/Full/Limited>

Auto: RGB quantization range is automatically selected based on video format according to CEA-861-E. CE video formats will use limited quantization range levels. IT video formats will use full quantization range levels.

Full: Full quantization range. The R, G, B quantization range includes all code values (0 - 255). This is defined in CEA-861-E.

Limited: Limited Quantization Range. R, G, B quantization range that excludes some code values at the extremes (16 - 235). This is defined in CEA-861-E.

Example: `xConfiguration Video Input 1 DVI 2 RGBQuantizationRange: Full`

xConfiguration Video Input DVI [2] Type

The official DVI standard supports both digital and analog signals. In most cases the default AutoDetect setting can detect whether the signal is analog RGB or digital. However, in some rare cases when DVI-I cables are used (these cables can carry both the analog and digital signals) the auto detection fails. This setting makes it possible to override the AutoDetect and select the correct DVI video input.

Requires user role: ADMIN

Value space: <AutoDetect/Digital/AnalogRGB/AnalogYPbPr>

AutoDetect: Set to AutoDetect to automatically detect if the signal is analog RGB or digital.

Digital: Set to Digital to force the DVI video input to Digital when using DVI-I cables with both analog and digital pins and AutoDetect fails.

AnalogRGB: Set to AnalogRGB to force the DVI video input to AnalogRGB when using DVI-I cables with both analog and digital pins and AutoDetect fails.

AnalogYPbPr: Set to AnalogYPbPr to force the DVI video input to AnalogYPbPr, as the component (YPbPr) signal cannot be auto detected.

Example: `xConfiguration Video Input DVI 2 Type: AutoDetect`

xConfiguration Video Layout Scaling

Define how the system shall adjust the aspect ratio for images or frames when there is a difference between the image and the frame it is to be placed in.

Requires user role: ADMIN

Value space: <Off/On>

Off: No adjustment of the aspect ratio.

On: Let the system automatically adjust aspect ratio.

Example: `xConfiguration Video Layout Scaling: On`

xConfiguration Video Layout ScaleToFrame

Define what to do if the aspect ratio of a video input source doesn't match the aspect ratio of the corresponding image frame in a composition. For example if you have a 4:3 input source (like XGA) to be displayed on a 16:9 output (like HD720).

Requires user role: ADMIN

Value space: <Manual/MaintainAspectRatio/StretchToFit>

Manual: If the difference in aspect ratio between the video input source and the target image frame is less than the Video Layout ScaleToFrameThreshold setting (in percent), the image is stretched to fit. If not, the system will maintain the original aspect ratio.

MaintainAspectRatio: Maintain the aspect ratio of the input source, and fill in black in the rest of the frame (letter boxing or pillar boxing).

StretchToFit: Stretch (horizontally or vertically) the input source to fit into the image frame. NOTE: The general limitation is that you cannot upscale in one direction and at the same time downscale in the other direction. In such situations the codec will apply letterboxing.

Example: `xConfiguration Video Layout ScaleToFrame: MaintainAspectRatio`

xConfiguration Video Layout ScaleToFrameThreshold

Only applicable if the Video Layout ScaleToFrame setting is set to manual. If the difference in aspect ratio between the video input source and the target image frame is less than the ScaleToFrameThreshold setting (in percent), the image is stretched to fit. If not, the system will maintain the original aspect ratio.

Requires user role: ADMIN

Value space: <0..100>

Range: Select a value from 0 to 100 percent.

Example: `xConfiguration Video Layout ScaleToFrameThreshold: 5`

xConfiguration Video Selfview

Determine if the main video source (self view) shall be displayed on screen.

This setting is obsoleted by the Video SelfviewDefault Mode setting as from TC6.0.

Requires user role: USER

Value space: <Off/On>

Off: Do not display self view on screen.

On: Display self view on screen.

Example: `xConfiguration Video Selfview: On`

xConfiguration Video SelfviewPosition

Select where the small self view PiP (Picture-in-Picture) will appear on screen.

This setting is obsoleted by the Video SelfviewDefault PIPPosition setting as from TC6.0.

Requires user role: ADMIN

Value space: <UpperLeft/UpperCenter/UpperRight/CenterLeft/CenterRight/LowerLeft/LowerRight>

UpperLeft: The self view PiP will appear in the upper left corner of the screen.

UpperCenter: The self view PiP will appear in the upper center of the screen.

UpperRight: The self view PiP will appear in the upper right corner of the screen.

CenterLeft: The self view PiP will appear on the left side of the screen, in center.

CenterRight: The self view PiP will appear on the right side of the screen, in center.

LowerLeft: The self view PiP will appear in the lower left corner of the screen.

LowerRight: The self view PiP will appear in the lower right corner of the screen.

Example: `xConfiguration Video SelfviewPosition: LowerRight`

xConfiguration Video SelfviewControl AutoResizing

The size of the self view frame can be configured to automatically change according to the following rules. The size is reduced from full screen to PiP (picture-in-picture) when there is a change in a frame that overlaps with the self view frame. The size is increased from PiP to full screen when nothing else is displayed on the monitor. The last rule does not apply to monitors with MonitorRole set to First.

Requires user role: ADMIN

Value space: <Off/On>

Off: Self view will not be automatically resized.

On: Self view is automatically resized as described above.

Example: `xConfiguration Video SelfviewControl AutoResizing: On`

xConfiguration Video SelfviewDefault Mode

Determine if the main video source (self view) shall be displayed on screen after a call. The position and size of the self view window is determined by the Video SelfviewDefault PIPPosition and the Video Selfview FullscreenMode settings respectively.

This setting obsoletes the Video Selfview setting as from TC6.0.

Requires user role: ADMIN

Value space: <Off/Current/On>

Off: Self view is switched off when leaving a call.

Current: Self view is left as is, i.e. if it was on during the call, it remains on after the call; if it was off during the call, it remains off after the call.

On: Self view is switched on when leaving a call.

Example: `xConfiguration Video SelfviewDefault Mode: Current`

xConfiguration Video SelfviewDefault FullscreenMode

Determine if the main video source (self view) shall be shown in full screen or as a small picture-in-picture (PiP) after a call. The setting only takes effect when self view is switched on (see the Video SelfviewDefault Mode setting).

Requires user role: ADMIN

Value space: <Off/Current/On>

Off: Self view will be shown as a PiP.

Current: The size of the self view picture will be kept unchanged when leaving a call, i.e. if it was a PiP during the call, it remains a PiP after the call; if it was fullscreen during the call, it remains fullscreen after the call.

On: The self view picture will be shown in fullscreen.

Example: `xConfiguration Video SelfviewDefault FullscreenMode: Current`

xConfiguration Video SelfviewDefault PIPPosition

Determine the position on screen of the small self view picture-in-picture (PiP) after a call. The setting only takes effect when self view is switched on (see the Video SelfviewDefault Mode setting) and fullscreen view is switched off (see the Video SelfviewDefault FullscreenMode setting).

This setting obsoletes the Video SelfviewPosition setting as from TC6.0.

Requires user role: ADMIN

Value space: <Current/UpperLeft/UpperCenter/UpperRight/CenterLeft/CenterRight/LowerLeft/ LowerRight >

Current: The position of the self view PiP will be kept unchanged when leaving a call.

UpperLeft: The self view PiP will appear in the upper left corner of the screen.

UpperCenter: The self view PiP will appear in the upper center position.

UpperRight: The self view PiP will appear in the upper right corner of the screen.

CenterLeft: The self view PiP will appear in the center left position.

CentreRight: The self view PiP will appear in the center right position.

LowerLeft: The self view PiP will appear in the lower left corner of the screen.

LowerRight: The self view PiP will appear in the lower right corner of the screen.

Example: `xConfiguration Video SelfviewDefault PIPPosition: Current`

xConfiguration Video SelfviewDefault OnMonitorRole

Determine which monitor/output to display the main video source (self view) on after a call. The value reflects the monitor roles set for the different outputs in the Video Output HDMI MonitorRole settings.

The setting applies both when self view is displayed in full screen, and when it is displayed as picture-in-picture (PiP), but only if the Video Monitors setting is set to Dual.

Requires user role: ADMIN

Value space: <First/Second/Current>

First: The self view picture will be shown on outputs with the Video Output HDMI MonitorRole set to First.

Second: The self view picture will be shown on outputs with the Video Output HDMI MonitorRole set to Second.

Current: When leaving the call, the self view picture will be kept on the same output as during the call.

Example: `xConfiguration Video SelfviewDefault OnMonitorRole: Current`

xConfiguration Video CamCtrlPip CallSetup Mode

This setting is used to switch on self view for a short while when setting up a call. The Video CamCtrlPip CallSetup Duration setting determines for how long it remains on. This applies when self view in general is switched off.

Requires user role: ADMIN

Value space: <Off/On>

Off: Self view is not shown automatically during call setup.

On: Self view is shown automatically during call setup.

Example: xConfiguration Video CamCtrlPip CallSetup Mode: Off

xConfiguration Video CamCtrlPip CallSetup Duration

This setting only has an effect when the Video CamCtrlPip CallSetup Mode setting is switched On. In this case, the number of seconds set here determines for how long self view is shown before it is automatically switched off.

Requires user role: ADMIN

Value space: <1..60>

Range: Choose for how long self view remains on. The valid range is between 1 and 60 seconds.

Example: xConfiguration Video CamCtrlPip CallSetup Duration: 10

xConfiguration Video PIP ActiveSpeaker DefaultValue Position

Determine the position on screen of the active speaker picture-in-picture (PiP). The setting only takes effect when using a video layout where the active speaker is a PiP, i.e. the Overlay layout, or possibly a Custom layout (see the Video Layout LocalLayoutFamily setting). The setting takes effect from the next call onwards; if changed during a call, it will have no effect on the current call.

Requires user role: ADMIN

Value space: <Current/UpperLeft/UpperCenter/UpperRight/CenterLeft/CenterRight/LowerLeft/LowerRight>

Current: The position of the active speaker PiP will be kept unchanged when leaving a call.

UpperLeft: The active speaker PiP will appear in the upper left corner of the screen.

UpperCenter: The active speaker PiP will appear in the upper center position.

UpperRight: The active speaker PiP will appear in the upper right corner of the screen.

CenterLeft: The active speaker PiP will appear in the center left position.

CentreRight: The active speaker PiP will appear in the center right position.

LowerLeft: The active speaker PiP will appear in the lower left corner of the screen.

LowerRight: The active speaker PiP will appear in the lower right corner of the screen.

Example: xConfiguration Video PIP ActiveSpeaker DefaultValue Position: Current

xConfiguration Video PIP Presentation DefaultValue Position

Determine the position on screen of the presentation picture-in-picture (PiP). The setting only takes effect when the presentation is explicitly minimized to a PiP, for example using the remote control or the Touch controller. The setting takes effect from the next call onwards; if changed during a call, it will have no effect on the current call.

Requires user role: ADMIN

Value space: <Current/UpperLeft/UpperCenter/UpperRight/CenterLeft/CenterRight/LowerLeft/LowerRight>

Current: The position of the presentation PiP will be kept unchanged when leaving a call.

UpperLeft: The presentation PiP will appear in the upper left corner of the screen.

UpperCenter: The presentation PiP will appear in the upper center position.

UpperRight: The presentation PiP will appear in the upper right corner of the screen.

CenterLeft: The presentation PiP will appear in the center left position.

CentreRight: The presentation PiP will appear in the center right position.

LowerLeft: The presentation PiP will appear in the lower left corner of the screen.

LowerRight: The presentation PiP will appear in the lower right corner of the screen.

Example: xConfiguration Video PIP Presentation DefaultValue Position: Current

xConfiguration Video Layout LocalLayoutFamily

Select which video layout family to use locally.

Requires user role: ADMIN

Value space: <Auto/FullScreen/Equal/PresentationSmallSpeaker/PresentationLargeSpeaker/ Prominent/Overlay/Single>

Auto: The default layout family, as given in the layout database provided by the system, will be used as the local layout.

FullScreen: The FullScreen layout family will be used as the local layout. It means that the active speaker or presentation will be shown in full screen. Using this value is not recommended as from TC6.0.

Equal: The Equal layout family will be used as the local layout. All videos have equal size, as long as there is space enough on the screen.

PresentationSmallSpeaker: The PresentationSmallSpeaker layout family will be used as the local layout. Using this value is not recommended as from TC6.0.

PresentationLargeSpeaker: The PresentationLargeSpeaker layout family will be used as the local layout. Using this value is not recommended as from TC6.0.

Prominent: The Prominent layout family will be used as the local layout. The active speaker, or the presentation if present, will be a large picture, while the other participants will be small pictures. Transitions between active speakers are voice switched.

Overlay: The Overlay layout family will be used as the local layout. The active speaker, or the presentation if present, will be shown in full screen, while the other participants will be small pictures-in-picture (PiP). Transitions between active speakers are voice switched.

Single: The active speaker, or the presentation if present, will be shown in full screen. The other participants are not shown. Transitions between active speakers are voice switched.

Example: xConfiguration Video Layout LocalLayoutFamily: Auto

xConfiguration Video Layout RemoteLayoutFamily

Select which video layout family to be used for the remote participants.

Requires user role: ADMIN

Value space: <Auto/FullScreen/Equal/PresentationSmallSpeaker/PresentationLargeSpeaker/ Prominent/Overlay/Single>

Auto: The default layout family, as given by the local layout database, will be used as the remote layout.

FullScreen: The FullScreen layout family will be used as the remote layout. It means that the active speaker or presentation will be shown in full screen. It is recommended not to use this value as from TC6.0.

Equal: The Equal layout family will be used as the remote layout. All videos have equal size, as long as there is space enough on the screen.

PresentationSmallSpeaker: The PresentationSmallSpeaker layout family will be used as the remote layout. Using this value is not recommended as from TC6.0.

PresentationLargeSpeaker: The PresentationLargeSpeaker layout family will be used as the remote layout. Using this value is not recommended as from TC6.0.

Prominent: The Prominent layout family will be used as the remote layout. The active speaker, or the presentation if present, will be a large picture, while the other participants will be small pictures. Transitions between active speakers are voice switched.

Overlay: The Overlay layout family will be used as the remote layout. The active speaker, or the presentation if present, will be shown in full screen, while the other participants will be small pictures-in-picture (PiP). Transitions between active speakers are voice switched.

Single: The active speaker, or the presentation if present, will be shown in full screen. The other participants are not shown. Transitions between active speakers are voice switched.

Example: xConfiguration Video Layout RemoteLayoutFamily: Auto

xConfiguration Video Monitors

Set the monitor layout mode.

Requires user role: ADMIN

Value space: <Single/Dual/DualPresentationOnly>

Single: The same layout is shown on all monitors.

Dual: The layout is distributed on two monitors.

DualPresentationOnly: All participants in the call will be shown on the first monitor, while the presentation (if any) will be shown on the second monitor.

Example: xConfiguration Video Monitors: Single

xConfiguration Video OSD Mode

The Video OSD (On Screen Display) Mode lets you define if information and icons should be displayed on screen.

Requires user role: ADMIN

Value space: <Off/On>

Off: Hide the on screen menus, icons and indicators.

On: Display the on screen menus, icons and indicators.

Example: xConfiguration Video OSD Mode: On

xConfiguration Video OSD WallPaperSelection

Applies to the on screen menu (OSD). The wall paper selection menu can be hidden outside the administrator settings menu and visible inside, or it can be visible both inside and outside the administrator settings menu. The administrator settings menu can be password protected.

Requires user role: ADMIN

Value space: <Off/On>

Example: xConfiguration Video OSD WallPaperSelection: On

xConfiguration Video OSD LanguageSelection

Applies to the on screen menu (OSD). The language selection menu can be hidden outside the administrator settings menu and visible inside, or it can be visible both inside and outside the administrator settings menu. The administrator settings menu can be password protected.

Requires user role: ADMIN

Value space: <Off/On>

Off: The language selection menu is hidden outside the administrator settings menu and visible inside. To change the language, navigate to: Home > Settings > Administrator Settings. If required log in with your username and password. Then navigate to Language.

On: The language selection menu is visible both inside and outside the administrator settings menu. To change the language, navigate to: Home > Settings > Language. Or navigate to Home > Settings > Administrator Settings. If required log in with your username and password. Then navigate to Language.

Example: xConfiguration Video OSD LanguageSelection: On

xConfiguration Video OSD MenuStartupMode

Configures the state of the OSD (On Screen Display) menu after a video system / codec boot. The setting only applies when the video system is operated by a remote control and the on-screen menu.

Requires user role: ADMIN

Value space: <Closed/Home>

Closed: The OSD menu will NOT expand automatically. This setting is recommended for 3rd party integrations that need full control of what is shown on the OSD.

Home: The OSD menu will show the home menu expanded.

Example: xConfiguration Video OSD MenyStartUpMode: Home

xConfiguration Video OSD VirtualKeyboard

Determine whether or not the virtual keyboard will automatically show on screen when text is to be entered in an input field. The setting only applies when the video system is operated by a remote control and the on-screen menu.

Requires user role: ADMIN

Value space: <UserSelectable/AlwaysOn>

UserSelectable: The user has to press a softbutton to open or close the virtual keyboard.

AlwaysOn: The virtual keyboard is automatically shown on screen as long as text can be entered in an input field.

Example: xConfiguration Video OSD VirtualKeyboard: UserSelectable

xConfiguration Video OSD EncryptionIndicator

Define for how long the encryption indicator (a padlock) will be shown on screen. The setting applies to both encrypted and non-encrypted calls, i.e. both to secure and non-secure conferences. The icon for encrypted calls is a locked padlock, and the icon for non-encrypted calls is a crossed out locked padlock.

Requires user role: ADMIN

Value space: <Auto/AlwaysOn/AlwaysOff>

Auto: If the Conference Encryption Mode setting is set to BestEffort and the call is encrypted, the encryption indicator is shown during the first seconds of a call. If the Conference Encryption Mode setting is set to BestEffort and the call is non-encrypted, the crossed out encryption indicator is shown during the entire call. If the Conference Encryption Mode setting is NOT set to BestEffort, the encryption indicator is not shown at all.

AlwaysOn: The encryption indicator is displayed on screen during the entire call. This applies to both encrypted and non-encrypted calls for all Conference Encryption Mode settings.

AlwaysOff: The encryption indicator is never displayed on screen. This applies to both encrypted and non-encrypted calls for all Conference Encryption Mode settings.

Example: xConfiguration Video OSD EncryptionIndicator: Auto

xConfiguration Video OSD MissedCallsNotification

Determine whether or not the OSD (On Screen Display) shall display a missed calls notification dialog box if there have been incoming calls that have not been answered. The setting only applies when the video system is operated by a remote control and the on-screen menu. When using a Touch controller the notification dialog box will appear on the Touch display, and not on the OSD.

Requires user role: ADMIN

Value space: <Off/On>

Off: The OSD will NOT show any indication that there have been any missed calls. This setting is recommended for 3rd party integrations that need full control of what is shown on the OSD.

On: The OSD will show a notification of missed calls.

Example: `xConfiguration Video OSD MissedCallsNotifications: On`

xConfiguration Video OSD AutoSelectPresentationSource

Determine if the presentation source should be automatically selected.

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable automatic selection of the presentation source.

On: Enable automatic selection of the presentation source.

Example: `xConfiguration Video OSD AutoSelectPresentationSource: Off`

xConfiguration Video OSD TodaysBookings

This setting can be used to display the system's bookings for today on the main OSD menu. This requires that the system is bookable by an external booking system, like Cisco TelePresence Management Suite (TMS).

Requires user role: ADMIN

Value space: <Off/On>

Off: Do not display todays bookings.

On: Displays information about this systems bookings on screen.

Example: `xConfiguration Video OSD TodaysBookings: Off`

xConfiguration Video OSD MyContactsExpanded

Set how the local contacts will be displayed in the phone book dialog in the OSD (On Screen Display).

Requires user role: ADMIN

Value space: <Off/On>

Off: The local contacts will be placed in a separate folder called MyContacts in the phonebook dialog.

On: The local contacts in the phone book will be shown in the top level of the phonebook dialog.

Example: `xConfiguration Video OSD MyContactsExpanded: Off`

xConfiguration Video OSD Output

The Video OSD (On Screen Display) Output lets you define which monitor should display the on screen menus, information and icons. By default the OSD is sent to the monitor connected to the Video OSD Output 1. If you cannot see the OSD on screen, then you must re-configure the OSD Output. You can do this by entering a key sequence on the remote control, from the web interface, or by a command line interface.

Using the remote control: Press the Disconnect key followed by: * # * # 0 x # (where x is output 1 to 2).

Using the web interface: Open a web browser and enter the IP address of the codec. Open the Advanced Configuration menu and navigate to Video OSD Output and select the video output.

Using a command line interface: Open a command line interface and connect to the codec (if in doubt of how to do this, see the API Guide for the codec). Enter the command: xConfiguration Video OSD Output [1..2] (select the OSD Output)

Requires user role: ADMIN

Value space: <1/2>

Range: Select 1 for HDMI output, or select 2 for DVI-I output.

Example: `xConfiguration Video OSD Output: 1`

xConfiguration Video OSD InputMethod InputLanguage

The codec can be enabled for Cyrillic input characters in the menus on screen. NOTE: Requires that xConfiguration Video OSD inputMethod Cyrillic is set to On.

Requires user role: ADMIN

Value space: <Latin/Cyrillic>

Latin: Latin characters can be entered when using the remote control (default).

Cyrillic: Cyrillic characters can be entered using the remote control. NOTE: Requires a Cisco TelePresence Remote Control with Cyrillic fonts.

Example: `xConfiguration Video OSD InputMethod InputLanguage: Latin`

xConfiguration Video OSD InputMethod Cyrillic

Set the Cyrillic mode for the menu input language in the menus on screen.

Requires user role: ADMIN

Value space: <Off/On>

Off: Cyrillic mode is NOT available as a menu input language in the menus on screen.

On: Cyrillic mode is available as a menu input language in the menus on screen. This will enable the setting Video OSD InputMethod InputLanguage.

Example: `xConfiguration Video OSD InputMethod Cyrillic: Off`

xConfiguration Video OSD LoginRequired

Determine if the system should require the user to login before accessing the On Screen Display (OSD). If enabled, the user must enter his username and his PIN. After the user has logged in he can only execute to the configurations changes and commands allowed by his Role.

Requires user role: ADMIN

Value space: <Off/On>

Off: No login to the OSD is required.

On: The user must log in to access the On Screen Display (OSD).

Example: `xConfiguration Video OSD LoginRequired: Off`

xConfiguration Video AllowWebSnapshots

Allow or disallow snapshots being taken of the local input sources, remote sites and presentation channel. If allowed, the web interface Call Control page will show snapshots both when idle and in a call.

NOTE: This feature is disabled by default, and must be enabled from the On Screen Display (OSD), from a directly connected Touch controller, or via the codec's serial port (USB port and RS-232 adapter).

Requires user role: ADMIN

Value space: <Off/On>

Off: Capturing web snapshots is not allowed.

On: Web snapshots can be captured and displayed on the web interface.

Example: `xConfiguration Video AllowWebSnapshots: Off`

xConfiguration Video Output HDMI [1,2] RGBQuantizationRange

All devices with HDMI outputs should follow the rules for RGB video quantization range defined in CEA-861. Unfortunately some devices do not follow the standard and this configuration may be used to override the settings to get a perfect image with any display. The default value is set to Full because most HDMI displays expects full quantization range.

Requires user role: ADMIN

Value space: <Auto/Full/Limited>

Auto: RGB quantization range is automatically selected based on the RGB Quantization Range bits (Q0, Q1) in the AVI infoframe. If no AVI infoframe is available, RGB quantization range is selected based on video format according to CEA-861-E.

Full: Full quantization range. The R, G, B quantization range includes all code values (0 - 255). This is defined in CEA-861-E.

Limited: Limited Quantization Range. R, G, B quantization range that excludes some code values at the extremes (16 - 235). This is defined in CEA-861-E.

Example: `xConfiguration Video Output HDMI 1 RGBQuantizationRange: Full`

xConfiguration Video Output HDMI [1,2] CEC Mode

The HDMI outputs support Consumer Electronics Control (CEC). When set to on (default is off), and the monitor connected to the HDMI output is CEC compatible and CEC is configured, the system will use CEC to set the monitor in standby when the system enters standby. Likewise the system will wake up the monitor when the system wakes up from standby. Please note that the different manufacturers uses different marketing names for CEC, for example Anynet+ (Samsung); Aquos Link (Sharp); BRAVIA Sync (Sony); HDMI-CEC (Hitachi); Kuro Link (Pioneer); CE-Link and Regza Link (Toshiba); RIHD (Onkyo); HDAVI Control, EZ-Sync, VIERA Link (Panasonic); EasyLink (Philips); and NetCommand for HDMI (Mitsubishi).

Requires user role: ADMIN

Value space: <Off/On>

Off: Disable CEC control.

On: Enable CEC control.

Example: `xConfiguration Video Output HDMI 1 CEC Mode: Off`

xConfiguration Video Output HDMI [1,2] MonitorRole

The HDMI monitor role describes which video stream will be shown on the monitor connected to the video output HDMI connector.

Requires user role: ADMIN

Value space: <First/Second/PresentationOnly>

First/Second: Define the role of the monitor in a multi-monitor setup. In a single-monitor setup, there is no difference between First and Second.

PresentationOnly: Show presentation video stream if active, and nothing else.

Example: xConfiguration Video Output HDMI 1 MonitorRole: First

xConfiguration Video Output HDMI [1,2] OverscanLevel

Some TVs or other monitors may not display the whole image sent out on the systems video output, but cuts the outer parts of the image. In this case this setting can be used to let the system not use the outer parts of video resolution. Both the video and the OSD menu will be scaled in this case.

Requires user role: ADMIN

Value space: <Medium/High/None>

Medium: The system will not use the outer 3% of the output resolution.

High: The system will not use the outer 6% of the output resolution

None: The system will use all of the output resolution.

Example: xConfiguration Video Output HDMI 1 OverscanLevel: None

xConfiguration Video Output HDMI [1,2] Resolution

Select the preferred resolution for the monitor connected to the video output HDMI connector. This will force the resolution on the monitor.

Requires user role: ADMIN

Value space: <Auto/1024_768_60/1280_1024_60/1280_720_60/1920_1080_60/1280_768_60/1360_768_60/1366_768_60>

Auto: The system will automatically try to set the optimal resolution based on negotiation with the connected monitor.

Range: 1024x768@60p, 1280x1024@60p, 1280x720@60p, 1920x1080@60p, 1280x768@60p, 1360x768@60p, 1366x768@60p

Example: xConfiguration Video Output HDMI 1 Resolution: Auto

xConfiguration Video WallPaper

Select a background image (wallpaper) for the video screen when idle.

Requires user role: USER

Value space: <None/Custom/Growing/Summersky/Waves>

None: There is no background image on the screen, i.e. the background is black.

Custom: Use the custom wallpaper that is stored on the system as background image on the screen. As default, there is no custom wallpaper stored and the background will be black. You can upload a custom wallpaper to the system using the web interface. The maximum supported resolution is 1920x1200.

Summersky, Growing, Waves: The chosen background image is shown on the screen.

Example: xConfiguration Video Wallpaper: Summersky

Experimental configuration

The Experimental settings are for testing only and should not be used unless agreed with Cisco.
These settings are not documented and WILL change in later releases.

Chapter 4

xCommand commands

Description of the xCommand commands

In this chapter, you can find a complete list of all xCommand type commands with parameters.

We recommend you visit our web site regularly for updated versions of the manual.
Go to: ▸ http://www.cisco.com/go/quickset-docs

Audio commands

xCommand Audio Microphones Mute

Mute all microphones.

Requires user role: USER

Example:
```
xCommand Audio Microphones Mute
*r AudioMicrophonesMuteResult (status=OK):
** end
```

xCommand Audio Microphones Unmute

Unmute microphones.

Requires user role: USER

Example:
```
xCommand Audio Microphones Unmute
*r AudioMicrophonesUnmuteResult (status=OK):
** end
```

xCommand Audio Sound Play

Play a specified audio sound.

Requires user role: USER

Parameters:
Sound(r): <Busy/CallWaiting/Dial/KeyTone/Ringing/SpecialInfo/TelephoneCall/VideoCall>
Loop: <On/Off>

Example:
```
xCommand Audio Sound Play Sound: Ringing
*r AudioSoundPlayResult (status=OK):
** end
```

xCommand Audio Sound Stop

Stop playing audio sound.

Requires user role: USER

Example:
```
xCommand Audio Sound Stop
*r AudioSoundStopResult (status=OK):
** end
```

Bookings commands

xCommand Bookings List

List the stored bookings for the system. The list of booking details is received from the management system. All parameters are optional, and can be used to limit the search result.

If no parameters are used, past, present and future bookings are all listed. To avoid listing bookings from yesterday and before, use DayOffset = 0.

Days: Number of days to retrieve bookings from.

DayOffset: Which day to start the search from (today: 0, tomorrow: 1…).

Limit: Max number of bookings to list.

Offset: Offset number of bookings for this search.

Requires user role: USER

Parameters:

Days: <1..365>

DayOffset: <0..365>

Limit: <1..65534>

Offset: <0..65534>

Example:

```
Example 1 (One booking in list)
xCommand Bookings List

OK
*r Bookings ResultInfo TotalRows: 1
*r Bookings LastUpdated: "2011-09-02T11:19:01Z"
*r Bookings Booking 1 Id: "273"
*r Bookings Booking 1 Title: "Sales meeting"
*r Bookings Booking 1 Agenda: "Describe this command"
*r Bookings Booking 1 Privacy: Public
*r Bookings Booking 1 Organizer FirstName: "Ola"
*r Bookings Booking 1 Organizer LastName: "Normann"
*r Bookings Booking 1 Organizer Email: "ola.normann@domain.com"
*r Bookings Booking 1 Time StartTime: "2011-09-02T13:00:00Z"
*r Bookings Booking 1 Time StartTimeBuffer: 600
*r Bookings Booking 1 Time EndTime: "2011-09-02T13:30:00Z"
*r Bookings Booking 1 Time EndTimeBuffer: 0
*r Bookings Booking 1 MaximumMeetingExtension: 30
*r Bookings Booking 1 MeetingExtensionAvailability: Guaranteed
*r Bookings Booking 1 BookingStatus: OK
```

```
*r Bookings Booking 1 BookingStatusMessage: ""
*r Bookings Booking 1 Webex Enabled: True
*r Bookings Booking 1 Webex Url: "http://webex.url"
*r Bookings Booking 1 Webex MeetingNumber: "webexNumber@cisco.com"
*r Bookings Booking 1 Webex Password: ""
*r Bookings Booking 1 Webex HostKey: ""
*r Bookings Booking 1 Webex DialInNumbers DialInNumber 1 Type: TollFree
*r Bookings Booking 1 Webex DialInNumbers DialInNumber 1 Number: "+1 987 654321"
*r Bookings Booking 1 Webex DialInNumbers DialInNumber 2 Type: Toll
*r Bookings Booking 1 Webex DialInNumbers DialInNumber 2 Number: "+1 987 654322"
*r Bookings Booking 1 InteropBridge Number: ""
*r Bookings Booking 1 InteropBridge ConferenceId: ""
*r Bookings Booking 1 ManualCallIn Number: ""
*r Bookings Booking 1 ManualCallIn ConferenceId: ""
*r Bookings Booking 1 ManualCallIn ConferencePassword: ""
*r Bookings Booking 1 Encryption: BestEffort
*r Bookings Booking 1 Role: Slave
*r Bookings Booking 1 Recording: Disabled
*r Bookings Booking 1 DialInfo Calls Call 1 Number: "91123456;conference-id=2100170569"
*r Bookings Booking 1 DialInfo Calls Call 1 Protocol: SIP
*r Bookings Booking 1 DialInfo Calls Call 1 CallRate: 3000
*r Bookings Booking 1 DialInfo Calls Call 1 CallType: Video
*r Bookings Booking 1 DialInfo ConnectMode: OBTP
** end
```

```
Example 2 (When no bookings have been received, or after all bookings have been removed by the command xCommand Bookings Clear)
xCommand Bookings List
OK
*r Bookings Error: "No bookings found."
*r Bookings ResultInfo TotalRows: 0
*r Bookings LastUpdated: Never
** end
```

xCommand Bookings Clear

Clear the current stored list of bookings.

Requires user role: USER

Example:

```
xCommand Bookings Clear
** end
OK
```

Boot commands

xCommand Boot

Reboot system.

Action: As a default the system restarts after a reboot. By selecting Shutdown, the system will not restart.

Requires user role: USER

Parameters:

Action: <Restart/Shutdown>

Example:

```
xCommand Boot
*r BootResult (status=OK):
** end
OK
CUIL reboot request, restarting
Connection closed by foreign host.
```

80

Call commands

xCommand Call Accept

Accept an incoming call. If no CallId is specified, all incoming calls are accepted. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Requires user role: USER

Parameters:

    CallId: <0..65534>

    CallType(o): <Audio/Video>

Example:

```
xCommand Call Accept CallId:19

OK

*r CallAcceptResult (status=OK):

** end
```

xCommand Call Reject

Reject incoming call. If no call id is specified, all incoming calls are rejected. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Requires user role: USER

Parameters:

    CallId: <0..65534>

Example:

```
xCommand Call Reject CallId:20

OK

*r CallRejectResult (status=OK):

** end
```

xCommand Call Disconnect

Disconnect a call. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Requires user role: USER

Parameters:

    CallId(r): <0..65534>

Example:

```
xCommand Call Disconnect CallId:17

OK

*r DisconnectCallResult (status=OK):

** end
```

xCommand Call DisconnectAll

Disconnect all active calls.

Requires user role: USER

Example:

```
xCommand Call DisconnectAll

OK

*r DisconnectAllResult (status=OK):

** end
```

xCommand Call Hold

Put a call on hold. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Requires user role: USER

Parameters:

    CallId(r): <0..65534>

    Reason(o): <Conference/Transfer/Other>

Example:

```
xCommand Call Hold CallId:19

OK

*r CallHoldResult (status=OK):

** end
```

xCommand Call Join

Join all existing calls, active and on hold. For this command to work, you have to configure Multiway™ first, see xConfiguration NetworkServices MultiWay Address and xConfiguration Conference 1 Multipoint Mode.

Requires user role: USER

Parameters:

CallId(r): <0..65534>

Example:

```
xCommand Call Join

OK

*r CallJoinResult (status=OK):

** end
```

xCommand Call Resume

Resume a call that have been put on hold. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Requires user role: USER

Parameters:

CallId(r): <0..65534>

Example:

```
xCommand Call Resume CallId:19

OK

*r CallResumeResult (status=OK):

** end
```

xCommand Call ExtendConference

Request to extend the conference for the call with the specified CallId.

For this command to apply, the following are required:

1) The result from the xStatus Conference Site <CallId> ConferenceExtended must be NotExtended: *s Conference Site <CallId> ConferenceExtended: NotExtended

2) The following tag from the bookings list command must have a value > 0:

*r Bookings Booking <bookingNumber> MaximumMeetingExtension: "30"

Note: The command returns the result from the command itself, not the result from the extension request to far end. A separate event returns the result from the extension request.

CallId: CallId for the call in question.

Requires user role: USER

Parameters:

CallId(r): <0..65534>

Example:

```
xCommand Call ExtendConference CallId: 1

OK

*r CallExtendConferenceResult (status=OK)

** end
```

xCommand Call UnattendedTransfer

Transfers an ongoing call to another participant. The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId. Fully supported for SIP calls only.

Requires user role: USER

Parameters:

CallId(r): <0..65534>
Number(r): <S: 0, 255>

Example:

```
xCommand Call UnattendedTransfer CallId: 2 Number: destination@example.
com

*r TransferUriResult (status=OK):

** end
```

xCommand Call HoldAll

Puts all ongoing calls on hold.

Requires user role: USER

Parameters:

Reason(0): <Conference/Transfer/Other>

Example:

```
xCommand Call HoldAll
```

xCommand Call Ignore

Turns off the ringtone for the incoming call. The call can still be answered.

Requires user role: USER

Parameters:

CallId(r): <0..65534>

Example:

```
xCommand Call Ignore CallId: 22
*r IgnoreResult (status=OK):
** end
```

xCommand Call Modify

Modifies the call type of the current call, to either an audio call or a video call. Not in use in this software version.

Requires user role: USER

Parameters:

CallId(r): <0..65534>
CallType(r): <Audio/Video>

Example:

```
xCommand Call Modify CallType: Audio CallId: 16
*r CallModifyResult (status=Error):
   Reason: Callescalation not enabled
** end
```

CallLog commands

xCommand CallLog Clear

Clear call logs stored in the system. If a logtag is given as argument, that specific call is deleted from the logs. If no logtag is given the system deletes all call logs. The LogTag values for the calls are found by issuing the xHistory CallLog Call command.

Requires user role: USER

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Clear
*r ClearResult (status=OK):
** end
```

xCommand CallLog Recent Delete

Delete the call log of recent calls. If a logtag is given as argument, that specific call is deleted from the log. If no logtag is given the system deletes all call logs. The LogTag values for recent calls are found by issuing the xHistory CallLog Recent command.

Requires user role: USER

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Recent Delete
*r DeleteResult (status=OK):
** end
```

xCommand CallLog Outgoing Delete

Delete the call log of outgoing calls. If a logtag is given as argument, that specific call is deleted from the log. If no logtag is given the system deletes the complete outgoing calls log. The LogTag values for outgoing calls are found by issuing the xHistory CallLog Outgoing command.

Requires user role: USER

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Outgoing Delete LogTag:202

*r DeleteResult (status=OK):

** end
```

xCommand CallLog Received Delete

Delete the call log of received calls. If a logtag is given as argument, that specific call is deleted from the log. If no logtag is given the system deletes the complete received calls log. The LogTag values for received calls are found by issuing the xHistory CallLog Received command.

Requires user role: USER

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Received Delete LogTag:126

*r DeleteResult (status=OK):

** end
```

xCommand CallLog Missed Delete

Delete the call log of missed calls. If a logtag is given as argument, that specific call is deleted from the log. If no logtag is given the system deletes the complete missed calls log. The LogTag values for missed calls are found by issuing the xHistory CallLog Missed command.

Requires user role: USER

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Missed Delete LogTag:119

*r DeleteResult (status=OK):

** end
```

xCommand CallLog Missed Dismiss

Review the call log of dismissed calls. If a logtag is given as argument, that specific call is deleted from the log. If no logtag is given the system deletes the complete missed calls log. The LogTag values for missed calls are found by issuing the xHistory CallLog Missed command.

Requires user role: USER

Parameters:

LogTag: <0..2147483647>

Example:

```
xCommand CallLog Missed Dimissed LogTag:119

*r DismissResult (status=OK):

** end
```

CamCtrlPip commands

xCommand CamCtrlPip

Show or hide the camera self-view in a small window (picture in picture).

Mode: Select whether the self-view picture in picture is on or off.

Duration: Set how long the self-view is shown for on screen.

Requires user role: USER

Parameters:

Mode(r): <On/Off>

Duration: <0..60>

Example:

```
xCommand CamCtrlPip Mode: On
*r CamCtrlPipResult (status=OK):
** end
```

Camera commands

xCommand Camera PanTiltReset

The camera is reset to its default values for pan and tilt. If the camera is daisy chained, the CameraId is given by its place in the chain.

Requires user role: USER

Parameters:

CameraId(r): <1..1>

Example:

```
xCommand Camera PanTiltReset CameraId:1
OK
*r PanTiltResetResult (status=OK):
** end
```

xCommand Camera PositionReset

Reset the camera position the the default position.

Requires user role: USER

Parameters:

CameraId(r): <1..1>

Example:

```
xCommand Camera PositionReset CameraId:1
OK
*r CameraPositionResetResult (status=OK):
** end
```

xCommand Camera PositionSet

Position the camera by defining the pan, tilt, zoom and focus parameters. If the camera is placed in a daisy chain you need to know the CameraId for the camera you want to address.

Requires user role: USER

Parameters:

CameraId(r): <1..1>

Pan: <-65535..65535>

Tilt: <-65535..65535>

Zoom: <0..65535>

Focus: <0..65535>

Example:

```
xCommand Camera PositionSet CameraId:1 Pan:200 Tilt:300
OK
*r CameraPositionSetResult (status=OK):
** end
```

xCommand Camera Ramp

Move the camera in a specified direction. The camera moves at specified speed until a stop command is issued. In a daisy chain, you need to know the CameraId for the camera you want to address. Be aware that pan and tilt can be operated simultaneously, but no other combinations. In the latter case only the first operation specified is executed. For example, if you try to run both zoom and pan at the same time, only zoom is executed.

NOTE: You must run a stop command to stop the camera, see the example below.

CameraId: Give the camera id.

Pan: Move the camera to the Left or Right, followed by Stop.

PanSpeed: Set the pan speed.

Tilt: Move the camera Up or Down, followed by Stop.

TiltSpeed: Set the tilt speed.

Zoom: Zoom the camera In or Out, followed by Stop.

ZoomSpeed: Set the zoom speed.

Focus: Focus the camera Far or Near, followed by Stop.

Requires user role: USER

Parameters:

CameraId(r): <1..1>

Pan: <Left/Right/Stop>

PanSpeed: <1..15>

Tilt: <Down/Up/Stop>

TiltSpeed: <1..15>

Zoom: <In/Out/Stop>

ZoomSpeed: <1..15>

Focus: <Far/Near/Stop>

Example:

```
xCommand Camera Ramp CameraId:1 Pan:left PanSpeed:1
OK
*r RampResult (status=OK):
** end
xCommand Camera Ramp CameraId:1 Pan:stop
OK
*r RampResult (status=OK):
** end
```

xCommand Camera TriggerAutofocus

Trigger the auto-focus functionality. The camera must support auto-focus functionality. If the camera is daisy chained, the CameraId is given by its place in the chain.

Requires user role: USER

Parameters:

 CameraId(r): <1..1>

Example:

```
xCommand Camera TriggerAutofocus CameraId:1

OK

*r TriggerAutofocusResult (status=OK):

** end
```

xCommand Camera PositionActivateFromPreset

Selects pan, tilt, zoom and focus parameters for the given camera id from the selected preset.

CameraId: Give the camera id.

PresetId: Select preset 1 to 15.

Requires user role: USER

Parameters:

 CameraId(r): <1..1>
 PresetId(r): <1..15>

Example:

```
xCommand Camera PositionActivateFromPreset CameraId:1 PresetID:1

OK

*r PositionActivateFromPresetResult (status=OK):

** end
```

xCommand Camera Preset Activate

Activate one of the stored camera preset.

Note: The xCommand Camera Preset commands applies to an individual camera. This is in contrast to the xCommand Preset commands where a single preset covers ALL connected cameras plus the Video Input switcher settings.

PresetId: The ID of the camera preset you want to activate.

Requires user role: USER

Parameters:

 PresetId(r): <1..35>

Example:

```
xCommand Camera Preset Activate PresetId: 1

OK

*r CameraPresetActivateResult (status=OK)

** end
```

xCommand Camera Preset Edit

Edit a stored camera preset. You can change the name of the camera preset and its position in the list that is returned by the xCommand Camera Preset List command.

Note: The xCommand Camera Preset commands applies to an individual camera. This is in contrast to the xCommand Preset commands where a single preset covers ALL connected cameras plus the Video Input switcher settings.

PresetId: The ID of the camera preset you want to edit.

ListPosition: The position in the list returned by the xCommand Camera Preset List command.

Name: The name of the camera preset. It will be used in the list returned by the xCommand Camera Preset List command.

Requires user role: ADMIN

Parameters:

 PresetId(r): <1..35>
 ListPosition: <1..35>
 Name: <S: 0, 255>

Example:

```
xCommand Camera Preset Edit PresetId: 1 ListPosition: 1 Name: ""

OK

*r CameraPresetEditResult (status=OK)

** end
```

xCommand Camera Preset List

List information about available camera presets.

Note: The xCommand Camera Preset commands are used to store camera positions for individual cameras. This is in contrast to the xCommand Preset commands where a single preset stores/ recalls ALL connected cameras plus the Video Input switcher settings. This makes it more usable for integrations where one wants to store multiple camera positions individually per camera rather than a complete camera position set.

PresetId: Filter on specified preset.

Requires user role: USER

Parameters:

PresetId: <1..35>

Example:

```
xCommand Camera Preset List PresetId: 1

OK

*r CameraPresetListResult (status=OK)

** end
```

xCommand Camera Preset Remove

Remove a camera preset.

Note: The xCommand Camera Preset commands applies to an individual camera. This is in contrast to the xCommand Preset commands where a single preset covers ALL connected cameras plus the Video Input switcher settings.

PresetId: The ID of the camera preset you want to remove.

Requires user role: USER

Parameters:

PresetId(r): <1..35>

Example:

```
xCommand Camera Preset Remove PresetId: 1

OK

*r CameraPresetRemoveResult (status=OK)

** end
```

xCommand Camera Preset Store

Store the current position of the chosen camera. The camera is identified by the CameraId parameter.

Note: The xCommand Camera Preset commands applies to an individual camera. This is in contrast to the xCommand Preset commands where a single preset covers ALL connected cameras plus the Video Input switcher settings. The xCommand Camera Preset commands are useful when you want to handle multiple camera positions individually per camera, rather than working with complete sets of camera positions.

PresetId: The ID of this camera preset. If not set explicitly, the codec will assign a preset ID automatically.

CameraId: Which camera to store the position of.

ListPosition: The new camera preset's position in the list returned by the xCommand Camera Preset List command.

Name: The name of the new camera preset. It will be used in the list returned by the xCommand Camera Preset List command.

Requires user role: ADMIN

Parameters:

PresetId: <1..35>
CameraId (r): <1..1>
ListPosition: <1..35>
Name: <S: 0, 255>

Example:

```
xCommand Camera Preset Store PresetId: 1 CameraId: 1 ListPosition: 1 Name
""

OK

*r CameraPresetStoreResult (status=OK)

** end
```

Conference commands

xCommand Conference DoNotDisturb Activate

This command switches on the Do Not Disturb mode, and the Timeout parameter allows you to control when it is switched off again. It only takes effect when xConfiguration Conference DoNotDisturb Mode is set to Timed.

When Do Not Disturb is switched on, all incoming calls are rejected and registered as missed calls. The calling side receives a busy signal.

Timeout: The number of minutes before Do Not Disturb is switched off. If not set, Do Not Disturb times out after 1440 minutes (24 hours).

Requires user role: USER

Parameters:

Timeout: <0..1440>

Example:

```
xCommand Conference DoNotDisturb Activate
*r ActivateResult:
** end
```

xCommand Conference DoNotDisturb Deactivate

This command switches off the Do Not Disturb mode. This command only takes effect when xConfiguration Conference DoNotDisturb Mode is set to Timed. When Do Not Disturb is switched off incoming calls come through as normal.

Requires user role: USER

Example:

```
xCommand Conference DoNotDisturb Deactivate
*r DeactivateResult:
** end
```

xCommand Conference ActiveSpeaker Reset

Resets ActiveSpeaker. Normal voice switching is switched back on.

Requires user role: USER

Example:

```
xCommand Conference ActiveSpeaker Reset
*r ConferenceActiveSpeakerResetResult (status=OK):
** end
```

xCommand Conference ActiveSpeaker Set

For manually setting the active speaker in a conference. Overrides normal voice switching.

Target: Identifies local or remote participant.

CallId: Identifies CallID for remote participant. Only relevant if Target is set to "remote".

Requires user role: USER

Parameters:

Target(r): <local/remote<

CallId: <0..65534>

Example:

```
Example 1
xCommand Conference ActiveSpeaker Set Target: local
*r ConferenceActiveSpeakerSetResult (status=OK):
** end


Example 2
xCommand Conference ActiveSpeaker Set Target: remote CallID: 3
*r ConferenceActiveSpeakerSetResult (status=OK):
** end
```

Dial commands

xCommand Dial

Dial out from the system. Returns information about the CallId and ConferenceId, which are required for some other commands.

Number: Enter the number or address.

Protocol: Select the SIP, H323 or H320 protocol.

CallRate: Set a call rate.

CallType: Select the audio or video call type.

BookingId: Any identifier that an external booking system (e.g. TMS, CTS-MAN) can use for its own references to match placed calls with the booking systems internal identifier for a meeting. This can be any string, e.g. a GUID. The booking Id is supplied in call logs, call events etc for the call.

Requires user role: USER

Parameters:

    Number(r): <S: 0, 255>

    Protocol: <H320/H323/Sip>

    CallRate: <64..6000>

    CallType: <Audio/Video>

    BookingId: <S: 0, 255>

    Appearance(o): <1..999999999>

Example:

```
xCommand Dial Number:543210 Protocol:h323

OK

*r DialResult (status=OK):

CallId: 2

ConferenceId: 1

** end
```

DTMFSend commands

xCommand DTMFSend

Send DTMF tones to the far end.

CallId: The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

DTMFString: Enter the DTMF string.

Requires user role: USER

Parameters:

    CallId: <0..65534>

    DTMFString(r): <S: 0, 32>

Example:

```
xCommand DTMFSend CallId:2 DTMFString:1234

*r DTMFSendResult (status=OK):

** end
```

FacilityService commands

xCommand FacilityService Dial

Dial out from the system to a facility service. A maximum of five facility services can be defined; which one of these five to dial to is identified by the ServiceId (ref. xConfiguration Facility Service [ServiceId] Type/Name/Number/CallType). The command returns information about the CallId and ConferenceId.

ServiceId: The identifier of the facility service.

Requires user role: USER

Parameters:

ServiceId(r): <1..5>

Example:

```
xCommand FacilityService Dial ServiceId: 1
OK
*r FacilityServiceDialResult (status=OK):
    CallId: 2
    ConferenceId: 1
** end
```

FarEndControl commands

xCommand FarEndControl Camera Move

Move the far end camera (the remote camera). NOTE: The far end camera moves in the specified direction until the stop command (ref: xCommand FarEndControl Camera Stop) is issued

CallId: The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Value: Select the action for how to move the camera.

Requires user role: USER

Parameters:

CallId: <0..65534>
Value(r): <Left/Right/Up/Down/ZoomIn/ZoomOut>

Example:

```
xCommand FarEndControl Camera Move CallId:3 Value:left
*r FECCMoveResult (status=OK):
** end
```

xCommand FarEndControl Camera Stop

Stop the far end camera after the xCommand FarEndControl Camera Move has been issued.

CallId: The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

Requires user role: USER

Parameters:

CallId: <0..65534>

Example:

```
xCommand FarEndControl Camera Stop CallId:3
*r FECCMoveResult (status=OK):
** end
```

xCommand FarEndControl Preset Activate

While in a call, this command is used to activate a preset on the far end codec. The preset covers the far end codec's camera positions and input video switcher settings.

The preset must be stored on the far end codec beforehand, either by using the xCommand Preset Store command locally on the far end codec, or by using the xCommand FarEndControl Preset Store command from a remote codec.

Note: The far end codec's xConfiguration Conference FarEndControl Mode setting must be switched On for the FarEndControl commands to work.

CallId: The CallId is required to identify the far end codec only when in a Multipoint call. The CallId is returned when issuing the xCommand Dial command. During the call you can run the xStatus Call command to see the CallId.

PresetId: The PresetId must be retrieved from the far end codec since this is the ID of the preset that is stored on that codec.

Requires user role: USER

Parameters:

   CallId: <0..65534>
   PresetId(r): <1..15>

Example:

```
xCommand FarEndControl Preset Activate CallId:3 PresetId:1

*r FECCPresetActivateResult (status=OK):

** end
```

xCommand FarEndControl Preset Store

While in a call, this command is used to store a preset on the far end codec. The preset covers the far end codec's camera positions and input video switcher settings.

Note: The far end codec's xConfiguration Conference FarEndControl Mode setting must be switched On for the FarEndControl commands to work.

CallId: The CallId is required to identify the far end codec only when in a Multipoint call. The CallId is returned when issuing the xCommand Dial command. During the call you can run the xStatus Call command to see the CallId.

PresetId: The PresetId must be retrieved from the far end codec since this is the ID of the preset that is stored on that codec.

Requires user role: ADMIN

Parameters:

   CallId: <0..65534>
   PresetId(r): <0..15>

Example:

```
xCommand FarEndControl Preset Store CallId:3 PresetId:1

*r FECCPresetStoreResult (status=OK):

** end
```

xCommand FarEndControl Source Select

Select which video input source to use as the main source on the far end system.

CallId: The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

SourceId: Select a video input source on the far end.

Requires user role: USER

Parameters:

   CallId: <0..65534>
   SourceId(r): <0..15>

Example:

```
xCommand FarEndControl Source Select CallId:3 SourceId:1

*r FECCSelectSourceResult (status=OK):

** end
```

HttpFeedback commands

xCommand HttpFeedback Register

Register the system to a HTTP(S) server to return XML feedback over HTTP(S) to specific URLs.

FeedbackSlot: You can have from 1 to 4 slots for feedback.

ServerUrl: Define the URL for the HTTP(S) server.

Expression[1..15]: What parts of the Status and Configuration XML documents to monitor are specified by XPath expressions. You can have from 1 to 15 XPath expressions.

Requires user role: USER

Parameters:

FeedbackSlot: <1..4>
ServerUrl(r): <S: 1, 2048>
Expression[1..15]: <S: 1, 255>

Example:

```
xCommand HttpFeedback Register FeedbackSlot:1 ServerUrl:10.47.19.41
Expression[1]:Status/Video Expression[2]:Status/Audio Expression[3]:Status/
Call Expression[4]:Status/Conference

OK

*r FeedbackRegisterResult (status=OK):

FeedbackSlot: 1

** end
```

xCommand HttpFeedback Deregister

Deregister XML feedback over HTTP(S).

Requires user role: USER

Parameters:

FeedbackSlot(r): <1..4>

Example:

```
xCommand HttpFeedback Deregister FeedbackSlot:1

OK

*r FeedbackDeregisterResult (status=OK):

FeedbackSlot: 1

** end
```

Key commands

xCommand Key Click

Emulates a remote control key press, followed by a key release.

Requires user role: ADMIN

Parameters:

Key(r): <0/1/2/3/4/5/6/7/8/9/C/Call/Disconnect/Down/F1/F2/F3/F4/F5/Grab/Home/Layout/
Left/Mute/MuteMic/Ok/PhoneBook/Presentation/Right/Selfview/Square/SrcAux/SrcCamera/
SrcDocCam/SrcPc/SrcVcr/Star/Up/VolumeDown/VolumeUp/ZoomIn/ZoomOut>

Example:

```
xCommand Key Click Key:Down

*r KeyClickResult (status=OK):

** end
```

xCommand Key Press

Emulates a remote control key press without releasing it. The Key Press command must be followed by a Key Release command to emulate releasing the key.

Requires user role: ADMIN

Parameters:

Key(r): <0/1/2/3/4/5/6/7/8/9/C/Call/Disconnect/Down/F1/F2/F3/F4/F5/Grab/Home/Layout/
Left/Mute/MuteMic/Ok/PhoneBook/Presentation/Right/Selfview/Square/SrcAux/SrcCamera/
SrcDocCam/SrcPc/SrcVcr/Star/Up/VolumeDown/VolumeUp/ZoomIn/ZoomOut>

Example:

```
xCommand Key Press Key:Home

*r KeyPressResult (status=OK):

** end
```

xCommand Key Release

Emulates a remote control key release. The Key Release command is issued after a Key Press command.

Requires user role: ADMIN

Parameters:

Key(r): <0/1/2/3/4/5/6/7/8/9/C/Call/Disconnect/Down/F1/F2/F3/F4/F5/Grab/Home/Layout/ Left/Mute/MuteMic/Ok/PhoneBook/Presentation/Right/Selfview/Square/SrcAux/SrcCamera/ SrcDocCam/SrcPc/SrcVcr/Star/Up/VolumeDown/VolumeUp/ZoomIn/ZoomOut>

Example:

```
xCommand Key Release Key:Home

*r KeyReleaseResult (status=OK):

** end
```

Message commands

xCommand Message Alert Display

Display a message on screen, for a specified duration of time (in seconds). NOTE: If Duration is not set, the command must be followed by xCommand Message Alert Clear.

Use the xFeedback commands to monitor the feedback from the user. Read more about the xFeedback commands in the API introduction section in this guide.

Title: Enter a message title.

Text: Enter the message to be displayed. The <p> and
 HTML tags will result in line breaks as normal; any other tags will appear as plain text.

Duration: Set how long (in seconds) the message is to be displayed on the screen. If set to 0 (zero) the message does not disappear until a xCommand Message Alert Clear message has been sent.

Requires user role: ADMIN

Parameters:

Title: <S: 0, 255>
Text(r): <S: 0, 255>
Duration: <0..3600>

Example:

```
xCommand Message Alert Display Title: "Message" Text: "The meeting will
end in 5 minutes." Duration: 20

OK

*r MessageAlertDisplayResult (status=OK):

** end
```

xCommand Message Alert Clear

Remove the message which was displayed using the xCommand Message Alert Display command. This is required when the Duration parameter is not set.

Requires user role: ADMIN

Example:

```
xCommand Message Alert Clear

OK

*r MessageAlertClearResult (status=OK):

** end
```

D14949.04 SX20 Codec API Reference Guide TC6.2, JULY 2013.

94

xCommand Message TextLine Display

Display a text line on screen. Optionally you can place the text line at a specified location and for a specified duration of time (in seconds). NOTE: If Duration is not set, the command must be followed by xCommand Message TextLine Clear.

Text: Enter the text line to be displayed. The <p> and
 HTML tags will result in line breaks as normal; any other tags will appear as plain text.

X: Enter the X-coordinate (horizontal) on screen. X=0 is in the upper left corner.

Y: Enter the Y-coordinate (vertical) on screen. Y=0 is in the upper left corner.

Duration: Set how long (in seconds) the text line is to be displayed on the screen. If set to 0 (zero) the text line is displayed until a xCommand Message TextLine Clear command has been sent.

Requires user role: ADMIN

Parameters:

    Text(r): <S: 0, 140>

    X: <1..10000>

    Y: <1..10000>

    Duration: <0..3600>

Example:

```
xCommand Message TextLine Display Text: "The message to be displayed."
X:500 Y:2000 Duration: 20

OK

*r MessageTextLineDisplayResult (status=OK):

** end
```

xCommand Message TextLine Clear

Clears the text line which was defined by the xCommand Message TextLine Display command.

Requires user role: ADMIN

Example:

```
xCommand Message TextLine Clear

OK

*r MessageTextLineClearResult (status=OK):

** end
```

xCommand Message Prompt Display

Display a small window on screen with a title, text and up to five options for response from the user. The message is displayed on screen until the user gives a response, or until the system receives the following command xCommand Message Prompt Clear.

Use the xFeedback commands to monitor the feedback from the user. Read more about the xFeedback commands in the API introduction section in this guide.

Title: Enter the title of the message.

Text: Enter the text line to be displayed. The <p> and
 HTML tags will result in line breaks as normal; any other tags will appear as plain text.

FeedbackId: To identify the feedback enter a FeedbackId.

Option.1 to Option.5: Enter the text to appear on the feedback options.

Requires user role: ADMIN

Parameters:

    Title: <S: 0, 255>

    Text(r): <S: 0, 255>

    FeedbackId: <S: 0, 255>

    Option.1: <S: 0, 255>

    Option.2: <S: 0, 255>

    Option.3: <S: 0, 255>

    Option.4: <S: 0, 255>

    Option.5: <S: 0, 255>

Example:

```
xCommand Message Prompt Display Title: "Meeting extension" Text: "The
meeting is about to end. Do you want to extend the meeting?" Option.1:
"No" Option.2: "Yes, extend with 5 minutes" Option.3: "Yes, extend with 10
minutes"

OK

*r MessagePromptDisplayResult (status=OK):

** end
```

95

xCommand Message Prompt Clear

Remove the window displayed using the xCommand Message Alert Display command.

Use the xFeedback commands to monitor the feedback from the user. Read more about the xFeedback commands in the API introduction section in this guide.

FeedbackId: The FeedbackId corresponds to the FeedbackId given by the xCommand Message Prompt Display command.

Requires user role: ADMIN

Parameters:

FeedbackId: <S: 0, 255>

Example:

```
xCommand Message Prompt Clear

OK

*r MessagePromptClearResult (status=OK):

** end
```

xCommand Message Prompt Response

Give a response to the xCommand Message Prompt Display.

Use the xFeedback commands to monitor the feedback from the user. Read more about the xFeedback commands in the API introduction section in this guide.

FeedbackId: The FeedbackId corresponds to the FeedbackId given by the xCommand Message Prompt Display command.

OptionId: The OptionId corresponds to the OptionIds given as possible responses in the xCommand Message Prompt Display command.

Requires user role: ADMIN

Parameters:

FeedbackId: <S: 0, 255>
OptionId(r): <1..5>

Example:

```
xCommand Message Prompt Response OptionId: 1

OK

*r MessagePromptResponseResult (status=OK):

** end
```

xCommand Message Echo

Issuing the command makes the API raise a message-echo event. The command has no other impact on the codec. Usage can be to poll the codec from a control system or any external device/system to check for connectivity. To monitor the feedback use the xFeedback command. You can read more about the xFeedback command in the general API introduction section.

Text: Enter the text to be echoed.

Requires user role: ADMIN

Parameters:

Text: <S: 0, 255>

Example:

```
xCommand Message Echo Text:" MyEchoListner99"

*** end

* e Message Echo Text: "MyEchoListner99"
```

xCommand Message FarendMessage

This command is used to send generic messages to the participants in a conference. Not in use in this software version.

CallId: The remote participant to whom the message needs to be sent.

Type: Type of the message.

Text: Enter the text line to be displayed.

Requires user role: ADMIN

Parameters:

CallId(o): <0..65534>
Type(r): <S: 0, 255>
Text(r): <S: 1, 1450>
LocalEcho(o): <On/Off>

Example:

```
xCommand Message FarendMessage CallId: 3 Type: 3 Text: Hello

*r FarendMessageResult (status=Error):

    Reason: Generic Message mode not enabled

** end
```

Phonebook commands

xCommand Phonebook Folder Add

Phonebook entries can be stored in folders. Use this command to add a folder to the local phonebook. The command returns the FolderId, which is a unique string that identifies the folder; typically the format is "localGroupId-n".

Name(r): The name of the folder.

ParentFolderId: The unique identifier for the parent folder. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook Folder Add command was issued to make the parent folder.

Requires user role: ADMIN

Parameters:

Name(r): <S: 0, 255>
ParentFolderId: <S: 0, 255>

Example:

```
xCommand Phonebook Folder Add Name: "New York Office"

OK

*r PhonebookFolderAddResult (status=OK):

 Name: localGroupId-3

** end
```

xCommand Phonebook Folder Modify

Modify an existing phonebook folder.

FolderId: The unique identifier for the folder. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook Folder Add command was issued to make the folder.

Name(r): The name of the contact.

ParentFolderId: The unique identifier for the parent folder. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook Folder Add command was issued to make the parent folder.

Requires user role: ADMIN

Parameters:

FolderId(r): <S: 0, 255>
Name: <S: 0, 255>
ParentFolderId: <S: 0, 255>

Example:

```
xCommand Phonebook Folder Modify FolderId:localGroupId-3 Name: "New York Head Office"

OK

*r PhonebookFolderModifyResult (status=OK):

** end
```

xCommand Phonebook Folder Delete

Delete an existing folder from the local phonebook.

FolderId: The unique identifier for the folder. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook Folder Add command was issued to make the folder.

Requires user role: ADMIN

Parameters:

FolderId(r): <S: 0, 255>

Example:

```
xCommand Phonebook Folder Delete FolderId:localGroupId-3

OK

*r PhonebookFolderDeleteResult (status=OK):

** end
```

xCommand Phonebook Contact Add

Add a new contact to the local phonebook. The command returns the ContactId, which is a unique string that identifies the contact; typically the format is "localContactId-n".

You can add several contact methods to a contact using the xCommand Phonebook ContactMethod Add command. Note that only the first contact method will appear in the Favorites list on the Cisco TelePresence Touch controller. All contact methods are available in the API, on the web interface and when using the remote control.

Name: The name of the contact.

FolderId: The unique identifier for the folder that you want to store the contact in. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook Folder Add command was issued to make the folder.

ImageURL: Currently not in use.

Title: The title of the contact.

Number: The phone number or address of the contact.

Protocol: Select Auto, SIP, H323 or H320 protocol.

CallRate: Set a call rate.

CallType: Select a call type (audio or video).

Device: Select the device type.

Requires user role: ADMIN

Parameters:

    Name(r): <S: 0, 255>

    FolderId: <S: 0, 255>

    ImageURL: <S: 0, 255>

    Title: <S: 0, 255>

    Number: <S: 0, 255>

    Protocol: <Auto/H320/H323/SIP>

    CallRate: <0..6000>

    CallType: <Audio/Video>

    Device: <Mobile/Other/Telephone/Video>

Example:

```
xCommand Phonebook Contact Add Name: "John Doe" Number:12345

OK

*r PhonebookContactAddResult (status=OK):

Name: localContactId-1

** end
```

xCommand Phonebook Contact Modify

Modify contact details of an existing contact in the local phonebook. The following parameters can be changed using this command: Name, FolderId, ImageURL and Title. You must use the xCommand Phonebook ContactMethod Modify command to change the other parameters: Number, Protocol, CallRate, CallType and Device.

ContactId: The unique identifier for the contact you want to modify. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook Contact Add command was issued to make the contact.

Name: The name of the contact.

FolderId: A unique identifier for the folder. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook Folder Add command was issued.

ImageURL: Currently not in use.

Title: The title of the contact.

Requires user role: ADMIN

Parameters:

    ContactId(r): <S: 0, 255>

    Name: <S: 0, 255>

    FolderId: <S: 0, 255>

    ImageURL: <S: 0, 255>

    Title: <S: 0, 255>

Example:

```
xCommand Phonebook Contact Modify ContactId:localContactId-1 Name: "John Doe - office"

OK

*r PhonebookContactModifyResult (status=OK):

** end
```

xCommand Phonebook Contact Delete

Delete an existing contact from the local phonebook.

ContactId: The unique identifier for the contact. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook Contact Add command was issued to make the contact.

Requires user role: ADMIN

Parameters:

ContactId(r): <S: 0, 255>

Example:

```
xCommand Phonebook Contact Delete ContactId:localContactId-1

OK

*r PhonebookContactDeleteResult (status=OK):

** end
```

xCommand Phonebook ContactMethod Add

Add contact details for an existing contact in the local phonebook. The command returns the ContactMethodId, which is a unique string that identifies the contact method; typically the format is "n".

You can add several contact methods to a contact. Note that only the first contact method will appear in the Favorites list on the Cisco TelePresence Touch controller. The first contact method may have been created when issuing the xCommand Phonebook Contact Add command to make the contact. All contact methods are available in the API, on the web interface and when using the remote control.

ContactId: The unique identifier for the contact that you want to add a contact method to. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook Contact Add command was issued to make the contact.

Device: Set which type of device to call to.

Number(r): The phone number or address of the contact.

Protocol: Select Auto, SIP, H323 or H320 protocol.

CallRate: Set a call rate.

CallType: Select a call type (audio or video).

Requires user role: ADMIN

Parameters:

ContactId(r): <S: 0, 255>

Device: <Mobile/Other/Telephone/Video>

Number(r): <S: 0, 255>

Protocol: <Auto/H320/H323/SIP>

CallRate: <0..6000>

CallType: <Audio/Video>

Example:

```
xCommand Phonebook ContactMethod Add ContactId:localContactId-2 Number:54321
Protocol:H323

OK

*r PhonebookContactMethodAddResult (status=OK):

Name: 1

** end
```

99

xCommand Phonebook ContactMethod Modify

Modify details about the contact method for an existing contact in the local phonebook.

ContactId: The unique identifier for the contact. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook Contact Add command was issued to make the contact.

ContactMethodId: The unique identifier for the contact method you want to modify. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook ContactMethod Add or xCommand Phonebook Contact Add commands were issued to make the contact method.

Device: Set which type of device to call to.

Number: The phone number or address of the contact.

Protocol: Select Auto, SIP, H323 or H320 protocol.

CallRate: Set a call rate.

CallType: Select a call type (audio or video).

Requires user role: ADMIN

Parameters:

 ContactId(r): <S: 0, 255>

ContactMethodId(r): <S: 0, 255>

Device: <Mobile/Other/Telephone/Video>

Number: <S: 0, 255>

Protocol: <Auto, H320/H323/SIP>

CallRate: <0..6000>

CallType: <Audio/Video>

Example:

```
xCommand Phonebook ContactMethod Modify ContactMethodId:117
ContactId:localContactId-10 Number:"newnumber@cisco.com"
OK
```

xCommand Phonebook ContactMethod Delete

Delete a contact method from an existing contact in the local phonebook.

ContactId: The unique identifier for the contact you want to change. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook Contact Add command was issued to make the contact.

ContactMethodId: The unique identifier for the contact method you want to delete. The identifier will be returned by an xCommand Phonebook Search command. It was also returned when the xCommand Phonebook ContactMethod Add command was issued to make the contact method.

Requires user role: ADMIN

Parameters:

ContactId(r): <S: 0, 255>

ContactMethodId(r): <S: 0, 255>

Example:

```
xCommand Phonebook ContactMethod Delete ContactId:localContactId-2
ContactMethodId:1
OK
*r PhonebookContactMethodDeleteResult (status=OK):
** end
```

xCommand Phonebook Search

The search command lets you search in both the local and corporate phone books. A search gives a ResultSet. More examples can be found on the Cisco Developer Network at http://developer.cisco.com/web/telepresence-developer/howtos/cseries-api/phonebook.

PhonebookId: The identifier of the phonebook server that will be used. See the xConfiguration Phonebook Server ID setting. Not necessary to use.

PhonebookType: Define whether to search the local phone book or the corporate phonebook.

SearchString: Search for entries containing this string (note that the entry does not have to begin with the string). If no FolderId is specified, all folders / phonebook directories will be searched. The SearchString parameter is optional for software version TC2.0 and later.

SearchField: Currently not in use.

Offset: Get records starting with this offset in a search. The default value is 0. Offset is used together with Limit to support paging.

FolderId: Search only in the specified folder. The FolderId (string) is listed in the ResultSet of a search result containing folders.

Limit: Limit the number of records in the result set to this number. For example, if the limit is set to 10, the ResultSet will contain only 10 entries (Contacts and Folders) even if the total number of hits is larger. The maximum limit is 1000.

Recursive: This parameter will only have effect when searching the local phone book. The setting determines whether a local phone book search should be limited to the given FolderId, or also recursively search in its subfolders. If not specified, the search will be recursive.

When issuing the command without specifying any parameters, all folders, contacts and contact methods in the local phone book will be returned.

Requires user role: USER

Parameters:

PhonebookId: <S: 0, 255>

PhonebookType: <Corporate/Local>

SearchString: <S: 0, 255>

SearchField: <Name/Number>

Offset: <0..65534>

FolderId: <S: 0, 255>

Limit: <0..65534>

Recursive: <False/True>

ContactType(o): <Any/Folder/Contact>

Example:

```
xCommand Phonebook Search PhonebookType:Corporate Limit:2
FolderId:"corporate _ 001"
OK
*r ResultSet ResultInfo TotalRows: 25
*r ResultSet Contact 1 Name: "John Doe (office)"
```

```
*r ResultSet Contact 1 ContactId: "localContactId-3"
*r ResultSet Contact 1 ContactMethod 1 ContactMethodId: "1"
*r ResultSet Contact 1 ContactMethod 1 Number: "H323:John.Doe.Office@
company.com"
*r ResultSet Contact 1 ContactMethod 1 Protocol: H323
*r ResultSet Contact 1 ContactMethod 2 ContactMethodId: "2"
*r ResultSet Contact 1 ContactMethod 2 Number: "SIP:John.Doe.Office@
company.com"
*r ResultSet Contact 1 ContactMethod 2 Protocol: SIP
*r ResultSet Contact 2 Name: "John Doe (home)"
*r ResultSet Contact 2 ContactId: "localContactId-4"
*r ResultSet Contact 2 ContactMethod 1 ContactMethodId: "3"
*r ResultSet Contact 2 ContactMethod 1 Number: "H323:John.Doe.Home@
company.com"
*r ResultSet Contact 2 ContactMethod 1 Protocol: H323
*r ResultSet Contact 2 ContactMethod 2 ContactMethodId: "4"
*r ResultSet Contact 2 ContactMethod 2 Number: "SIP:John.Doe.Home@company.
com"
*r ResultSet Contact 2 ContactMethod 2 Protocol: SIP
** end
```

Presentation commands

xCommand Presentation Start

Open a media stream from the selected presentation source.

PresentationSource: Select the video input source to be used for presentation.

SendingMode: Select whether the presentation is shown locally or locally and remotely.

Requires user role: USER

Parameters:

PresentationSource: <1..2>

SendingMode: <LocalRemote/LocalOnly>

Example:

```
xCommand Presentation Start PresentationSource:2

OK

*r PresentationStartResult (status=OK):

** end
```

xCommand Presentation Stop

Stop the media stream from the presentation source.

Requires user role: USER

Example:

```
xCommand Presentation Stop

OK

*r PresentationStopResult (status=OK):

** end
```

Preset commands

xCommand Preset Store

Store the connector selections for all video input sources and the current position for all cameras. Note that information about all video input sources and all camera positions are included in the same preset. The system may hold 15 predefined video input presets. These presets are available for far end control, i.e. they are referred in the PresetId parameter of the xCommand FarEndControl Preset Activate command. In contrast, the xCommand Camera Preset commands applies to individual cameras only. Those presets are not available for far end control.

PresetId: The ID of this preset.

Type: Currently it makes no difference if you select Camera or All. However, this may change in the future (other audio or video settings may be included in a preset). If you want to ensure that a preset only affects camera positions we recommend that you select Camera.

Description: Enter a description of the camera preset.

Requires user role: ADMIN

Parameters:

PresetId(r): <1..15>

Type(r): <All/Camera>

Description: <S: 0, 255>

Example:

```
xCommand Preset Store PresetId:3 Type:Camera Description:"Left view"

OK

*r PresetStoreResult (status=OK):

** end
```

xCommand Preset Activate

Activate one of the stored local presets.

PresetId: The ID of the preset you want to activate.

Requires user role: USER

Parameters:

PresetId(r): <1..15>

Example:

```
xCommand Preset Activate PresetId:3

OK

*r PresetActivateResult (status=OK):

** end
```

xCommand Preset Clear

Delete a preset.

PresetId: The ID of the preset you want to delete.

Requires user role: USER

Parameters:

PresetId(r): <1..15>

Example:

```
xCommand Preset Clear PresetId:3
OK
*r PresetClearResult (status=OK):
** end
```

Provisioning commands

xCommand Provisioning StartUpgrade

The codec software can be upgraded from the provisioning server. When starting the upgrade the software is automatically downloaded and installed. The codec reboots to complete the software upgrade.

Requires user role: ADMIN

Example:

```
xCommand Provisioning StartUpgrade
*r StartUpgradeResult (status=OK):
** end
```

xCommand Provisioning CompleteUpgrade

Starts installing the software upgrade if you wish to install it before it is set to do so.

Requires user role: ADMIN

Example:

```
xCommand Provisioning CompleteUpgrade
*r CompleteUpgradeResult (status=OK):
** end
```

xCommand Provisioning PostponeUpgrade

Postpones the installing of the software upgrade.

Requires user role: ADMIN

Parameters:

SecondsToPostpone(r): <0..65534>

Example:

```
xCommand Provisioning PostponeUpgrade SecondsToPostpone: 60
*r PostponeUpgradeResult (status=OK):
** end
```

xCommand Provisioning CUCM CAPF OperationStart

Starts a pending CAPF (Certificate Authority Proxy Function) operation toward the CUCM. When the CUCM administrator initiates an operation that the endpoint should perform (like installing, updating or deleting certificates), this command can be used to start the CAPF operation.

AuthString: Set the authentication string required for the CAPF operation.

Requires user role: USER

Parameters:

AuthString(o): <S: 4, 10>

Example:

```
xCommand Provisioning CUCM CAPF OperationStart AuthString: 123456

*r CAPFStartResult (status=OK):

** end
```

xCommand Provisioning CUCM CTL Delete

Delete the stored CTL and ITL files (CTL: Certificate Trust List, ITL: Identity Trust List).

Requires user role: USER

Example:

```
xCommand Provisioning CUCM CTL Delete

*r DeleteResult (status=OK):

** end
```

Security commands

xCommand Security FIPSMode Activate

Activate FIPS (140-2) mode. NOTE: Activating FIPS mode implies a reset to factory defaults.

While in FIPS mode, software upgrade is disabled and the following limitations apply: All calls are encrypted, unencrypted communication protocols like Telnet and HTTP cannot be used. IEEE802.1x and SNMP are disabled. The SIP Profile Type setting cannot be Microsoft. The root user is not available (root settings cannot be changed).

To exit FIPS mode, perform a factory reset.

Confirm: <Yes>

Requires user role: ADMIN

Parameters:

Confirm(r): <Yes>

Example:

```
xCommand Security FIPSMode Activate Confirm: "Yes"

OK

*r SecurityFIPSModeActivateConfirmResult (status=OK)

** end
```

Standby commands

xCommand Standby Activate

Set the system in standby mode, which turns off the video outputs and put the camera into sleep mode.

Requires user role: USER

Example:

```
xCommand Standby Activate
*r ActivateResult (status=OK):
** end
```

xCommand Standby Deactivate

Bring the system out of standby mode.

Requires user role: USER

Example:

```
xCommand Standby Deactivate
*r DeactivateResult (status=OK):
** end
```

xCommand Standby ResetTimer

Set a temporary standby delay (in minutes). If the system is in standby mode when the reset timer is set, the system is brought out of standby mode. When left idle for the given delay the system goes into standby mode. Setting the reset timer does not affect the Standby Delay in the Advanced configuration menu (or by xConfiguration Standby Delay). Next time this delay is the valid standby delay.

Requires user role: USER

Parameters:

Delay(r): <1..480>

Example:

```
xCommand Standby ResetTimer Delay:10
*r ResetResult (status=OK):
** end
```

SystemUnit commands

xCommand SystemUnit OptionKey Add

Add an option key to support additional features.

Requires user role: ADMIN

Parameters:

Key(r): <S: 16, 24>

Example:

```
xCommand SystemUnit OptionKey Add Key:******************
*r OptionKeyResult (status=OK):
** end
```

xCommand SystemUnit OptionKey RemoveAll

Removes all Option keys.

Requires user role: ADMIN

Parameters:

Confirm(r): <Yes>

Example:

```
xCommand SystemUnit OptionKey RemoveAll Confirm: Yes
*r OptionKeyResult (status=OK):
** end
```

xCommand SystemUnit ReleaseKey Add

Add the software release key. Used to enable new software (applicable for main software releases).

Requires user role: ADMIN

Parameters:

Key(r): <S: 16, 24>

Example:

```
xCommand SystemUnit ReleaseKey Add Key:******************
*r ReleaseKeyResult (status=OK):
** end
```

xCommand SystemUnit ReleaseKey RemoveAll

Removes all Release keys.

Requires user role: ADMIN

Parameters:

Confirm(r): <Yes>

Example:

```
xCommand SystemUnit ReleaseKey RemoveAll Confirm: Yes
*r ReleaseKeyResult (status=OK):
** end
```

xCommand SystemUnit AdminPassword Set

Set an administrator password to restrict access the codec. After a restart of the codec this password also applies to the web interface.

Requires user role: ADMIN

Parameters:

Password(r): <S: 0, 64>

Example:

```
xCommand SystemUnit AdminPassword Set Password:***********
*r AdminPasswordSetResult (status=OK):
** end
```

xCommand SystemUnit Diagnostics Run

This command runs all self-diagnostics commands on the system.

Requires user role: ADMIN

Parameters:

ResultSet(o): <Alerts/All/None>

Example:

```
xCommand SystemUnit Diagnostics Run
*r DiagnosticsResult (status=OK):
** end
```

xCommand SystemUnit MenuPassword Set

Set a menu password to restrict access to Administrator Settings menu. If you have a remote control the password can also be set from the on screen menu. If you have a Cisco TelePresence Touch controller the menu password is set from the command line interface.

Requires user role: ADMIN

Parameters:

Password(r): <S: 0, 255>

Example:

```
xCommand SystemUnit MenuPassword Set Password:***********
*r MenuPasswordSetResult (status=OK):
** end
```

xCommand SystemUnit MenuPassword Validate

Validate that the supplied password is correct.

Requires user role: ADMIN

Parameters:

Password(r): <S: 0, 255>

Example:

```
xCommand SystemUnit MenuPassword Validate Password:***********
*r MenuPasswordValidateResult (status=OK):
** end
```

xCommand SystemUnit DateTime Set

Set the date and time for the system, if not available from NTP (Network Time Protocol).

Requires user role: ADMIN

Parameters:

Year: <2008..2037>

Month: <1..12>

Day: <1..31>

Hour: <0..23>

Minute: <0..59>

Second: <0..59>

Example:

```
xCommand SystemUnit DateTime Set Year:2012 Month:7 Day:3 Hour:12 Minute:0
Second:0
*r DateTimeSetResult (status=OK):
** end
```

xCommand SystemUnit DateTime Get

Read the time and date from the system.

Requires user role: USER

Example:

```
xCommand SystemUnit DateTime get
*r DateTimeGetResult (status=OK):
 Year: 2012
 Month: 7
 Day: 3
 Hour: 12
 Minute: 0
 Second: 0
** end
```

xCommand SystemUnit FactoryReset

Reset the codec to factory default settings. The call logs are deleted and all system parameters are reset to default values. All files that have been uploaded to the codec are deleted. The Release key and Option key are not affected.

As a default the system restarts after the factory reset, but other behavior can be forced by selecting a different TrailingAction.

TrailingAction: Select Shutdown or NoAction to override the default behavior (Restart).

Requires user role: ADMIN

Parameters:

Confirm(r): <Yes>

TrailingAction: <NoAction/Restart/Shutdown>

Example:

```
xCommand SystemUnit FactoryReset Confirm: Yes
*r FactoryResetConfirmResult (status=OK):
** end
```

xCommand SystemUnit SoftwareUpgrade

Initiate a software upgrade by fetching the software on a given URL. If the server requires username and password these parameters must be included.

Requires user role: ADMIN

Parameters:

URL(r): <S: 0, 255>

UserName: <S: 0, 255>

Password: <S: 0, 255>

Example:

```
xCommand SystemUnit SoftwareUpgrade URL: "ftp://<ftp_server_ip_
address>/s52000tc4_0_0.pkg" UserName: testDownload Password: 1234
*r SystemUnitSoftwareUpgradeResult (status=OK):
** end
```

xCommand SystemUnit ConfigurationProfile Change

Select a previously saved configuration profile. NOTE: Requires a restart of the codec.

Requires user role: ADMIN

Parameters:

Name(r): <S: 0, 255>

Example:

```
xCommand SystemUnit ConfigurationProfile Change Name: "My _
ConfigurationProfile _ 1"
*r ConfigurationProfileChangeResult (status=OK):
   Warning: New configuration profile will be active after next boot.
** end
```

xCommand SystemUnit ConfigurationProfile Remove

Delete a configuration profile that has been stored in the system.

Requires user role: ADMIN

Parameters:

Name(r): <S: 0, 255>

Example:

```
xCommand SystemUnit ConfigurationProfile Remove Name: "My _
ConfigurationProfile _ 1"
*r ConfigurationProfileRemoveResult (status=OK):
** end
```

xCommand SystemUnit ConfigurationProfile SaveCurrentConfigurationAs

Save the current system settings into a configuration profile. Assign a name to the new profile. The name is the unique identifier of the profile.

Requires user role: ADMIN

Parameters:

Name(r): <S: 0, 255>

Example:

```
xCommand SystemUnit ConfigurationProfile SaveCurrentConfigurationAs Name:
"My _ ConfigurationProfile _ 1"
*r ConfigurationProfileSaveCurrentConfigurationResult (status=OK):
** end
```

xCommand SystemUnit ConfigurationProfile List

List configuration profiles that has been stored in the system.

Requires user role: ADMIN

Example:

```
xCommand SystemUnit ConfigurationProfile List
*r ConfigurationProfileListResult (status=OK):
   Profile: My _ ConfigurationProfile _ 1
   Profile: My _ ConfigurationProfile _ 2
** end
```

xCommand SystemUnit ConfigurationProfile CancelChange

Cancel the "ConfigurationProfile Change" command, that would otherwise take effect after next system boot.

Requires user role: ADMIN

Example:

```
xCommand SystemUnit ConfigurationProfile CancelChange
*r ConfigurationProfileCancelChangeResult (status=OK):
** end
```

xCommand SystemUnit Notifications RemoveAll

Clears the list of system notifications that are reported by xStatus SystemUnit Notifications Text/ Type.

Requires user role: ADMIN

Example:

```
xCommand SystemUnit Notifications RemoveAll
*r NotificationsRemoveAllResult (status=OK):
** end
```

xCommand SystemUnit ReleaseKey List

List all software versions that the system has a valid release key for.

Requires user role: ADMIN

Example:

```
xCommand SystemUnit ReleaseKey List

*r SystemUnit ReleaseKey 1 Version: "TC1"

*r SystemUnit ReleaseKey 2 Version: "TC2"

*r SystemUnit ReleaseKey 3 Version: "TC3"

*r SystemUnit ReleaseKey 4 Version: "TC4"

*r SystemUnit ReleaseKey 5 Version: "TC5"

** end
```

Video commands

xCommand Video PictureLayoutSet

Select the screen layout mode.

Target: Select if the target is the local layout or the remote layout.

CallId: The CallID is returned when the xCommand Dial command is run. During the call you can run the xStatus Call command to see the CallId.

LayoutFamily: Select a layout family.

CustomLayoutName: Enter a name for the layout.

Requires user role: USER

Parameters:

Target: <local/remote>

CallId: <0..65534>

LayoutFamily(r): <auto/custom/equal/fullscreen/overlay/presentationlargespeaker/presentationsmallspeaker/prominent/single/speaker_full>

CustomLayoutName: <S: 1, 128>

Example:

```
xCommand Video PictureLayoutSet Target: Local LayoutFamily: equal

*r SetPictureLayoutResult (status=OK):

** end
```

xCommand Video Layout SetPresentationView

Set the presentation view mode.

View: Select Default when you want the presentation to be viewed with the default settings for the codec. Select Maximized when you want the presentation to be displayed in full screen. Select Minimized when you want the presentation to be displayed in a small picture on screen.

Requires user role: ADMIN

Parameters:

View(r): <Default/Maximized/Minimized>

Example:

```
xCommand Video Layout SetPresentationView View:Default

OK

*r VideoLayoutSetPresentationViewResult (status=OK):

** end
```

xCommand Video OSD Close

Closes all menus on the on–screen display.

Element: The on–screen menu.

Requires user role: ADMIN

Parameters:

Element(r): <Menu>

Example:

```
xCommand Video OSD Close Element: Menu
*r OsdCloseElementResult (status=OK):
** end
```

xCommand Video PIP ActiveSpeaker Set

Sets position for the active speakers PiP (picture in picture).

Position: Predefined positions.

CenterLeft: The ActiveSpeaker PiP appears on the left side of the screen, in center.

CenterRight: The ActiveSpeaker PiP appears on the right side of the screen, in center.

LowerLeft: The ActiveSpeaker PiP appears in the lower left corner of the screen.

LowerRight: The ActiveSpeaker PiP appears in the lower right corner of the screen.

UpperCenter: The ActiveSpeaker PiP appears on top of the screen, in center.

UpperLeft: The ActiveSpeaker PiP appears in the upper left corner of the screen.

UpperRight: The ActiveSpeaker PiP appears in the upper right corner of the screen.

Requires user role: ADMIN

Parameters:

Position(r): <CenterLeft/CenterRight/LowerLeft/LowerRight/UpperCenter/UpperLeft/UpperRight>

Example:

```
xCommand Video PIP ActiveSpeaker Set Position: UpperRight
*r ActiveSpeakerPipSetResult (status=OK):
** end
```

xCommand Video PIP Presentation Set

Sets position for the presentation PiP (picture in picture).

Position: Predefined positions.

CenterLeft: The presentation PiP appears on the left side of the screen, in center.

CenterRight: The presentation PiP appears on the right side of the screen, in center.

LowerLeft: The presentation PiP appears in the lower left corner of the screen.

LowerRight: The presentation PiP appears in the lower right corner of the screen.

UpperCenter: The presentation PiP appears on top of the screen, in center.

UpperLeft: The presentation PiP appears in the upper left corner of the screen.

UpperRight: The presentation PiP appears in the upper right corner of the screen.

Requires user role: ADMIN

Parameters:

Position(r): <CenterLeft/CenterRight/LowerLeft/LowerRight/UpperCenter/UpperLeft/UpperRight>

Example:

```
xCommand Video PIP Presentation Set Position: LowerLeft
*r PresentationPipSetResult (status=OK):
** end
```

xCommand Video Selfview Set

Sets self-view on/off and specifies its size and position. If the parameter is not specified, current value is used.

Mode: Selfview on/off

FullscreenMode: Self-view in fullscreen or pip.

PIPPosition: Predefined PIP positions.

FullscreenMode: Choose between displaying the self-view in full screen or not.

PIPPosition: Select the position for the self-view image.

CenterLeft: The self-view PiP appears on the left side of the screen, in center.

CenterRight: The self-view PiP appears on the right side of the screen, in center.

LowerLeft: The self-view PiP appears in the lower left corner of the screen.

LowerRight: The self-view PiP appears in the lower right corner of the screen.

UpperCenter: The self-view PiP appears on top of the screen, in center.

UpperLeft: The self-view PiP appears in the upper left corner of the screen.

UpperRight: The self-view PiP appears in the upper right corner of the screen.

OnMonitorRole: Displays self-view on monitors with this role.

Requires user role: ADMIN

Parameters:

Mode(o): <On/Off>

FullscreenMode(o): <On/Off>

PIPPosition(o): <CenterLeft/CenterRight/LowerLeft/LowerRight/UpperCenter/UpperLeft/UpperRight>

OnMonitorRole(o): <First/Fourth/Second/Third>

Example:

```
Example 1
xCommand Video Selfview Set Mode: On FullscreenMode: Off PIPPosition:
CenterLeft OnMonitorRole: First
OK
*r SelfviewSetResult (status=OK):
** end

Example 2
xCommand Video Selfview Set FullscreenMode: On OnMonitorRole: Second
OK
*r SelfviewSetResult (status=OK):
** end
```

xCommand Video PIP Presentation Set

Sets position for the presentation PiP (picture in picture).

Position: Predefined positions.

CenterLeft: The presentation PiP appears on the left side of the screen, in center.

CenterRight: The presentation PiP appears on the right side of the screen, in center.

LowerLeft: The presentation PiP appears in the lower left corner of the screen.

LowerRight: The presentation PiP appears in the lower right corner of the screen.

UpperCenter: The presentation PiP appears on top of the screen, in center.

UpperLeft: The presentation PiP appears in the upper left corner of the screen.

UpperRight: The presentation PiP appears in the upper right corner of the screen.

Requires user role: ADMIN

Parameters:

Position(r): <CenterLeft/CenterRight/LowerLeft/LowerRight/UpperCenter/UpperLeft/UpperRight>

Example:

```
xCommand Video PIP Presentation Set Position: LowerLeft
*r PresentationPipSetResult (status=OK):
** end
```

xCommand Video PreviewFilmstrip Set

Moves the location of the filmstrip associated with the layouts used in presentation preview mode. Not in use in this software version.

LowerCenter: The preview appears on the bottom of the screen, in center.

LowerLeft: The preview appears in the lower left corner of the screen.

LowerRight: The preview appears in the lower right corner of the screen.

UpperCenter: The preview appears on top of the screen, in center.

UpperLeft: The preview appears in the upper left corner of the screen.

UpperRight: The preview appears in the upper right corner of the screen.

Requires user role: ADMIN

Parameters:

Position(r): <LowerCenter/LowerLeft/LowerRight/UpperCenter/UpperLeft/UpperRight>

Example:

```
xCommand Video PreviewFilmstrip Set Position: LowerCenter
*r PreviewFilmstripSetResult (status=OK):
** end
```

Experimental commands

The Experimental commands are for testing only and should not be used unless agreed with Cisco.
These commands are not documented and WILL change in later releases.

Chapter 5

xStatus commands

Description of the xStatus commands

In this chapter, you can find all of the xStatus commands and the responses. Status type commands return information about the system and system processes. You can query all information or just some of it.

We recommend you visit our web site regularly for updated versions of the manual.
Go to: ▸ http://www.cisco.com/go/quickset-docs

116

Audio status

xStatus Audio

Shows the top level overview of the audio status. The identities of the LocalInput, RemoteInput, LocalOutput and RemoteOutput are used when querying additional information.

xStatus Audio Microphones Mute

Shows the microphones mute mode.

Value space of the result returned:

<On/Off>

Example:

```
xStatus Audio Microphones Mute
*s Audio Microphones Mute: Off
** end
```

xStatus Audio Volume

Shows the volume level (dB) of the loudspeaker output.

Value space of the result returned:

<0..100>

Example:

```
xStatus Audio Volume
*s Audio Volume: 70
** end
```

xStatus Audio Input RemoteInput [1..n] CallId

Shows the CallId for the remote audio input.
You can run the command xStatus Audio Input RemoteInput to find the identity [1..n] of the input.

Value space of the result returned:

<0..65534>

Example:

```
xStatus Audio Input RemoteInput 8 CallId
*s Audio Input RemoteInput 8 CallId: 28
** end
```

xStatus Audio Input Connectors Microphone [1..n] ConnectionStatus

Indicates whether a microphone is detected on the microphone input connector.

Value space of the result returned:

<NotConnected/HeadsetMicConnected/GenesisConnected/Connected>

Example:

```
xStatus Audio Input Connectors Microphone ConnectionStatus
*s Audio Input Connectors Microphone 1 ConnectionStatus: NotConnected
*s Audio Input Connectors Microphone 2 ConnectionStatus: Connected
** end
```

Call status

xStatus Call

Shows the top level overview of the call status. The call identity is used when query for additional information about the call.

xStatus Call [1..n] Status

Shows the status of a call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Idle/Dialling/Ringing/Connecting/Connected/Disconnecting/OnHold/EarlyMedia/Preserved/ RemotePreserved>

Example:

```
xStatus Call 27 Status
*s Call 27 Status: Connected
** end
```

xStatus Call [1..n] Direction

States the direction of the call initiation. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Incoming/Outgoing>

Example:

```
xStatus Call 27 Direction
*s Call 27 Direction: Outgoing
** end
```

xStatus Call [1..n] Protocol

Shows the call protocol of the incoming or outgoing call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<H323/SIP>

Example:

```
xStatus Call 27 Protocol
*s Call 27 Protocol: "h323"
** end
```

xStatus Call [1..n] CallType

Shows the call type of the incoming or outgoing call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Video/Audio/AudioCanEscalate/ForwardAllCall/Unknown>

Example:

```
xStatus Call 27 CallType
*s Call 27 CallType: Video
** end
```

xStatus Call [1..n] RemoteNumber

Shows the remote (far end) number or URI of an incoming or outgoing call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<String>

Example:

```
xStatus Call 27 RemoteNumber
*s Call 27 RemoteNumber: "5585232"
** end
```

xStatus Call [1..n] CallbackNumber

Shows the remote (far end) number or URI of an incoming or outgoing call, including the call protocol, for call back. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<String>

Example:

```
xStatus Call 27 CallbackNumber
*s Call 27 CallbackNumber: "h323:firstname.lastname@company.com"
** end
```

xStatus Call [1..n] DisplayName

Shows the name of the remote (far end) participant in an incoming or outgoing call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<String>

Example:

```
xStatus Call 27 DisplayName
*s Call 27 DisplayName: "firstname.lastname@company.com"
** end
```

xStatus Call [1..n] TransmitCallRate

Shows the transmit bandwidth in the call in kilobits per second (kbps). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Integer>

Example:

```
xStatus Call 27 TransmitCallRate
*s Call 27 TransmitCallRate: 768
** end
```

xStatus Call [1..n] ReceiveCallRate

Shows the receive bandwidth in the call in kilobits per second (kbps). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Integer>

Example:

```
xStatus Call 27 ReceiveCallRate
*s Call 27 ReceiveCallRate: 4000
** end
```

xStatus Call [1..n] FacilityServiceId

When calling a facility service, the facility service id is shown. Otherwise the value 0 is returned.

Value space of the result returned:

<0..5>

Example:

```
xStatus Call FacilityServiceId
*s Call 3 FacilityServiceId: 1
** end
```

xStatus Call [1..n] Encryption Type

Shows the encryption type of the call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<"None"/"Aes-128">

Example:

```
xStatus Call 27 Encryption Type
*s Call 27 Encryption Type: "None"
** end
```

xStatus Call [1..n] PlacedOnHold

Shows the placed on hold status of the call. You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<True/False>

Example:

```
xStatus Call 27 PlacedOnHold
*s Call 27 PlacedOnHold: False
** end
```

xStatus Call [1..n] Duration

Shows the duration of a call (in seconds). You can run the command xStatus Call to find the call identity.

Value space of the result returned:

<Integer>

Example:

```
xStatus Call 27 Duration
*s Call 27 Duration: 2354
** end
```

xStatus Call [1..n] SecurityStatus

Security status of a call represents the least of the security statuses of all the endpoints involved in a CUCM call.

Value space of the result returned:

<Invalid/Unknown/NotAuthenticated/Authenticated>

xStatus Call [1..n] AnswerState

Indicates if a call is answered, ignored or has been automatically answered by a system.

Value space of the result returned:

<Unanswered/Ignored/Autoanswered/Answered>

Example:

```
xStatus Call AnswerState
*s Call 5 AnswerState: Answered
** end
```

xStatus Call [1..n] ModifyState

Shows the current state of a pending Call Modify operation. Not in use in this software version.
OutgoingModify: a request CallEscalate to Video is pending.
IncomingModify: a peer has requested to escalate the call to Video.
Idle: No Call Modify operation is pending.

Value space of the result returned:

<Idle/Outgoing/Incoming>

Example:

```
xStatus Call ModifyState
*s Call 11 ModifyState: Idle
** end
```

xStatus Call [1..n] DeviceType

Indicates whether the remote system is a single endpoint or a MCU. Some Cisco endpoints (EX90, C40, C60, C90, SX20) have built-in MCU capabilities.

Value space of the result returned:

<Endpoint/MCU>

Example:

```
xStatus Call DeviceType
*s Call 4 DeviceType: Endpoint
** end
```

Camera status

xStatus Camera

Shows the top level overview of the camera status.

xStatus Camera [1..1] Connected

Shows if the camera is connected or not.

Value space of the result returned:

    <True/False>

Example:

```
xStatus Camera 1 Connected
*s Camera 1 Connected: True
** end
```

xStatus Camera [1..1] HardwareID

Shows the hardware identity of the camera.

Value space of the result returned:

    <String>

Example:

```
xStatus Camera 1 HardwareID
*s Camera 1 HardwareID: "50000000"
** end
```

xStatus Camera [1..1] Manufacturer

Shows the manufacturer of the camera.

Value space of the result returned:

    <String>

Example:

```
xStatus Camera 1 Manufacturer
*s Camera 1 Manufacturer: "TANDBERG"
** end
```

xStatus Camera [1..1] Model

Shows the camera model.

Value space of the result returned:

    <String>

Example:

```
xStatus Camera 1 Model
*s Camera 1 Model: "PrecisionHD 1080p 12X"
** end
```

xStatus Camera [1..1] SoftwareID

Shows the software identity of the camera.

Value space of the result returned:

    <String>

Example:

```
xStatus Camera 1 SoftwareID
*s Camera 1 SoftwareID: "S01718-4.0FINAL [ID:40063] 2010-10-20"
** end
```

xStatus Camera [1..1] SerialNumber

Shows the camera serial number.

Value space of the result returned:

    <String>

Example:

```
xStatus Camera 1 SerialNumber
*s Camera 1 SerialNumber: "B1AB26B00010"
** end
```

xStatus Camera [1..1] IpAddress

Shows the camera IP address.

Value space of the result returned:

    <String>

Example:

```
xStatus Camera 1 IpAddress
*s Camera 1 IpAddress: ""
** end
```

xStatus Camera [1..1] MacAddress

Shows the MAC (Media Access Control) address for the camera.

Value space of the result returned:

<String>

Example:

```
xStatus Camera 1 MacAddress
*s Camera 1 MacAddress: ""
** end
```

xStatus Camera [1..1] Position Pan

Shows the current pan (move left and right) position of the camera. The value range depends on camera type.

Value space of the result returned:

<-65535..65535>

Example:

```
xStatus Camera 1 Position Pan
*s Camera 1 Position Pan: 412
** end
```

xStatus Camera [1..1] Position Tilt

Shows the current tilt (move up and down) position of the camera. The value range depends on camera type.

Value space of the result returned:

<-65535..65535>

Example:

```
xStatus Camera 1 Position Tilt
*s Camera 1 Position Tilt: 106
** end
```

xStatus Camera [1..1] Position Zoom

Shows the current zoom (zoom in and out) position of the camera. The value range depends on camera type.

Value space of the result returned:

<0..65535>

Example:

```
xStatus Camera 1 Position Zoom
*s Camera 1 Position Zoom: 828
** end
```

xStatus Camera [1..1] Position Focus

Shows the current focus position of the camera. The value range depends on camera type.

Value space of the result returned:

<0..65535>

Example:

```
xStatus Camera 1 Position Focus
*s Camera 1 Position Focus: 4597
** end
```

xStatus Camera [1..1] Capabilities Options

Shows the camera capabilities (ptzf = pan, tilt, zoom, focus).

Value space of the result returned:

<String>

Example:

```
xStatus Camera 1 Capabilities Options
*s Camera 1 Capabilities Options: "ptzf"
** end
```

xStatus Camera [1..1] Flip

In Flip mode (vertical flip) the image can be flipped upside down.

Value space of the result returned:

<"Auto"/"On"/"Off">

Example:

```
xStatus Camera 1 Flip
*s Camera 1 Flip: "Off"
** end
```

Conference status

xStatus Conference

Shows the top level overview of the conference status. The identity of the Conference Site can only be read during a call.

xStatus Conference Presentation Mode

Shows the status of the secondary video stream.

Value space of the result returned:

<On/Off>

Example:

```
xStatus Conference Presentation Mode
*s Conference Presentation Mode: Off
** end
```

xStatus Conference Presentation Protocol

Shows the video protocol used when transmitting the presentation.

Value space of the result returned:

<String>

Example:

```
xStatus Conference Presentation Protocol
*s Conference Presentation Protocol: "H264"
** end
```

xStatus Conference Presentation Resolution Height

Shows the height of the the presentation.

Value space of the result returned:

<0..3000>

Example:

```
xStatus Conference Presentation Resolution Height
*s Conference Presentation Resolution Height: 0
** end
```

xStatus Conference Presentation Resolution Width

Shows the width of the the presentation.

Value space of the result returned:

<0..4000>

Example:

```
xStatus Conference Presentation Resolution Width
*s Conference Presentation Resolution Width: 0
** end
```

xStatus Conference Presentation SiteId

Shows the identity of the system that sends the presentation.

Value space of the result returned:

<0..65535>

Example:

```
xStatus Conference Presentation SiteId
*s Conference Presentation SiteId: 0
** end
```

xStatus Conference Presentation LocalSource

Shows the local video input source that is used when the presentation is sent from the local site.

Value space of the result returned:

<1..5>

Example:

```
xStatus Conference Presentation LocalSource
*s Conference Presentation LocalSource: 0
** end
```

xStatus Conference Presentation LocalSendingMode

Shows whether the presentations are set to be shown locally and sent to remote side, only shown locally or not shown at all.

Value space of the result returned:

<Off/LocalRemote/LocalOnly>

Example:

```
xStatus Conference Presentation LocalSendingMode
*s Conference Presentation LocalSendingMode: Off
** end
```

xStatus Conference Presentation LastLocalSource

Identifies the last used local presentation source.

Value space of the result returned:

    <Integer>

Example:

```
xStatus Conference Presentation LastLocalSource
*s Conference Presentation LastLocalSource: 2
** end
```

xStatus Conference Site [1..n] Capabilities FECC NumberOfPresets

Shows the number of presets available for the input sources at a far end site.

Value space of the result returned:

    <1..15>

Example:

```
xStatus Conference Site 2 Capabilities FECC NumberOfPresets
*s Conference Site 2 Capabilities FECC NumberOfPresets: 15
** end
```

xStatus Conference Site [1..n] Capabilities FECC NumberOfSources

Shows the number of input sources that can be connected at a far end site.

Value space of the result returned:

    <1..5>

Example:

```
xStatus Conference Site 2 Capabilities FECC NumberOfSources
*s Conference Site 2 Capabilities FECC NumberOfSources: 5
** end
```

xStatus Conference Site [1..n] Capabilities FECC Source [1..n] SourceId

Shows the ID of an input source that can be connected at a far end site.

Value space of the result returned:

    <Integer>

Example:

```
xStatus Conference Site 2 Capabilities FECC Source 1 SourceId
*s Conference Site 2 Capabilities FECC Source 1 SourceId: 6
** end
```

xStatus Conference Site [1..n] Capabilities FECC Source [1..n] Name

Shows the name of an input source that can be connected at a far end site.

Value space of the result returned:

    <String>

Example:

```
xStatus Conference Site 2 Capabilities FECC Source 1 Name
*s Conference Site 2 Capabilities FECC Source 1 Name: "Main camera"
** end
```

xStatus Conference Site [1..n] Capabilities FECC Source [1..n] Options

Shows available options for an input source that can be connected at a far end site (for a camera: p=pan; t=tilt; z=zoom; f=focus).

Value space of the result returned:

    <String>

Example:

```
xStatus Conference Site 2 Capabilities FECC Source 1 Options
*s Conference Site 2 Capabilities FECC Source 1 Options: "ptzf"
** end
```

xStatus Conference Site [1..n] Capabilities FECC Mode

Shows whether or not you have permission to control the input sources at a far end site.

On: Far end input source control is permitted.

Off: Far end input source control is not permitted.

Value space of the result returned:

    <On/Off>

Example:

```
xStatus Conference Site 2 Capabilities FECC Mode
*s Conference Site 2 Capabilities FECC Mode: On
** end
```

xStatus Conference Site [1..n] Capabilities Presentation

Lists the presentation capabilities for other participants in the conference.

Value space of the result returned:

<True/False>

Example:

```
xStatus Conference Site 2 Capabilities Presentation
*s Conference Site 2 Capabilities Presentation: True
** end
```

xStatus Conference Site [1..n] MicrophonesMuted

Lists the audio mute status for other participants in the conference.

Value space of the result returned:

<True/False>

Example:

```
xStatus Conference Site 2 MicrophonesMuted
*s Conference Site 2 MicrophonesMuted: True
** end
```

xStatus Conference Site [1..n] Manufacturer

Shows the manufacturer of the video system at a far end site.

Value space of the result returned:

<String>

Example:

```
xStatus Conference Site 2 Manufacturer
*s Conference Site 2 Manufacturer: "Cisco"
** end
```

xStatus Conference Site [1..n] SoftwareID

Shows the ID of the software running of the video system at a far end site.

Value space of the result returned:

<String>

Example:

```
xStatus Conference Site 2 SoftwareID
*s Conference Site 2 SoftwareID: "TC5"
** end
```

xStatus Conference Site [1..n] BlackScreenCause

Shows the reason why the screen is black while in a CTMS (Cisco TelePresence Multipoint Switch) managed conference.

FirstParticipant: You are the first and only participant in the meeting.

LastParticipant: You are the only remaining participant in the meeting.

NoResources: The conference started with insufficient resources.

EarlyJoin: The scheduled conference hast not started yet.

HostNotJoined: The conference is waiting for the host to join.

SecurityIssues: The video is blocked due to security issues.

NoVideo: It is a conference without video.

WebexOnly: It is a Webex only conference.

Value space of the result returned:

<None/Other/FirstParticipant/LastParticipant/NoResources/EarlyJoin/HostNotJoined/SecurityIssues/NoVideo/WebexOnly>

Example:

```
xStatus Conference Site 2 BlackScreenCause
*s Conference Site 17 BlackScreenCause: None
** end
```

xStatus Conference Site [1..n] ConferenceExtended

Shows the status of conference extension (only relevant for conferences scheduled from CTS-MAN (Cisco TelePresence Manager); they can be extended only once).

Unsupported: Conference extension is not supported.

NotExtended: The conference is not (yet) extended.

Extended: The conference has been extended, and is currently in the extended phase.

Value space of the result returned:

<Unsupported/NotExtended/Extended>

Example:

```
xStatus Conference Site 2 ConferenceExtended
*s Conference Site 2 ConferenceExtended: NotExtended
** end
```

xStatus Conference Site [1..n] BookingId

Shows the booking ID of a conference (if assigned). The booking ID can be used for easy identification of a call or conference.

Value space of the result returned:

    <String>

Example:

```
xStatus Conference Site 2 BookingId
*s Conference Site 33 BookingId: "MyConference"
** end
```

xStatus Conference Site [1..n] SecurityStatus

Security status of a call represents the least of the security statuses of all the endpoints involved in a CUCM call.

Invalid: Initial status.

Unknown: In the absence of the Call-Info header, that CUCM uses to pass this status to the phone, the phone assumes a value of Unknown.

NotAuthenticated: The current connected call is not signaling authenticated. This call is not secure.

Authenticated: The current connected call is signaling authenticated. This is not a fully secured call.

Encrypted: Encrypted indicates the current connected call is signaling/media encrypted. This call is secure.

Value space of the result returned:

    <Invalid/Unknown/Authenticated/NotAuthenticated>

Example:

```
xStatus Conference Site SecurityStatus
*s Conference Site 21 SecurityStatus: Unknown
** end
```

xStatus Conference Site [1..n] UnattendedTransfer

Indicates whether Unattended Transfer is available for the call.

Value space of the result returned:

    <True/False>

Example:

```
xStatus Conference Site UnattendedTransfer
*s Conference Site 14 UnattendedTransfer: True
** end
```

xStatus Conference Site [1..n] AttendedTransfer

Indicates whether Attended Transfer is available for the call.

Value space of the result returned:

    <True/False>

Example:

```
xStatus Conference Site AttendedTransfer
*s Conference Site 14 AttendedTransfer: True
** end
```

xStatus Conference Site [1..n] Appearance

Returns identification for the shared line appearance in the call.

Value space of the result returned:

    <Integer>

Example:

```
xStatus Conference Site Appearance
*s Conference Site 14 Appearance: 0
** end
```

xStatus Conference Site [1..n] GCI

Returns Global Call Identification (GCI) of the call. The GCI is used to identify whether the call is remote or local and it is provided by CUCM.

Value space of the result returned:

    <String>

Example:

```
xStatus Conference Site GCI
*s Conference Site 14 GCI: ""
** end
```

xStatus Conference Site [1..n] CalText

Returns the Confidential Access Level (CAL) of the session sent by vendors that support AS-SIP.

Value space of the result returned:

    <String>

Example:

```
xStatus Conference Site CalText
*s Conference Site 14 CalText: ""
** end
```

xStatus Conference Site [1..n] Hold

Indicates whether the far-end site can be placed on hold or not.

Value space of the result returned:

    <True/False>

Example:

```
xStatus Conference Site Hold
*s Conference Site 2 Hold: True
** end
```

xStatus Conference Site [1..n] Preserved

Indicates if the far end site is having connection issues, and has been preserved to avoid invoking features that can break the call.

Value space of the result returned:

    <Off/Local/Remote>

Example:

```
xStatus Conference Site Preserved
*s Conference Site 3 Preserved: Off
** end
```

xStatus Conference Line [1..n] Mode

Indicates whether the system is configured as private or shared line on CUCM.

Value space of the result returned:

    <Shared/Private>

Example:

```
xStatus Conference Line Mode
*s Conference Line 1 Mode: Private
** end
```

xStatus Conference Line [1..n] Appearance Status

Returns the status of the shared line call.

Value space of the result returned:

    <Idle/Private/Trying/Progressing/Alerting/Active/Held/HeldPrivate/Unknown>

Example:

```
xStatus Conference Line 1 Appearance Status
*s Conference Line 1 Appearance 5464 Status: Held
** end
```

xStatus Conference Line [1..n] Appearance URI

Returns the URI of the shared line call. Not visible for a private call.

Value space of the result returned:

    <String>

Example:

```
xStatus Conference Line 1 Appearance URI
*s Conference Line 1 Appearance 5464 URI: "12345@cisco.com"
** end
```

xStatus Conference Line [1..n] Appearance GCI

Returns Global Call Identification (GCI) of the shared line call. The GCI is used to identify whether the call is remote or local and it is provided by CUCM.

Value space of the result returned:

    <String>

Example:

```
xStatus Conference Line 1 Appearance GCI
*s Conference Line 1 Appearance 5464 GCI: "1-161386"
** end
```

xStatus Conference Multipoint Mode

Shows how the Multipoint video conferences are handled. See xConfiguration Conference Multipoint Mode for more information.

Value space of the result returned:

    <Off/MultiWay/MultiSite/Auto/CUCMMediaResourceGroupList>

Example:

```
xStatus Conference Multipoint Mode
*s Conference Multipoint Mode: "MultiWay"
** end
```

xStatus Conference DoNotDisturb

Shows whether DoNotDisturb mode is switched on or not.

Value space of the result returned:

<Active/Inactive>

Example:

```
xStatus Conference DoNotDisturb
*s Conference DoNotDisturb: Inactive
** end
```

xStatus Conference ActiveSpeaker Mode

Shows whether the ActiveSpeaker mode is switched on or not.

Value space of the result returned:

<On/Off>

Example:

```
xStatus Conference ActiveSpeaker Mode
*s Conference ActiveSpeaker Mode: Off
** end
```

xStatus Conference ActiveSpeaker SiteId

Returns the SiteId (CallId) of the current active speaker.

Value space of the result returned:

<Integer>

Example:

```
xStatus Conference ActiveSpeaker SiteId
*s Conference ActiveSpeaker SiteId: 0
** end
```

Diagnostics status

xStatus Diagnostics

Shows the top level overview of the diagnostics. The example shows the status for an ongoing call. The identities of the call and channels are used when querying additional information.

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 Jitter

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Jitter

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Jitter

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Jitter

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Jitter

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Jitter

Shows the jitter at the present moment in the incoming/outgoing channel, as specified by RFC 3550.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
Jitter
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Jitter:
0
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 Packets

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Packets

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Packets

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Packets

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Packets

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Packets
Shows the number of packets received/sent in the incoming/outgoing channels.

Value space of the result returned:
<Integer>

Example:
```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
Packets
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Packets:
405
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 Loss

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Loss

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Loss

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Loss

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Loss

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Loss
Shows the number of packets lost in the incoming/outgoing channels.

Value space of the result returned:
<Integer>

Example:
```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
Loss
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Loss: 96
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 LastIntervalLost

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 LastIntervalLost

Shows the number of packets lost during the last interval for the incoming/outgoing channels.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels IncomingDataChannel 327 Netstat 1
LastIntervalLost
*s Diagnostics Call 27 Channels IncomingDataChannel 327 Netstat 1
LastIntervalLost: 0
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 LastIntervalReceived

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 LastIntervalReceived

Shows the number of packets received during the last interval for the incoming/outgoing channels.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels IncomingDataChannel 327 Netstat 1
LastIntervalReceived
*s Diagnostics Call 27 Channels IncomingDataChannel 327 Netstat 1
LastIntervalReceived: 84
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 Drop

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Drop

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Drop

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Drop

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Drop

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Drop

Shows the number of packets dropped in the incoming/outgoing channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
Drop
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Drop: 0
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 Bytes

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 Bytes

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 Bytes

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 Bytes

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 Bytes

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 Bytes

Shows the number of bytes received/sent in the incoming/outgoing channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
Bytes
*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1 Bytes:
129920
** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 ChannelRate

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 ChannelRate

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 ChannelRate

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 ChannelRate

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 ChannelRate

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 ChannelRate

Shows the bandwidth for the incoming/outgoing channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
ChannelRate

*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
ChannelRate: 128000

** end
```

xStatus Diagnostics Call [1..n] Channels IncomingAudioChannel [1..n] Netstat 1 MaxJitter

xStatus Diagnostics Call [1..n] Channels IncomingVideoChannel [1..n] Netstat 1 MaxJitter

xStatus Diagnostics Call [1..n] Channels IncomingDataChannel [1..n] Netstat 1 MaxJitter

xStatus Diagnostics Call [1..n] Channels OutgoingAudioChannel [1..n] Netstat 1 MaxJitter

xStatus Diagnostics Call [1..n] Channels OutgoingVideoChannel [1..n] Netstat 1 MaxJitter

xStatus Diagnostics Call [1..n] Channels OutgoingDataChannel [1..n] Netstat 1 MaxJitter

Shows the maximum jitter that has been measured during last time interval (5 seconds).

Value space of the result returned:

<Integer>

Example:

```
xStatus Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
MaxJitter

*s Diagnostics Call 27 Channels OutgoingDataChannel 327 Netstat 1
MaxJitter: 0

** end
```

H320 status

xStatus H320 Gateway Status

Returns the state of the H320 Gateway, if the codec is paired with an ISDN Link.

Value space of the result returned:

<OK/OKWithWarning/Error/Inactive/Warning>

Example:

```
xStatus H320 Gateway status
*s H320 Gateway Status: Error
** end
```

xStatus H320 Gateway Address

Returns the IPv4 address of the ISDN Gateway, if the endpoint is paired to one.

Value space of the result returned:

<OK/OKWithWarning/Error/Inactive>

Example:

```
xStatus H320 Gateway Status
*s H320 Gateway Status: Inactive
** end
```

xStatus H320 Gateway Number

Returns the IPv6 address of the ISDN Gateway if the endpoint is paired to one.

Value space of the result returned:

<String>

Example:

```
xStatus H320 Gateway Number
*s H320 Gateway Number: ""
** end
```

xStatus H320 Gateway Mode

Returns information on the type of calls the ISDN Gateway is configured for, if the codec is paired with an ISDN Link.

Value space of the result returned:

<Unknown/PRI/BRI/External/G703>

Example:

```
xStatus H320 Gateway Mode
*s H320 Gateway Mode: Unknown
** end
```

xStatus H320 Gateway Reason

Shows the reason for rejected Gateway registration. Only available if the codec is connected to an ISDN Link.

Value space of the result returned:

<String>

Example:

```
xStatus H320 Gateway Reason
*s H320 Gateway Reason: ""
** end
```

xStatus H320 Gateway Id

Returns the unique identification of the H320 Gateway, if the codec is paired with an ISDN Link. .

Value space of the result returned:

<String>

Example:

```
xStatus H320 Gateway Id
*s H320 Gateway Id: "00:50:60:0B:EF:11"
** end
```

H323 status

xStatus H323

Shows the top level overview of the H323 status.

xStatus H323 Gatekeeper Status

Shows the gatekeeper registration status.

Value space of the result returned:

<Required/Discovering/Discovered/Authenticating/Authenticated/Registering/Registered/Inactive/Rejected>

Example:

```
xStatus H323 Gatekeeper Status
*s H323 Gatekeeper Status: Registered
** end
```

xStatus H323 Gatekeeper Address

Displays the IP address of the gatekeeper where the system is registered.

Value space of the result returned:

<String>

Example:

```
xStatus H323 Gatekeeper Address
*s H323 Gatekeeper Address: "192.0.1.20"
** end
```

xStatus H323 Gatekeeper Port

Shows the port which is used when connecting to on the gatekeeper.

Value space of the result returned:

<Integer>

Example:

```
xStatus H323 Gatekeeper Port
*s H323 Gatekeeper Port: 1719
** end
```

xStatus H323 Gatekeeper Reason

Shows the reason for rejected registration.

Value space of the result returned:

<String>

Example:

```
xStatus H323 Gatekeeper Reason
*s H323 Gatekeeper Reason: ""
** end
```

HttpFeedback status

xStatus HttpFeedback

Shows the top level overview of the HTTP status.

xStatus HttpFeedback [1..4] URL

Shows the URL (Uniform Resource Locator) of the HTTP server. There can be up to three HTTP servers, specified by the URL.

Value space of the result returned:

    <String>

Example:

```
xStatus HttpFeedback 1 URL
*s HttpFeedback 1 URL: "http://tms.company.com/tms/public/feedback/code.aspx"
** end
```

xStatus HttpFeedback [1..4] Expression [1..15]

Shows the feedback from the HTTP server. There can be up to 15 expressions for each URL. See the xCommand HttpFeedback commands for more information.

Value space of the result returned:

    <String>

ICE status

xStatus ICE Configured

Shows the ICE configuration status. ICE (Interactive Connectivity Establishment, RFC 5245) is a NAT traversal solution that the endpoints can use to discover the optimized media path. Thus the shortest route for audio and video is always secured between the endpoints.

Value space of the result returned:

    <On/Off>

Example:

```
xStatus ICE Configured
*s ICE Configured: "Off"
** end
```

xStatus ICE Defaultcandidate

Returns the default candidate where the endpoint initially receives media.

Value space of the result returned:

    <Host/Rflx/Relay>

Example:

```
xStatus ICE Defaultcandidate
*s ICE Defaultcandidate: "Host"
** end
```

xStatus ICE Turn IP

Returns the IP address of the TURN server that is currently in use.

Value space of the result returned:

    <String>

Example:

```
xStatus ICE Turn IP
*s ICE Turn IP: "192.0.1.20"
** end
```

xStatus ICE Turn Hostname

Returns the hostname, or IP address, that is configured as the TURN server.

Value space of the result returned:

<String>

Example:

```
xStatus ICE Turn Hostname
*s ICE Turn Hostname: "callway.medianetworkservices.com"
** end
```

xStatus ICE Turn Username

Returns the user name used for accessing the TURN server.

Value space of the result returned:

<String>

Example:

```
xStatus ICE Turn Username
*s ICE Turn Username: "username"
** end
```

xStatus ICE Call Result

Shows the state of the ICE routing in the present call.

Succeeded: The ICE routing is in use.

Failed: The ICE routing has failed.

Unsupported: ICE routing is not supported.

Mangled: ICE cannot route the media, because a server, such as VCS, is set to handle it.

Value space of the result returned:

<Succeeded/Failed/Unsupported/Mangled>

Example:

```
xStatus ICE Call Result
*s ICE Call 0 Result: "Unsupported"
** end
```

xStatus ICE Call Local Candidate

Returns the network interface type the local system is using to send media.

HOST: The endpoint sends media from its own IP address.

PRFLX: The endpoint sends media from its public IP address as seen by the remote Peer.

SRFLX: The endpoint sends media from its public IP address as seen by the TURN server.

RELAY: The endpoint sends media from the IP address and port allocated on the TURN server, and is used as a fallback until ICE has concluded.

Value space of the result returned:

<HOST/PRFLX/SRLFX/RELAY>

Example:

```
xStatus ICE Call Local Candidate
*s ICE Call 1 Local Candidate: "HOST"
** end
```

xStatus ICE Call Local IP

Returns the IP address the local system is using for media routing in the present call.

Value space of the result returned:

<String>

Example:

```
xStatus ICE Call Local IP
*s ICE Call 1 Local IP: "192.0.1.20"
** end
```

xStatus ICE Call Remote Candidate

Returns the network interface type the remote system is using to send media.

HOST: The far-end receives media on its own IP address.

PRFLX: The far-end receives media on its public IP address as seen by the remote Peer.

SRFLX: The far-end receives media on its public IP address as seen by the TURN server.

RELAY: The far-end receives media on the IP address and port allocated on the TURN server, and is used as a fallback until ICE has concluded.

Value space of the result returned:

<HOST/PRFLX/SRFLX/RELAY>

Example:

```
xStatus ICE Call Remote Candidate
*s ICE Call 1 Remote Candidate: "SRFLX"
** end
```

xStatus ICE Call Remote IP

Returns the IP address the remote system is using for media routing in the present call.

Value space of the result returned:

<String>

Example:

```
xStatus ICE Call Remote IP
*s ICE Call 1 Remote IP: "192.0.1.20"
** end
```

MediaChannels status

xStatus MediaChannels

Shows the top level overview of the media channel status. The example shows the status for an ongoing call. The identities of the call and channels are used when querying additional information.

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Encryption Status

Shows the encryption status on the incoming channel.

Value space of the result returned:

<On/Off>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Encryption Status
*s MediaChannels Call 27 IncomingAudioChannel 327 Encryption Status: Off
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Audio Protocol

Shows the audio algorithm for the incoming audio channel.

AACLD: The AAC-LD is an MPEG-4 Low Delay Audio Coder audio compression format.

G722: The G.722 algorithm is an ITU standard.

G7221: The G.722.1 algorithm is a licensed royalty-free ITU-T standard.

G711Mu: The G.711 Mu-law compression algorithm is used in North America and Japan.

G711A: The G.711 A-law compression algorithm is used in Europe and the rest of the world.

G729: The G.729 algorithm is an ITU standard.

G729AB: The G.729 annex A and B is an ITU standard.

Value space of the result returned:

<AACLD/G722/G7221/G711Mu/G711A/G729/G729AB>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Audio Protocol
*s MediaChannels Call 27 IncomingAudioChannel 327 Audio Protocol: AACLD
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Audio Mute

Audio mute status of incoming audio.

Value space of the result returned:

<True/False>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Audio Mute
*s MediaChannels Call 27 IncomingAudioChannel 327 Audio Mute: True
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Audio Channels

Shows the number of incoming audio channels.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Audio Channels
*s MediaChannels Call 27 IncomingAudioChannel 327 Audio Channels: 1
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTP Local IpAddress

Shows the local IP address of the Real-time Transport Protocol (RTP) port for the incoming audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP
Local IpAddress
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTP Local Port

Shows the local UDP port number of the Real-time Transport Protocol (RTP) port for the incoming audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP
Local Port
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Local
Port: 16404
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTP Remote IpAddress

Shows the remote IP address of the Real-time Transport Protocol (RTP) port for the incoming audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP
Remote IpAddress
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTP Remote Port

Shows the remote UDP port number of the Real-time Transport Protocol (RTP) port for the incoming audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP
Remote Port
*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTP Remote
Port: 50932
** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTCP Local IpAddress

Shows the local IP address of the Real-time Transport Control Protocol (RTCP) port for the incoming audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP
Local IpAddress

*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP Local
IpAddress: "192.168.24.190"

** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTCP Local Port

Shows the local UDP port number of the Real-time Transport Control Protocol (RTCP) port for the incoming audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP
Local Port

*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP Local
Port: 16405

** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RTCP Remote IpAddress

Shows the remote IP address of the Real-time Transport Control Protocol(RTCP) port for the incoming audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP
Remote IpAddress

*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP Remote
IpAddress: "192.168.136.130"

** end
```

xStatus MediaChannels Call [1..n] IncomingAudioChannel [1..n] Transport RCTP Remote Port

Shows the remote UDP port number of the Real-time Transport Control Protocol(RTCP) port for the incoming audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP
Remote Port

*s MediaChannels Call 27 IncomingAudioChannel 327 Transport RTCP Remote
Port: 50933

** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Encryption Status

Shows the encryption status on the incoming channel.

Value space of the result returned:

<On/Off>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Encryption Status

*s MediaChannels Call 27 IncomingVideoChannel 330 Encryption Status: Off

** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] ChannelRole

Shows if the incoming channel is the main video channel or presentation channel.

Value space of the result returned:

<Main/Presentation/Legacy>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 ChannelRole

*s MediaChannels Call 27 IncomingVideoChannel 330 ChannelRole: Main

** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video Protocol

Shows the video algorithm for the incoming video channel.

H264: The H.264 algorithm is an ITU-T standard for video compression.

H263pp: The H.263pp algorithm is an ITU-T standard for video compression.

H263: The H.263 algorithm is an ITU-T standard for video compression.

H261: The H.261 algorithm is an ITU-T standard for video compression.

Value space of the result returned:

<Off/Raw/H261/H263/H263p/H263pp/H264/MPV4/JPEG/HEVC>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video Protocol
*s MediaChannels Call 27 IncomingVideoChannel 330 Video Protocol: H264
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video FrameRate

Shows the video frame rate of the incoming channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video FrameRate
*s MediaChannels Call 27 IncomingVideoChannel 330 Video FrameRate: 25
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video ResolutionX

Shows the width (resolution in direction X) of the incoming video.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionX
*s MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionX: 768
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Video ResolutionY

Shows the height (resolution in direction Y) of the incoming video.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionY
*s MediaChannels Call 27 IncomingVideoChannel 330 Video ResolutionY: 448
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTP Local IpAddress

Shows the local IP address of the Real-time Transport Protocol (RTP) port for the incoming video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP
Local IpAddress
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTP Local Port

Shows the local UDP port number of the Real-time Transport Protocol (RTP) port for the incoming video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP
Local Port
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Local
Port: 16404
** end
```

143

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTP Remote IpAddress

Shows the remote IP address of the Real-time Transport Protocol (RTP) port for the incoming video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP
Remote IpAddress
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTP Remote Port

Shows the remote UDP port number of the Real-time Transport Protocol (RTP) port for the incoming video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP
Remote Port
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTP Remote
Port: 50932
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTCP Local IpAddress

Shows the local IP address of the Real-time Transport Control Protocol (RTCP) port for the incoming video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP
Local IpAddress
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTCP Local Port

Shows the local UDP port number of the Real-time Transport Control Protocol (RTCP) port for the incoming video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP
Local Port
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP Local
Port: 16405
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RTCP Remote IpAddress

Shows the remote IP address of the Real-time Transport Control Protocol(RTCP) port for the incoming video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP
Remote IpAddress
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] IncomingVideoChannel [1..n] Transport RCTP Remote Port

Shows the remote UDP port number of the Real-time Transport Control Protocol(RTCP) port for the incoming video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP
Remote Port
*s MediaChannels Call 27 IncomingVideoChannel 330 Transport RTCP Remote
Port: 50933
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Encryption Status

Shows the encryption status on the outgoing channel.

Value space of the result returned:

<On/Off>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Encryption Status
*s MediaChannels Call 27 OutgoingAudioChannel 328 Encryption Status: Off
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Audio Protocol

Shows the audio algorithm for the outgoing audio channel.

AACLD: The AAC-LD is an MPEG-4 Low Delay Audio Coder audio compression format.

G722: The G.722 algorithm is an ITU standard.

G7221: The G.722.1 algorithm is a licensed royalty-free ITU-T standard.

G711Mu: The G.711 Mu-law compression algorithm is used in North America and Japan.

G711A: The G.711 A-law compression algorithm is used in Europe and the rest of the world.

G729: The G.729 algorithm is an ITU standard.

G729AB: The G.729 annex A and B is an ITU standard.

Value space of the result returned:

<AACLD/G722/G7221/G711Mu/G711A/G729/G729AB>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Audio Protocol
*s MediaChannels Call 27 OutgoingAudioChannel 328 Audio Protocol: AACLD
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Audio Channels

Shows the number of outgoing audio channels.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Audio Channels
*s MediaChannels Call 27 OutgoingAudioChannel 328 Audio Channels: 1
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTP Local IpAddress

Shows the local IP address of the Real-time Transport Protocol (RTP) port for the outgoing audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP
Local IpAddress
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTP Local Port

Shows the local UDP port number of the Real-time Transport Protocol (RTP) port for the outgoing audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP
Local Port
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Local
Port: 16404
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTP Remote IpAddress

Shows the remote IP address of the Real-time Transport Protocol (RTP) port for the outgoing audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP
Remote IpAddress
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTP Remote Port

Shows the remote UDP port number of the Real-time Transport Protocol (RTP) port for the outgoing audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP
Remote Port
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTP Remote
Port: 50932
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTCP Local IpAddress

Shows the local IP address of the Real-time Transport Control Protocol (RTCP) port for the outgoing audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP
Local IpAddress
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTCP Local Port

Shows the local UDP port number of the Real-time Transport Control Protocol (RTCP) port for the outgoing audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP
Local Port
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP Local
Port: 16405
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RTCP Remote IpAddress

Shows the remote IP address of the Real-time Transport Control Protocol(RTCP) port for the outgoing audio in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP
Remote IpAddress
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] OutgoingAudioChannel [1..n] Transport RCTP Remote Port

Shows the remote UDP port number of the Real-time Transport Control Protocol(RTCP) port for the outgoing audio in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP
Remote Port
*s MediaChannels Call 27 OutgoingAudioChannel 328 Transport RTCP Remote
Port: 50933
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Encryption Status

Shows the encryption status on the outgoing channel.

Value space of the result returned:

<On/Off>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Encryption Status
*s MediaChannels Call 27 OutgoingVideoChannel 331 Encryption Status: Off
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] ChannelRole

Shows if the outgoing channel is the main video channel or presentation channel.

Value space of the result returned:

<Main/Presentation/Legacy>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole
*s MediaChannels Call 27 OutgoingVideoChannel 331 ChannelRole: Main
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video Protocol

Shows the video algorithm for the outgoing video channel.

H264: The H.264 algorithm is an ITU-T standard for video compression.

H263pp: The H.263pp algorithm is an ITU-T standard for video compression.

H263: The H.263 algorithm is an ITU-T standard for video compression.

H261: The H.261 algorithm is an ITU-T standard for video compression.

Value space of the result returned:

<Off/Raw/H261/H263/H263p/H263pp/H264/MPV4/JPEG/HEVC>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video Protocol
*s MediaChannels Call 27 OutgoingVideoChannel 331 Video Protocol: "H264"
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video FrameRate

Shows the video frame rate of the outgoing channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video FrameRate
*s MediaChannels Call 27 OutgoingVideoChannel 331 Video FrameRate: 30
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video ResolutionX

Shows the width (resolution in direction X) of the outgoing video.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionX
*s MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionX: 768
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Video ResolutionY

Shows the height (resolution in direction Y) of the outgoing video.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionY
*s MediaChannels Call 27 OutgoingVideoChannel 331 Video ResolutionY: 448
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTP Local IpAddress

Shows the local IP address of the Real-time Transport Protocol (RTP) port for the outgoing video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP
Local IpAddress
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTP Local Port

Shows the local UDP port number of the Real-time Transport Protocol (RTP) port for the outgoing video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP
Local Port
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Local
Port: 16404
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTP Remote IpAddress

Shows the remote IP address of the Real-time Transport Protocol (RTP) port for the outgoing video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP
Remote IpAddress
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTP Remote Port

Shows the remote UDP port number of the Real-time Transport Protocol (RTP) port for the outgoing video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP
Remote Port
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTP Remote
Port: 50932
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTCP Local IpAddress

Shows the local IP address of the Real-time Transport Control Protocol (RTCP) port for the outgoing video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP
Local IpAddress
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Local
IpAddress: "192.168.24.190"
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTCP Local Port

Shows the local UDP port number of the Real-time Transport Control Protocol (RTCP) port for the outgoing video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP
Local Port
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Local
Port: 16405
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RTCP Remote IpAddress

Shows the remote IP address of the Real-time Transport Control Protocol (RTCP) port for the outgoing video in the media channel.

Value space of the result returned:

<String>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP
Remote IpAddress
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Remote
IpAddress: "192.168.136.130"
** end
```

xStatus MediaChannels Call [1..n] OutgoingVideoChannel [1..n] Transport RCTP Remote Port

Shows the remote UDP port number of the Real-time Transport Control Protocol (RTCP) port for the outgoing video in the media channel.

Value space of the result returned:

<Integer>

Example:

```
xStatus MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP
Remote Port
*s MediaChannels Call 27 OutgoingVideoChannel 331 Transport RTCP Remote
Port: 50933
** end
```

Network status

xStatus Network

Shows the top level overview of the network status.

xStatus Network 1 Ethernet MacAddress

Shows the MAC (Media Access Control) address for the Ethernet interface.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 Ethernet MacAddress
*s Network 1 Ethernet MacAddress: "00:50:60:02:FD:C7"
** end
```

xStatus Network 1 Ethernet Speed

Shows the Ethernet speed in Mbps. The speed can be in full-duplex or half-duplex.

Value space of the result returned:

<"10half"/"10full"/"100half"/"100full"/"1000full">

Example:

```
xStatus Network 1 Ethernet Speed
*s Network 1 Ethernet Speed: "100full"
** end
```

xStatus Network 1 IPv4 Address

Shows the IPv4 address that uniquely identifies this system.

Value space of the result returned:

<String>

Example:

```
xStatus Network 1 IPv4 Address
*s Network 1 IPv4 Address: "192.0.2.149"
** end
```

xStatus Network 1 IPv4 SubnetMask

Shows the subnet mask which determines which subnet an IPv4 address belongs to.

Value space of the result returned:

    <String>

Example:

```
xStatus Network 1 IPv4 SubnetMask
*s Network 1 IPv4 SubnetMask: "255.255.255.0"
** end
```

xStatus Network 1 IPv4 Gateway

Shows the address of the IPv4 gateway.

Value space of the result returned:

    <String>

Example:

```
xStatus Network 1 IPv4 Gateway
*s Network 1 IPv4 Gateway: "192.0.2.10"
** end
```

xStatus Network 1 IPv4 DNS Domain Name

Shows the domain name.

Value space of the result returned:

    <String>

Example:

```
xStatus Network 1 IPv4 DNS Domain Name
*s Network 1 IPv4 DNS Domain Name: "www.example.com www.example.int"
** end
```

xStatus Network 1 IPv4 DNS Server [1..5] Address

Shows the IP address of the DNS server.

Value space of the result returned:

    <String>

Example:

```
xStatus Network 1 IPv4 DNS Server 1. Address
*s Network 1 IPv4 DNS Server 1 Address: "192.0.2.60"
** end
```

xStatus Network 1 IPv4 DHCP TftpServerAddress

Returns the IP address of the TFTP server (provisioning server) as assigned by DHCP, provided that the DHCP server supports option 150. This status is used only when xConfiguration Provisioning Mode is CUCM.

Value space of the result returned:

    <String>

Example:

```
xStatus Network 1 IPv4 DHCP TftpServerAddress
*s Network 1 IPv4 DHCP TftpServerAddress: "192.0.2.0"
** end
```

xStatus Network 1 IPv4 DHCP TmsServer

Returns the IP address of the TMS server (provisioning server) as assigned by DHCP, provided that the DHCP server supports option 242. This status is used only when xConfiguration Provisioning Mode is TMS.

Value space of the result returned:

    <String>

Example:

```
xStatus Network 1 IPv4 DHCP TmsServer
*s Network 1 IPv4 DHCP TmsServer: "192.0.2.0"
** end
```

xStatus Network 1 IPv4 DHCP ProvisioningServer

Returns the IP address of the VCS provisioning server as assigned by DHCP, provided the provisioning server address is defined as a vendor encapsulated option in the DHCP server. This status is used only when xConfiguration Provisioning Mode is VCS.

Value space of the result returned:

    <String>

Example:

```
xStatus Network 1 IPv4 DHCP ProvisioningServer
*s Network 1 IPv4 DHCP ProvisioningServer: "192.0.2.0"
** end
```

xStatus Network 1 IPv4 DHCP ProvisioningDomain

Returns the SIP domain of the VCS provisioning server as assigned by DHCP, provided the provisioning server address is defined as a vendor encapsulated option in the DHCP server. This status is used only when xConfiguration Provisioning Mode is VCS.

Value space of the result returned:

    <String>

Example:

```
xStatus Network 1 IPv4 DHCP ProvisioningDomain
*s Network 1 IPv4 DHCP ProvisioningDomain: "1234@company.com"
** end
```

xStatus Network 1 IPv4 DHCP TftpServer

Returns the IP address or DNS name of the TFTP server (provisioning server) as assigned by DHCP, provided the DHCP server sets the option 66. This status is used only when xConfiguration Provisioning Mode is CUCM.

Value space of the result returned:

    <String>

Example:

```
xStatus Network  IPv4 DHCP TftpServer
*s Network 1 IPv4 DHCP TftpServerAddress: "192.0.2.0"
*s Network 1 IPv4 DHCP TftpServer: "1234@company.com"
** end
```

xStatus Network 1 IPv6 Address

Shows the IPv6 address that uniquely identifies this system.

Value space of the result returned:

    <String>

Example:

```
xStatus Network 1 IPv6 Address
*s Network 1 IPv6 Address: ""
** end
```

xStatus Network 1 IPv6 Gateway

Shows the address of the IPv6 gateway.

Value space of the result returned:

    <String>

Example:

```
xStatus Network 1 IPv6 Gateway
*s Network 1 IPv6 Gateway: ""
** end
```

xStatus Network 1 MTU

Shows the MTU (Maximum Transmission Unit) size for the network.

Value space of the result returned:

    <Integer>

Example:

```
xStatus Network 1 MTU
*s Network 1 MTU: 1500
** end
```

xStatus Network 1 VLAN Voice VlanId

The feedback shows the VLAN Voice ID; or Off if the VLAN Voice Mode is not enabled.

Value space of the result returned:

    <"Off"/"1".."4094">

Example:

```
xStatus Network 1 VLAN Voice VlanId
*s Network 1 VLAN Voice VlanId: "Off"
** end
```

xStatus Network VLAN Native VlanId

Identifies the VLAN associated with any untagged traffic between the switch and the endpoint. For more information see documentation for the IEEE 802.1Q protocol.

Value space of the result returned:

    <String>

Example:

```
xStatus Network VLAN Native VlanId
*s Network 1 VLAN Native VlanId: "200"
** end
```

xStatus Network CDP Platform

Returns the hardware platform name of the switch connected to the endpoint.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP Platform
*s Network 1 CDP Platform: "cisco WS-C3750X-48P"
** end
```

xStatus Network CDP Version

Returns information about the software release version the switch is running.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP Version
*s Network 1 CDP Version: "Cisco IOS Software, C3750E Software (C3750E-
UNIVERSALK9-M), Version 15.0(1)SE2, RELEASE SOFTWARE (fc3)*Technical
Support: http://www.cisco.com/techsupport*Copyright (c) 1986-2011 by Cisco
Systems, Inc.*Compiled Thu 22-Dec-11 00:05 by prod_rel_team"
** end
```

xStatus Network CDP Capabilities

Describes the functional capability for the switch in form of a device type. See documentation for CDP protocol for more information.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP Capabilities
*s Network 1 CDP Capabilities: "0x0029"
** end
```

xStatus Network CDP DeviceId

Identifies the name of the switch in form of a character string.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP DeviceId
*s Network 1 CDP DeviceId: "123456.company.com"
** end
```

xStatus Network CDP PortID

Returns the identification the switch uses of the port the endpoint is connected to.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP PortID
*s Network 1 CDP PortID: "GigabitEthernet1/0/23"
** end
```

xStatus Network CDP Duplex

Indicates the status (duplex configuration) of the CDP broadcast interface. Used by network operators to diagnose connectivity problems between adjacent network elements.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP Duplex
*s Network 1 CDP Duplex: "Full"
** end
```

xStatus Network CDP VTPMgmtDomain

Returns the switch's configured VTP management domain name-string.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP VTPMgmtDomain
*s Network 1 CDP VTPMgmtDomain: "anyplace"
** end
```

xStatus Network CDP Address

Returns the first network address of both receiving and sending devices.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP Address
*s Network 1 CDP Address: "192.0.1.20"
** end
```

xStatus Network CDP PrimaryMgmtAddress

Returns the management address used to configure and monitor the switch the endpoint is connected to.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP PrimaryMgmtAddress
*s Network 1 CDP PrimaryMgmtAddress: "10.1.1.2"
** end
```

xStatus Network CDP SysName

Returns the SysName as configured in the switch the endpoint is connected to.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP SysName
*s Network 1 CDP SysName: ""
** end
```

xStatus Network CDP SysObjectID

Returns the SysObjectID as configured in the switch the endpoint is connected to.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP SysObjectID
*s Network 1 CDP SysObjectID: ""
** end
```

xStatus Network CDP VoIPApplianceVlanID

Identifies the VLAN used for VoIP traffic from the endpoint to the switch. For more information see documentation of the IEEE 802.1Q protocol.

Value space of the result returned:

<String>

Example:

```
xStatus Network CDP VoIPApplianceVlanID
*s Network 1 CDP VoIPApplianceVlanID: "300"
** end
```

NetworkServices status

xStatus NetworkServices NTP Address

Returns the address of the NTP server(s) the codec is using.

Value space of the result returned:

<String>

Example:

```
xStatus NetworkServices NTP Address
*s NetworkServices NTP Address: "64.104.193.12 64.104.222.16 144.254.15.121"
** end
```

Preset status

xStatus Preset

Shows the top level overview of the camera presets status.

xStatus Preset [1..15] Defined

Shows if a camera preset is stored at this position.

Value space of the result returned:

<True/False>

Example:

```
xStatus Preset 1 Defined
*s Preset 1 Defined: True
** end
```

xStatus Preset [1..15] Type

Shows the camera preset type.

Value space of the result returned:

<All/Camera>

Example:

```
xStatus Preset 1 Type
*s Preset 1 Type: All
** end
```

xStatus Preset [1..15] Description

Lists the configured name for the specific preset.

Value space of the result returned:

<String>

Example:

```
xStatus Preset 1 Description
*s Preset 1 Description: "Zoom in"
** end
```

Provisioning status

xStatus Provisioning

Shows the top level overview of the provisioning status.

xStatus Provisioning Status

Shows the status of the provisioning.

Failed: The provisioning failed.

AuthenticationFailed: The authentication failed.

Provisioned: The endpoint is provisioned.

Idle: The provisioning is not active.

NeedConfig: The endpoint needs to be configured.

Value space of the result returned:

    <Failed/AuthenticationFailed/Provisioned/Idle/NeedConfig>

Example:

```
xStatus Provisioning Status
*s Provisioning Status: Provisioned
** end
```

xStatus Provisioning Reason

Shows the cause when provisioning has failed.

Value space of the result returned:

    <String>

Example:

```
xStatus Provisioning Reason
*s Provisioning Reason: ""
** end
```

xStatus Provisioning Server

Identifies the server the system is provisioned to. This address may either be defined in the xConfiguration ExternalManager Address setting, or provided by DHCP.

Value space of the result returned:

    <String>

Example:

```
xStatus Provisioning Server
*s Provisioning Server: "192.0.2.0"
** end
```

xStatus Provisioning Software UpgradeStatus SessionId

Shows the ID of the session for the software upgrade.

Value space of the result returned:

    <String>

Example:

```
xStatus Provisioning Software UpgradeStatus SessionId
*s Provisioning Software UpgradeStatus SessionId: ""
** end
```

xStatus Provisioning Software UpgradeStatus LastChange

Shows the date and time for the latest software upgrade.

Value space of the result returned:

    <String>

Example:

```
xStatus Provisioning Software UpgradeStatus LastChange
*s Provisioning Software UpgradeStatus LastChange: "2011-06-07T07:20:03Z"
** end
```

xStatus Provisioning Software UpgradeStatus Status

Shows the status of the software upgrade.

Value space of the result returned:

    <None/InProgress/Failed/InstallationFailed/Succeeded>

Example:

```
xStatus Provisioning Software UpgradeStatus Status
*s Provisioning Software UpgradeStatus Status: None
** end
```

xStatus Provisioning Software UpgradeStatus Phase

Shows the phase of the software upgrade.

Value space of the result returned:

<None/DownloadPending/Downloading/DownloadPaused/DownloadDone/
AboutToInstallUpgrade/Postponed/Installing>

Example:

```
xStatus Provisioning Software UpgradeStatus Phase
*s Provisioning Software UpgradeStatus Phase: None
** end
```

xStatus Provisioning Software UpgradeStatus Message

Shows the system message for the software upgrade.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software UpgradeStatus Message
*s Provisioning Software UpgradeStatus Message: ""
** end
```

xStatus Provisioning Software UpgradeStatus VersionId

Shows the version ID of the software currently being uploaded and installed.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software UpgradeStatus VersionId
*s Provisioning Software UpgradeStatus VersionId: "s52000tc5_1_0.pkg"
** end
```

xStatus Provisioning Software UpgradeStatus URL

Shows the URL that the new software currently is being uploaded and installed from.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software UpgradeStatus URL
*s Provisioning Software UpgradeStatus URL: "http://.../s52000tc5_1_0.pkg"
** end
```

xStatus Provisioning Software UpgradeStatus SecondsUntilUpgrade

Indicates how many seconds remain before the software upgrade is automatically installed. The upgrade installation can be started manually by issuing the command xCommand Provisioning CompleteUpgrade or postponed with the command xCommand Provisioning PostponeUpgrade.

Value space of the result returned:

<Integer>

Example:

```
xStatus Provisioning Software UpgradeStatus SecondsUntilUpgrade
*s Provisioning Software UpgradeStatus SecondsUntilUpgrade: 0
** end
```

xStatus Provisioning Software Current VersionId

Shows the version ID of the current software.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software Current VersionId
*s Provisioning Software Current VersionId: "s52000tc5_1_0.pkg"
** end
```

xStatus Provisioning Software Current URL

Shows the URL that the current software was uploaded from.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software Current URL
*s Provisioning Software Current URL: "http://.../s52000tc5_1_0.pkg"
** end
```

xStatus Provisioning Software Current CompletedAt

Shows date and time for when the current software upgrade was completed.

Value space of the result returned:

<String>

Example:

```
xStatus Provisioning Software Current CompletedAt
*s Provisioning Software Current CompletedAt: "2011-06-07T07:20:03Z"
** end
```

xStatus Provisioning CUCM CAPF Mode

Shows the authentication mode required for a pending CAPF operation, if any.

IgnoreAuth: No pending or ongoing CAPF operation.

NullAuth: Pending or ongoing CAPF operation that does not require authentication.

StringAuth: Pending or ongoing CAPF operation that requires string authentication.

LSCAuth: Pending or ongoing CAPF operation that requires authentication by Locally Significant Certificate.

MICAUth: Pending or ongoing CAPF operation that requires authentication by Manufacturer Installed Certificate.

Value space of the result returned:

<IgnoreAuth/NullAuth/StringAuth/LSCAuth/MICAuth>

Example:

```
xStatus Provisioning CUCM CAPF Mode
*s Provisioning CUCM CAPF Mode: IgnoreAuth
** end
```

xStatus Provisioning CUCM CAPF ServerName

Shows the CAPF server name. This can be a hostname, a FQDN or an IP address.

Value space of the result returned:

<IgnoreAuth/NullAuth/StringAuth/LSCAuth/MICAuth>

Example:

```
xStatus Provisioning CUCM CAPF ServerName
*s Provisioning CUCM CAPF ServerName: "192.168.0.1"
** end
```

xStatus Provisioning CUCM CAPF ServerPort

Shows the port number to be used for CAPF operations.

Value space of the result returned:

<Integer>

Example:

```
xStatus Provisioning CUCM CAPF ServerPort
*s Provisioning CUCM CAPF ServerPort: 3804
** end
```

xStatus Provisioning CUCM CAPF LSC

Shows whether a Locally Significant Certificate (LSC) is installed or not.

Value space of the result returned:

<NotInstalled/Installed>

Example:

```
xStatus Provisioning CUCM CAPF LSC
*s Provisioning CUCM CAPF LSC: NotInstalled
** end
```

xStatus Provisioning CUCM CAPF OperationState

Shows the state of the current CAPF operation, if any.

Pending: A CAPF operation is pending.

NonPending: There is no CAPF operation in process.

InProgress: A CAPF operation is in progress.

Failed: The CAPF operation has failed.

Value space of the result returned:

<Pending/NonPending/InProgress/Failed>

Example:

```
xStatus Provisioning CUCM CAPF OperationState
*s Provisioning CUCM CAPF OperationState: NonPending
** end
```

xStatus Provisioning CUCM CAPF OperationResult

Shows the result of the last CAPF operation.

NotSet: No CAPF operation has been executed yet.

CAPFUnknown: Unknown CAPF failure.

CAPFInvalidAuthStrLen: Invalid length of authentication string (should be 4-10 digits).

CAPFInvalidInitReason: Invalid init reason specified.

CAPFInvalidAuthMode: Invalid authentication mode.

CAPFNotNone: Last CAPF session is still running.

CAPFNullCInt: Invalid client.

CAPFNullCIntcontext: Invalid client context.

CAPFInvalidUsageStartSession: Session is in wrong state.

CAPFNullSessionData: Session context data is empty.

CAPFInvalidSession: Invalid CAPF session.

CAPFNullParameters: Invalid server parameters.

CAPFNoStat: Empty status.

CAPFTimeout: The operation has timed out.

CAPFConnLost: Connection to the CAPF server has been lost.

CAPFAbort: CAPF operation has been terminated.

CAPFKeyGenFailed: Key generation has failed.

CAPFConnectFailed: Failed to connect to CAPF server.

CAPFFenceError: Unknown internal CAPF failure.

CAPFServerBusy: CAPF server is busy.

CAPFAuthRejected: Authentication has been rejected.

CAPFInvalidParms: Invalid parameters.

CAPFCancelled: CAPF operation has been canceled.

CAPFFailed: CAPF operation has failed.

CAPFSuccess: CAPF operation has succeeded.

CAPFLSCUpdated: Locally Significant Certificate (LSC) has been updated. CAPF operation has succeeded.

Value space of the result returned:

<NotSet/CAPFUnknown/CAPFInvalidAuthStrLen/CAPFInvalidInitReason/CAPFInvalidAuthMode/
CAPFNotNone/CAPFNullCInt/CAPFNullCIntcontext/CAPFInvalidUsageStartSession/
CAPFNullSessiondata/CAPFInvalidSession/CAPFNullParameters/CAPFNoStat/CAPFTimeout/
CAPFConnLost/CAPFAbort/CAPFKeyGenFailed/CAPFConnectFailed/CAPFFenceError/
CAPFServerBusy/CAPFAuthRejected/CAPFInvalidParms/CAPFCancelled/CAPFFailed/
CAPFSuccess/CAPFLSCUpdated>

Example:

```
xStatus Provisioning CUCM CAPF OperationResult
```

```
*s Provisioning CUCM CAPF OperationResult: NotSet
** end
```

xStatus Provisioning CUCM ProvisionSecurity

Shows the provisioned configuration file type.

None: The provisioned configuration file is plain text or the endpoint has not been provisioned.

Signed: The provisioned configuration file is signed.

Encrypted: The provisioned configuration file is signed and encrypted.

Value space of the result returned:

<None/Signed/Encrypted>

Example:

```
xStatus Provisioning CUCM ProvisionSecurity
*s Provisioning CUCM ProvisionSecurity: None
** end
```

xStatus Provisioning CUCM CTL State

Shows whether the Certificate Trust List (CTL) file is installed or not.

Value space of the result returned:

<NotInstalled/Installed>

Example:

```
xStatus Provisioning CUCM CTL State
*s Provisioning CUCM CTL State: Installed
** end
```

Security status

xStatus Security FIPS Mode

Shows the FIPS mode status.

Value space of the result returned:

<On/Off>

Example:

```
xStatus Security FIPS Mode
*s Security FIPS Mode: Off
** end
```

SIP status

xStatus SIP

Shows the top level overview of the SIP status.

xStatus SIP Proxy [1] Status

Shows the status of the communication between the endpoint and the SIP Proxy server.

Active: The communication between the endpoint and the SIP Proxy is active.

DNSFailed: The attempt to establish communication to the DNS server failed.

Off: There is no communication between the endpoint and the SIP Proxy.

Timeout: The attempt to establish communication to the SIP Proxy timed out.

UnableTCP: The system is unable to use TCP as the transport method.

UnableTLS: The system is unable to use TLS as the transport method.

Unknown: The status of the communication is not known.

Value space of the result returned:

<Active/DNSFailed/Off/Timeout/UnableTCP/UnableTLS/Unknown>

Example:

```
xStatus SIP Proxy 1 Status
*s SIP Proxy 1 Status: Active
** end
```

xStatus SIP Proxy [1] Address

Shows the address of the SIP Proxy that the system communicates with.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Proxy 1 Address
*s SIP Proxy 1 Address: "192.0.2.50"
** end
```

xStatus SIP Proxy [1] Secure

Shows the encryption status of the signaling with the SIP Proxy server.

Value space of the result returned:

<True/False>

Example:

```
xStatus SIP Proxy 1 Secure
*s SIP Proxy 1 Secure: True
** end
```

xStatus SIP Proxy [1] Verified

Shows whether or not the SSL certificate of the server that the video system / codec tries to register to is included in the codec's trusted CA-list. The server is typically a Cisco VCS or CUCM.

True: The server's SIP certificate is checked against the trusted CA-list on the codec and found valid. Additionally, the fully qualified domain name of the server matches the valid certificate.

False: A TLS connection is not set up because the SIP certificate verification failed or the domain name did not match. Note that the status also returns False when TLS is not used (xConfiguration SIP Profile 1 DefaultTransport not set to TLS) or certificate verification is switched off (xConfiguration SIP Profile TlsVerify set to Off).

Value space of the result returned:

<True/False>

Example:

```
xStatus SIP Proxy 1 Verified
*s SIP Proxy 1 Verified: False
** end
```

xStatus SIP Registration [1..n] Status

Shows the status of the registration to the SIP Proxy Server.

Deregister: The system is in the process of de-registering to the SIP Proxy.

Failed: The system failed to register to the SIP Proxy.

Inactive: The system is not registered to any SIP Proxy.

Registered: The system is registered to the SIP Proxy.

Registering: The system is in the process of registering to the SIP Proxy.

Value space of the result returned:

<Deregister/Failed/Inactive/Registered/Registering>

Example:

```
xStatus SIP Registration 1 Status
*s SIP Registration 1 Status: Registered
** end
```

xStatus SIP Registration [1..n] Reason

Shows a message to explain the reason why the SIP registration failed.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Registration 1 Reason
*s SIP Registration 1 Reason: "404 Not Found"
** end
```

xStatus SIP Registration [1..n] URI

Shows the URI used for registration to the SIP Proxy server.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Registration 1 URI
*s SIP Registration 1 URI: "firstname.lastname@company.com"
** end
```

xStatus SIP Registration [1..n] Authentication

Shows which authentication mechanism is used when registering to the SIP Proxy Server.

Digest: Uses the Digest access authentication method, as specified by RFC 2069.

NTLM: Uses the NTLM authentication method, which is a Microsoft authentication protocol.

Off: No authentication mechanism is used.

Value space of the result returned:

    <Digest/NTLM/Off>

Example:

```
xStatus SIP Registration 1 Authentication
*s SIP Registration 1 Authentication: Off
** end
```

xStatus SIP Profile 1 Proxy [1] Status

Shows the status of the communication between the endpoint and the SIP Proxy server.

Active: The communication between the endpoint and the SIP Proxy is active.

DNSFailed: The attempt to establish communication to the DNS server failed.

Off: There is no communication between the endpoint and the SIP Proxy.

Timeout: The attempt to establish communication to the SIP Proxy timed out.

UnableTCP: The system is unable to use TCP as the transport method.

UnableTLS: The system is unable to use TLS as the transport method.

Unknown: The status of the communication is not known.

Value space of the result returned:

    <Active/DNSFailed/Off/Timeout/UnableTCP/UnableTLS/Unknown>

Example:

```
xStatus SIP Profile 1 Proxy 1 Status
*s SIP Profile 1 Proxy 1 Status: Active
** end
```

xStatus SIP Profile 1 Proxy [1] Address

Shows the address of the SIP Proxy that the system communicates with.

Value space of the result returned:

    <String>

Example:

```
xStatus SIP Profile 1 Proxy 1 Address
*s SIP Profile 1 Proxy 1 Address: "192.0.2.50"
** end
```

xStatus SIP Profile 1 Secure

Shows the encryption status of the signaling with the SIP Proxy server.

Value space of the result returned:

    <True/False>

Example:

```
xStatus SIP Profile 1 Secure
*s SIP Profile 1 Secure: True
** end
```

xStatus SIP Profile 1 Verified

Shows whether or not the SSL certificate of the server that the video system / codec tries to register to is included in the codec's trusted CA-list. The server is typically a Cisco VCS or CUCM.

True: The server's SIP certificate is checked against the trusted CA-list on the codec and found valid. Additionally, the fully qualified domain name of the server matches the valid certificate.

False: A TLS connection is not set up because the SIP certificate verification failed or the domain name did not match. Note that the status also returns False when TLS is not used (xConfiguration SIP Profile 1 DefaultTransport not set to TLS) or certificate verification is switched off (xConfiguration SIP Profile TlsVerify set to Off).

Value space of the result returned:

    <True/False>

Example:

```
xStatus SIP Profile 1 Verified
*s SIP Profile 1 Verified: False
** end
```

xStatus SIP Profile 1 Authentication

Shows which authentication mechanism is used when registering to the SIP Proxy Server.

Digest: Uses the Digest access authentication method, as specified by RFC 2069.

NTLM: Uses the NTLM authentication method, which is a Microsoft authentication protocol.

Off: No authentication mechanism is used.

Value space of the result returned:

    <Digest/NTLM/Off>

Example:

```
xStatus SIP Profile 1 Authentication
*s SIP Profile 1 Authentication: Off
** end
```

xStatus SIP Profile 1 Mailbox MessagesWaiting

Indicates how many new messages are in the mailbox.

Value space of the result returned:

<Integer>

Example:

```
xStatus SIP Profile 1 Mailbox MessagesWaiting
*s SIP Profile 1 Mailbox MessagesWaiting: 0
** end
```

xStatus SIP Profile 1 CallForward Mode

Indicates whether the call forward mode for SIP is set to on or off.

Value space of the result returned:

<On/Off>

Example:

```
xStatus SIP Profile 1 CallForward Mode
*s SIP Profile 1 CallForward Mode: Off
** end
```

xStatus SIP Profile 1 CallForward URI

Indicates the address the incoming calls are directed to when call forward mode is set on.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile 1 CallForward URI
*s SIP Profile 1 CallForward URI: ""
** end
```

xStatus SIP Profile 1 CallForward DisplayName

Returns the URI that is displayed on the user interface for the forwarded call.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile 1 CallForward DisplayName
*s SIP Profile 1 CallForward DisplayName: ""
** end
```

xStatus SIP Profile 1 Registration [1..n] Status

Shows the status of the registration to the SIP Proxy Server.

Deregister: The system is in the process of de-registering to the SIP Proxy.

Failed: The system failed to register to the SIP Proxy.

Inactive: The system is not registered to any SIP Proxy.

Registered: The system is registered to the SIP Proxy.

Registering: The system is in the process of registering to the SIP Proxy.

Value space of the result returned:

<Deregister/Failed/Inactive/Registered/Registering>

Example:

```
xStatus SIP Profile 1 Registration 1 Status
*s SIP Profile 1 Registration 1 Status: Registered
** end
```

xStatus SIP Profile 1 Registration [1..n] Reason

Shows a message to explain the reason why the SIP registration failed.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile 1 Registration 1 Reason
*s SIP Profile 1 Registration 1 Reason: "404 Not Found"
** end
```

xStatus SIP Profile 1 Registration [1..n] URI

Shows the URI used for registration to the SIP Proxy server.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile 1 Registration 1 URI
*s SIP Profile 1 Registration 1 URI: "firstname.lastname@company.com"
** end
```

xStatus SIP Profile 1 Turn ServerAddress

Not in use in this release.

Value space of the result returned:

<String>

xStatus SIP Profile 1 DirectoryURI Primary URI

Returns the primary directory URI set for the endpoint in UCM. Directory URI is associated with a directory number, and can be used to make calls and identify callers. Up to five directory URIs can be set, but only one of them can be set as primary.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile DirectoryURI Primary URI
*s SIP Profile 1 DirectoryURI Primary URI: "123456@company.com"
** end
```

xStatus SIP Profile 1 DirectoryURI Primary URI

Returns the primary directory URI set for the endpoint in UCM. Directory URI is associated with a directory number, and can be used to make calls and identify callers. Up to five directory URIs can be set, but only one of them can be set as primary.

Value space of the result returned:

<String>

Example:

```
xStatus SIP Profile DirectoryURI Primary URI
*s SIP Profile 1 DirectoryURI Primary URI: "123456@company.com"
** end
```

Standby status

xStatus Standby Active

Shows if the system is in standby or not.

Value space of the result returned:

<On/Off>

Example:

```
xStatus Standby Active
*s Standby Active: Off
** end
```

SystemUnit status

xStatus SystemUnit

Shows the top level overview of the system unit status.

xStatus SystemUnit ProductType

Shows the product type.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit ProductType
*s SystemUnit ProductType: "Cisco TelePresence Codec"
** end
```

xStatus SystemUnit ProductId

Shows the product identity.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit ProductId
*s SystemUnit ProductId: "Cisco TelePresence Codec C90"
** end
```

xStatus SystemUnit ProductPlatform

Shows the product platform.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit ProductPlatform
*s SystemUnit ProductPlatform: "C90"
** end
```

xStatus SystemUnit Uptime

Shows the number of seconds since the last restart of the codec.

Value space of the result returned:

<Integer>

Example:

```
xStatus SystemUnit Uptime
*s SystemUnit Uptime: 597095
** end
```

xStatus SystemUnit Software Application

Shows which software application is running on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software Application
*s SystemUnit Software Application: "Endpoint"
** end
```

xStatus SystemUnit Software Version

Shows the software version installed on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software Version
*s SystemUnit Software Version: "TC5.1.0"
** end
```

xStatus SystemUnit Software Name

Shows the name of the software that is installed on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software Name
*s SystemUnit Software Name: "s52010"
** end
```

xStatus SystemUnit Software ReleaseDate

Shows the release date of the software installed on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software ReleaseDate
*s SystemUnit Software ReleaseDate: "2012-02-22"
** end
```

xStatus SystemUnit Software MaxVideoCalls

Shows the the maximum number of simultaneous video calls that is supported.

Value space of the result returned:

<Integer>

Example:

```
xStatus SystemUnit Software MaxVideoCalls
*s SystemUnit Software MaxVideoCalls: 3
** end
```

xStatus SystemUnit Software MaxAudioCalls

Shows the the maximum number of simultaneous audio calls that is supported.

Value space of the result returned:

<Integer>

Example:

```
xStatus SystemUnit Software MaxAudioCalls
*s SystemUnit Software MaxAudioCalls: 3
** end
```

xStatus SystemUnit Software ReleaseKey

Shows if there is a valid releasekey for the software version that is installed on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software ReleaseKey
*s SystemUnit Software ReleaseKey: "true"
** end
```

xStatus SystemUnit Software OptionKeys NaturalPresenter

Shows if the system has the option key installed that supports the NaturalPresenter functionality.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software OptionKeys NaturalPresenter
*s SystemUnit Software OptionKeys NaturalPresenter: "true"
** end
```

xStatus SystemUnit Software OptionKeys MultiSite

Shows if the system has the option key installed that supports the MultiSite functionality.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software OptionKeys MultiSite
*s SystemUnit Software OptionKeys MultiSite: "true"
** end
```

xStatus SystemUnit Software OptionKeys PremiumResolution

Shows if the system has the option key installed that supports the PremiumResolution functionality.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software OptionKeys PremiumResolution
*s SystemUnit Software OptionKeys PremiumResolution: "true"
** end
```

xStatus SystemUnit Software OptionKeys HighDefinition

Shows if the system has the option key installed that supports the HighDefinition functionality.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software OptionKeys HighDefinition
*s SystemUnit Software OptionKeys HighDefinition: "true"
** end
```

xStatus SystemUnit Software OptionKeys DualDisplay

Shows if the system has the option key installed that supports the DualDisplay functionality.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Software OptionKeys DualDisplay
*s SystemUnit Software OptionKeys DualDisplay: "true"
** end
```

xStatus SystemUnit Hardware Module SerialNumber

Shows the serial number of the hardware module in the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware Module SerialNumber
*s SystemUnit Hardware Module SerialNumber: "F9AA99A00090"
** end
```

xStatus SystemUnit Hardware Module Identifier

Shows the revision of the hardware module in the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware Module Identifier
*s SystemUnit Hardware Module Identifier: "1"
** end
```

xStatus SystemUnit Hardware MainBoard SerialNumber

Shows the serial number of the main board in the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware MainBoard SerialNumber
*s SystemUnit Hardware MainBoard SerialNumber: "PH0999989"
** end
```

xStatus SystemUnit Hardware MainBoard Identifier

Shows the revision of the main board in the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware MainBoard Identifier
*s SystemUnit Hardware MainBoard Identifier: "101400-5 [06]"
** end
```

xStatus SystemUnit Hardware BootSoftware

Shows the version of the boot software that is installed on the codec.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware BootSoftware
*s SystemUnit Hardware BootSoftware: "U-Boot 2010.04-30"
** end
```

xStatus SystemUnit Hardware MonitoringSoftware

The feedback shows the monitoring software id.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware MonitoringSoftware
*s SystemUnit Hardware MonitoringSoftware: "39"
** end
```

xStatus SystemUnit Hardware Monitoring Fan [1..n] Status

The feedback shows the speed (rpm) for the specified fan.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Hardware Monitoring Fan 1 Status
*s SystemUnit Hardware Monitoring Fan 1 Status: "locked on 1096 rpm"
** end
```

xStatus SystemUnit Hardware Temperature

The feedback shows the current maximum temperature (degree Celsius) measured in the codec/ system.

Value space of the result returned:

    <String>

Example:

```
xStatus SystemUnit Hardware Temperature
*s SystemUnit Hardware Temperature: "64.0"
** end
```

xStatus SystemUnit Hardware TemperatureThreshold

Returns information on the maximum temperature for the codec. If this temperature is exceeded the system automatically shuts down.

Value space of the result returned:

    <String>

Example:

```
xStatus SystemUnit Hardware TemperatureThreshold
*s SystemUnit Hardware TemperatureThreshold: "85"
** end
```

xStatus SystemUnit State System

Shows what state the system is in.

InCall: The system is in a call.

Initialized: The system is ready for use.

Initializing: The system is initializing.

Multisite: The system is in a Multisite conference.

Sleeping: The system is in sleep mode.

Value space of the result returned:

    <InCall/Initialized/Initializing/Multisite/Sleeping>

Example:

```
xStatus SystemUnit State System
*s SystemUnit State System: Initialized
** end
```

xStatus SystemUnit State MaxNumberOfCalls

Shows the the maximum number of simultaneous calls.

Value space of the result returned:

    <0..5>

Example:

```
xStatus SystemUnit State MaxNumberOfCalls
*s SystemUnit State MaxNumberOfCalls: 3
** end
```

xStatus SystemUnit State MaxNumberOfActiveCalls

Shows the the maximum number of simultaneous active calls. Calls that are set on hold/transfer are not counted as active.

Value space of the result returned:

    <0..5>

Example:

```
xStatus SystemUnit State MaxNumberOfActiveCalls
*s SystemUnit State MaxNumberOfActiveCalls: 3
** end
```

xStatus SystemUnit State NumberOfActiveCalls

Shows the number of active calls.

Value space of the result returned:

    <0..5>

Example:

```
xStatus SystemUnit State NumberOfActiveCalls
*s SystemUnit State NumberOfActiveCalls: 0
** end
```

xStatus SystemUnit State NumberOfSuspendedCalls

Shows the number of suspended calls.

Value space of the result returned:

    <0..5>

Example:

```
xStatus SystemUnit State NumberOfSuspendedCalls
*s SystemUnit State NumberOfSuspendedCalls: 0
** end
```

xStatus SystemUnit State NumberOfInProgressCalls

Shows the number of calls in progress.

Value space of the result returned:

<0..5>

Example:

```
xStatus SystemUnit State NumberOfInProgressCalls
*s SystemUnit State NumberOfInProgressCalls: 0
** end
```

xStatus SystemUnit ContactInfo

Returns the system's active contact information. This is the address which is used to reach this system.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit ContactInfo
*s SystemUnit ContactInfo: "firstname.lastname@company.com"
** end
```

xStatus SystemUnit ContactName

Returns the system's active contact name. The result depends on which protocol, if any, the system is registered on. Unlike Contact Info, Contact Name is not configurable and is automatically set by the system.

Value space of the result returned:

<String>

Example:

```
Not registered to a protocol
xStatus SystemUnit ContactName: "192.0.2.0"
Example 2: Registered on SIP
xStatus SystemUnit ContactName: "1234@192.0.2.0"
```

xStatus SystemUnit Notifications Notification [1..n] Type

Lists the system notification types. Notifications are issued e.g. when a system is rebooted because of a software upgrade, or when a factory reset is performed.

FactoryResetOK: This value is returned after a successful factory reset.

FactoryResetFailed: This value is returned after a failed factory reset attempt.

SoftwareUpgradeOK: This value is returned after a successful software upgrade.

SoftwareUpgradeFailed: This value is returned after a failed software upgrade attempt.

RebootRequired: This value is returned when a reboot is required.

Other: This value is returned for any other notifications.

All the notifications can be removed from the list by issuing the xCommand SystemUnit Notifications RemoveAll command.

Value space of the result returned:

<FactoryResetOK, FactoryResetFailed, SoftwareUpgradeOK, SoftwareUpgradeFailed, RebootRequired, Other>

Example:

```
xStatus SystemUnit Notifications Notification 1 Type
*s SystemUnit Notifications Notification 1 Type: SoftwareUpgradeOK
** end
```

xStatus SystemUnit Notifications Notification [1..n] Text

Lists text related to important system notifications. Notifications are issued e.g. when a system was rebooted because of a software upgrade, or when a factory reset has been performed.

All the notifications can be removed from the list by issuing the xCommand SystemUnit Notifications RemoveAll command.

Value space of the result returned:

<String>

Example:

```
xStatus SystemUnit Notifications Notification 1 Text
*s SystemUnit Notifications Notification 1 Text: "OK"
** end
```

xStatus SystemUnit Diagnostics LastRun

Shows when the SystemUnit diagnostics was last run on the codec.

Value space of the result returned:

    <String>

Example:

```
xStatus SystemUnit Diagnostics LastRun
*s SystemUnit Diagnostics LastRun: "2012-08-17, 16:23:27"
** end
```

xStatus SystemUnit Diagnostics Message Type

Returns information on the results of the latest diagnostics on the system.

Value space of the result returned:

    <ValidAdminPassword/CamerasDetected/H320GatewayStatus/ISDNLinkCompatibility/
SIPProfileRegistration/SIPListenPortAndOutboundMode/TLSVerifyRequiredCerts/
DefaultCallProtocolRegistered/NetSpeedAutoNegotiated/HasValidReleaseKey/
EthernetDuplexMatches/IPv4Assignment/IPv6Assignment/SIPProfileType/
CallProtocolIPStackPlatformCompatability/CameraStatus/CameraPairing/
CameraSoftwareVersion/SelectedVideoInputSourceConnected/OSDVideoOutput/
VideoFromInternalCamera/H323GatekeeperStatus>

Example:

```
xStatus SystemUnit Diagnostics Message type
*s SystemUnit Diagnostics Message 1 Type: InvalidAdminPassword
** end
```

xStatus SystemUnit Diagnostics Message Level

Returns information on the level of the diagnostics message.

Error: There is an error in the system. The system can still be used, but there can be some restrictions.

Warning: A problem is detected and a more specific report follows indicating the exact problem.

Critical: The warning level is critical. The system cannot be used.

Value space of the result returned:

    <Error/Warning/Critical>

Example:

```
xStatus SystemUnit Diagnostics Message Level
*r Status (status=Error):
    Reason: No match on address expression
    XPath: Status/SystemUnit/Diagnostics/Message/Level
** end
```

xStatus SystemUnit Diagnostics Message Description

A description of the current diagnostics alerts.

Value space of the result returned:

    <String>

Example:

```
xStatus SystemUnit Diagnostics Message Description
*s SystemUnit DiagnosticsResult Message 1 Description: "IP configuration
incomplete"
** end
```

xStatus SystemUnit Diagnostics Message References

Additional information on the diagnostics alert, if available.

Value space of the result returned:

    <String>

Example:

```
xStatus SystemUnit Diagnostics Message References
*s SystemUnit DiagnosticsResult Message 1 References: ""
** end
```

Time status

xStatus Time ZoneOlson

Shows the current time zone on Olson format.

Value space of the result returned:

<String>

Example:

```
xStatus Time ZoneOlson
*s Time ZoneOlson: Europe/Berlin
** end
```

Video status

xStatus Video Input

Shows the top level overview of the video input status.

xStatus Video Input LastConnectedSource

Shows the last connected video input source.

Value space of the result returned:

<1..1>

Example:

```
xStatus Video Input LastConnectedSource
*s Video Input LastConnectedSource: 0
** end
```

xStatus Video Input Source [1..3] Resolution Height

Shows the resolution height (in pixels) for the video input source.

Value space of the result returned:

<0..3000>

Example:

```
xStatus Video Input Source 1 Resolution Height
*s Video Input Source 1 Resolution Height: 1080
** end
```

xStatus Video Input Source [1..3] Resolution Width

Shows the resolution width (in pixels) for the video input source.

Value space of the result returned:

<0..4000>

Example:

```
xStatus Video Input Source 1 Resolution Width
*s Video Input Source 1 Resolution Width: 1920
** end
```

xStatus Video Input Source [1..3] Resolution RefreshRate

Shows the resolution refresh rate (Hz) for the video input source.

Value space of the result returned:

<0..300>

Example:

```
xStatus Video Input Source 1 Resolution RefreshRate
*s Video Input Source 1 Resolution RefreshRate: 50
** end
```

xStatus Video Input Source [1..3] Resolution FormatType

Shows the resolution format type for the video input source.

Value space of the result returned:

<Unknown/AnalogCVTBlanking/AnalogCVTReducedBlanking/AnalogGTFDefault/
AnalogGTFSecondary/AnalogDiscreteTiming/AnalogDMTBlanking/AnalogCEABlanking/Digital>

Example:

```
xStatus Video Input Source 1 Resolution FormatType
*s Video Input Source 1 Resolution FormatType: Digital
** end
```

xStatus Video Input Source [1..3] Resolution FormatStatus

Shows the resolution format status for the video input source.

Value space of the result returned:

<Ok/AnalogOutOfRange/AnalogNotFound/Interlaced/Error/Unknown>

Example:

```
xStatus Video Input Source 1 Resolution FormatStatus
*s Video Input Source 1 Resolution FormatStatus: Ok
** end
```

xStatus Video Input Source [1..3] MediaChannelId

Not in use in this release.

Value space of the result returned:

<Integer>

Example:

```
xStatus Video Input Source MediaChannelId
*s Video Input Source 1 MediaChannelId: 2
*s Video Input Source 2 MediaChannelId: 3
** end
```

xStatus Video Input HDMI [1] Connected

Shows if there is something connected to the HDMI connector. Not all connections can be detected.

Value space of the result returned:

<True/False/Unknown>

Example:

```
xStatus Video Input HDMI 1 Connected
*s Video Input HDMI 1 Connected: True
** end
```

xStatus Video Input HDMI [1] SignalState

Shows the signal state for the HDMI input.

Unknown: The signal format is unknown.

OK: A signal is detected and the signal format is supported.

Unsupported: A signal is detected, but the signal format is not supported.

Value space of the result returned:

<Unknown/OK/Unsupported>

Example:

```
xStatus Video Input HDMI 1 SignalState
*s Video Input HDMI 1 SignalState: OK
** end
```

xStatus Video Input DVI [2] Connected

Shows if there is something connected to the DVI connector. Not all connections can be detected.

Value space of the result returned:

<True/False/Unknown>

Example:

```
xStatus Video Input DVI 2 Connected
*s Video Input DVI 2 Connected: False
** end
```

xStatus Video Input DVI [2] SignalState

Shows the signal state for the DVI-I input.

Unknown: The signal format is unknown.

OK: A signal is detected and the signal format is supported.

Unsupported: A signal is detected, but the signal format is not supported.

Value space of the result returned:

<Unknown/OK/Unsupported>

Example:

```
xStatus Video Input DVI 2 SignalState
*s Video Input DVI 2 SignalState: OK
** end
```

xStatus Video Input USB 3 Connected

Shows if there is something connected to the USB connector. Not all connections can be detected.

Value space of the result returned:

<True/False/Unknown>

Example:

```
 xStatus Video Input USB Connected
*s Video Input USB 3 Connected: False
** end
```

xStatus Video Input USB 3 SignalState

Shows the signal state for the USB input.

Unknown: The signal format is unknown.

OK: A signal is detected and the signal format is supported.

Unsupported: A signal is detected, but the signal format is not supported.

Value space of the result returned:

<Unknown/OK/Unsupported>

Example:

```
xStatus Video Input USB SignalState
*s Video Input USB 3 SignalState: Unknown
** end
```

xStatus Video Output

Shows the top level overview of the video output status.

xStatus Video Output HDMI [1, 2] Resolution Height

Shows the resolution height (in pixels) for the video output HDMI.

Value space of the result returned:

<120..3000>

Example:

```
xStatus Video Output HDMI 1 Resolution Height
*s Video Output HDMI 1 Resolution Height: 720
** end
```

xStatus Video Output HDMI [1, 2] Resolution Width

Shows the resolution width (in pixels) for the video output HDMI.

Value space of the result returned:

<176..4000>

Example:

```
xStatus Video Output HDMI 1 Resolution Width
*s Video Output HDMI 1 Resolution Width: 1280
** end
```

xStatus Video Output HDMI [1, 2] Resolution RefreshRate

Shows the resolution refresh rate (Hz) for the video output HDMI.

Value space of the result returned:

<1..300>

Example:

```
xStatus Video Output HDMI 1 Resolution RefreshRate
*s Video Output HDMI 1 Resolution RefreshRate: 60
** end
```

xStatus Video Output HDMI [1, 2] Connected

Indicates whether a display is connected to the HDMI video output or not. NOTE: When a display enters standby mode, the endpoint may not be able to detect it. The connector status will then return False/Unknown even if the display is physically connected.

True - A display is connected to the video output connector.

False - No display is connected to the video output.

Value space of the result returned:

<True/False>

Example:

```
xStatus Video Output HDMI Connected
*s Video Output HDMI 1 Connected: False
** end
```

xStatus Video Layout

Shows the top level overview of the video layout status.

xStatus Video Layout Mode

Indicates type of layout currently used by the codec, custom or default.

Value space of the result returned:

<Default/Custom>

Example:

```
xStatus Video Layout Mode
*s Video Layout Mode: Default
** end
```

xStatus Video Layout PresentationView

Returns information about the presentation view mode.

Value space of the result returned:

<Default/Maximized/Minimized>

Example:

```
xStatus Video Layout PresentationView
*s Video Layout PresentationView: "Default"
** end
```

xStatus Video Layout Site [1..n] Output [1..2] FamilyName

Shows the name of the video layout family.

Value space of the result returned:

<String>

Example:

```
xStatus Video Layout Site 1 Output 1 FamilyName
*s Video Layout Site 1 Output 1 FamilyName: "fullscreen"
** end
```

xStatus Video Layout Site [1..n] Output [1..2] FullFamilyName

Shows the name, included information about self-view on/off, for the video layout family.

Value space of the result returned:

<String>

Example:

```
xStatus Video Layout Site 1 Output 1 FullFamilyName
*s Video Layout Site 1 Output 1 FullFamilyName: "fullscreen-local-single-
camctrl"
** end
```

xStatus Video Layout Site [1..n] Output [1..2] GraphicName

Shows the name of the graphic layout. The name identifies the layout used right now at the specified output.

NOTE: Note that while the FamilyName is constant as the long as the configurations on the system does not change, the GraphicName varies depending on system state (the number of participants for instance).

Value space of the result returned:

    \<String>

Example:

```
xStatus Video Layout Site 1 Output 1 GraphicName
*s Video Layout Site 1 Output 1 GraphicName: "full-pip"
** end
```

xStatus Video Layout Site [1..n] Output [1..2] Frame [1..6] PositionX

Shows the horizontal position of the upper left corner of the frame.

Value space of the result returned:

    \<0..10000>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 PositionX
*s Video Layout Site 1 Output 1 Frame 1 PositionX: 0
** end
```

xStatus Video Layout Site [1..n] Output [1..2] Frame [1..6] PositionY

Shows the vertical position of the upper left corner of the frame.

Value space of the result returned:

    \<0..10000>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 PositionY
*s Video Layout Site 1 Output 1 Frame 1 PositionY: 0
** end
```

xStatus Video Layout Site [1..n] Output [1..2] Frame [1..6] Width

Shows the width of the frame.

Value space of the result returned:

    \<0..10000>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 Width
*s Video Layout Site 1 Output 1 Frame 1 Width: 10000
** end
```

xStatus Video Layout Site [1..n] Output [1..2] Frame [1..6] Height

Shows the height of the frame.

Value space of the result returned:

    \<0..10000>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 Height
*s Video Layout Site 1 Output 1 Frame 1 Height: 10000
** end
```

xStatus Video Layout Site [1..n] Output [1..2] Frame [1..6] Layer

Shows the layer of the frame.

Value space of the result returned:

    \<1..6>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 Layer
*s Video Layout Site 1 Output 1 Frame 1 Layer: 1
** end
```

xStatus Video Layout Site [1..n] Output [1..2] Frame [1..6] VideoSourceType

Describes the video source type in the frame.

Value space of the result returned:

    \<String>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 VideoSourceType
*s Video Layout Site 1 Output 1 Frame 1 VideoSourceType: "graphic"
** end
```

xStatus Video Layout Site [1..n] Output [1..2] Frame [1..6] VideoSourceId

Shows the video source Id which is used when adding or updating frames. See the xCommand Video Layout Frame Add and xCommand Video Layout Frame Update.

Value space of the result returned:

<1..2>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 VideoSourceId
*s Video Layout Site 1 Output 1 Frame 1 VideoSourceId: 0
** end
```

xStatus Video Layout Site [1..n] Output [1..2] Frame [1..6] InputNumber

Shows the layout input number.

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 InputNumber
*s Video Layout Site 1 Output 1 Frame 1 InputNumber: 0
** end
```

xStatus Video Layout Site [1..n] Output [1..2] Frame [1..6] Filename

Shows the filename of the layout frame.

Value space of the result returned:

<String>

Example:

```
xStatus Video Layout Site 1 Output 1 Frame 1 Filename
*s Video Layout Site 1 Output 1 Frame 1 Filename: "/user/posters/
wallpaper.png"
** end
```

xStatus Video Layout Site Output Frame VideoSourceContent

Shows the video source content type of each layout frame.

noSource: There is no source.

black: The frame is black.

main: The frame displays the main image from a local or a remote user.

self-view: The frame displays the self-view image.

selfviewPip: The frame displays the self-view image in PiP format.

speaker: The frame displays the image of the speaker.

speakerPip: The frame displays the image of the speaker in PiP format.

presentation: The frame displays the image of the presentation.

presentationPip: The frame displays the image of the presentation in PiP format.

presentationPreview: The frame displays the preview image of the presentation.

localInput: The frame displays the image of the local input source.

wallpaper: The frame displays the wallpaper.

gui: The frame displays the user interface.

internal: For internal use.

Value space of the result returned:

<noSource/black/main/selfview/selfviewPip/speaker/speakerPip/presentation/presentationPip/ presentationPreview/localInput/wallpaper/gui/internal>

Example:

```
xStatus Video Layout Site Output Frame VideoSourceContent
*s Video Layout Site 1 Output 1 Frame 1 VideoSourceContent: wallpaper
*s Video Layout Site 1 Output 2 Frame 1 VideoSourceContent: wallpaper
*s Video Layout Site 1 Output 3 Frame 1 VideoSourceContent: wallpaper
*s Video Layout Site 1 Output 3 Frame 2 VideoSourceContent: gui
*s Video Layout Site 1 Output 4 Frame 1 VideoSourceContent: wallpaper
*s Video Layout Site 1 Output 5 Frame 1 VideoSourceContent: wallpaper
** end
```

xStatus Video Selfview Mode

Shows whether selfview mode is set on or not.

Value space of the result returned:

<On/Off>

Example:

```
xStatus Video Selfview Mode
*s Video Selfview Mode: Off
** end
```

xStatus Video Selfview FullscreenMode

Shows whether selfview is set on full screen mode or not.

Value space of the result returned:

<On/Off>

Example:

```
xStatus Video Selfview FullscreenMode
*s Video Selfview FullscreenMode: Off
** end
```

xStatus Video Selfview PIPPosition

Shows the position of the selfview image on the screen.

Value space of the result returned:

<UpperLeft/UpperCenter/UpperRight/CenterLeft/CenterRight/LowerLeft/LowerRight>

Example:

```
xStatus Video Selfview PIPPosition
*s Video Selfview PIPPosition: LowerRight
** end
```

xStatus Video Selfview OnMonitorRole

Identifies which monitor(s) contains the selfview, if present.

Value space of the result returned:

<First/Second/Third/Fourth>

Example:

```
xStatus Video Selfview OnMonitorRole
*s Video Selfview OnMonitorRole: First
** end
```

xStatus Video PIP ActiveSpeaker Position

Shows the position of the active speaker's image on the screen.

Value space of the result returned:

<UpperLeft/UpperCenter/UpperRight/CenterLeft/CenterRight/LowerLeft/LowerRight>

Example:

```
xStatus Video PIP ActiveSpeaker Position
*s Video PIP ActiveSpeaker Position: UpperCenter
** end
```

xStatus Video PIP Presentation Position

Shows the position of the presentation image on the screen.

Value space of the result returned:

<UpperLeft/UpperCenter/UpperRight/CenterLeft/CenterRight/LowerLeft/LowerRight>

Example:

```
xStatus Video PIP Presentation Position
*s Video PIP Presentation Position: CenterLeft
** end
```

Experimental status

The Experimental status is for testing only and should not be used unless agreed with Cisco. These
settings are not documented and WILL change in later releases.

Chapter 6

Appendices

Startup script

You can add a startup script on the codec to execute certain commands from the API during boot up.

Adding a startup script

The startup script can be used to execute certain commands from the API during boot up. To enable this feature one must log in to the codec as root and follow the below points.

Login to the codec

1. Connect to the codec through the network, using a command line interface (SSH, Telnet or SCP) and login as `root`

2. Make a user directory using the following command: "`mkdir /user/scripts.d`"

3. Put an executable file (permission must be changed to executable) in this directory.

   Example of the text in such a file:

   ```
   #!/usr/bin/env tsh

   xCommand Audio LocalInput Update InputId: 1
   MixerMode: Fixed
   ```

The startup script file

- The file must start with the following sequence: `#!/usr/bin/env tsh`
- The file can contain any xCommand or xConfiguration command
- The system executes the commands and configurations in sequence.
- The file can have any name as long as it is placed in this directory.
- For multiple commands you must use Unix end of line (LF). Windows end of line does not work.

Cisco TelePresence Remote Control

Function keys: Represents shortcuts and advanced functions. Each key reflects a soft key on screen.

Microphone: Press the 🎤 key to toggle the microphones on/off.

Volume: Press plus (+) or minus (−) on the volume key to adjust the volume.

Mute the ringtone: Press minus (−) on the volume key to mute the ringtone on an incoming call.

OK/Select: Press the ✓ key to confirm your choice or selection.

Phone book: Press the 📖 key to display the local Phone book.

Home: Press the 🏠 key to key to show the menu on screen.

Call: Using the 📞 key:
INITIATE A CALL: Select a name from the Phone book or enter the name, number or URI and press the Call key to initiate the call.

SHORTCUT TO RECENT CALLS: Use the Call key as a shortcut to Recent Calls when the Call menu is not visible.

Clear: Press the C key to remove characters in a text field.

Waking up the system

Grab the remote control and make sure your hand touches the rubber line sensors going along both sides of the remote control.

or: Press any key on the remote control.
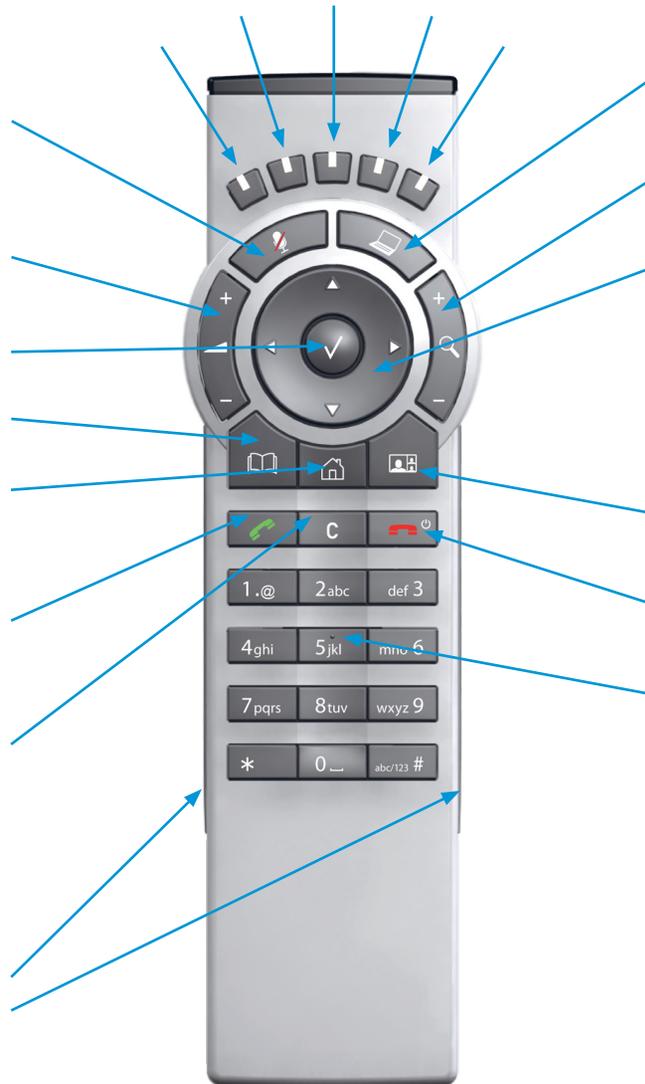
Presentation: Press the 💻 key to show/hide a presentation.

Zoom: Press the + or – on the 🔍 key to zoom the camera in and out.

Arrows:

· Up/Down: Use the ▲ and ▼ arrow keys to navigate in the menu.
· Arrow Right: Press the ▶ arrow key to expand the selected menu item or to move to the right in a text field.
· Arrow Left: Press the ◀ arrow key to go one step back in the menu or to move to the left in a text field.

Layout: Press the key to display the Layout menu, then select a view in the menu.

End call/Standby: Press the key to end a call, or when idle, press and hold the key to go into standby mode.

Alphanumeric keypad: Use the keypad in the same way as you would use a cellular phone.

· 0-9, a-z, period (.), @, space, *: Press a key repeatedly to toggle between the options displayed on each key.
· abc/123 #: Press the # key to toggle between touch tones mode (long press), lower case characters and numbers.

IR transmitter range (DIP switch setting)

The IR transmitter has a short and long range. Open the battery cover and remove the batteries to set the DIP switch.

· Short range (1 m): Move the DIP switch down.
· Longer range: Move the DIP switch up.

Remote control key map

The Cisco TelePresence Remote Control 5 has the following button codes and IR signal parameters.

Button codes – Remote control 5			
Dec	*Hex*	*Address*	*Button name*
0	0		
1	1	0	Number 1
2	2	0	Number 2
3	3	0	Number 3
4	4	0	Number 4
5	5	0	Number 5
6	6	0	Number 6
7	7	0	Number 7
8	8	0	Number 8
9	9	0	Number 9
10	0A	0	Number 0
11	0B	0	*
12	0C	0	#
13	0D		
14	0E		
15	0F		
16	10		
17	11	0	Presenter
18	12		
19	13		
20	14		
21	15		
22	16	0	Zoom out
23	17	0	Zoom in
24	18		
25	19	0	Volume down
26	1A	0	Volume up
27	1B	0	Microphone off
28	1C		
29	1D	0	Arrow up
30	1E	0	Arrow down

Button codes – Remote control 5			
Dec	*Hex*	*Address*	*Button name*
31	1F	0	Arrow left
32	20	0	Arrow right
33	21	0	Ok
34	22	0	Call
35	23	0	End call
36	24	0	Phone book
37	25	0	Layout
38	26	0	Clear
39	27		
–	–		
41	29		
42	2A	0	Soft key 1
43	2B	0	Soft key 2
44	2C	0	Soft key 3
45	2D	0	Soft key 4
46	2E	0	Soft key 5
–	–		
–	–		
–	–		
–	–		
51	33	0	Home

IR Signal parameters	
Name	*Description*
Protocol	Siemens SDA2208
Reference frequency	485 kHz
Address	4 and 7
IR wavelength	940 nm
IR carrier frequency	30 kHz

The SystemTools commands

Note: The systemtools commands are used for administrative control of the codec and is only available from a command line interface. Systemtools should not be used to program the codec.

Required parameters in angle brackets: <text>

Optional parameters in square brackets: [text]

To get an overview of the supported commands type "systemtools ?".

Example:

```
systemtools ?
authorizedkeys
boothalt
license
network
pairing
passwd
pki
rootsettings
securitysettings
securitystatus


OK
```

To see the usage of the commands add a question mark after the command.

Example:

```
systemtools authorizedkeys ?
usage: authorizedkeys <add <method> <key>
[comment] | delete <id> | list | clear>


OK
```

systemtools authorizedkeys add <method> <key> [comment]

Add the SSH keys on the codec.

method(r): The encryption method used, which can be SSH-rsa or SSH-dss.

key(r): The public key as it is in the SSH public key file.

comment: Optional comment.

systemtools authorizedkeys delete <id>

Delete the given SSH key on the codec, defined by the id.

id(r): The id as as displayed in the authorized keys list.

systemtools authorizedkeys list

List the SSH keys on the codec.

systemtools authorizedkeys clear

Clear all SSH keys on the codec.

systemtools boothalt allow

Allow the user to stop the system during the boot loader startup sequence using a serial console.

systemtools boothalt prevent

Prevent the user from stopping the system during the boot loader startup sequence using a serial console.

systemtools boothalt status

Show whether or not the system can be stopped during a boot loader startup sequence.

systemtools license list

Lists all the licenses for the codec.

systemtools license show <name>

Shows the content of a license file, define by the name.

name(r): The name of the license file.

systemtools network ping <hostname>

Network debug command.

hostname(r): The IP address or URL of the host.

systemtools network traceroute <hostname>

Network debug command.

hostname(r): The IP address or URL of the host.

systemtools network netstat

Network debug command.

systemtools network addrs

Check the systems IP address.

systemtools pairing unpair

Remove association with Cisco TelePresence Touch controller.

systemtools passwd

Change the password for the logged in user.

systemtools pki list

Lists the codec certificate and CA list if they exist.

systemtools pki delete <cert-name>

Delete the codec certificate and CA list if they exist.

cert-name(r): The name of the certificate.

systemtools rootsettings get

Obtain the current setting for the systemtools rootsetting.

systemtools rootsettings on [password]

Command to control the root user availability.

Enable access to the system for the root user on all ports.
password: The root user password.

systemtools rootsettings serial [password]

Command to control the root user availability.

Enable access to the system for the root user on the serial port.
password: The root user password.

systemtools rootsettings off

Command to control the root user availability.

Disable access to the system for the root user on all ports.

systemtools rootsettings never

Command to control the root user availability.

NOTE: The root user is permanently turned off.

To get back the root user the system must be reset to factory defaults, ref. xCommand SystemUnit FactoryReset.

systemtools securitysettings jitc

Set up security requirements so they meet JITC.

Set password and PIN polices enforced on the codec.

systemtools securitysettings isjitc

Check if the current settings are JTIC compiant.

systemtools securitysettings default

Revert to default security settings.

systemtools securitysettings ask

Query for the separate configurations. When issuing this command you can see each policy separately.

- Press enter to keep the current value.
- Enter a number and press enter to change the given policy.
- The default value "0" indicates no restrictions.

Max failed login attempts [0]?
- Number of failed logins until a user is set inactive.

Suspend-time after max failed login attempts (minutes) [0]?
- Number of minutes the user is set inactive after maximum failed login attempts have been exceeded.

Max simultaneous sessions total [0]?
- Maximum number of users that can be logged in simultaneous to web and maximum number of users that can be logged in simultaneous to ssh/Telnet.

Max simultaneous sessions per user [0]?
- Maximum number of simultaneous sessions per user.

Number of passwords to remember [0]?
- Number of previous passwords that the new password must differ from.

Number of PINs to remember [0]?
- Number of previous PINs that the new PIN must differ from.

Maximum time between password renewals (days) [0]?
- If the user has not changed the password within the renewal time the user will be set inactive.

Minimum time between password renewals (hours) [0]?
- The user can only change password once within this limit.

Maximum time between PIN renewals (days) [0]?
- If the user has not changed the PIN within the renewal time the user will be set inactive.

Minimum time between PIN renewals (hours) [0]?
- The user can only change PIN once within this limit.

Maximum time between logins (days) [0]?
- If the user has not logged in within this limit the user will be set inactive.

Max consecutive equal digits in PINs [0]?
- Maximum consecutive equal digits in PINs.

Minimum number of digits in PINs [0]?
- Minimum number of digits in PINs.

Maximum number of digits in PINs [0]?
- Maximum number of digits in PINs.

Max consecutive identical characters in passwords [0]?
- Maximum consecutive identical characters in passwords.

Minimum number of characters in passwords [0]?
- Minimum number of characters in passwords.

Maxmum number of characters in passwords [0]?
- Maxmum number of characters in passwords.

Minimum number of lower-case letters in passwords [0]?
- Minimum number of lower-case letters in passwords.

Minimum number of upper-case letters in passwords [0]?
- Minimum number of upper-case letters in passwords.

Minimum number of numerical characters in passwords [0]?
- Minimum number of numerical characters in passwords.

Minimum number of special characters in passwords [0]?
- Minimum number of special characters in passwords.

Minimum number of character groups in passwords [0]?
- Minimum number of character groups in passwords.

Minimum number of character changed from previous password [0]?
- Minimum number of character changed from previous password.

systemtools securitystatus

Shows the security status for the codec.

183

About disconnect cause types

The following parameters are logged when a call is disconnected. The disconnect cause types are used in disconnect events (xEvent) and also logged in xHistory CallLogs.

CauseType	Describes why the call was disconnected. The value space is { OtherLocal, LocalDisconnect, UnknownRemoteSite, LocalBusy, LocalReject, InsufficientSecurity, OtherRemote, RemoteDisconnect, RemoteBusy, RemoteRejected, RemoteNoAnswer, CallForwarded, NetworkRejected }
CauseString	Describes the Cause Code.
CauseCode	The disconnect Cause Codes are defined in SIP and Q.850.
CauseOrigin	SIP, Q.850, internal.

Example 1:

```
xHistory CallLogs Call 694
...
*h xHistory CallLogs Call 694 DisconnectCause: "Normal"
*h xHistory CallLogs Call 694 DisconnectCauseType: RemoteDisconnect
*h xHistory CallLogs Call 694 DisconnectCauseCode: 16
*h xHistory CallLogs Call 694 DisconnectCauseOrigin: Q850
...
** end
```

On our web site you will find an overview of the worldwide Cisco contacts.

Go to: ▸ http://www.cisco.com/web/siteassets/contacts

Corporate Headquarters

Cisco Systems, Inc.

170 West Tasman Dr.

San Jose, CA 95134 USA